4: Linear Time Invariant Systems

LTI Systems

$$x[n] \longrightarrow \boxed{\mathscr{H}} \longrightarrow y[n] = \mathscr{H}(x[n])$$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$

Time Invariant: $y[n]=\mathscr{H}\left(x[n]\right)\Rightarrow y[n-r]=\mathscr{H}\left(x[n-r]\right)\forall r$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$
Time Invariant: $y[n]=\mathscr{H}\left(x[n]\right)\Rightarrow y[n-r]=\mathscr{H}\left(x[n-r]\right)\forall r$

The behaviour of an LTI system is completely defined by its impulse response: $h[n]=\mathscr{H}\left(\delta[n]\right)$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$
Time Invariant: $y[n]=\mathscr{H}\left(x[n]\right)\Rightarrow y[n-r]=\mathscr{H}\left(x[n-r]\right)\forall r$

The behaviour of an LTI system is completely defined by its impulse response: $h[n]=\mathscr{H}\left(\delta[n]\right)$

Proof:
We can always write $x[n]=\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$
Time Invariant: $y[n]=\mathscr{H}\left(x[n]\right)\Rightarrow y[n-r]=\mathscr{H}\left(x[n-r]\right)\forall r$

The behaviour of an LTI system is completely defined by its impulse response: $h[n]=\mathscr{H}\left(\delta[n]\right)$

Proof:
We can always write $x[n]=\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]$

Hence $\mathscr{H}\left(x[n]\right)=\mathscr{H}\left(\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]\right)$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$
Time Invariant: $y[n]=\mathscr{H}\left(x[n]\right)\Rightarrow y[n-r]=\mathscr{H}\left(x[n-r]\right)\forall r$

The behaviour of an LTI system is completely defined by its impulse response: $h[n]=\mathscr{H}\left(\delta[n]\right)$

Proof:
We can always write $x[n]=\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]$

Hence
$$\begin{aligned}\mathscr{H}\left(x[n]\right)&=\mathscr{H}\left(\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]\right)\\&=\sum_{r=-\infty}^{\infty}x[r]\mathscr{H}\left(\delta[n-r]\right)\end{aligned}$$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n]+\beta v[n]\right)=\alpha\mathscr{H}\left(u[n]\right)+\beta\mathscr{H}\left(v[n]\right)$

Time Invariant: $y[n]=\mathscr{H}\left(x[n]\right)\Rightarrow y[n-r]=\mathscr{H}\left(x[n-r]\right)\forall r$

The behaviour of an LTI system is completely defined by its impulse response: $h[n]=\mathscr{H}\left(\delta[n]\right)$

Proof:

We can always write $x[n]=\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]$

Hence
$$\begin{aligned}\mathscr{H}\left(x[n]\right)&=\mathscr{H}\left(\sum_{r=-\infty}^{\infty}x[r]\delta[n-r]\right)\\&=\sum_{r=-\infty}^{\infty}x[r]\mathscr{H}\left(\delta[n-r]\right)\\&=\sum_{r=-\infty}^{\infty}x[r]h[n-r]\end{aligned}$$

LTI Systems

Linear Time-invariant (LTI) systems have two properties:

Linear: $\mathscr{H}\left(\alpha u[n] + \beta v[n]\right) = \alpha \mathscr{H}\left(u[n]\right) + \beta \mathscr{H}\left(v[n]\right)$
Time Invariant: $y[n] = \mathscr{H}\left(x[n]\right) \Rightarrow y[n-r] = \mathscr{H}\left(x[n-r]\right) \forall r$

The behaviour of an LTI system is completely defined by its impulse response: $h[n] = \mathscr{H}\left(\delta[n]\right)$

Proof:
We can always write $x[n] = \sum_{r=-\infty}^{\infty} x[r]\delta[n-r]$

Hence
$$\begin{aligned}\mathscr{H}\left(x[n]\right) &= \mathscr{H}\left(\sum_{r=-\infty}^{\infty} x[r]\delta[n-r]\right)\\ &= \sum_{r=-\infty}^{\infty} x[r]\mathscr{H}\left(\delta[n-r]\right)\\ &= \sum_{r=-\infty}^{\infty} x[r]h[n-r]\\ &= x[n] * h[n]\end{aligned}$$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \stackrel{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \overset{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \overset{(i)}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

$\Rightarrow x[n] * v[n] * w[n]$ is unambiguous

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \overset{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

$\Rightarrow x[n] * v[n] * w[n]$ is unambiguous

Proof: $\sum_{r,s} x[n-r]v[r-s]w[s] \overset{\text{(i)}}{=} \sum_{p,q} x[p]v[q-p]w[n-q]$

(i) substitute $p = n - r$, $q = n - s$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \stackrel{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

$\Rightarrow x[n] * v[n] * w[n]$ is unambiguous

Proof: $\sum_{r,s} x[n-r]v[r-s]w[s] \stackrel{\text{(i)}}{=} \sum_{p,q} x[p]v[q-p]w[n-q]$

(i) substitute $p = n - r$, $q = n - s$

Distributive over $+$:

$x[n] * (\alpha v[n] + \beta w[n]) = (x[n] * \alpha v[n]) + (x[n] * \beta w[n])$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \overset{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

$\Rightarrow x[n] * v[n] * w[n]$ is unambiguous

Proof: $\sum_{r,s} x[n-r]v[r-s]w[s] \overset{\text{(i)}}{=} \sum_{p,q} x[p]v[q-p]w[n-q]$

(i) substitute $p = n - r$, $q = n - s$

Distributive over $+$:

$x[n] * (\alpha v[n] + \beta w[n]) = (x[n] * \alpha v[n]) + (x[n] * \beta w[n])$

Proof: $\sum_r x[n-r](\alpha v[r] + \beta w[r]) =$

$\alpha \sum_r x[n-r]v[r] + \beta \sum_r x[n-r]w[r]$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \stackrel{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

$\Rightarrow x[n] * v[n] * w[n]$ is unambiguous

Proof: $\sum_{r,s} x[n-r]v[r-s]w[s] \stackrel{\text{(i)}}{=} \sum_{p,q} x[p]v[q-p]w[n-q]$

(i) substitute $p = n - r$, $q = n - s$

Distributive over $+$:

$x[n] * (\alpha v[n] + \beta w[n]) = (x[n] * \alpha v[n]) + (x[n] * \beta w[n])$

Proof: $\sum_r x[n-r]\left(\alpha v[r] + \beta w[r]\right) =$

$\alpha \sum_r x[n-r]v[r] + \beta \sum_r x[n-r]w[r]$

Identity: $x[n] * \delta[n] = x[n]$

Convolution Properties

Convolution: $x[n] * v[n] = \sum_{r=-\infty}^{\infty} x[r]v[n-r]$

Convolution obeys normal arithmetic rules for multiplication:

Commutative: $x[n] * v[n] = v[n] * x[n]$

Proof: $\sum_r x[r]v[n-r] \overset{\text{(i)}}{=} \sum_p x[n-p]v[p]$

(i) substitute $p = n - r$

Associative: $x[n] * (v[n] * w[n]) = (x[n] * v[n]) * w[n]$

$\Rightarrow x[n] * v[n] * w[n]$ is unambiguous

Proof: $\sum_{r,s} x[n-r]v[r-s]w[s] \overset{\text{(i)}}{=} \sum_{p,q} x[p]v[q-p]w[n-q]$

(i) substitute $p = n - r$, $q = n - s$

Distributive over $+$:

$x[n] * (\alpha v[n] + \beta w[n]) = (x[n] * \alpha v[n]) + (x[n] * \beta w[n])$

Proof: $\sum_r x[n-r]\left(\alpha v[r] + \beta w[r]\right) =$

$\alpha \sum_r x[n-r]v[r] + \beta \sum_r x[n-r]w[r]$

Identity: $x[n] * \delta[n] = x[n]$

Proof: $\sum_r \delta[r]x[n-r] \overset{\text{(i)}}{=} x[n]$ (i) all terms zero except $r = 0$.

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

Proof $(1) \Rightarrow (2)$:

Define $x[n] = \begin{cases} 1 & h[-n] \geq 0 \\ -1 & h[-n] < 0 \end{cases}$

then $y[0] = \sum x[0-n]h[n] = \sum |h[n]|$.

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

Proof $(1) \Rightarrow (2)$:

Define $x[n] = \begin{cases} 1 & h[-n] \geq 0 \\ -1 & h[-n] < 0 \end{cases}$

then $y[0] = \sum x[0-n]h[n] = \sum |h[n]|$.

But $|x[n]| \leq 1 \forall n$ so BIBO $\Rightarrow y[0] = \sum |h[n]| < \infty$.

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

Proof $(1) \Rightarrow (2)$:

Define $x[n] = \begin{cases} 1 & h[-n] \geq 0 \\ -1 & h[-n] < 0 \end{cases}$

then $y[0] = \sum x[0-n]h[n] = \sum |h[n]|$.

But $|x[n]| \leq 1 \forall n$ so BIBO $\Rightarrow y[0] = \sum |h[n]| < \infty$.

Proof $(2) \Rightarrow (1)$:

Suppose $\sum |h[n]| = S < \infty$ and $|x[n]| \leq B$ is bounded.

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

Proof $(1) \Rightarrow (2)$:

Define $x[n] = \begin{cases} 1 & h[-n] \geq 0 \\ -1 & h[-n] < 0 \end{cases}$

then $y[0] = \sum x[0-n]h[n] = \sum |h[n]|$.

But $|x[n]| \leq 1 \forall n$ so BIBO $\Rightarrow y[0] = \sum |h[n]| < \infty$.

Proof $(2) \Rightarrow (1)$:

Suppose $\sum |h[n]| = S < \infty$ and $|x[n]| \leq B$ is bounded.

Then $|y[n]| = \left|\sum_{r=-\infty}^{\infty} x[n-r]h[r]\right|$

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

Proof $(1) \Rightarrow (2)$:

Define $x[n] = \begin{cases} 1 & h[-n] \geq 0 \\ -1 & h[-n] < 0 \end{cases}$

then $y[0] = \sum x[0-n]h[n] = \sum |h[n]|$.

But $|x[n]| \leq 1 \forall n$ so BIBO $\Rightarrow y[0] = \sum |h[n]| < \infty$.

Proof $(2) \Rightarrow (1)$:

Suppose $\sum |h[n]| = S < \infty$ and $|x[n]| \leq B$ is bounded.

Then $|y[n]| = \left| \sum_{r=-\infty}^{\infty} x[n-r]h[r] \right|$

$$\leq \sum_{r=-\infty}^{\infty} |x[n-r]| \, |h[r]|$$

BIBO Stability

BIBO Stability: Bounded Input, $x[n] \Rightarrow$ Bounded Output, $y[n]$

The following are equivalent:

(1) An LTI system is BIBO stable

(2) $h[n]$ is absolutely summable, i.e. $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$

(3) $H(z)$ region of absolute convergence includes $|z| = 1$.

Proof $(1) \Rightarrow (2)$:

Define $x[n] = \begin{cases} 1 & h[-n] \geq 0 \\ -1 & h[-n] < 0 \end{cases}$

then $y[0] = \sum x[0-n]h[n] = \sum |h[n]|$.

But $|x[n]| \leq 1 \forall n$ so BIBO $\Rightarrow y[0] = \sum |h[n]| < \infty$.

Proof $(2) \Rightarrow (1)$:

Suppose $\sum |h[n]| = S < \infty$ and $|x[n]| \leq B$ is bounded.

$$\begin{aligned}
\text{Then } |y[n]| &= \left|\sum_{r=-\infty}^{\infty} x[n-r]h[r]\right| \\
&\leq \sum_{r=-\infty}^{\infty} |x[n-r]|\,|h[r]| \\
&\leq B \sum_{r=-\infty}^{\infty} |h[r]| \leq BS < \infty
\end{aligned}$$

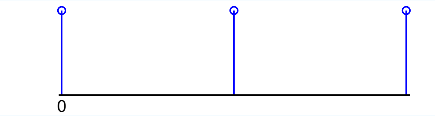
Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

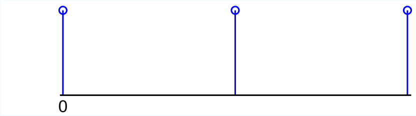
Example: $h[n] = \left[\begin{array}{ccc} 1 & 1 & 1 \end{array}\right]$



Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

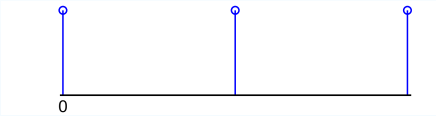


$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$

Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

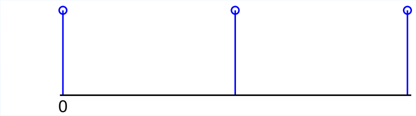


$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$
$$= e^{-j\omega}(1 + 2\cos\omega)$$

Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$ is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

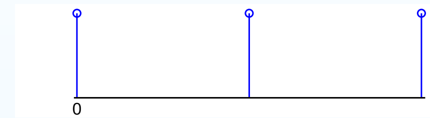


$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$
$$= e^{-j\omega}(1 + 2\cos\omega)$$

$$\left|H(e^{j\omega})\right| = |1 + 2\cos\omega|$$

Frequency Response

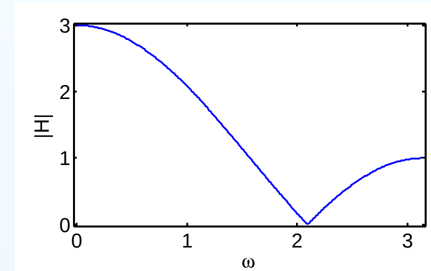
For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.



Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$
$$= e^{-j\omega}\left(1 + 2\cos\omega\right)$$

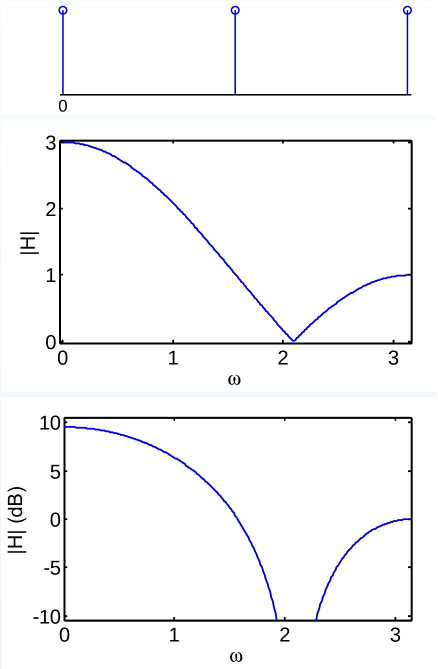
$$\left|H(e^{j\omega})\right| = |1 + 2\cos\omega|$$



Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$



$$\begin{aligned} H(e^{j\omega}) &= 1 + e^{-j\omega} + e^{-j2\omega} \\ &= e^{-j\omega}\left(1 + 2\cos\omega\right) \end{aligned}$$

$$\left|H(e^{j\omega})\right| = \left|1 + 2\cos\omega\right|$$

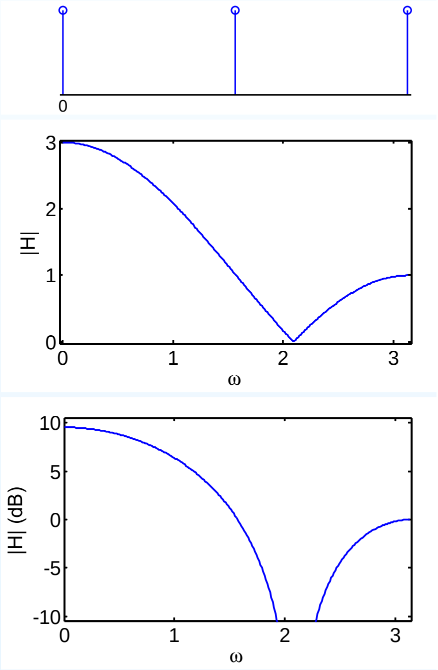
Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$
$$= e^{-j\omega}\left(1 + 2\cos\omega\right)$$

$$\left|H(e^{j\omega})\right| = \left|1 + 2\cos\omega\right|$$
$$\angle H(e^{j\omega}) = -\omega + \pi\frac{1-\operatorname{sgn}(1+2\cos\omega)}{2}$$



Frequency Response

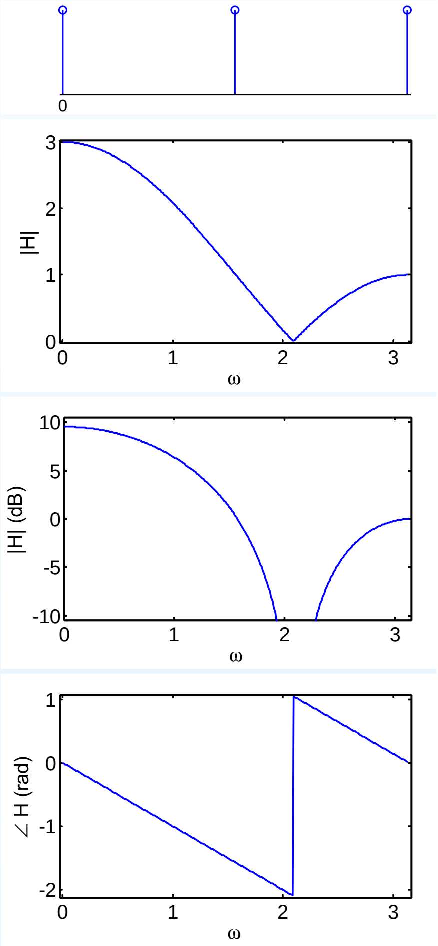
For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$\begin{aligned} H(e^{j\omega}) &= 1 + e^{-j\omega} + e^{-j2\omega} \\ &= e^{-j\omega}\left(1 + 2\cos\omega\right) \end{aligned}$$

$$\left|H(e^{j\omega})\right| = \left|1 + 2\cos\omega\right|$$

$$\angle H(e^{j\omega}) = -\omega + \pi\frac{1-\operatorname{sgn}(1+2\cos\omega)}{2}$$



Frequency Response

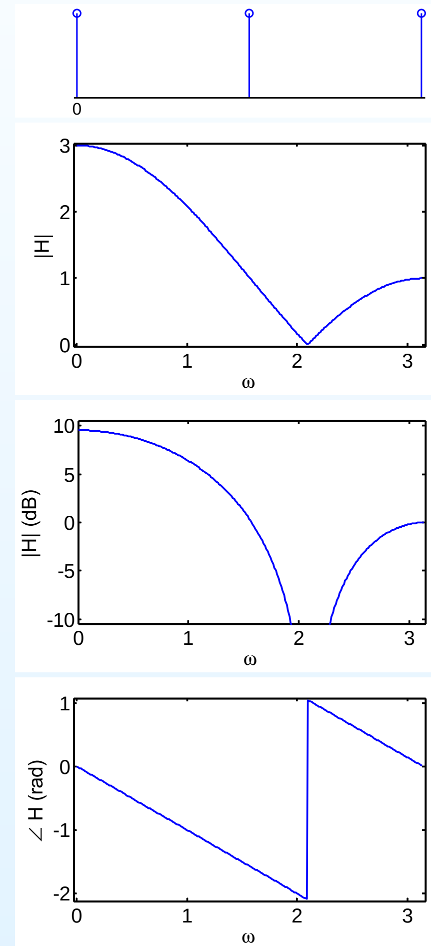
For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$\begin{aligned} H(e^{j\omega}) &= 1 + e^{-j\omega} + e^{-j2\omega} \\ &= e^{-j\omega}\left(1 + 2\cos\omega\right) \end{aligned}$$

$$\left|H(e^{j\omega})\right| = |1 + 2\cos\omega|$$

$$\angle H(e^{j\omega}) = -\omega + \pi\frac{1-\operatorname{sgn}(1+2\cos\omega)}{2}$$



Sign change in $(1 + 2\cos\omega)$ at $\omega = 2.1$ gives

(a) gradient discontinuity in $\left|H(e^{j\omega})\right|$

(b) an abrupt phase change of $\pm\pi$.

Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

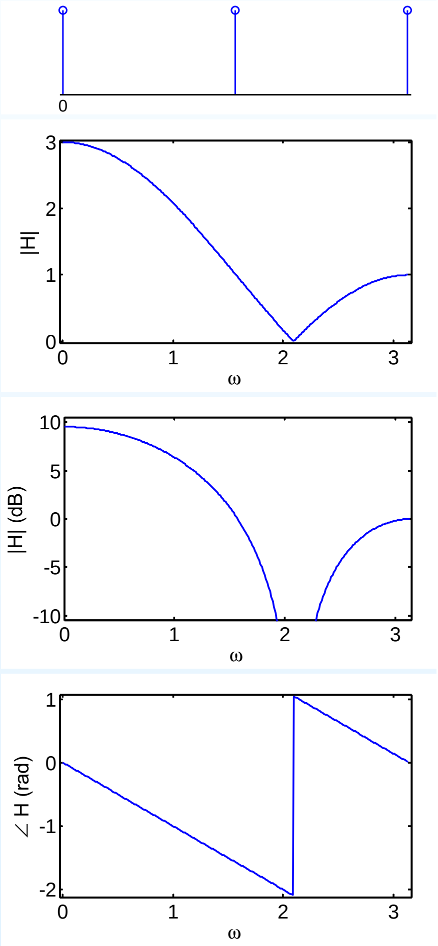
Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$
$$= e^{-j\omega}\left(1 + 2\cos\omega\right)$$

$$\left|H(e^{j\omega})\right| = \left|1 + 2\cos\omega\right|$$
$$\angle H(e^{j\omega}) = -\omega + \pi\frac{1-\operatorname{sgn}(1+2\cos\omega)}{2}$$

Sign change in $(1 + 2\cos\omega)$ at $\omega = 2.1$ gives
(a) gradient discontinuity in $\left|H(e^{j\omega})\right|$
(b) an abrupt phase change of $\pm\pi$.

Group delay is $-\frac{d}{d\omega}\angle H(e^{j\omega})$: gives delay of the modulation envelope at each ω.



Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.

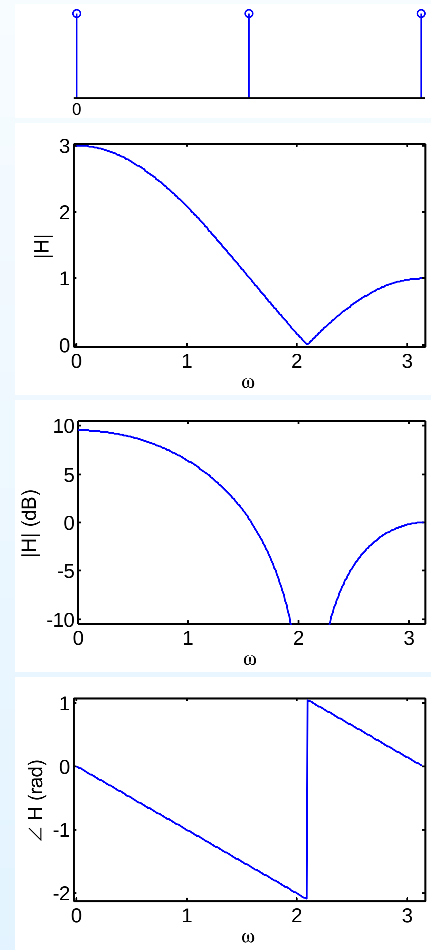
Example: $h[n] = \left[\begin{array}{ccc} 1 & 1 & 1 \end{array}\right]$

$$\begin{aligned} H(e^{j\omega}) &= 1 + e^{-j\omega} + e^{-j2\omega} \\ &= e^{-j\omega}\left(1 + 2\cos\omega\right) \end{aligned}$$

$$\left|H(e^{j\omega})\right| = \left|1 + 2\cos\omega\right|$$
$$\angle H(e^{j\omega}) = -\omega + \pi\frac{1-\operatorname{sgn}(1+2\cos\omega)}{2}$$

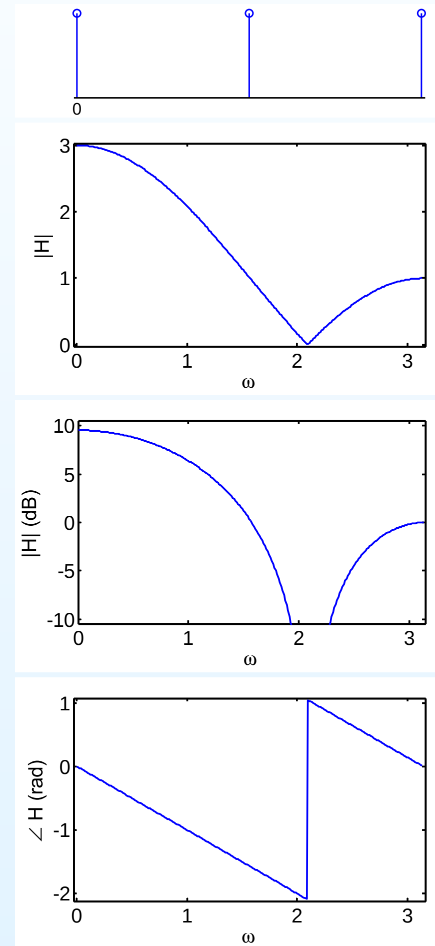
Sign change in $(1 + 2\cos\omega)$ at $\omega = 2.1$ gives
(a) gradient discontinuity in $\left|H(e^{j\omega})\right|$
(b) an abrupt phase change of $\pm\pi$.

Group delay is $-\frac{d}{d\omega}\angle H(e^{j\omega})$: gives delay of the modulation envelope at each ω.
Normally varies with ω but for a symmetric filter it is constant: in this case $+1$ samples.



Frequency Response

For a BIBO stable system $Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$
where $H(e^{j\omega})$is the DTFT of $h[n]$ i.e. $H(z)$ evaluated at $z = e^{j\omega}$.



Example: $h[n] = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$$H(e^{j\omega}) = 1 + e^{-j\omega} + e^{-j2\omega}$$
$$= e^{-j\omega}\left(1 + 2\cos\omega\right)$$

$$\left|H(e^{j\omega})\right| = |1 + 2\cos\omega|$$
$$\angle H(e^{j\omega}) = -\omega + \pi\frac{1-\operatorname{sgn}(1+2\cos\omega)}{2}$$

Sign change in $(1 + 2\cos\omega)$ at $\omega = 2.1$ gives

(a) gradient discontinuity in $|H(e^{j\omega})|$

(b) an abrupt phase change of $\pm\pi$.

Group delay is $-\frac{d}{d\omega}\angle H(e^{j\omega})$: gives delay of the modulation envelope at each ω.
Normally varies with ω but for a symmetric filter it is constant: in this case $+1$ samples.
Discontinuities of $\pm k\pi$ do not affect group delay.

Causality

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Causality

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Formal definition:

If $v[n] = x[n]$ for $n \leq n_0$ then $\mathscr{H}(v[n]) = \mathscr{H}(x[n])$ for $n \leq n_0$.

Causality

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Formal definition:

If $v[n] = x[n]$ for $n \leq n_0$ then $\mathscr{H}\left(v[n]\right) = \mathscr{H}\left(x[n]\right)$ for $n \leq n_0$.

The following are equivalent:

(1) An LTI system is causal

Causality

+

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Formal definition:

If $v[n] = x[n]$ for $n \leq n_0$ then $\mathscr{H}\left(v[n]\right) = \mathscr{H}\left(x[n]\right)$ for $n \leq n_0$.

The following are equivalent:

(1) An LTI system is causal

(2) $h[n]$ is causal $\Leftrightarrow h[n] = 0$ for $n < 0$

Causality

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Formal definition:

If $v[n] = x[n]$ for $n \leq n_0$ then $\mathscr{H}\left(v[n]\right) = \mathscr{H}\left(x[n]\right)$ for $n \leq n_0$.

The following are equivalent:

(1) An LTI system is causal

(2) $h[n]$ is causal $\Leftrightarrow h[n] = 0$ for $n < 0$

(3) $H(z)$ converges for $z = \infty$

Causality

+

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Formal definition:

If $v[n] = x[n]$ for $n \leq n_0$ then $\mathscr{H}(v[n]) = \mathscr{H}(x[n])$ for $n \leq n_0$.

The following are equivalent:

(1) An LTI system is causal

(2) $h[n]$ is causal $\Leftrightarrow h[n] = 0$ for $n < 0$

(3) $H(z)$ converges for $z = \infty$

Any right-sided sequence can be made causal by adding a delay.

Causality

+

Causal System: cannot see into the future

i.e. output at time n depends only on inputs up to time n.

Formal definition:

If $v[n] = x[n]$ for $n \le n_0$ then $\mathscr{H}(v[n]) = \mathscr{H}(x[n])$ for $n \le n_0$.

The following are equivalent:

(1) An LTI system is causal

(2) $h[n]$ is causal $\Leftrightarrow h[n] = 0$ for $n < 0$

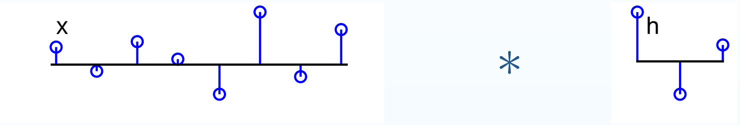
(3) $H(z)$ converges for $z = \infty$

Any right-sided sequence can be made causal by adding a delay.

All the systems we will deal with are causal.

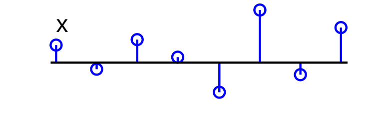
Convolution Complexity

$y[n] = x[n] * h[n]$: convolve $x[0:N-1]$ with $h[0:M-1]$

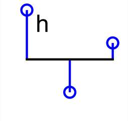


Convolution Complexity

$y[n] = x[n] * h[n]$: convolve $x[0 : N-1]$ with $h[0 : M-1]$



$*$

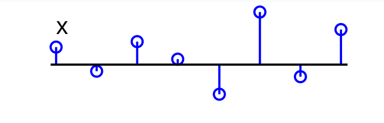


Convolution sum:

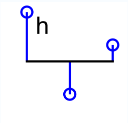
$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

Convolution Complexity

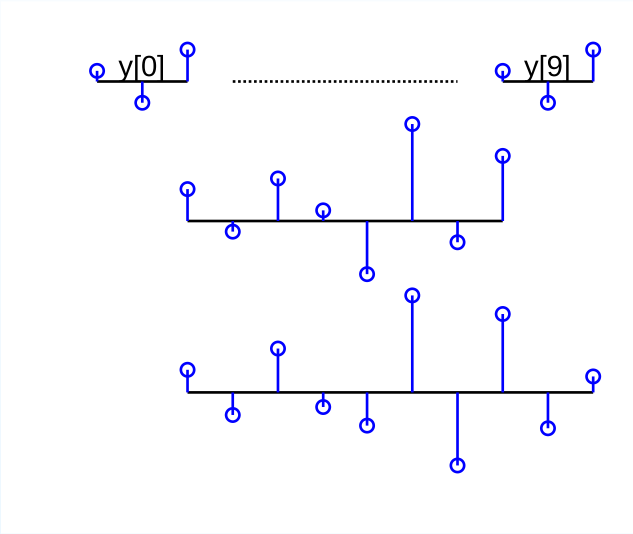
$y[n] = x[n] * h[n]$: convolve $x[0:N-1]$ with $h[0:M-1]$



$*$



$\rightarrow$



Convolution sum:

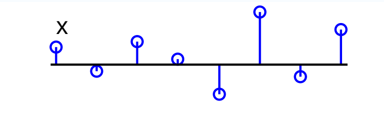
$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

$N = 8, M = 3$

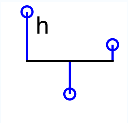
$M + N - 1 = 10$

Convolution Complexity

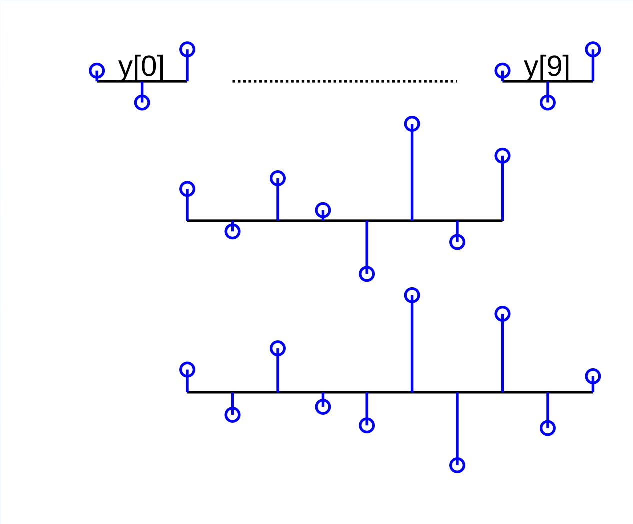
$y[n] = x[n] * h[n]$: convolve $x[0:N-1]$ with $h[0:M-1]$



$*$



$\rightarrow$



Convolution sum:

$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$

$y[n]$ is only non-zero in the range

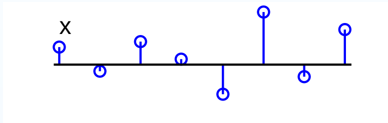
$$0 \le n \le M+N-2$$

$N = 8$, $M = 3$

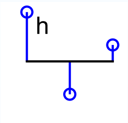
$M + N - 1 = 10$

Convolution Complexity

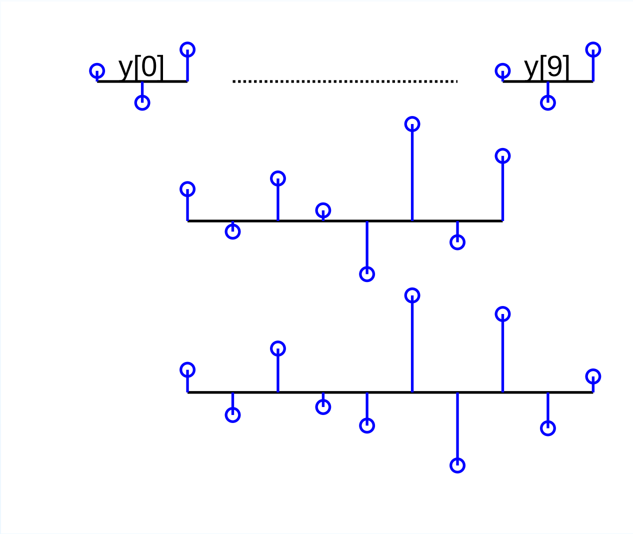
$y[n] = x[n] * h[n]$: convolve $x[0 : N-1]$ with $h[0 : M-1]$



$*$



$\rightarrow$



Convolution sum:

$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

$y[n]$ is only non-zero in the range

$$0 \le n \le M + N - 2$$

Thus $y[n]$ has only

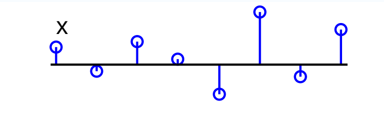
$M + N - 1$ non-zero values

$N = 8$, $M = 3$

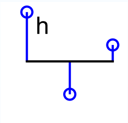
$M + N - 1 = 10$

Convolution Complexity

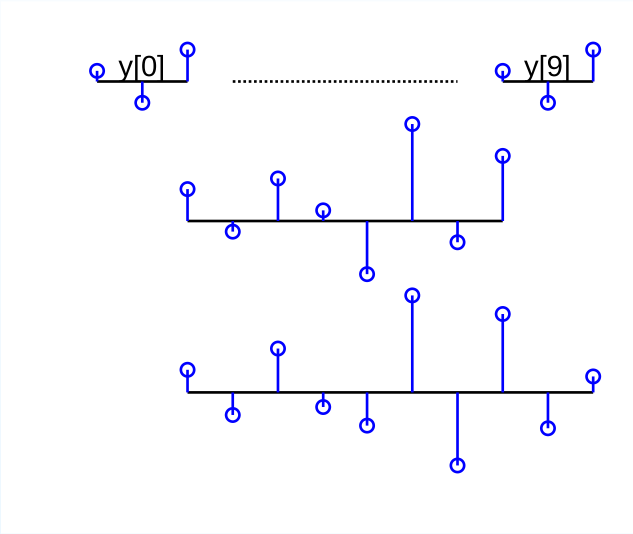
$y[n] = x[n] * h[n]$: convolve $x[0:N-1]$ with $h[0:M-1]$



$*$



$\rightarrow$



Convolution sum:

$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

$y[n]$ is only non-zero in the range

$$0 \le n \le M+N-2$$

Thus $y[n]$ has only

$M+N-1$ non-zero values

$N = 8$, $M = 3$

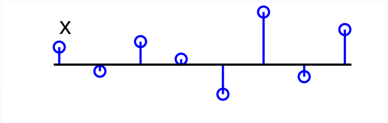
$M+N-1 = 10$

Algebraically:

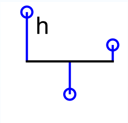
$$x[n-r] \neq 0 \Rightarrow 0 \le n-r \le N-1$$

Convolution Complexity

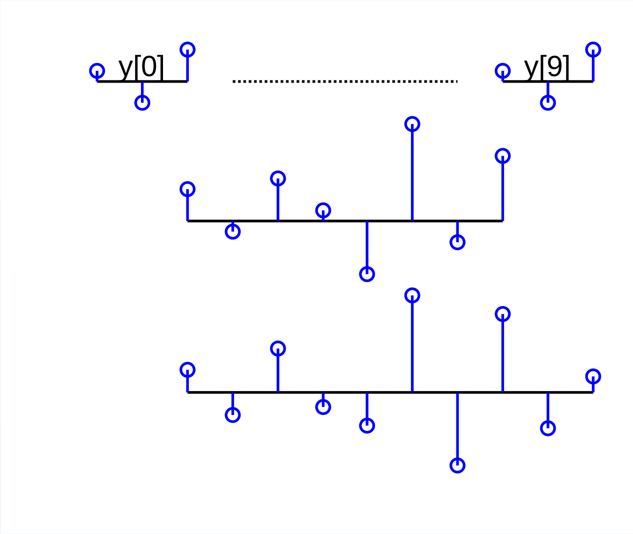
$y[n] = x[n] * h[n]$: convolve $x[0 : N-1]$ with $h[0 : M-1]$



$*$



$\rightarrow$



Convolution sum:

$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

$y[n]$ is only non-zero in the range

$$0 \le n \le M + N - 2$$

Thus $y[n]$ has only

$M + N - 1$ non-zero values

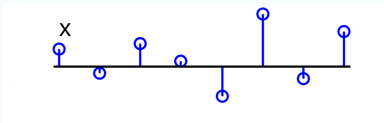
$N = 8$, $M = 3$
$M + N - 1 = 10$

Algebraically:

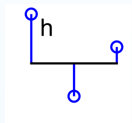
$$x[n-r] \ne 0 \Rightarrow 0 \le n - r \le N - 1$$
$$\Rightarrow n + 1 - N \le r \le n$$

Convolution Complexity

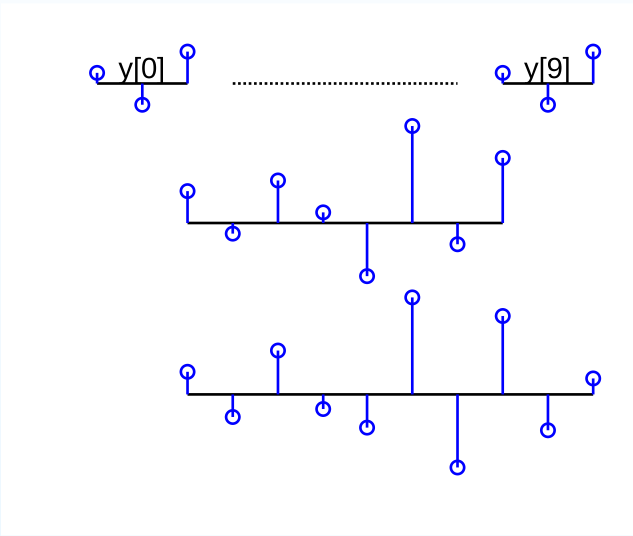
$y[n] = x[n] * h[n]$: convolve $x[0:N-1]$ with $h[0:M-1]$



$*$



$\rightarrow$



Convolution sum:

$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

$y[n]$ is only non-zero in the range

$$0 \le n \le M+N-2$$

Thus $y[n]$ has only

$M+N-1$ non-zero values

$N=8$, $M=3$
$M+N-1=10$

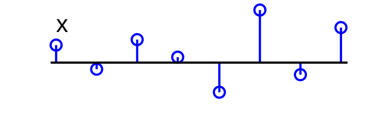
Algebraically:

$$x[n-r] \neq 0 \Rightarrow 0 \le n-r \le N-1$$
$$\Rightarrow n+1-N \le r \le n$$

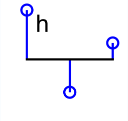
Hence: $y[n] = \sum_{r=\max(0,n+1-N)}^{\min(M-1,n))} h[r]x[n-r]$

Convolution Complexity

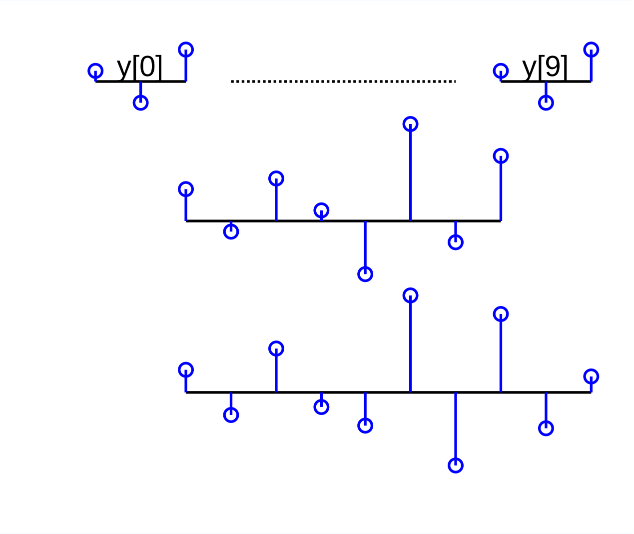
$y[n] = x[n] * h[n]$: convolve $x[0 : N-1]$ with $h[0 : M-1]$



$*$



$\rightarrow$



Convolution sum:

$$y[n] = \sum_{r=0}^{M-1} h[r]x[n-r]$$

$y[n]$ is only non-zero in the range

$$0 \le n \le M+N-2$$

Thus $y[n]$ has only

$M+N-1$ non-zero values

$N = 8$, $M = 3$
$M + N - 1 = 10$

Algebraically:

$$x[n-r] \neq 0 \Rightarrow 0 \le n-r \le N-1$$
$$\Rightarrow n+1-N \le r \le n$$

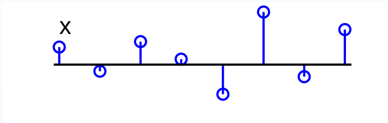
Hence: $y[n] = \sum_{r=\max(0,n+1-N)}^{\min(M-1,n))} h[r]x[n-r]$

We must multiply each $h[n]$ by each $x[n]$ and add them to a total

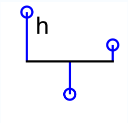
$\Rightarrow$ total arithmetic complexity ($\times$ or $+$ operations) $\approx 2MN$

Circular Convolution

$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0 : N-1]$ with $h[0 : M-1]$

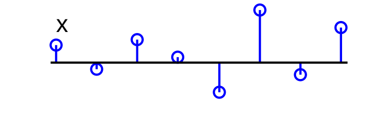


$\circledast_N$

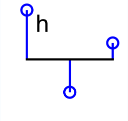


Circular Convolution

$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$

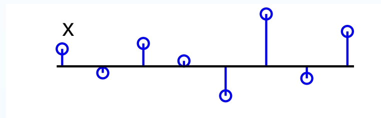


Convolution sum:

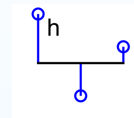
$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

Circular Convolution

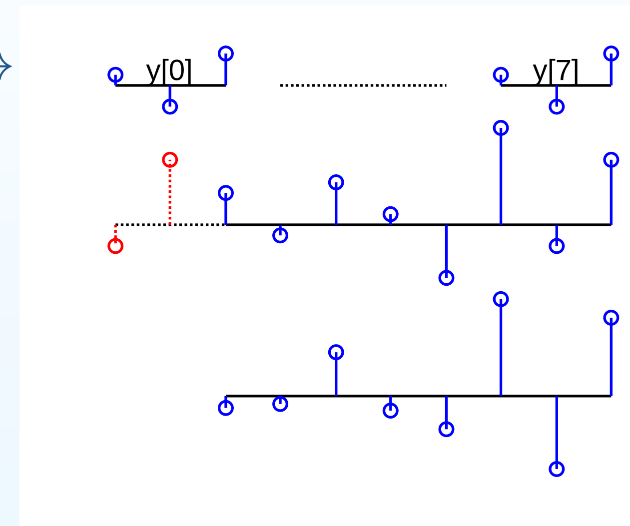
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$



$\rightarrow$



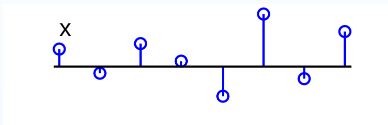
Convolution sum:

$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

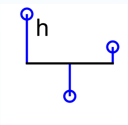
$N = 8, M = 3$

Circular Convolution

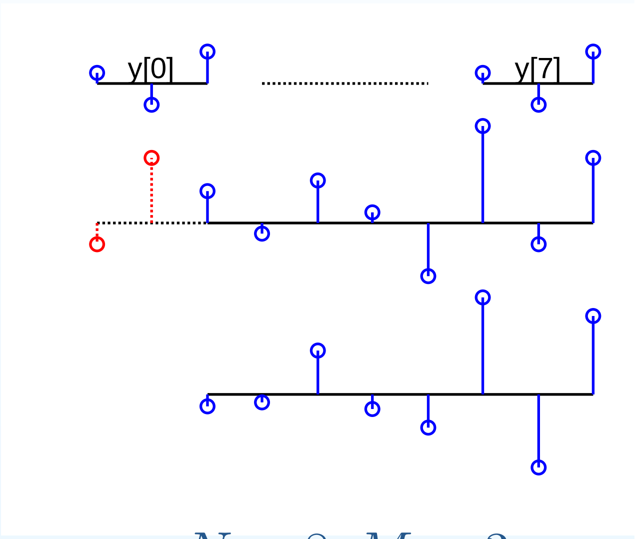
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$



$\rightarrow$



Convolution sum:

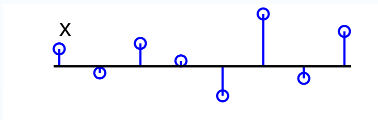
$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

$y_{\circledast_N}[n]$ has period N

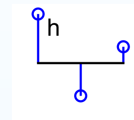
$N = 8$, $M = 3$

Circular Convolution

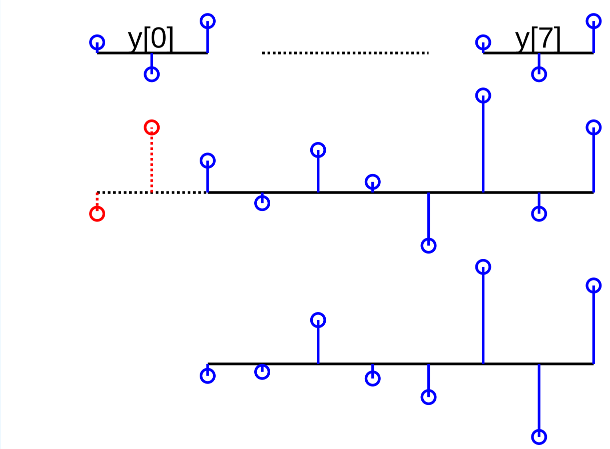
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$



$\rightarrow$



Convolution sum:

$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r] x[(n-r)_{\text{mod } N}]$$

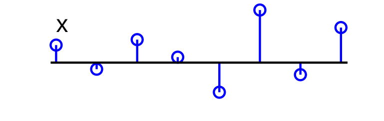
$y_{\circledast_N}[n]$ has period N

$\Rightarrow y_{\circledast_N}[n]$ has N distinct values

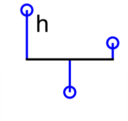
$N = 8, M = 3$

Circular Convolution

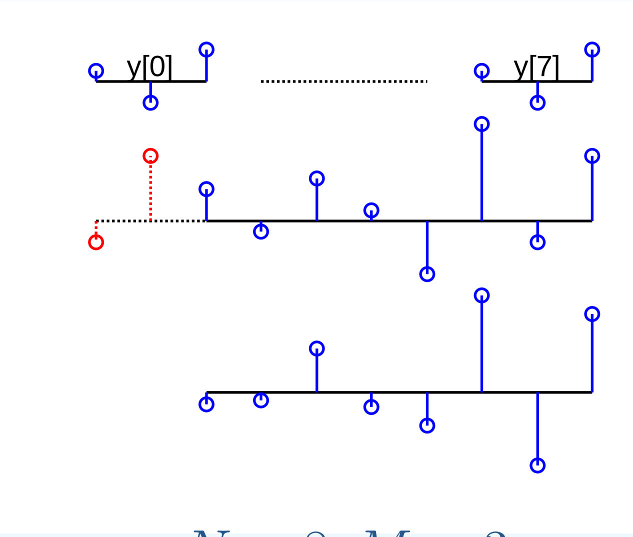
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0 : N-1]$ with $h[0 : M-1]$



$\circledast_N$



$\rightarrow$



$N = 8, M = 3$

Convolution sum:

$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

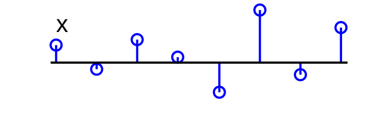
$y_{\circledast_N}[n]$ has period N

$\Rightarrow y_{\circledast_N}[n]$ has N distinct values

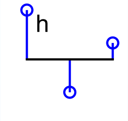
- Only the first $M-1$ values are affected by the circular repetition:

Circular Convolution

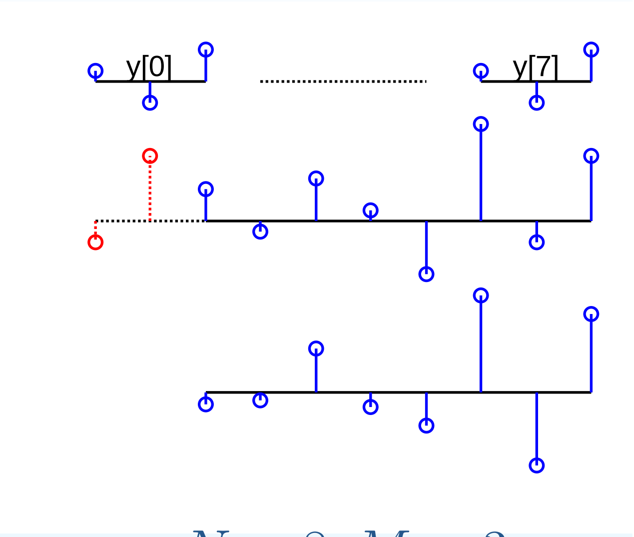
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$



$\rightarrow$



Convolution sum:

$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

$y_{\circledast_N}[n]$ has period N

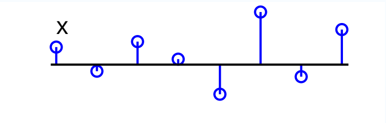
$\Rightarrow y_{\circledast_N}[n]$ has N distinct values

$N = 8, M = 3$

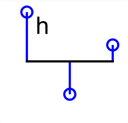
- Only the first $M-1$ values are affected by the circular repetition:
  $y_{\circledast_N}[n] = y[n]$ for $M-1 \le n \le N-1$

Circular Convolution

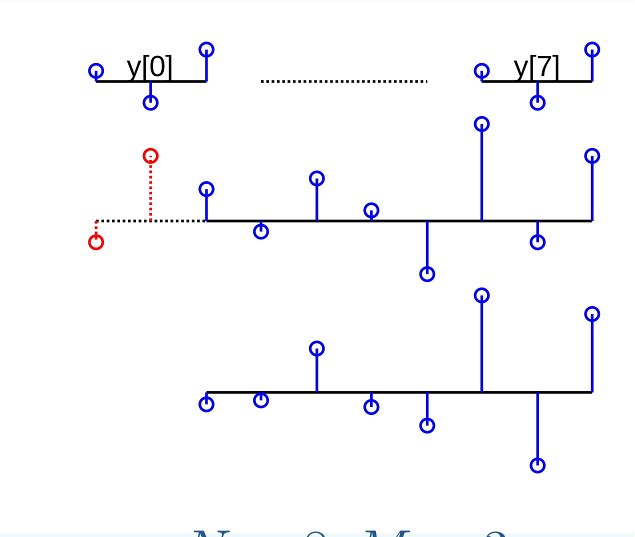
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$



$\rightarrow$



$N = 8$, $M = 3$

Convolution sum:

$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

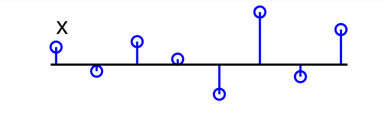
$y_{\circledast_N}[n]$ has period N

$\Rightarrow y_{\circledast_N}[n]$ has N distinct values

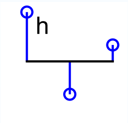
- Only the first $M-1$ values are affected by the circular repetition:
  $$y_{\circledast_N}[n] = y[n] \text{ for } M-1 \le n \le N-1$$
- If we append $M-1$ zeros (or more) onto $x[n]$, then the circular repetition has no effect at all and:
  $$y_{\circledast_{N+M-1}}[n] = y[n] \text{ for } 0 \le n \le N+M-2$$

Circular Convolution

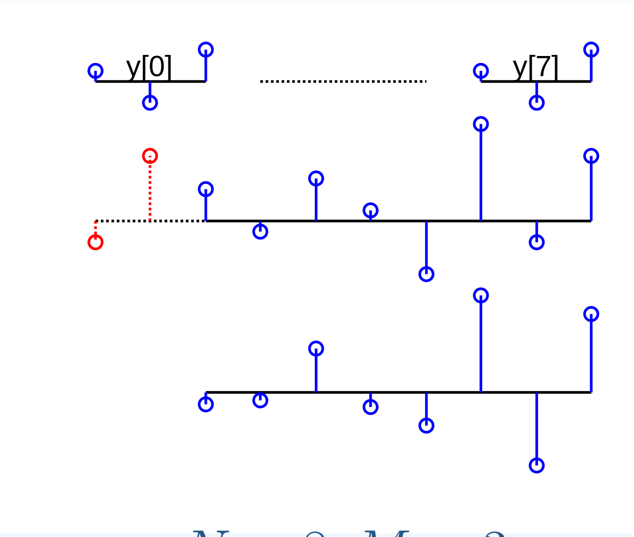
$y_{\circledast}[n] = x[n] \circledast_N h[n]$: circ convolve $x[0:N-1]$ with $h[0:M-1]$



$\circledast_N$



$\rightarrow$



$N = 8$, $M = 3$

Convolution sum:

$$y_{\circledast_N}[n] = \sum_{r=0}^{M-1} h[r]x[(n-r)_{\text{mod } N}]$$

$y_{\circledast_N}[n]$ has period N

$\Rightarrow y_{\circledast_N}[n]$ has N distinct values

- Only the first $M-1$ values are affected by the circular repetition:
  $y_{\circledast_N}[n] = y[n]$ for $M-1 \le n \le N-1$
- If we append $M-1$ zeros (or more) onto $x[n]$, then the circular repetition has no effect at all and:
  $y_{\circledast_{N+M-1}}[n] = y[n]$ for $0 \le n \le N+M-2$

Circular convolution is a necessary evil in exchange for using the DFT

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

DFT or IDFT take $4L \log_2 L$ operations if L is a power of 2

(or $16L \log_2 L$ if not).

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

DFT or IDFT take $4L \log_2 L$ operations if L is a power of 2
(or $16L \log_2 L$ if not).
Total operations: $\approx 12L \log_2 L \approx 12\,(M+N)\log_2 (M+N)$

Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

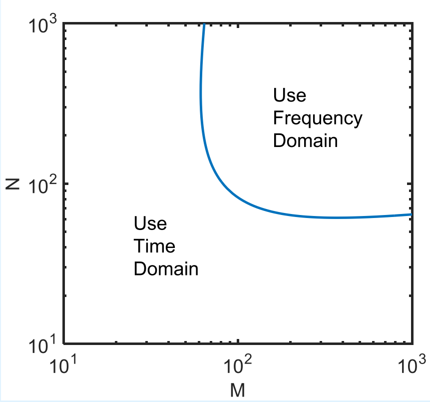
(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

DFT or IDFT take $4L\log_2 L$ operations if L is a power of 2
(or $16L\log_2 L$ if not).

Total operations: $\approx 12L\log_2 L \approx 12(M+N)\log_2(M+N)$

Beneficial if both M and N are $>\sim 70$.



Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

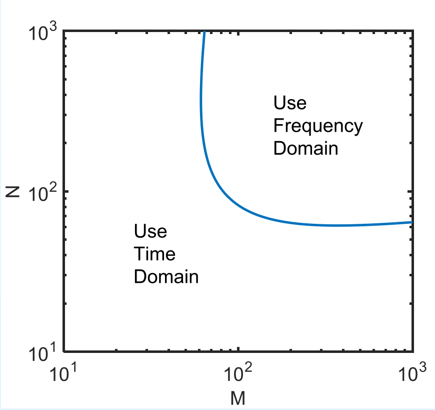
DFT or IDFT take $4L\log_2 L$ operations if L is a power of 2

(or $16L\log_2 L$ if not).

Total operations: $\approx 12L\log_2 L \approx 12\,(M+N)\log_2 (M+N)$

Beneficial if both M and N are $>\sim 70$.

Example: $M = 10^3$, $N = 10^4$:



Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

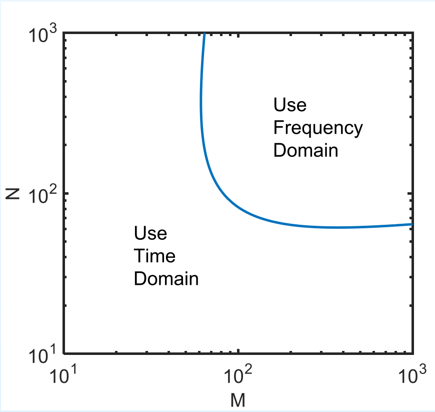
DFT or IDFT take $4L\log_2 L$ operations if L is a power of 2
(or $16L\log_2 L$ if not).

Total operations: $\approx 12L\log_2 L \approx 12\,(M+N)\log_2\,(M+N)$

Beneficial if both M and N are $>\sim 70$.

Example: $M = 10^3$, $N = 10^4$:

Direct: $2MN = 2 \times 10^7$



Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

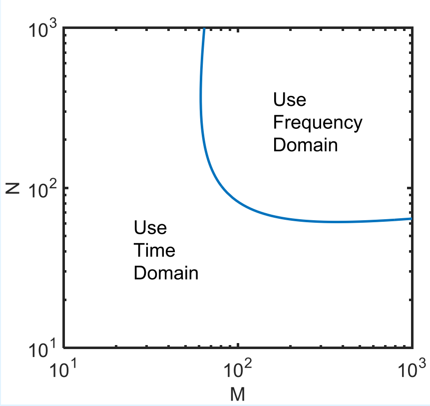
(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

DFT or IDFT take $4L \log_2 L$ operations if L is a power of 2
(or $16L \log_2 L$ if not).
Total operations: $\approx 12L \log_2 L \approx 12\,(M + N) \log_2 (M + N)$
Beneficial if both M and N are $>\sim 70$.

Example: $M = 10^3$, $N = 10^4$:
Direct: $2MN = 2 \times 10^7$
with DFT: $= 1.8 \times 10^6$ ☺



Frequency-domain convolution

Idea: Use DFT to perform circular convolution - less computation

(1) Choose $L \geq M + N - 1$ (normally round up to a power of 2)

(2) Zero pad $x[n]$ and $h[n]$ to give sequences of length L: $\tilde{x}[n]$ and $\tilde{h}[n]$

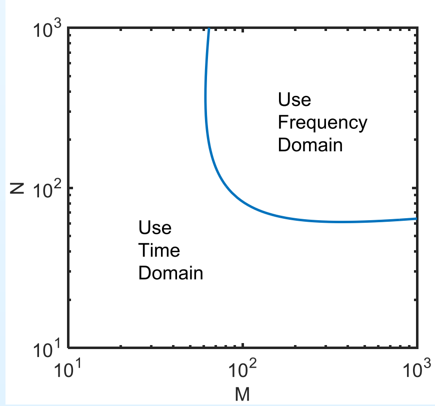
(3) Use DFT: $\tilde{y}[n] = \mathcal{F}^{-1}(\tilde{X}[k]\tilde{H}[k]) = \tilde{x}[n] \circledast_L \tilde{h}[n]$

(4) $y[n] = \tilde{y}[n]$ for $0 \leq n \leq M + N - 2$.

Arithmetic Complexity:

DFT or IDFT take $4L\log_2 L$ operations if L is a power of 2
(or $16L\log_2 L$ if not).
Total operations: $\approx 12L\log_2 L \approx 12\,(M+N)\log_2\,(M+N)$
Beneficial if both M and N are $>\sim 70$.

Example: $M = 10^3$, $N = 10^4$:
Direct: $2MN = 2 \times 10^7$
with DFT: $= 1.8 \times 10^6$ ☺



But: (a) DFT may be very long if N is large
(b) No outputs until all $x[n]$ has been input.

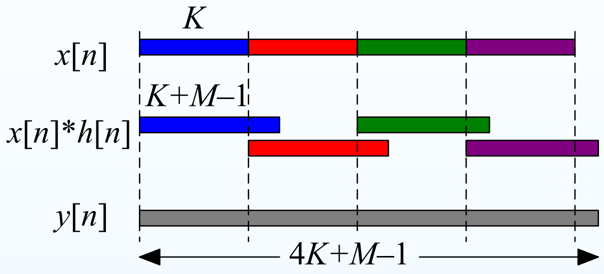
Overlap Add

If N is very large:

Overlap Add

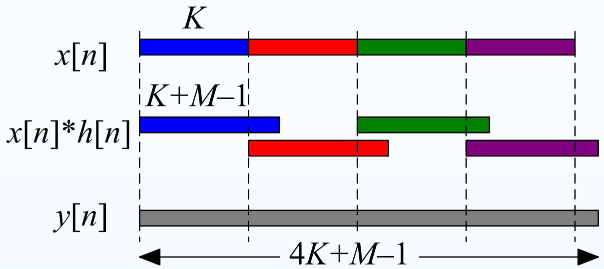
If N is very large:

(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K



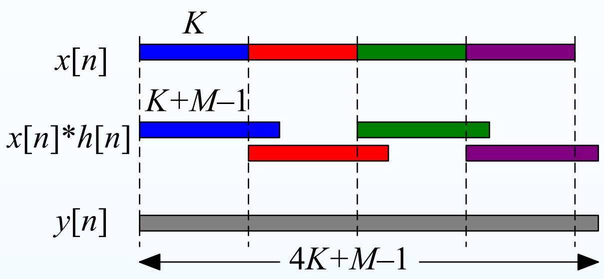
Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$



Overlap Add

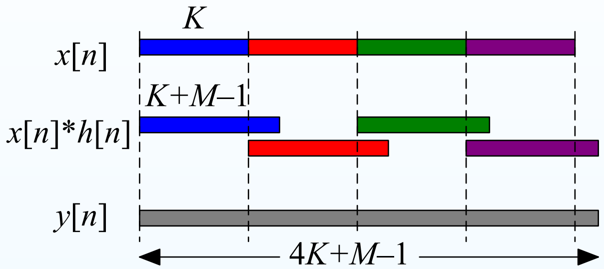
If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$



Each output chunk is of length $K + M - 1$ and overlaps the next chunk

Overlap Add

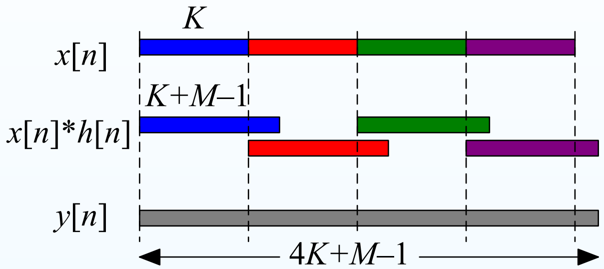
If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



Each output chunk is of length $K + M - 1$ and overlaps the next chunk

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results

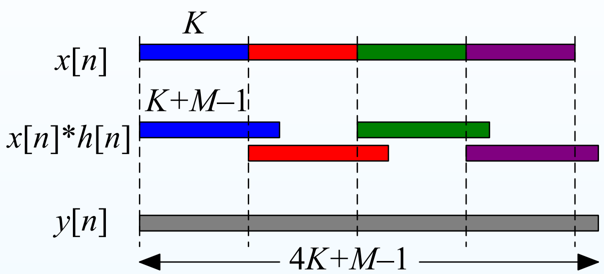


Each output chunk is of length $K+M-1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



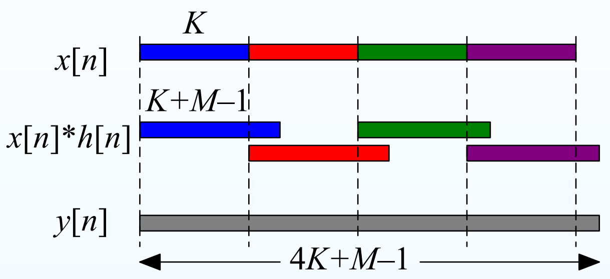
Each output chunk is of length $K+M-1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\left(M+K\right)\log_2\left(M+K\right)$

Computational saving if $\approx 100 < M \ll K \ll N$

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



Each output chunk is of length $K+M-1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$

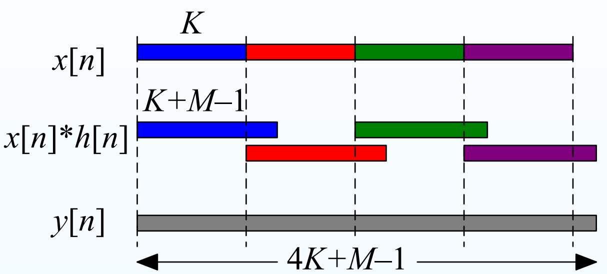
Computational saving if $\approx 100 < M \ll K \ll N$

Example: $M = 500$, $K = 10^4$, $N = 10^7$

Direct: $2MN = 10^{10}$

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



Each output chunk is of length $K + M - 1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$

Computational saving if $\approx 100 < M \ll K \ll N$

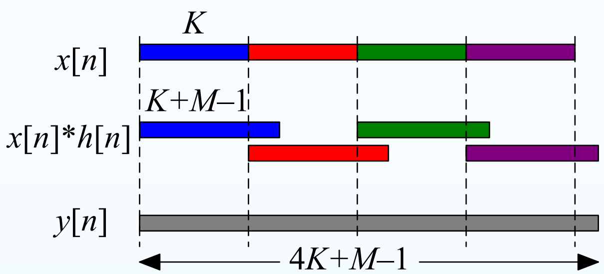
Example: $M = 500$, $K = 10^4$, $N = 10^7$

Direct: $2MN = 10^{10}$

single DFT: $12\,(M+N)\log_2(M+N) = 2.8 \times 10^9$

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



Each output chunk is of length $K+M-1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$

Computational saving if $\approx 100 < M \ll K \ll N$

Example: $M = 500$, $K = 10^4$, $N = 10^7$

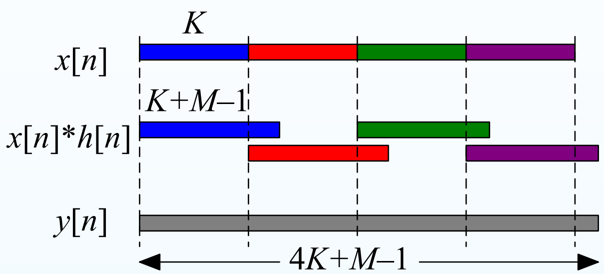
Direct: $2MN = 10^{10}$

single DFT: $12\,(M+N)\log_2(M+N) = 2.8 \times 10^9$

overlap-add: $\frac{N}{K} \times 8\,(M+K)\log_2(M+K) = 1.1 \times 10^9$ ☺

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



Each output chunk is of length $K + M - 1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$

Computational saving if $\approx 100 < M \ll K \ll N$

Example: $M = 500$, $K = 10^4$, $N = 10^7$

Direct: $2MN = 10^{10}$

single DFT: $12\,(M+N)\log_2(M+N) = 2.8 \times 10^9$

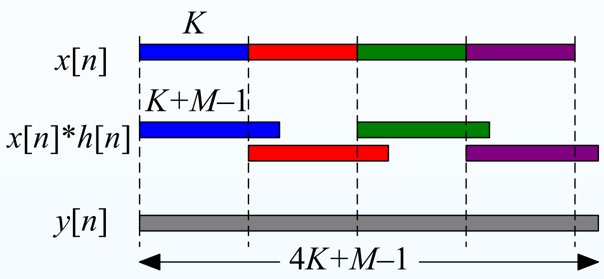
overlap-add: $\frac{N}{K} \times 8\,(M+K)\log_2(M+K) = 1.1 \times 10^9$ ☺

Other advantages:

(a) Shorter DFT

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



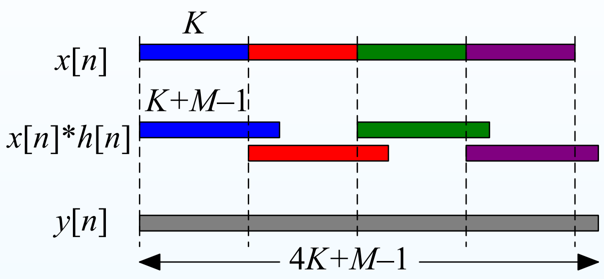
Each output chunk is of length $K + M - 1$ and overlaps the next chunk
Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$
Computational saving if $\approx 100 < M \ll K \ll N$

Example: $M = 500$, $K = 10^4$, $N = 10^7$
Direct: $2MN = 10^{10}$
single DFT: $12\,(M+N)\log_2(M+N) = 2.8 \times 10^9$
overlap-add: $\frac{N}{K} \times 8\,(M+K)\log_2(M+K) = 1.1 \times 10^9$ ☺

Other advantages:
(a) Shorter DFT
(b) Can cope with $N = \infty$

Overlap Add

If N is very large:
(1) chop $x[n]$ into $\frac{N}{K}$ chunks of length K
(2) convolve each chunk with $h[n]$
(3) add up the results



Each output chunk is of length $K + M - 1$ and overlaps the next chunk

Operations: $\approx \frac{N}{K} \times 8\,(M+K)\log_2(M+K)$

Computational saving if $\approx 100 < M \ll K \ll N$

Example: $M = 500$, $K = 10^4$, $N = 10^7$

Direct: $2MN = 10^{10}$

single DFT: $12\,(M+N)\log_2(M+N) = 2.8 \times 10^9$

overlap-add: $\frac{N}{K} \times 8\,(M+K)\log_2(M+K) = 1.1 \times 10^9$ ☺

Other advantages:

(a) Shorter DFT

(b) Can cope with $N = \infty$

(c) Can calculate $y[0]$ as soon as $x[K-1]$ has been read:

algorithmic delay = $K - 1$ samples

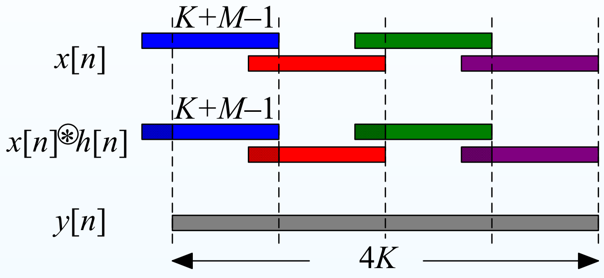
Overlap Save

Alternative method:

Overlap Save

Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K+M-1$

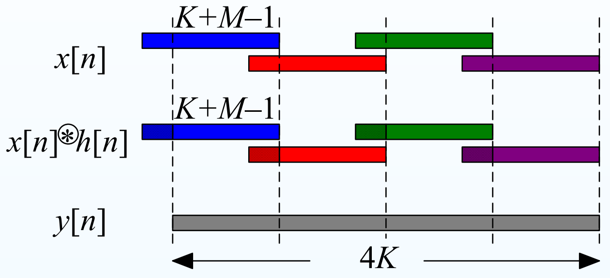


Overlap Save

Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K+M-1$

(2) $\circledast_{K+M-1}$ each chunk with $h[n]$

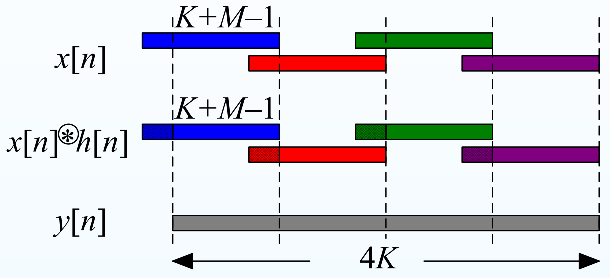


Overlap Save

Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K+M-1$

(2) $\circledast_{K+M-1}$ each chunk with $h[n]$



The first $M-1$ points of each output chunk are invalid

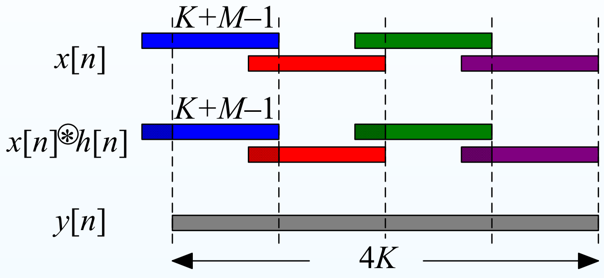
Overlap Save

Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K+M-1$

(2) $\circledast_{K+M-1}$ each chunk with $h[n]$

(3) discard first $M-1$ from each chunk



The first $M-1$ points of each output chunk are invalid

Overlap Save

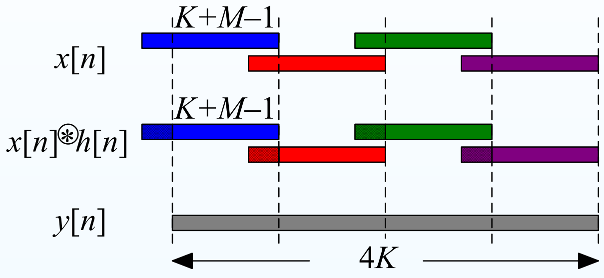
Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K + M - 1$

(2) $\circledast_{K+M-1}$ each chunk with $h[n]$

(3) discard first $M - 1$ from each chunk

(4) concatenate to make $y[n]$



The first $M - 1$ points of each output chunk are invalid

Overlap Save

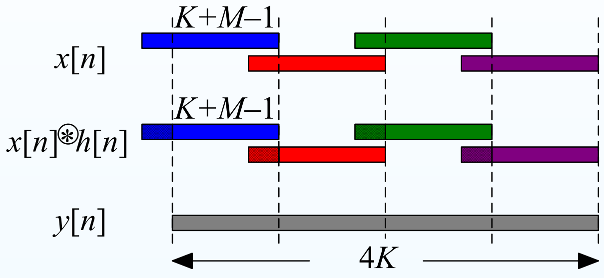
Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K+M-1$

(2) $\circledast_{K+M-1}$ each chunk with $h[n]$

(3) discard first $M-1$ from each chunk

(4) concatenate to make $y[n]$



The first $M-1$ points of each output chunk are invalid

Operations: slightly less than overlap-add because no addition needed to create $y[n]$

Overlap Save

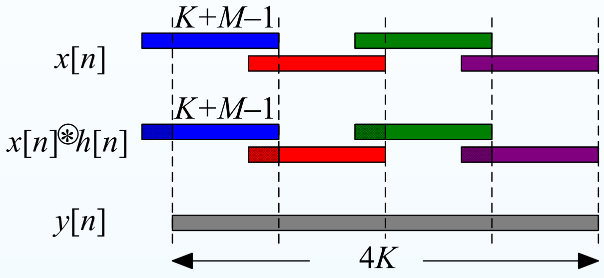
Alternative method:

(1) chop $x[n]$ into $\frac{N}{K}$ overlapping chunks of length $K + M - 1$

(2) $\circledast_{K+M-1}$ each chunk with $h[n]$

(3) discard first $M - 1$ from each chunk

(4) concatenate to make $y[n]$



The first $M - 1$ points of each output chunk are invalid

Operations: slightly less than overlap-add because no addition needed to create $y[n]$

Advantages: same as overlap add

Strangely, rather less popular than overlap-add

Summary

- LTI systems: impulse response, frequency response, group delay

Summary

- LTI systems: impulse response, frequency response, group delay
- BIBO stable, Causal, Passive, Lossless systems

Summary

- LTI systems: impulse response, frequency response, group delay
- BIBO stable, Causal, Passive, Lossless systems
- Convolution and circular convolution properties

Summary

- LTI systems: impulse response, frequency response, group delay
- BIBO stable, Causal, Passive, Lossless systems
- Convolution and circular convolution properties
- Efficient methods for convolution
  - single DFT
  - overlap-add and overlap-save

Summary

- LTI systems: impulse response, frequency response, group delay
- BIBO stable, Causal, Passive, Lossless systems
- Convolution and circular convolution properties
- Efficient methods for convolution
  - single DFT
  - overlap-add and overlap-save

For further details see Mitra: 4 & 5.

MATLAB routines

fftfilt	Convolution using overlap add
$x[n] \circledast y[n]$	real(ifft(fft(x).*fft(y)))