Multi-Rate DSP

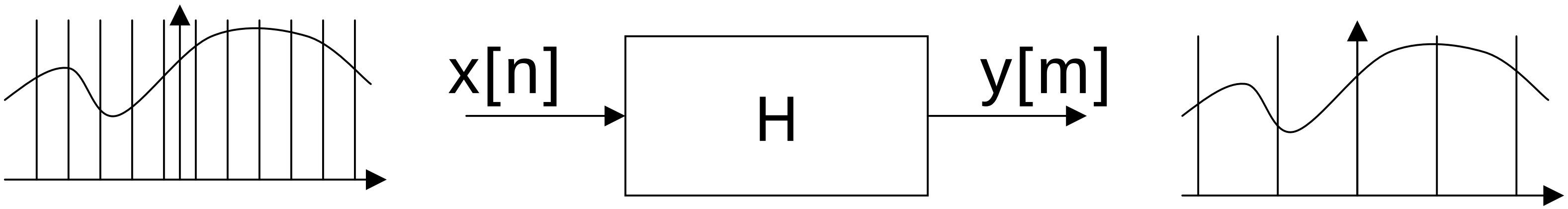
Fundamentals:

Decimation, Interpolation,

Polyphase Decomposition

Multi-Rate DSP

- What? Sampling period at output (T_m) $\neq$ at input (T_n)



- Applications:
  - Conversion between different media or different audio/video formats (CD, DAB, DAT, DVD-R, DVD-R+, BlueRay, etc.): different rates for different bandwidths, etc.

Multi-Rate DSP

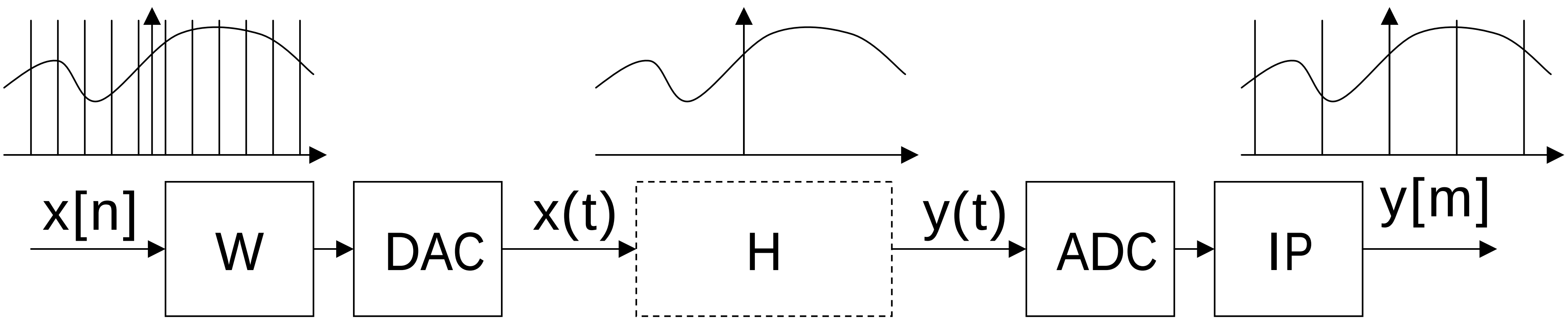
- Why sample rate conversion? (I)
  - *Compatibility:* convert sample frequencies of different stds.
    - E.g. CD (44.1 kHz) $\leftrightarrow$ DAB (32 kHz) $\leftrightarrow$ DAT (48 kHz)
    - Composite video signals: NTSC (14.818 MHz) $\leftrightarrow$ PAL (17.734 MHz)
    - TV Luminance (13.5 MHz) $\leftrightarrow$ Colour difference (6.75 MHz)
  - *Efficiency:* easier data processing (computationally more efficient), less storage, lower transmission speed, ...
    - E.g. reduce computational overheads of oversampled signal: compression by retaining only essential data
    - Narrowband FIR filters: more efficient than large # coefficients of analogue filters (steep skirts!)

Multi-Rate DSP

- *All-digital:* Change sample frequency in an efficient manner
  - Avoid use of analogue filters (bulky, difficult to realize higher-order)
- *Cost:* Avoid need for expensive analogue anti-aliasing filters
  - Oversampling ($F_s >> 2F_B$) allows for simpler anti-aliasing filters
- Allows for exploiting *high-resolution ADCs:*
  - Less expensive overall
  - Less quantization noise (because spread over wider frequency band & lower in-band) & less aliasing
  - Simpler modulation methods (delta-sigma) possible; no S&H amp

Multi-Rate DSP Methods

- Methods for rate conversion
  - Method A: convert to intermediate analogue signal

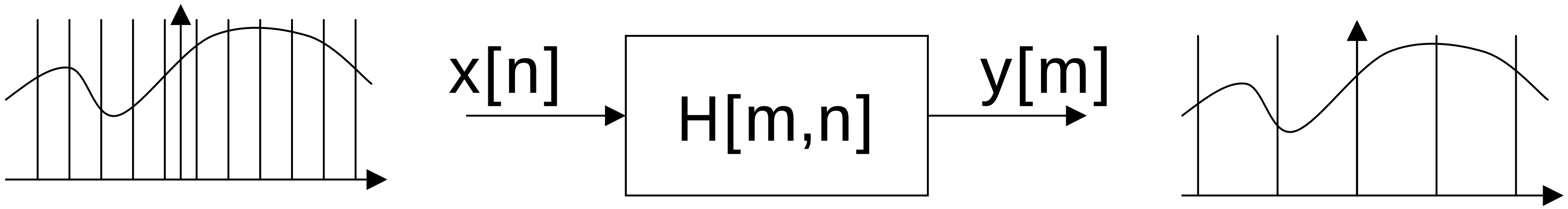


- +: simple concept, low computation (fast)
- +: high ratio of pre- vs. post-sampling rates is possible, in 1 stage

- -: extra distortion (W, DAC & IP)
- -: extra quantization error due to extra ADC

Multi-Rate DSP Methods

- Method B: direct sampling conversion within digital domain



  - +: no additional distortion (only single ADC+DAC pair)
  - -: for rational (fractional) rates only (except Farrow)
  - -: high non-integer ratios require multi-stage approach

Reconstruct & Re-sample

- A(1) Reconstruct continuous trace from samples
  - In continuous time: interpolated continuous-time signal *y* constructed from samples *x*:

$$y(t) = \sum_{n=-\infty}^{+\infty} x(nT_x)h(t - nT_x)$$

  - Provided

$$x(t) \leftrightarrow X(f), \qquad -F_x/2 < f < F_x/2 \text{ (bandlimited)}$$

$$h(t) = \text{sinc}\left(\frac{\pi t}{T_x}\right) \leftrightarrow H(f) = T_x, \quad -F_x/2 < f < F_x/2 \quad \text{(IP pulses)}$$

then $y(t) = x(t)$ (no ISI, but this $h(t)$ is not realisable!)

  - In practice: <u>finite</u> summation & realisable $h(t) \rightarrow$ distortion

Reconstruct & Re-sample

- A(2) Re-sample reconstructed signal at new times $t=mT_y$, i.e.,

$$y(t=mT_y)=\sum_{n=-\infty}^{+\infty}x(nT_x)h(mT_y-nT_x)$$

  - If $F_y > F_x$: accurate reconstruction
  - If $F_y < F_x$: $x(t)$ must be limited to $|f|<F_y/2$ to avoid aliasing (signal ambiguity) due to undersampling of original signal
  - If $T_y = T_x$: $y(t)$ is discrete convolution (LTI)
- Thus, $x(nT_x)\to y(t)\to y(mT_y)$, $y(t)\cong x(t)$

Reconstruct & Re-sample

- Re-arranging $y(mT_y)$:

$$y(mT_y) = \sum_{n=-\infty}^{+\infty} x(nT_x)h((k_m + \delta_m - n)T_x) = \sum_{n=-\infty}^{+\infty} x((k_m - n)T_x)h(nT_x + \delta_m T_x)$$

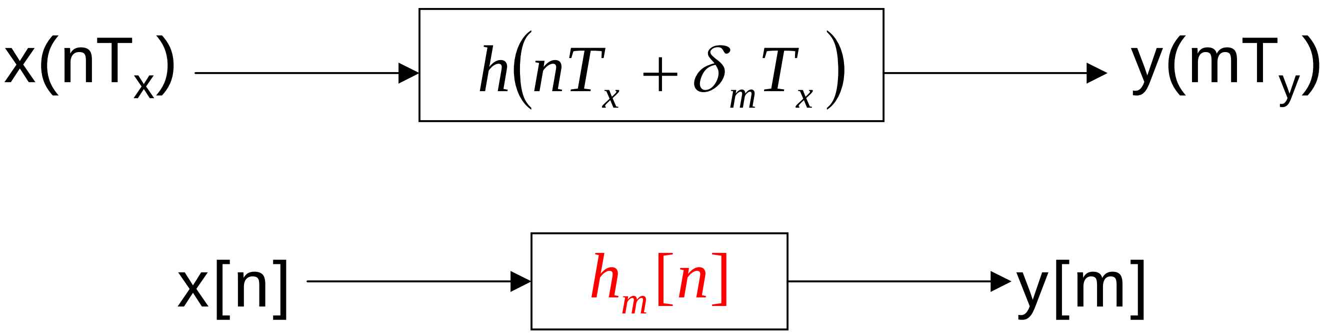
where $k_m = \left\lfloor \frac{mT_y}{T_x} \right\rfloor$ $\qquad \delta_m = \frac{mT_y}{T_x} - k_m$

- Recipe for multi-rate conversion via reconstruction:
  - (1) shift $h(.)$ such that $h(\delta_m T_x)$ now falls at $t = mT_y$
  - (2) adjust value of $x(nT_x)$ to shifted value $x(k_m T_x - nT_x)$
- Issue: new ∞ set of cffs. of $h(.)$ needed for each new m !

Reconstruct & Re-sample

- System representation: use form

$$y(mT_y) = \sum_{n=-\infty}^{+\infty} x(nT_x)\, h\big((k_m + \delta_m - n)T_x\big)$$

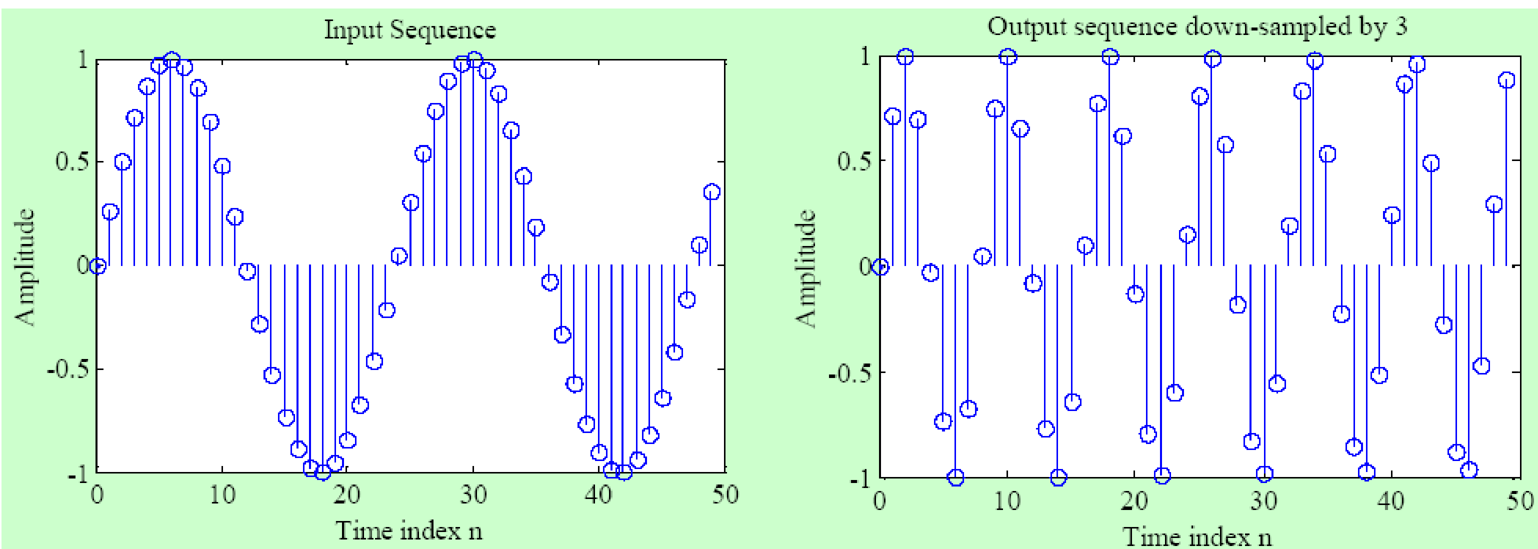


LTV filter (nonstationary system)

Direct Conversion (All-Digital)

- Two basic operations:
  - Decimation (down-sampling): $T_y = DT_x, \qquad (D > 1)$

$$y(mT_y) = \sum_{n=-\infty}^{+\infty} x(nT_x)h((mD-n)T_x)$$



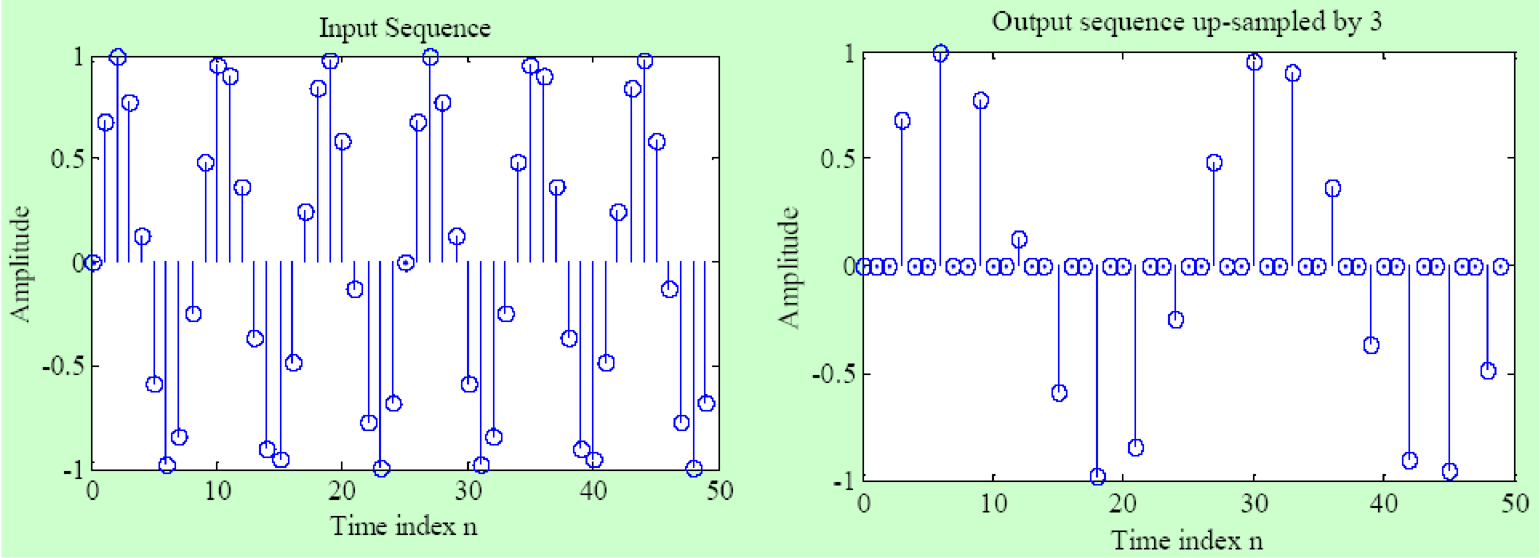
Decimation reduces data size & processing time

"compression" of time axis & subsampled

Direct Conversion (All-Digital)

- Two basic operations:
  - Interpolation (up-sampling): $T_y = T_x / I, \quad (I > 1)$

$$y(mT_y) = \sum_{n=-\infty}^{+\infty} x(nT_x)\, h\big((m/I - n)T_x\big)$$



"dilation" of time axis & padding with zeroes

Down- and Up-Sampling in z-Domain

- Down-sampling: $x_d[n] = x[2n], \quad n = 0, \pm 2, \pm 4, \ldots$
  - E.g. D=2:

$$X_d(z) = \sum_{n=-\infty}^{+\infty} x_d[n]\, z^{-n} = \sum_{n=-\infty}^{+\infty} x[2n]\, z^{-n} = \sum_{m=-\infty}^{+\infty} x[m]\, z^{-m/2} = \frac{1}{D}\sum_{k=0}^{D-1} H\left(e^{-\frac{j2\pi k}{D}} z^{\frac{1}{D}}\right) X\left(e^{-\frac{j2\pi k}{D}} z^{\frac{1}{D}}\right), \; D = 2$$

(multi-valued complex square root)

- Up-sampling:
  - E.g. I=2: $x_u[n] = x[n/2], \quad n = 0, \pm 2, \pm 4, \ldots$

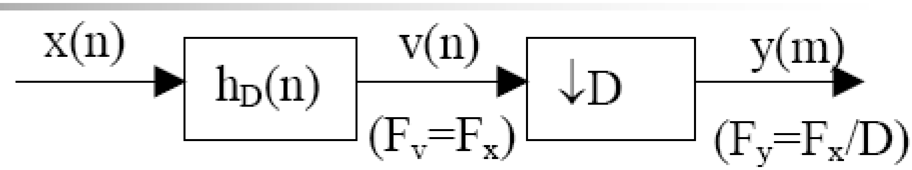
$$X_u(z) = \sum_{n=-\infty}^{+\infty} x_u[n]\, z^{-n} = \sum_{n=-\infty}^{+\infty} x[n/2]\, z^{-n} = \sum_{m=-\infty}^{+\infty} x[m]\, z^{-2m} = X\left(z^2\right)$$

Decimation (I)

- Down-sampling by integer factor D :
  - subsampling: $$x_d[n] = x[nD], \qquad n = 0, \pm D, \pm 2D, \ldots$$
  - sampling theorem: $$-F_X/2 \leq F \leq +F_X/2 \quad \Leftrightarrow \quad -\pi \leq \omega \leq +\pi$$
  - requires *anti-aliasing LP pre-filter*, because lower output sampling rate (missing samples may cause ambiguity): $$-F_x/(2D) \leq F \leq +F_x/(2D) \Leftrightarrow -\pi/D \leq \omega_x \leq +\pi/D$$
    - LPF produces potentially distorted output signal $V(\omega_x)$!

Decimation (II)

- Time-variant filtering:



$$v(n) = \sum_{k=0}^{+\infty} h_D(k)\, x(n-k),$$

$$y(m) = v(mD) = v(mD)\,\delta(mD) \equiv \sum_{k=0}^{+\infty} h_D(k)\, x(mD-k),$$

$$\delta(n) = \frac{1}{D}\sum_{k=0}^{D-1} \exp\left(\frac{j2\pi n}{D}k\right)$$

$$Y(z) = \sum_{m=-\infty}^{+\infty} v(m)\left[\frac{1}{D}\sum_{k=0}^{D-1}\exp\left(\frac{j2\pi\, m}{D}k\right)\right] z^{-\frac{m}{D}} = \frac{1}{D}\sum_{k=0}^{D-1} V\left(\exp\left(-\frac{j2\pi\, k}{D}\right) z^{\frac{1}{D}}\right), \quad z' = \exp\left(-\frac{j2\pi k}{D}\right) z^{\frac{1}{D}}$$

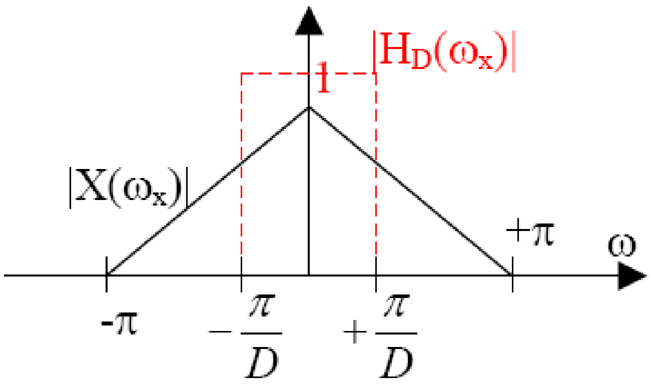
$$\Rightarrow \quad Y(z) = \frac{1}{D}\sum_{k=0}^{D-1} H_D\left(\exp\left(-\frac{j2\pi\, k}{D}\right) z^{\frac{1}{D}}\right) X\left(\exp\left(-\frac{j2\pi\, k}{D}\right) z^{\frac{1}{D}}\right)$$

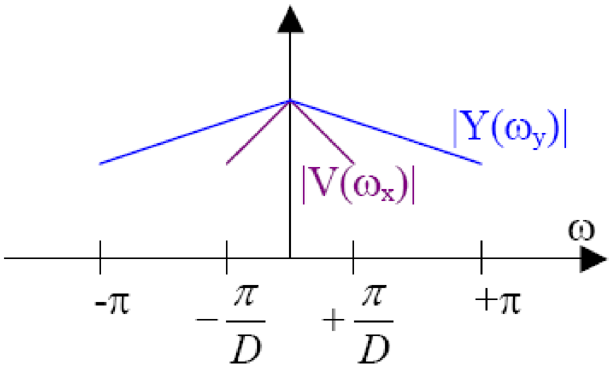
  - Anti-aliasing: retain $k=0$ term only
  - Filtered spectrum $V(\omega_x)$ is “stretched” in frequency by factor D:

$$z = e^{j\omega} \quad \Rightarrow \quad Y(\omega_y) = \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) X\left(\frac{\omega_y}{D}\right) = \frac{1}{D} X\left(\frac{\omega_y}{D}\right)$$

Decimation (III)

- Spectra:



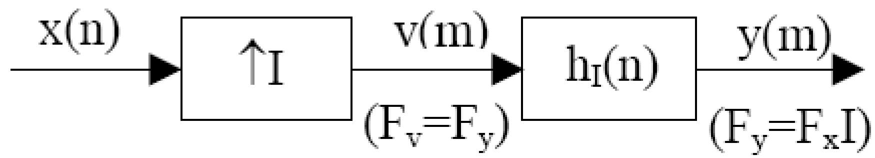


Interpolation (I)

- Up-sampling by integer factor *I* :
  - zero padding: $v(n) = x_u[n] = x[n/I], \quad n = 0, \pm I, \pm 2I, \ldots$
  - sampling theorem: $-F_X/2 \leq F \leq +F_X/2 \quad \Leftrightarrow \quad -\pi \leq \omega \leq +\pi$
  - no anti-aliasing pre-filtering because now higher output sampling rate (hence no potential ambiguity as a result of conversion)... $F_y = F_x I \geq F_x$
  - ... but need *anti-imaging post-filtering* to eliminate periodic images of $X(\omega)$: $-F_x I/2 \leq F \leq +F_x I/2, \; \omega_y = 2\pi F / F_X \Leftrightarrow -\pi / I \leq \omega_y \leq +\pi / I$

Interpolation (II)

- Time-variant filtering:



$$v(kI) = x(k),$$

$$V(z) = \sum_{n=-\infty}^{+\infty} v(n) z^{-n} = \sum_{n=-\infty}^{+\infty} x(n) z^{-nI} = X\left(z^{I}\right)$$

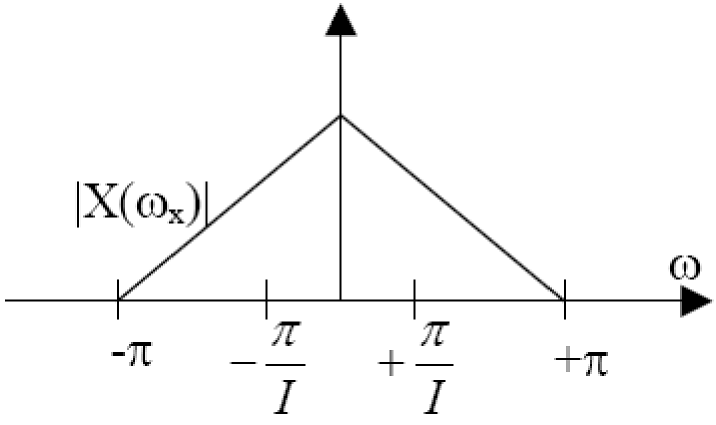
$$y(m) = \sum_{k=-\infty}^{+\infty} h_I(m-k)\, v(k) = \sum_{k=-\infty}^{+\infty} h_I(m-kI)\, x(k)$$

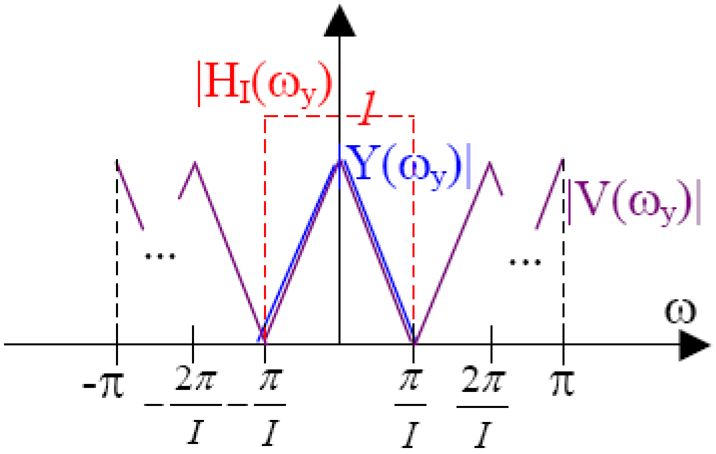
  - Spectrum $X(\omega_x)$ is “compressed” in frequency by factor I :

$$z = e^{j\omega} \qquad \Rightarrow \boxed{Y(\omega_y) = V(\omega_y) = X(\omega_y I) = X(\omega_x),} \qquad -\pi / I \le \omega_y \le +\pi / I$$

Interpolation (III)

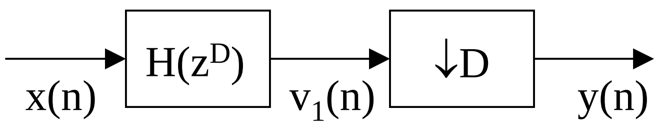
- Spectra:





Noble Identity for Decimation

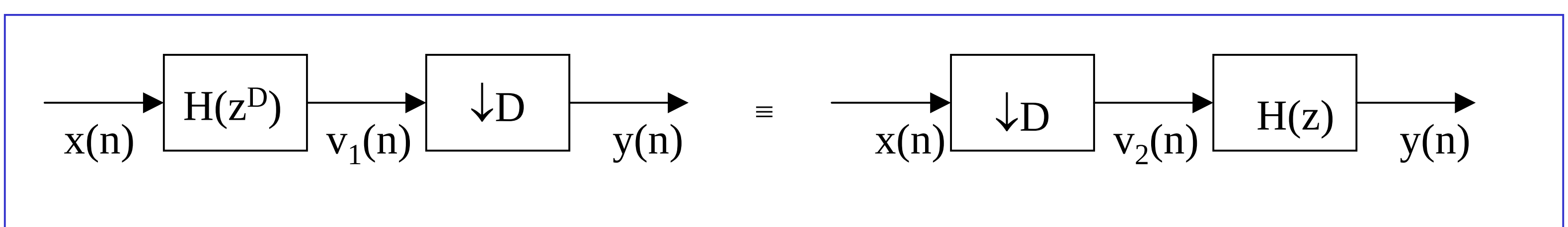
- LTI and LTV D-systems do <u>not</u> commute



$$V_1(z) = H\left(z^D\right)X(z)$$

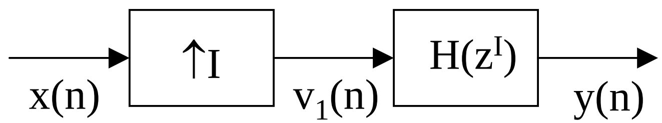
$$Y(z) = \frac{1}{D}\sum_{i=0}^{D-1} V_1\left(z^{1/D}W_D^i\right) = \frac{1}{D}\sum_{i=0}^{D-1} H\left(zW_D^{iD}\right)X\left(z^{1/D}W_D^i\right), \qquad W_D^{iD} = \left[\exp\left(-j2\pi/D\right)\right]^{iD} = 1$$

$$\Rightarrow \quad Y(z) = \frac{1}{D}H(z)\sum_{i=0}^{D-1} X\left(z^{1/D}W_D^i\right) = H(z)V_2(z)$$



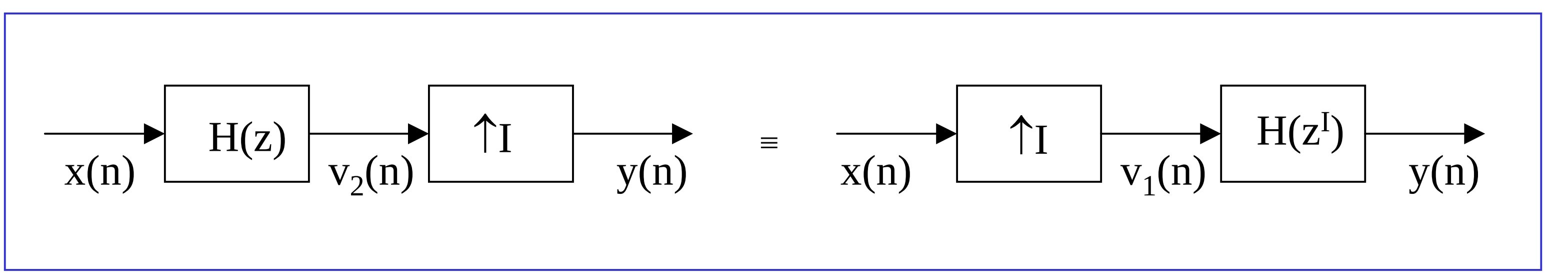
Noble Identity for Interpolation

- LTI and LTV I-systems do <u>not</u> commute



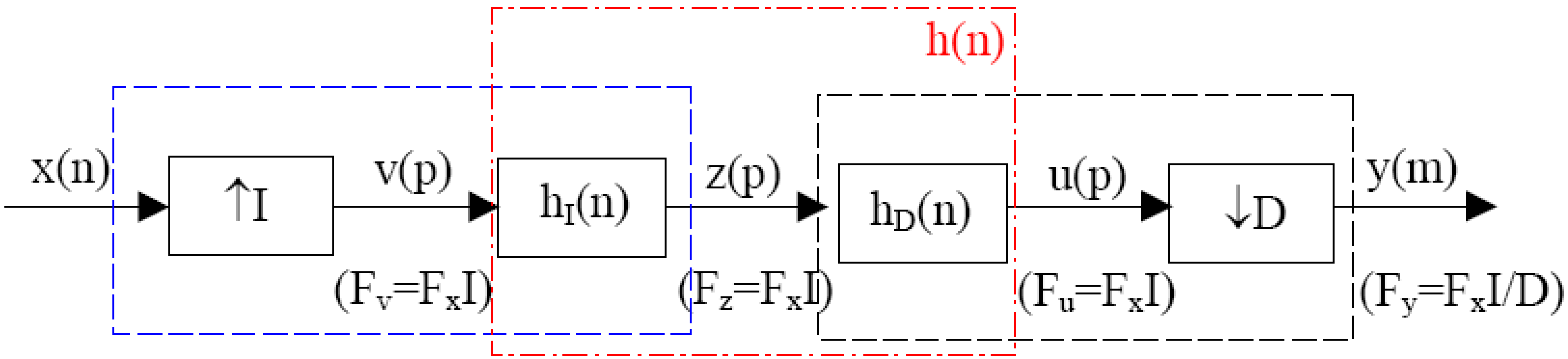
$$V_1(z) = X\left(z^I\right)$$

$$Y(z) = H\left(z^I\right)V_1(z) = H\left(z^I\right)X\left(z^I\right) = V_2\left(z^I\right)$$



Rational Multirate Conversion

- Order is important: first up-conversion (*I*), then down-conversion (*D*)
  - this order avoids distortion due to anti-aliasing / loss of data
  - yields rational (non-integer) conversion rate *I* / *D*
  - irrational factor requires analogue intermediate stage



combination possible because v, u have same sampling rate

Rational Multirate Conversion

- Anti-aliasing and anti-imaging filters can be combined into single *h*(*n*) according to most stringent bound:

$$H(\omega_z) = \begin{cases} I, & -\min\left(\frac{\pi}{D},\frac{\pi}{I}\right) \le \omega_z \le \min\left(\frac{\pi}{D},\frac{\pi}{I}\right) \\ 0, & |\omega_z| > \min\left(\frac{\pi}{D},\frac{\pi}{I}\right) \end{cases}$$

- Conversion output:

$$v(p) = x(p/I), \quad p = 0, \pm I, \pm 2I, \ldots$$

$$u(p) = h(p) * v(p) = \sum_{k=-\infty}^{+\infty} h(p-kI)\,x(k) \quad \Rightarrow \quad y(m) = u(mD) = \sum_{k=-\infty}^{+\infty} h(mD-kI)\,x(k)$$

After change of summation variable & manipulation:

$$y(m) = \sum_{k=-\infty}^{+\infty} h\left[(mD)\bmod I + kI\right] x\left(\left\lfloor \frac{mD}{I} \right\rfloor - k\right)$$

$$\lfloor mD/I \rfloor = \{mD - [(mD)\bmod I]\}/I$$

Rational Multirate Conversion

- Interpretation & properties of $y(m)/x(n)$:
  - LTV 2-D filtering: $h\left[(mD)\bmod I + kI\right] = f(k,m)$
  - Periodic in m with period I:

    for any integer α:

$$f(k,m+\alpha I) = h\left[(\alpha DI + mD)\bmod I + kI\right] = h\left[(mD)\bmod I + kI\right] = f(k,m)$$

- Spectrum: $U(\omega_u) = H(\omega_u)X(\omega_u I) = \begin{cases} I\,X(\omega_u I), & -\min\left(\frac{\pi}{D},\frac{\pi}{I}\right) \le \omega_u \le +\min\left(\frac{\pi}{D},\frac{\pi}{I}\right) \\ 0, & |\omega_u| > \min\left(\frac{\pi}{D},\frac{\pi}{I}\right) \end{cases}$

$$\Rightarrow \quad Y(\omega_y) = \frac{1}{D}\sum_{k=0}^{D-1} V\left(\frac{\omega_y - 2\pi k}{D}\right) = \begin{cases} \frac{I}{D} X\left(\frac{\omega_y}{D}\right), & -\min\left(\pi,\frac{D}{I}\pi\right)' \le \omega_y \le +\min\left(\pi,\frac{D}{I}\pi\right)' \\ 0, & |\omega_y| \ge +\min\left(\pi,\frac{D}{I}\pi\right) \end{cases}$$

Rational Multirate Conversion

- Example: CD- to DAT-format conversion

  - $I / D = F_y / F_x$ = 48 kHz / 44.1 kHz

  - Rational (or rational approximation): I=160, D=147

    $\Rightarrow F_v = I\, F_x$ = 160 × 44.1 kHz = 7.056 MHz

Polyphase Decomposition

- Decomposition of $H(z) = \Sigma h_m z^{-m}$ in blocks of M:

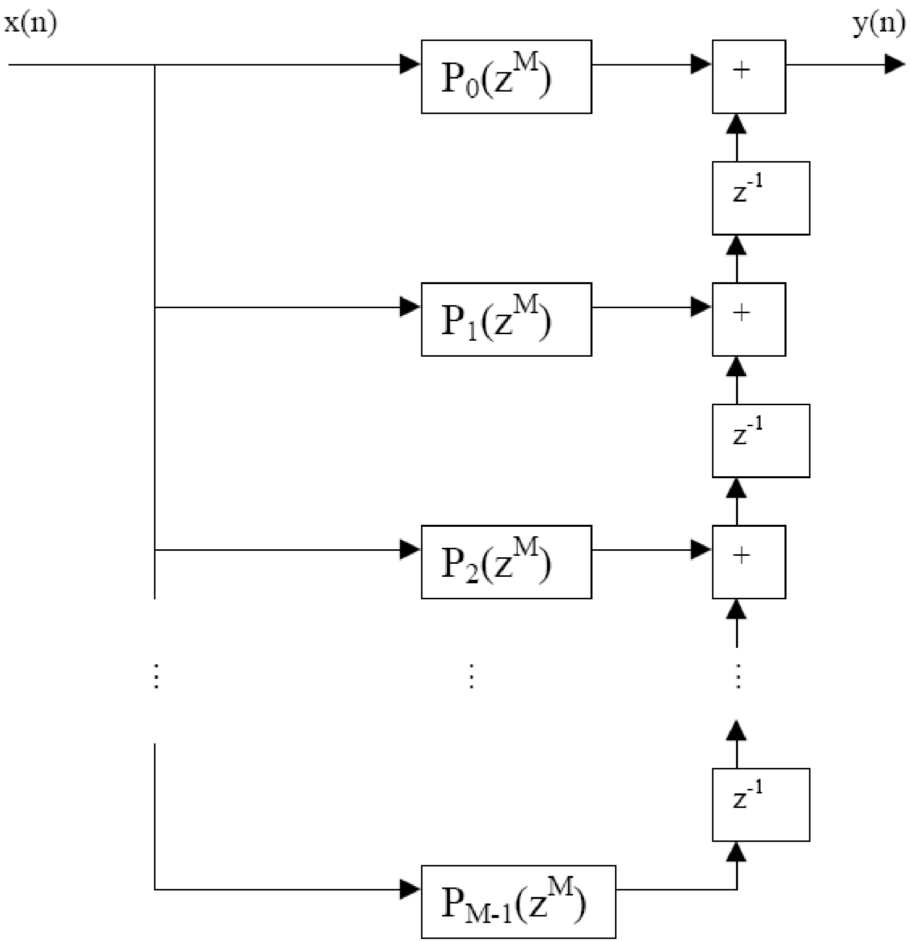
$$H(z) = \ldots + h(-M)z^{M} + h(-M+1)z^{M-1} + \ldots + h(-1)z^{1}$$
$$+ h(0)z^{0} + h(1)z^{-1} + \ldots + h(M-1)z^{-(M-1)}$$
$$+ h(M)z^{-M} + h(M+1)z^{-(M+1)} + \ldots + h(2M-1)z^{-(2M-1)}$$
$$+ h(2M)z^{-2M} + h(2M+1)z^{-(2M+1)} + \ldots + h(3M-1)z^{-(3M-1)} + \ldots$$

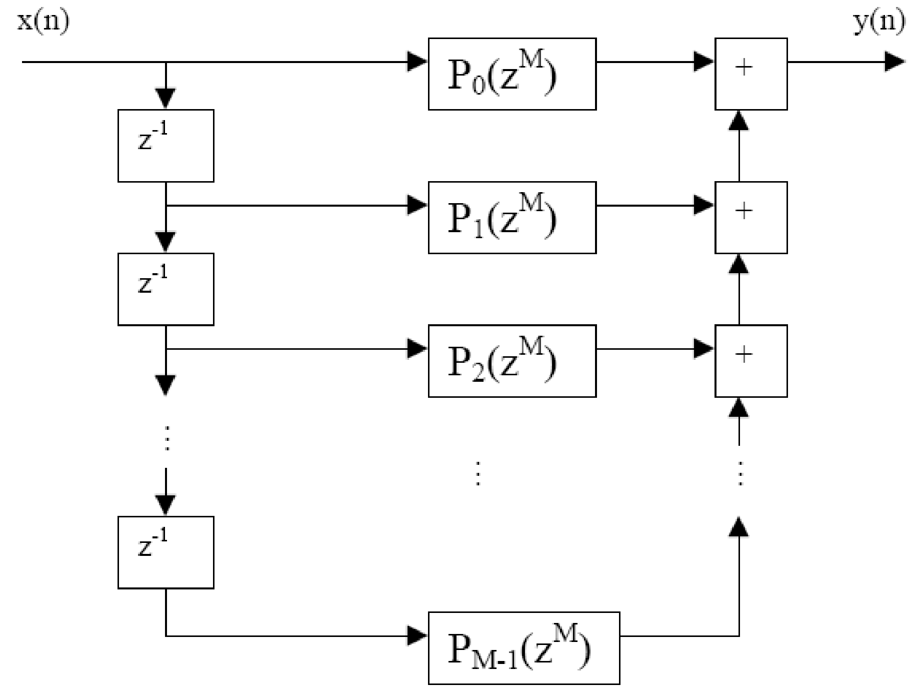
$$= z^{0}[\ldots + h(0)z^{0} + h(M)z^{-M} + \ldots] + z^{-1}[\ldots + h(1) + h(M+1)z^{-M} + \ldots]$$
$$+ z^{-2}[\ldots + h(2) + h(M+2)z^{-M} + \ldots] + \ldots$$
$$+ z^{-(M-1)}[\ldots + h(M-1) + h(2M-1)z^{-M} + \ldots]$$

$$\Rightarrow \quad H(z) = \sum_{i=0}^{M-1} z^{-i} P_i\left(z^{M}\right) \quad \text{where} \quad P_i(z) = \sum_{n=-\infty}^{+\infty} z^{-n} h(nM+i)$$

Polyphase Decomposition

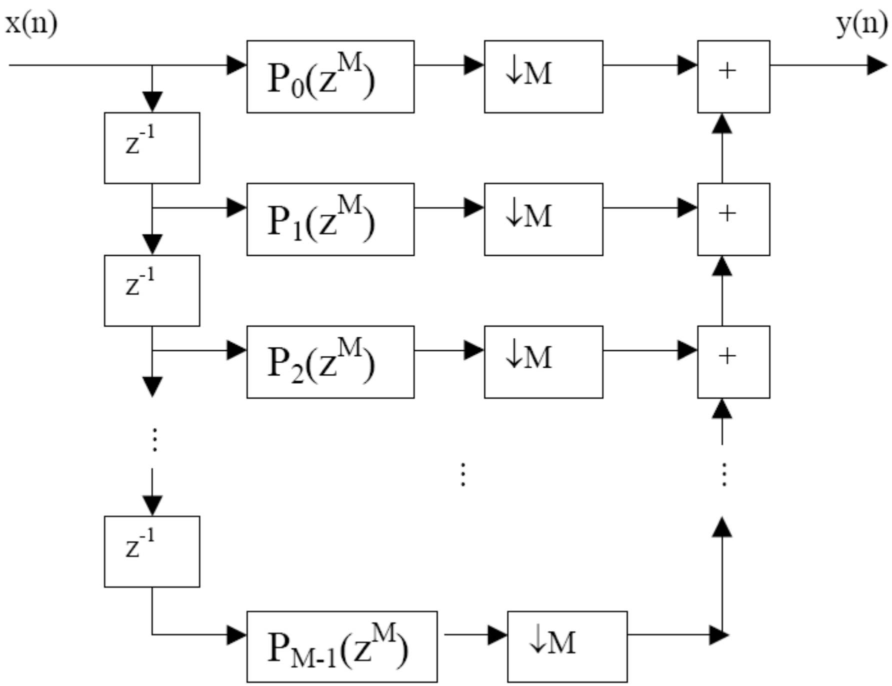
- Realisations:



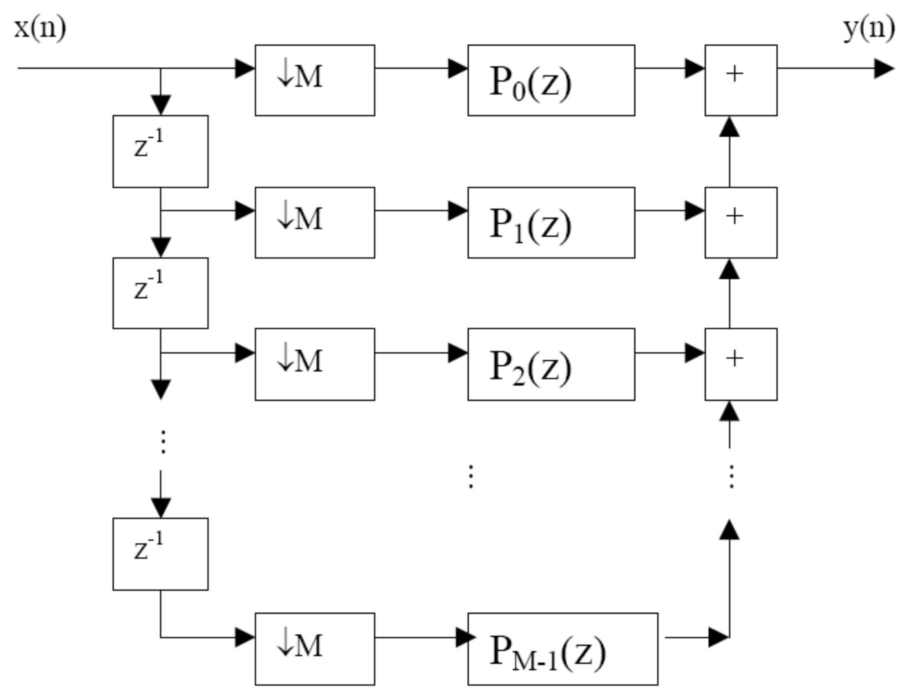


Implementation of Decimation

- Using noble identity:



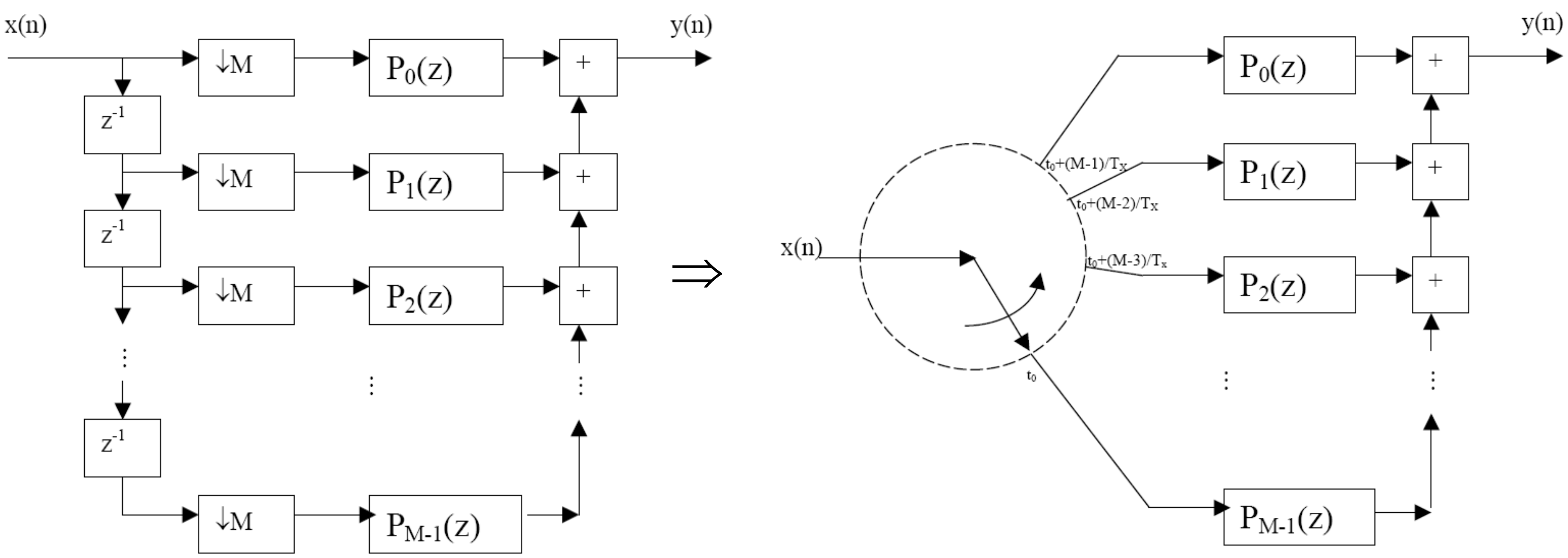
Operations performed at high rate



Operations at low rate ⇒ more efficient

Implementation of Decimation

- Using commutator:



one input per *D* pulses;
counter-clockwise rotation

Multi-Stage Sampling Conversion: Optimisation

- If required multi-rate conversion ratio is very large: more efficient implementation (= reduced computation & less storage) is obtained by using several smaller-ratio convertors in cascade
  - Permits low-cost anti-aliasing and anti-imaging filters
- Method: factorisation of decimation factor:

$$D = D_1 \times D_2 \times \ldots \times D_m >> 1$$

  - How to choose m, D_i?

- Metrics for deciding on multi-stage optimisation:
  - (1) Number of multiplications per second: $O = \sum_{i=1}^{m} p_i F_i$
  - (2) Filter storage requirement: $S = \sum_{i=1}^{m} p_i$

Multi-Stage Sampling Conversion: Optimisation

- Guidelines for choosing m, D_i:
  - Usually $m \leq 4$, typically $m = 3$ most efficient
    - Largest improvement in O and S for $m = 1 \rightarrow 2$
    - Significant extra improvement in O only for $m = 2 \rightarrow 3$
  - Find all integer factorisations of $\prod_{i=1}^{m} D_i$ for given M
  - Calculate & compare O and S for each factorisation
  - In general, choose sequencing as $D_1 > D_2 > \ldots > D_m$
  - For m=2: closed-form optimum choice is known:

$$D_1 = \frac{2D}{2 - \Delta f + \sqrt{2D\Delta f}}, \qquad D_2 = \frac{D}{D_1}$$

  - For other m, optimisation routines exist