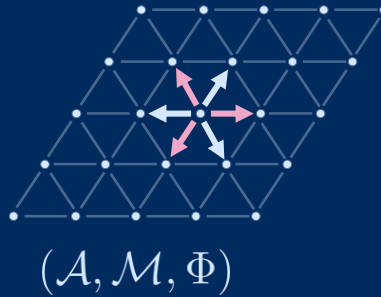


Algebraic Signal Processing Theory

Markus Püschel
Computer Science
ETH zürich

Collaboration with:
José Moura

and
Jelena Kovacevic
Martin Rötteler
Aliaksei Sandryhaila




Picture: http://www.christianch.ch/images/andere_gefahren.png

2

Preliminaries

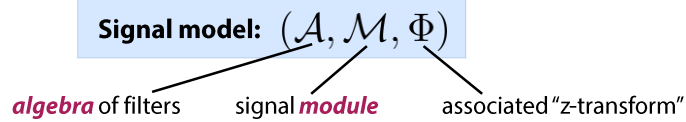
- *ASP = algebraic signal processing theory*
- *Algebra*: theory of groups, rings, and fields
- Scope of ASP: *linear signal processing*
- This talk: Focus on the *discrete* case

3

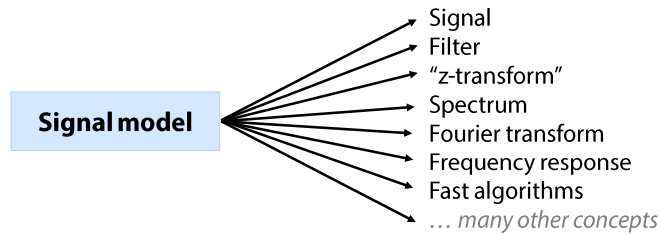
Signal processing concepts	Infinite time	Finite time	Infinite space	Finite space	ASP: Generic case	Other/new models
	z-transform	Finite z-transform	C-transform	Finite C-transform	Φ	
Set of signals	Laurent series in z^n	Polynomials in z^n	Series in C_n	Polynomials in C_n	\mathcal{M}	
Set of filters	Laurent series in z^n	Polynomials in z^n	Series in T_n	Polynomials in T_n	\mathcal{A}	
Fourier transform	DTFT	DFT	DSFT	DCTs/DSTs	\mathcal{F}	
Convolution					 Derivation	
Spectrum						
Frequency response						
Fast algorithms						
Filter banks						
<many others>						4

Big Picture

- Key concept in ASP:



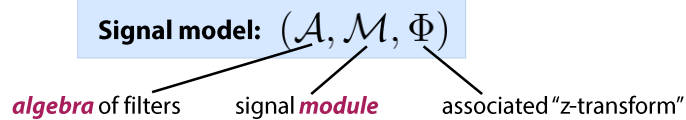
- Examples: Infinite and finite time or space, many others
- Signal model defined: all other concepts follow



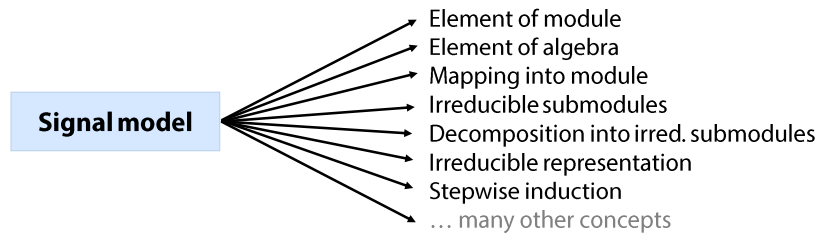
5

Big Picture

- Key concept in ASP:



- Examples: Infinite and finite time or space, many others
- Signal model defined: all other concepts follow



6

Organization

- *The algebraic structure underlying linear signal processing*
- From shift to signal model: Time and space
- From infinite to finite signal models
- Fast algorithms
- Conclusions

7

Algebraic Structure of Signal and Filter Space

- **Signal space, available operations:**

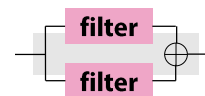
- $\alpha \cdot \text{signal} = \text{signal}$
 - $\text{signal} + \text{signal} = \text{signal}$
- } **vector space**

- **Filter space, available operations:**

- $\alpha \cdot \text{filter} = \text{filter}$
 - $\text{filter} + \text{filter} = \text{filter}$
 - $\text{filter} \cdot \text{filter} = \text{filter}$
- } **ring**

- **Filters operate on signals:**

- $\text{filter} \cdot \text{signal} = \text{signal}$



Set of filters = an algebra \mathcal{A}

Set of signals = an \mathcal{A} -module \mathcal{M}

8

(Algebraic) Signal Model

- Signals arise as sequences of numbers $(s_n)_{n \in I} \in \mathbb{C} \times \mathbb{C} \times \dots = \mathbb{C}^I$
- Need to assign module and algebra
- Example infinite discrete time: $(s_n)_{n \in \mathbb{Z}}$

$$\begin{aligned} \text{z-transform: } \Phi : (s_n)_{n \in \mathbb{Z}} &\rightarrow \sum s_n z^{-n} \in \mathcal{M} \\ \mathcal{M} &= \{ \sum s_n z^{-n} \} \quad \mathcal{A} = \{ \sum h_k z^{-k} \} \end{aligned}$$

- **Signal model (definition):** $(\mathcal{A}, \mathcal{M}, \Phi)$

\mathcal{A} algebra of filters

\mathcal{M} an \mathcal{A} -module of signals

Φ linear mapping $\mathbb{C}^I \rightarrow \mathcal{M}$

9

Algebras Occurring in SP: Shift-Invariance

- What is the shift?
 - A special filter $x (=z^{-1})$ is an element of \mathcal{A}
 - Filters expressible as polynomials/series in x

shift(s) = generator(s) of \mathcal{A}

- Shift-invariance: $x \cdot h = h \cdot x$ for all $h \in \mathcal{A}$

signal model $(\mathcal{A}, \mathcal{M}, \Phi)$ is shift-invariant $\iff \mathcal{A}$ is commutative

- Shift-invariant + finite-dimensional (+ one shift only):

$\mathcal{A} = \mathbb{C}[x]/p(x)$ is a polynomial algebra

10

Example: Finite Time Model and DFT

- **Finite signals:** $(s_0, s_1, \dots, s_{n-1})$ $\dim(\mathcal{M}), \dim(\mathcal{A}) < \infty$
- **Signal model:** $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n - 1)$

$$\text{signals } s(x) = \sum_{i=0}^{n-1} s_i x^i \in \mathcal{M} \quad \text{filters } h(x) = \sum_{k=0}^{n-1} h_k x^k \in \mathcal{A}$$

$$\text{filtering} = \text{cyclic convolution } h(x) \cdot s(x) \bmod x^n - 1$$

$$\text{finite z-transform } \Phi : \mathbb{C}^n \rightarrow \mathcal{M} \\ (s_0, s_1, \dots, s_{n-1}) \mapsto \sum s_i x^i \in \mathcal{M}$$

- **Spectrum/Fourier transform: Chinese remainder theorem**

$$\mathcal{F} : \mathbb{C}[x]/(x^n - 1) \rightarrow \mathbb{C}[x]/(x - \omega_n^0) \oplus \dots \oplus \mathbb{C}[x]/(x - \omega_n^{n-1}) \\ s(x) \mapsto (s(\omega_n^0), \dots, s(\omega_n^{n-1}))$$

$$\mathcal{F} = \text{DFT}_n$$

11

Summary so far

- **Signal model** $(\mathcal{A}, \mathcal{M}, \Phi)$
- **Shift-invariance:** \mathcal{A} is commutative
 - in addition finite makes \mathcal{A} a polynomial algebra
- **Infinite and finite time are special cases of signal models**

How to go beyond time?

Answer: Derivation of signal model from shift



12

Organization

- The algebraic structure underlying linear signal processing
- *From shift to signal model: Time and space*
- From infinite to finite signal models
- Fast algorithms
- Conclusions

13

	<i>Time</i>	<i>Space</i>
	$x \cdot t_n = t_{n+1}$	$x \cdot t_n = \frac{1}{2}(t_{n-1} + t_{n+1})$
<i>shift</i>		
<i>(time) marks</i>		
<i>k-fold shift</i>		
<i>realization of (time) marks</i>	$t_0 = 1 \Rightarrow t_n = x^n$	$t_0 = 1 \Rightarrow t_n = C_n$
<i>signals</i>	$s = \sum s_n x^n$	$s = \sum s_n C_n$
<i>filters</i>	$h = \sum h_k x^k$	$h = \sum h_k T_k$
		Chebyshev polynomials

14

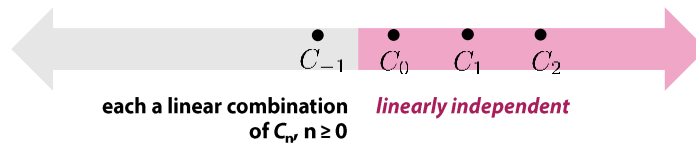
Time and Space (cont'd)

Chebyshev polynomials

- Time: done $\mathcal{M} = \{\sum s_n x^n\}$ $\mathcal{A} = \{\sum h_k x^k\}$

$$\Phi: (s_n)_{n \in \mathbb{Z}} \rightarrow \sum s_n x^n \in \mathcal{M} \quad \text{z-transform}$$

- Space: $\mathcal{M} = \{\sum s_n C_n\}$ $\mathcal{A} = \{\sum h_k T_k\}$



- Signal model only for right-sided sequences:

$$\Phi: (s_n)_{n \in \mathbb{Z}} \rightarrow \sum_{n \geq 0} s_n C_n \in \mathcal{M} \quad \text{C-transform}$$

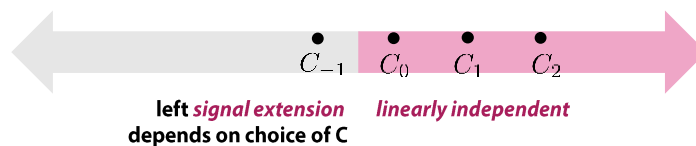
15

Left Signal Extension

Chebyshev polynomials

- Infinite space model: $\mathcal{M} = \{\sum_{n \geq 0} s_n C_n\}$ $\mathcal{A} = \{\sum_{k \geq 0} h_k T_k\}$

$$\Phi: (s_n)_{n \in \mathbb{Z}} \rightarrow \sum_{n \geq 0} s_n C_n \in \mathcal{M}$$

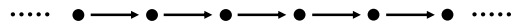


- Simplest signal extension: *monomial* $C_{-n} = a C_k$
- Monomial if and only if $C \in \{T, U, V, W\}$

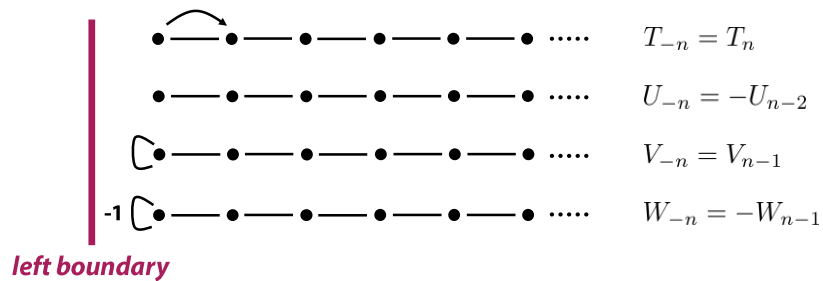
16

Visualization

- Infinite discrete time (z-transform)



- Infinite discrete space (C-transform, $C = T, U, V, W$)



17

Organization

- The algebraic structure underlying linear signal processing
- From shift to signal model: Time and space
- *From infinite to finite signal models*
- Fast algorithms
- Conclusions

18

Derivation: *Finite* Time Model



- **Solution: *Right boundary condition***

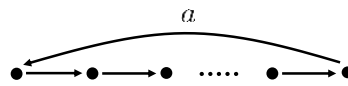
$$x^n = a_{n-1}x^{n-1} + \dots + a_0x^0$$

$$p(x) = x^n - a_{n-1}x^{n-1} - \dots - a_0x^0 = 0$$

$$\mathcal{M} = \mathcal{A} = \mathbb{C}[x]/p(x)$$

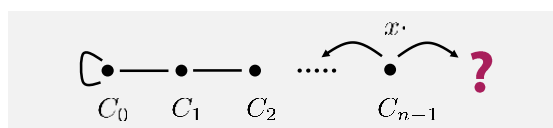
- **Monomial *signal extension*:** $p(x) = x^n - a$, $a \neq 0$
($a = 1$: finite z-transform)

- **Visualization:**



19

Derivation: *Finite* Space Model



$$C \in \{T, U, V, W\}$$

- **Monomial *signal extension*:** For each $C \in \{T, U, V, W\}$
four cases

$$C_n = C_{n-2}$$

$$C_n = 0$$

$$C_n = C_{n-1}$$

$$C_n = -C_{n-1}$$

- **16 finite space models \iff 16 DCTs/DSTs as Fourier transforms**

20

16 Finite Space Models

	$s_n - s_{n-2}$	s_n	$s_n - s_{n-1}$	$s_n + s_{n-1}$	f	C
$s_{-1} = s_1$	DCT-1 $2(x^2 - 1)U_{n-2}$	DCT-3 T_n	DCT-5 $(x - 1)W_{n-1}$	DCT-7 $(x + 1)V_{n-1}$	1	T
$s_{-1} = 0$	DST-3 $2T_n$	DST-1 U_n	DST-7 V_n	DST-5 W_n	$\sin \theta$	U
$s_{-1} = s_0$	DCT-6 $2(x - 1)W_{n-1}$	DCT-4 V_n	DCT-2 $2(x - 1)U_{n-1}$	DCT-4 $2T_n$	$\cos \frac{1}{2}\theta$	V
$s_{-1} = -s_0$	DST-8 $2(x + 1)V_{n-1}$	DST-6 W_n	DST-4 $2T_n$	DST-2 $2(x + 1)U_{n-1}$	$\sin \frac{1}{2}\theta$	W

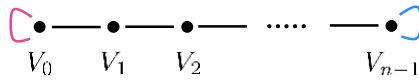
■ **Example: Signal model for DCT, type 2:**

$$\mathcal{M} = \mathbb{C}[x]/(x - 1)U_{n-2} = \{\sum_{i=0}^{n-1} s_i V_i\}$$

$$\mathcal{A} = \mathbb{C}[x]/(x - 1)U_{n-2} = \{\sum_{k=0}^{n-1} h_k T_k\}$$

$$\Phi : (s_i)_{0 \leq i < n} \mapsto \sum_{i=0}^{n-1} s_i V_i$$

■ **Visualization:**



Time (complex): complex finite z-transform				Section VI-B
Φ	\mathcal{M}	\mathcal{A}	$\mathcal{F} = \mathcal{P}_{b,\alpha}$	other \mathcal{F}
$s \mapsto \sum s_k x^k$	$\mathbb{C}[z]/(x^n - \alpha)$	regular	DFT $_n, D$	—
	$\mathbb{C}[z]/(x^n - 1)$	regular	DFT $_n = \text{DFT-1}_n$	DFT-2 $_n$
	$\mathbb{C}[z]/(x^n + 1)$	regular	DFT-3 $_n$	DFT-4 $_n$
Time (real): real finite z-transform				Section VI-G
Φ	\mathcal{M}	\mathcal{A}	$\mathcal{F} = \mathcal{P}_{b,\alpha}$	other \mathcal{F}
$s \mapsto \sum s_k x^k$	$\mathbb{R}[z]/(x^n - 1)$	regular	n.a.	RDFT $_n = \text{RDFT-1}_n$
	$\mathbb{R}[z]/(x^n - 1)$	regular	n.a.	DHT $_n = \text{DHT-1}_n$ (DWT-1 $_n$)
	$\mathbb{R}[z]/(x^n - 1)$	regular	n.a.	DHT-2 $_n$ (DWT-2 $_n$)
	$\mathbb{R}[z]/(x^n + 1)$	regular	n.a.	RDFT-3 $_n$
	$\mathbb{R}[z]/(x^n + 1)$	regular	n.a.	RDFT-4 $_n$
	$\mathbb{R}[z]/(x^n + 1)$	regular	n.a.	DHT-3 $_n$ (DWT-3 $_n$)
	$\mathbb{R}[z]/(x^n + 1)$	regular	n.a.	DHT-4 $_n$ (DWT-4 $_n$)
	$\mathbb{R}[z]/(x^n + 1)$	regular	n.a.	—
Space (complex/real): finite C-transform (C = TU,V,W)				Sections VIII-B, IX-B, XI-B
Φ	\mathcal{M}	\mathcal{A}	$\mathcal{F} = \mathcal{P}_{b,\alpha}$	other \mathcal{F}
$s \mapsto \sum s_k T_k$	$\mathbb{C}[z]/(x^2 - 1)U_{n-2}$	regular	DCT-1 $_n = \text{DCT-1}_n$	—
	$\mathbb{C}[z]/T_n$	regular	DCT-3 $_n = \text{DCT-3}_n$	—
	$\mathbb{C}[z]/(x - 1)W_{n-1}$	regular	DCT-5 $_n = \text{DCT-5}_n$	—
	$\mathbb{C}[z]/(x + 1)V_{n-1}$	regular	DCT-7 $_n = \text{DCT-7}_n$	—
	$\mathbb{C}[z]/(T_n - \cos r\pi)$	regular	DCT-3 $_n(r) = \text{DCT-3}_n(r)$	—
$s \mapsto \sum s_k U_k$	$\mathbb{C}[z]/T_n$	regular	DST-3 $_n$	DST-3 $_n$
	$\mathbb{C}[z]/W_n$	regular	DST-1 $_n$	DST-1 $_n$
	$\mathbb{C}[z]/V_n$	regular	DCT-7 $_n$	DCT-7 $_n$
	$\mathbb{C}[z]/W_n$	regular	DST-5 $_n$	DST-5 $_n$
	$\mathbb{C}[z]/(T_n - \cos r\pi)$	regular	DST-3 $_n(r)$	DST-3 $_n(r)$
$s \mapsto \sum s_k V_k$	$\mathbb{C}[z]/(x - 1)W_{n-1}$	regular	DCT-6 $_n$	DCT-6 $_n$
	$\mathbb{C}[z]/V_n$	regular	DCT-8 $_n$	DCT-8 $_n$
	$\mathbb{C}[z]/(x - 1)U_{n-1}$	regular	DCT-2 $_n$	DCT-2 $_n$
	$\mathbb{C}[z]/T_n$	regular	DCT-4 $_n$	DCT-4 $_n$
	$\mathbb{C}[z]/(T_n - \cos r\pi)$	regular	DCT-4 $_n(r)$	DCT-4 $_n(r)$
$s \mapsto \sum s_k W_k$	$\mathbb{C}[z]/(x + 1)V_{n-1}$	regular	DST-8 $_n$	DST-8 $_n$
	$\mathbb{C}[z]/W_n$	regular	DST-6 $_n$	DST-6 $_n$
	$\mathbb{C}[z]/T_n$	regular	DCT-4 $_n$	DCT-4 $_n$
	$\mathbb{C}[z]/(x + 1)U_{n-1}$	regular	DST-2 $_n$	DST-2 $_n$
	$\mathbb{C}[z]/(T_n - \cos r\pi)$	regular	DST-4 $_n(r)$	DST-4 $_n(r)$
$s \mapsto \sum s_k x^k$	$\mathbb{C}[z]/(x^n - 1)$	$((x^{-1} + x)/2)$	n.a.	RDFT $_n = \text{RDFT-1}_n$
	$\mathbb{C}[z]/(x^n - 1)$	$((x^{-1} + x)/2)$	n.a.	RDFT-2 $_n$
	$\mathbb{C}[z]/(x^n - 1)$	$((x^{-1} + x)/2)$	n.a.	DHT $_n = \text{DHT-1}_n$
	$\mathbb{C}[z]/(x^n - 1)$	$((x^{-1} + x)/2)$	n.a.	DHT-2 $_n$
	$\mathbb{C}[z]/(x^n + 1)$	$((x^{-1} + x)/2)$	n.a.	RDFT-3 $_n$
	$\mathbb{C}[z]/(x^n + 1)$	$((x^{-1} + x)/2)$	n.a.	RDFT-4 $_n$
	$\mathbb{C}[z]/(x^n + 1)$	$((x^{-1} + x)/2)$	n.a.	DHT-3 $_n$
	$\mathbb{C}[z]/(x^n + 1)$	$((x^{-1} + x)/2)$	n.a.	DHT-4 $_n$
	$\mathbb{C}[z]/(x^n + 1)$	$((x^{-1} + x)/2)$	n.a.	—

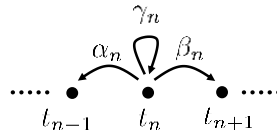
1D Trigonometric Transforms

- Signal models for all existing (and some newly introduced) trigonometric transforms (~30)
- Explains all existing trigonometric transforms
- Gives for each transform associated "z-transform", filters, etc.

source: "Algebraic Theory of Signal Processing," Arxiv

More Exotic 1-D Model

- Generic next neighbor shift

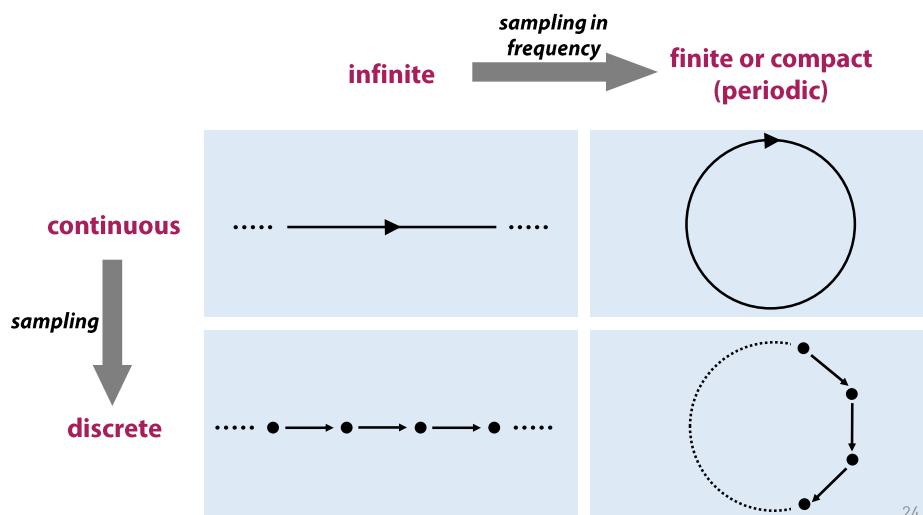


- Space *variant* but shift invariant
- Connects to orthogonal polynomials
- Applications?

23

Top-Down: 1-D Time (Directed) Models

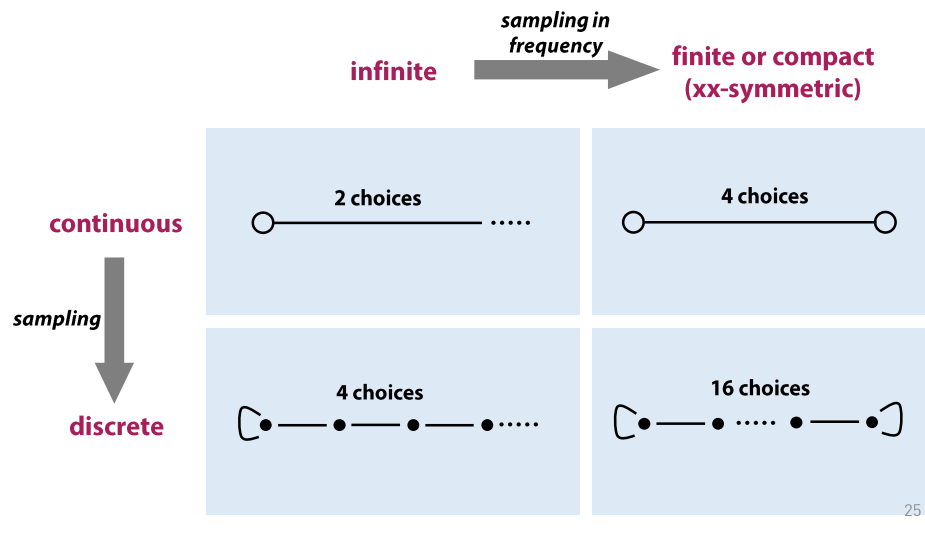
$$h * s = \int s(\tau) h(t - \tau) d\tau$$



24

Top-Down: 1-D Space (Undirected) Models

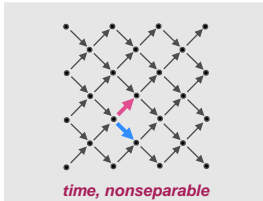
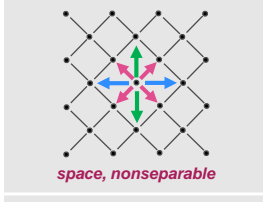
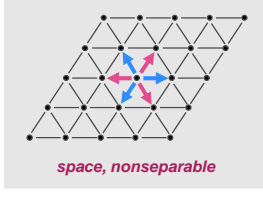
$$h * s = \int s(\tau) \frac{1}{2} (h(t - \tau) + h(t + \tau)) d\tau$$



Finite Signal Models in Two Dimensions

Visualization (without b.c.)	Signal Model $\mathcal{A} = \mathcal{M}$	Fourier Transform
<p>2-D time, separable</p>	$\mathbb{C}[x, y] / \langle x^n - 1, y^n - 1 \rangle$ time shifts: x, y	$\text{DFT}_n \otimes \text{DFT}_n$
<p>2-D space, separable</p>	$\mathbb{C}[x, y] / \langle T_n(x), T_n(y) \rangle$ space shifts: x, y	$\text{DCT}_n \otimes \text{DCT}_n$ 16 types

26

 <p><i>time, nonseparable</i></p>	$\mathbb{C}[u, v] / \langle u^n - 1, u^{n/2} - v^{n/2} \rangle$ time shifts: u, v	$\text{DDQT}_{n \times n/2}$
 <p><i>space, nonseparable</i></p>	$\mathbb{C}[u, v, w] / \langle T_{n/2}(u), T_{n/2}(v), 4w^2 - (u+1)(v+1) \rangle$ space shifts: u, v, w	$\text{DQT}_{n \times n/2}$
 <p><i>space, nonseparable</i></p>	$\mathbb{C}[x, y] / \langle C_n(x, y), \overline{C}_n(x, y) \rangle$ space shifts: u, v	$\text{DTT}_{n \times n}$

27

Organization

- The algebraic structure underlying linear signal processing
- From shift to signal model: Time and space
- From infinite to finite signal models
- *Fast algorithms*
- Conclusions

28

Fast Algorithms: Two Schools

$$\omega_n = e^{-2\pi j/n}$$

DFT

$$y_k = \sum_{\ell=0}^{n-1} \omega_n^{k\ell} s_\ell$$

$$y = \text{DFT}_n \cdot s$$

$$\text{DFT}_n = [\omega_n^{k\ell}]_{0 \leq k, \ell < n}$$

Cooley-Tukey FFT (general radix)

$$y_{n_2 j_1 + j_2} = \sum_{k_1=0}^{n_1-1} (\omega_n^{j_2 k_1}) \left(\sum_{k_2=0}^{n_2-1} s_{n_1 k_2 + k_1} \omega_n^{j_2 k_2} \right) \omega_n^{j_1 k_1}$$

$$\text{DFT}_n = L_{n_2}^n (I_{n_1} \otimes \text{DFT}_{n_2}) T_{n_1} (\text{DFT}_{n_1} \otimes I_{n_2})$$

DCT, type III

II. THE ODD-FACTOR ALGORITHM

The length- N IDCT of input sequence $X(k)$ is defined by

$$x(n) = \sum_{k=0}^{N-1} X(k) \cos \frac{\pi(2n+1)k}{2N} \quad 0 \leq n \leq N-1 \quad (1)$$

where sequence length N is an arbitrarily composite integer expressed by

$$N = 2^m \times q = 2^m \times \prod_{i=1}^p (2i+1)^{r_i} \quad (2)$$

Algorithm derivation

annually prime). The IDCT can be decomposed into

$$x \left(\left(\frac{qn+1}{2} \right) \right) = \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)} \quad (3)$$

$$x(qn+m) = \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi[(q(2n+1) + (q-1-2m)k]}{2N} \quad (4)$$

$$x(qn+q+m-1) = \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi[(q(2n+1) + (q-1-2m)k]}{2N} \quad (5)$$

where for (3)-(5), $n = 0$ to $N/q - 1$ and $m = 0$ to $(q-3)/2$. Equation (3) can be rewritten into

$$x \left(\left(\frac{qn+1}{2} \right) \right) = \sum_{k=0}^{N/2-1} \left(\sum_{\ell=0}^{q-1} X \left(\frac{2\ell N}{q} + k \right) \cos \frac{\pi(2n+1)(2\ell N/q + k)}{2N} \right) \cos \frac{\pi(2n+1)k}{2(N/q)} \quad (6)$$

$$x(qn+m) = \sum_{k=0}^{N/2-1} \left(\sum_{\ell=0}^{q-1} X \left(\frac{2\ell N}{q} + k \right) \cos \frac{\pi(2n+1)(2\ell N/q + k)}{2N} \right) \cos \frac{\pi(2n+1)k}{2(N/q)} \quad (7)$$

$$x(qn+q+m-1) = \sum_{k=0}^{N/2-1} \left(\sum_{\ell=0}^{q-1} X \left(\frac{2\ell N}{q} + k \right) \cos \frac{\pi(2n+1)(2\ell N/q + k)}{2N} \right) \cos \frac{\pi(2n+1)k}{2(N/q)} \quad (8)$$

$$\begin{aligned} & \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)} \\ & + \sum_{k=0}^{N/2-1} X \left(\frac{2N}{q} + k \right) \cos \frac{\pi(2n+1)(2N/q + k)}{2N} \\ & = \sum_{k=0}^{N/2-1} \left\{ X(k) + \sum_{\ell=1}^{q-1} (-1)^\ell X \left(\frac{2\ell N}{q} + k \right) \right. \\ & \quad \left. + X \left(\frac{2N}{q} + k \right) \right\} \cos \frac{\pi(2n+1)k}{2(N/q)} \\ & + \sum_{k=0}^{N/2-1} (-1)^k X \left(\frac{2N}{q} \right) \cos \frac{\pi(2n+1)k}{2(N/q)}. \end{aligned} \quad (6)$$

It is noted that input $\pi(2n+1)N/q$ is excluded from (6). By defining $S(k) = X(2N/q+k) + X(2N/q-k)$ and $T(k) = X(2N/q+k) - X(2N/q-k)$, where $k = 1, \dots, (q-1)/2$, we need to rewrite (6). Without presenting the details of derivation, we have

$$F(k) = \begin{cases} X(k) + \sum_{\ell=1}^{(q-1)/2} (-1)^\ell S(k) & k = 1, \dots, N/q - 1 \\ \sum_{\ell=1}^{(q-1)/2} (-1)^\ell X \left(\frac{2\ell N}{q} \right) & k = 0 \end{cases}$$

Therefore, (6) can be computed by a length- $(q-1)N/q$ DFT.

$$F(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)} + \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$G(n, m) = x(qn+m) - x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$H(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)}$$

$$I(n, m) = x(qn+m) - x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$J(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)}$$

$$K(n, m) = x(qn+m) - x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$L(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)}$$

$$\begin{aligned} & \sum_{k=0}^{N/2-1} X \left(\frac{2N}{q} + k \right) \cos \frac{\pi(2n+1)(2N/q + k)}{2N} \\ & = \sum_{k=0}^{N/2-1} \left\{ X(k) + \sum_{\ell=1}^{q-1} (-1)^\ell X \left(\frac{2\ell N}{q} + k \right) \right. \\ & \quad \left. + X \left(\frac{2N}{q} + k \right) \right\} \cos \frac{\pi(2n+1)k}{2(N/q)} \\ & + \sum_{k=0}^{N/2-1} (-1)^k X \left(\frac{2N}{q} \right) \cos \frac{\pi(2n+1)k}{2(N/q)}. \end{aligned} \quad (6)$$

It is noted that input $\pi(2n+1)N/q$ is excluded from (6). By defining $S(k) = X(2N/q+k) + X(2N/q-k)$ and $T(k) = X(2N/q+k) - X(2N/q-k)$, where $k = 1, \dots, (q-1)/2$, we need to rewrite (6). Without presenting the details of derivation, we have

$$F(k) = \begin{cases} X(k) + \sum_{\ell=1}^{(q-1)/2} (-1)^\ell S(k) & k = 1, \dots, N/q - 1 \\ \sum_{\ell=1}^{(q-1)/2} (-1)^\ell X \left(\frac{2\ell N}{q} \right) & k = 0 \end{cases}$$

Therefore, (6) can be computed by a length- $(q-1)N/q$ DFT.

$$F(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)} + \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$G(n, m) = x(qn+m) - x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$H(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)}$$

$$I(n, m) = x(qn+m) - x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$J(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)}$$

$$K(n, m) = x(qn+m) - x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \sin \frac{\pi(2n+1)k}{2(N/q)}$$

$$L(n, m) = x(qn+m) + x(qn+q+m-1)$$

$$= \sum_{k=0}^{N/2-1} X(k) \cos \frac{\pi(2n+1)k}{2(N/q)}$$

sequence length that is a power of odd integers. Therefore, the odd-factor algorithm is general and particularly suited to sequence length containing any possible combination of odd factors. Fig. 1 shows an example for $N = 27$. In principle, the proposed odd-factor algorithm is the reverse process of the FFT algorithm presented in [12]. For a composite sequence length containing both odd and even factors, the radix-2 and the odd-factor algorithms can be jointly used. In principle, the decomposition process can be carried out in many ways. However, a lower count of operations is obtained if the decomposition process starts with the ascending order of the factors of N . To minimize the required number of arithmetic operations, we generally prefer a computational complexity whose growth rate with the sequence length is as small as possible. In [12], it was proved that the growth rate of the computational complexity is proportional to the values of the odd factors. From Fig. 2, which shows the computational complexity in terms of the number of arithmetic operations per transform point, it can be observed that the growth rate of the computational complexity with the sequence length for $N = 2^m$ is larger than that for $N = 3^m$, and the smallest growth rate is achieved for $N = 2^m$. This observation indicates that the smallest growth rate can be achieved by using the radix-2 algorithm before the odd-factor algorithm in the decomposition process. In summary, the following steps can efficiently compute the IDCT of arbitrarily composite sequence length:

1. Decompose the sequence length N into a product of prime factors.

2. Decompose the sequence length N into a product of prime factors.

3. Decompose the sequence length N into a product of prime factors.

4. Decompose the sequence length N into a product of prime factors.

5. Decompose the sequence length N into a product of prime factors.

6. Decompose the sequence length N into a product of prime factors.

7. Decompose the sequence length N into a product of prime factors.

8. Decompose the sequence length N into a product of prime factors.

9. Decompose the sequence length N into a product of prime factors.

10. Decompose the sequence length N into a product of prime factors.

11. Decompose the sequence length N into a product of prime factors.

Typical derivation (> 200 such papers)

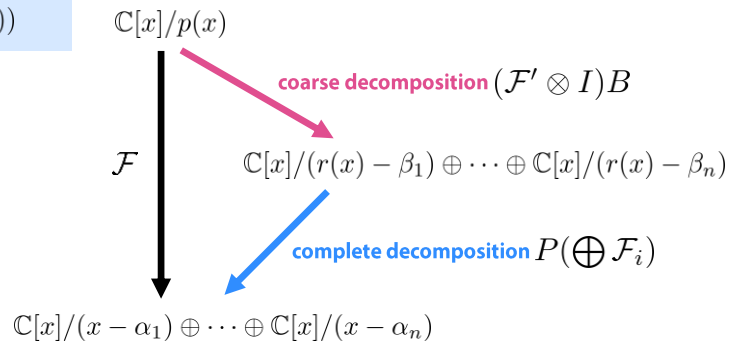
- Reason for existence?
- Underlying principle?
- All algorithms found?

$$V(k, m) = \begin{cases} X(k) + \sum_{\ell=1}^{(q-1)/2} (-1)^\ell S(k) \cos \frac{\pi \ell k}{2N} - T(k) \sin \frac{\pi \ell k}{2N} & k = 1, \dots, N/q - 1 \\ \sum_{\ell=1}^{(q-1)/2} (-1)^\ell X \left(\frac{2\ell N}{q} \right) \cos \frac{\pi \ell k}{2N} & k = 0 \end{cases} \quad (11)$$

$$W(k, m) = \begin{cases} X(k) + \sum_{\ell=1}^{(q-1)/2} (-1)^\ell S(k) \cos \frac{\pi \ell k}{2N} & \sin \frac{\pi \ell k}{2N} \\ + \sum_{\ell=1}^{(q-1)/2} (-1)^\ell T(k) \sin \frac{\pi \ell k}{2N} & \cos \frac{\pi \ell k}{2N} & k = 1, \dots, N/q - 1 \\ \sum_{\ell=1}^{(q-1)/2} (-1)^\ell X \left(\frac{2\ell N}{q} \right) \sin \frac{\pi \ell k}{2N} & \cos \frac{\pi \ell k}{2N} & k = N/q \end{cases} \quad (12)$$

Cooley-Tukey FFT Type Algorithms

assume p decomposes
 $p(x) = q(r(x))$



Example: $x^n - 1 = (x^m)^k - 1$ yields Cooley-Tukey FFT

$$\text{DFT}_n = L_{n_2}^n (I_{n_1} \otimes \text{DFT}_{n_2}) T_{n_1}^n (\text{DFT}_{n_1} \otimes I_{n_2})$$

Application to DCTs/DSTs

- Decomposition properties of Chebyshev polynomials

$$\begin{aligned} T_{km} &= T_k(T_m) \\ U_{km-1} &= U_{m-1} \cdot U_{k-1}(T_m) \\ V_{(k-1)/2+km} &= V_m \cdot V_{(k-1)/2}(T_{2m+1}) \\ W_{(k-1)/2+km} &= W_m \cdot W_{(k-1)/2}(T_{2m+1}) \\ T_{km+m/2} &= T_{m/2} \cdot V_k(T_m) \\ U_{km+m/2-1} &= U_{m/2-1} \cdot W_k(T_m) \end{aligned}$$

**Induced algorithms
DCTs/DSTs**

$$\text{DCT-3}_n = \text{DCT-3}_n(1/2), \quad \text{DCT-3}_{km}(r) = K_m^{\text{DCT}} \left(\bigoplus_{0 \leq c < k} \text{DCT-3}_m(r_c) \right) \text{DCT-3}_k(r) \otimes I_m B_{k,m}^{(C2)}$$

$$\text{DST-3}_n = \text{DST-3}_n(1/2), \quad \text{DST-3}_{km}(r) = K_m^{\text{DST}} \left(\bigoplus_{0 \leq c < k} \text{DST-3}_m(r_c) \right) \text{DST-3}_k(r) \otimes I_m B_{k,m}^{(C2)}$$

$$\text{DCT-4}_n = \text{DCT-4}_n(1), \quad \text{DCT-4}_{km}(r) = K_m^{\text{DCT}} \left(\bigoplus_{0 \leq c < k} \text{DCT-4}_m(r_c) \right) \text{DCT-4}_k(r) \otimes I_m B_{k,m}^{(C4)}$$

$$\text{DST-4}_n = \text{DST-4}_n(1), \quad \text{DST-4}_{km}(r) = K_m^{\text{DST}} \left(\bigoplus_{0 \leq c < k} \text{DST-4}_m(r_c) \right) \text{DST-4}_k(r) \otimes I_m B_{k,m}^{(S4)}$$

$$\text{DCT-1}_{km+1} = P_{k,m}^{(C1)} \left(\text{DCT-1}_{m+1} \oplus \left(\bigoplus_{0 \leq c < k} \text{DCT-3}_m(\frac{1}{2}) \right) (\overline{\text{DST}}^{-1}_{k-1} \otimes I_m) \right) B_{k,m}^{(C1)}$$

$$\text{DST-1}_{km+1} = P_{k,m}^{(S1)} \left(\text{DST-1}_{m+1} \oplus \left(\bigoplus_{0 \leq c < k} \text{DST-3}_m(\frac{1}{2}) \right) (\overline{\text{DST}}^{-1}_{k-1} \otimes I_m) \oplus \text{DST-1}_{m-1} \right) B_{k,m}^{(S1)}$$

$$\text{DCT-2}_{km} = P_{k,m}^{(C2)} \left(\text{DCT-2}_m \oplus \left(\bigoplus_{0 \leq c < k} \text{DCT-4}_m(\frac{1}{2}) \right) (\overline{\text{DST}}^{-1}_{k-1} \otimes I_m) \right) B_{k,m}^{(C2)}$$

$$\text{DST-2}_{km} = P_{k,m}^{(S2)} \left(\text{DST-2}_m \oplus \left(\bigoplus_{0 \leq c < k} \text{DST-4}_m(\frac{1}{2}) \right) (\overline{\text{DST}}^{-1}_{k-1} \otimes I_m) \oplus \text{DST-2}_m \right) B_{k,m}^{(S2)}$$

$$\text{DCT-7}_{km+(k+1)/2} = P_{k,m}^{(C7)} \left(\left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DCT-3}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-7}_{k+1} \otimes I_{2m+1}) \oplus \text{DCT-7}_{m+1} \right) B_{k,m}^{(C7)}$$

$$\text{DST-7}_{km+(k+1)/2} = P_{k,m}^{(S7)} \left(\left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DST-3}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-7}_{k+1} \otimes I_{2m+1}) \oplus \text{DST-7}_m \right) B_{k,m}^{(S7)}$$

$$\text{DCT-8}_{km+(k+1)/2} = P_{k,m}^{(C8)} \left(\left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DCT-4}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-8}_{k+1} \otimes I_{2m+1}) \oplus \text{DCT-8}_m \right) B_{k,m}^{(C8)}$$

$$\text{DST-8}_{km+(k+1)/2} = P_{k,m}^{(S8)} \left(\left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DST-4}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-8}_{k+1} \otimes I_{2m+1}) \oplus \text{DST-8}_{m+1} \right) B_{k,m}^{(S8)}$$

$$\text{DCT-5}_{km+(k+1)/2} = P_{k,m}^{(C5)} \left(\text{DCT-5}_{m+1} \oplus \left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DCT-3}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-5}_{k+1} \otimes I_{2m+1}) \right) B_{k,m}^{(C5)}$$

$$\text{DST-5}_{km+(k+1)/2} = P_{k,m}^{(S5)} \left(\text{DST-5}_m \oplus \left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DST-3}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-5}_{k+1} \otimes I_{2m+1}) \right) B_{k,m}^{(S5)}$$

$$\text{DCT-6}_{km+(k+1)/2} = P_{k,m}^{(C6)} \left(\text{DCT-6}_{m+1} \oplus \left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DCT-4}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-6}_{k+1} \otimes I_{2m+1}) \right) B_{k,m}^{(C6)}$$

$$\text{DST-6}_{km+(k+1)/2} = P_{k,m}^{(S6)} \left(\text{DST-6}_m \oplus \left(\bigoplus_{0 \leq c < \frac{k+1}{2}} \text{DST-4}_{2m+1}(\frac{2c+1}{k}) \right) (\overline{\text{DST}}^{-6}_{k+1} \otimes I_{2m+1}) \right) B_{k,m}^{(S6)}$$

<many more>

**Journal paper shown:
special case k = 3**

Real DFTs/DHTs

$$\begin{pmatrix} \text{RDFT}_n \\ \text{RDFT}_{2n} \\ \text{DHT}_n \\ \text{DHT}_{2n} \end{pmatrix} = P_n \begin{pmatrix} \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \end{pmatrix} \begin{pmatrix} B_{2m}^{(1)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(2)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(3)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(4)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \end{pmatrix} \begin{pmatrix} \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \end{pmatrix} \otimes I_n$$

$$\begin{pmatrix} \text{RDFT}_n \\ \text{RDFT}_{2n} \\ \text{DHT}_n \\ \text{DHT}_{2n} \end{pmatrix} = P_n \begin{pmatrix} \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \end{pmatrix} \begin{pmatrix} B_{2m}^{(1)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(2)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(3)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(4)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \end{pmatrix} \begin{pmatrix} \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \end{pmatrix} \otimes I_n$$

$$\begin{pmatrix} \text{RDFT}_n \\ \text{RDFT}_{2n} \\ \text{DHT}_n \\ \text{DHT}_{2n} \\ \text{BRDFT}_n \\ \text{BRDFT}_{2n} \end{pmatrix} = P_n \begin{pmatrix} \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \end{pmatrix} \begin{pmatrix} B_{2m}^{(1)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(2)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(3)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(4)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(5)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(6)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \end{pmatrix} \begin{pmatrix} \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \end{pmatrix} \otimes I_n$$

$$\begin{pmatrix} \text{RDFT}_n \\ \text{RDFT}_{2n} \\ \text{DHT}_n \\ \text{DHT}_{2n} \\ \text{BRDFT}_n \\ \text{BRDFT}_{2n} \end{pmatrix} = P_n \begin{pmatrix} \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \end{pmatrix} \begin{pmatrix} B_{2m}^{(1)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(2)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(3)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(4)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(5)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(6)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \end{pmatrix} \begin{pmatrix} \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \end{pmatrix} \otimes I_n$$

$$\begin{pmatrix} \text{RDFT}_n \\ \text{RDFT}_{2n} \\ \text{DHT}_n \\ \text{DHT}_{2n} \\ \text{BRDFT}_n \\ \text{BRDFT}_{2n} \end{pmatrix} = P_n \begin{pmatrix} \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \end{pmatrix} \begin{pmatrix} B_{2m}^{(1)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(2)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(3)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(4)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(5)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(6)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \end{pmatrix} \begin{pmatrix} \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \end{pmatrix} \otimes I_n$$

$$\begin{pmatrix} \text{RDFT}_n \\ \text{RDFT}_{2n} \\ \text{DHT}_n \\ \text{DHT}_{2n} \\ \text{BRDFT}_n \\ \text{BRDFT}_{2n} \end{pmatrix} = P_n \begin{pmatrix} \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{RDFT}_{2m} & \text{RDFT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{DHT}_{2m} & \text{DHT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \\ \text{BRDFT}_{2m} & \text{BRDFT}_{2m} \end{pmatrix} \begin{pmatrix} B_{2m}^{(1)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(2)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(3)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(4)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(5)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \\ B_{2m}^{(6)} S_{2m+1+1/2}(k(-e^{-\pi/2})) \end{pmatrix} \begin{pmatrix} \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \\ \text{URDFT}_n \end{pmatrix} \otimes I_n$$

Algebraic Theory of Transform Algorithms

- Consolidates the area
- Few general principles account for practically all existing algorithms
 - General Cooley-Tukey type algorithms
 - General prime-factor type algorithms
 - General Rader type algorithms
- Derivation is greatly simplified
- Many (dozens) new algorithms discovered
- Applicable to existing and novel linear transforms
 - DCTs/DSTs, MDCTs, RDFT, DHT, DQT, DTT, ...

Organization

- **The algebraic structure underlying linear signal processing**
- **From shift to signal model: Time and space**
- **From infinite to finite signal models**
- **Fast algorithms**
- **Conclusions**

35

Markus Püschel and José M. F. Moura

Algebraic Signal Processing Theory: Foundation and 1-D Time
IEEE Transactions on Signal Processing, Vol. 56, No. 8, pp. 3572-3585, 2008

Markus Püschel and José M. F. Moura

Algebraic Signal Processing Theory: 1-D Space
IEEE Transactions on Signal Processing, Vol. 56, No. 8, pp. 3586-3599, 2008

Markus Püschel and José M. F. Moura

Algebraic Signal Processing Theory: Cooley-Tukey Type Algorithms for DCTs and DSTs
IEEE Transactions on Signal Processing, Vol. 56, No. 4, pp. 1502-1521, 2008

Yevgen Voronenko and Markus Püschel

Algebraic Signal Processing Theory: Cooley-Tukey Type Algorithms for Real DFTs
IEEE Transactions on Signal Processing, Vol. 57, No. 1, pp. 205-222, 2009

Jelena Kovacevic and Markus Püschel

Algebraic Signal Processing Theory: Sampling for Infinite and Finite 1-D Space
IEEE Transactions on Signal Processing, Vol. 58, No. 1, pp. 242-257, 2010

Markus Püschel and Martin Rötteler

Algebraic Signal Processing Theory: Cooley-Tukey Type Algorithms on the 2-D Spatial Hexagonal Lattice
Applicable Algebra in Engineering, Communication and Computing, special issue on "The memory of Thomas Beth", Vol. 19, No. 3, pp. 259-292, 2008

Markus Püschel and Martin Rötteler

Algebraic Signal Processing Theory: 2-D Spatial Hexagonal Lattice
IEEE Transactions on Image Processing, Vol. 16, No. 6, pp. 1506-1521, 2007

Aliaksai Sandryhaila, Jelena Kovacevic and Markus Püschel

Algebraic Signal Processing Theory: Cooley-Tukey Type Algorithms for Polynomial Transforms Based on Induction
SIAM Journal on Matrix Analysis and Applications, Vol. 32, No. 2, pp. 364-384, 2011

Aliaksei Sandryhaila, Jelena Kovacevic and Markus Püschel

Algebraic Signal Processing Theory: 1-D Nearest-Neighbor Models
IEEE Transactions on Signal Processing, Vol. 60, No. 5, pp. 2247-2259, 2012

36

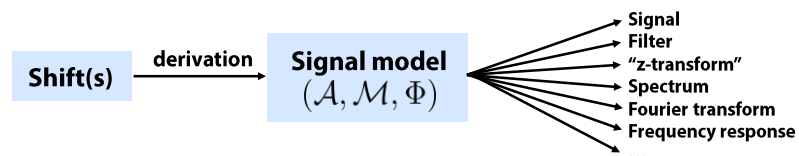
Related Work on Algebraic Methods in SP

- **Fourier analysis/Fourier transforms on groups G** (Beth, Rockmore, Clausen, Maslen, Healy, Terras, ...)
 - In the algebraic theory the special case $\mathcal{A} = \mathcal{M} = \mathbb{C}[G]$
 - If G non-commutative, necessarily non-shift-invariant
 - Algebraic theory provides associated filters etc., ties to SP concepts
- **Algebraic methods to derive DFT algorithms** (Nicholson, Winograd, Nussbaumer, Auslander, Feig, Burrus, ...)
 - Recognizes algebra/module for DFT, but only used for deriving algorithms
- **Origin of this work**
 - Beth (84), Minkwitz (93), Egner/Püschel (97/98)
 - Helpful hints: Steidl (93), Moura/Bruno (98), Strang (99)
- **Algebraic systems theory** (Kalman, Basile/Marro, Wonham/Morse, Willems/Mitter, Fuhrmann, Fliess, ...)
 - Focuses on infinite discrete time; different type of questions

37

Algebraic Signal Processing Theory: Conclusions

- **Signal model: One concept instantiating different SP methods**



- **General (axiomatic) approach to linear SP**
- **Deeper insight into SP**
- **Applications to date:**
 - New signal models (non-separable 2-D)
 - Comprehensive theory of fast algorithms

SMART project: www.ece.cmu.edu/~smart

Chebyshev Polynomials

[back1](#)
[back2](#)
[back3](#)

■ **Defining three-term recurrence:**

initial: $C_0 = 1, C_1 = ax + b$

$$C_{n+1} = 2xC_n + C_{n-1} \iff xC_n = \frac{1}{2}(C_{n+1} + C_{n-1})$$

■ **Special cases:**

C	...	$n = -1$	$n = 0$	$n = 1$	$n = 2$...
T	...	x	1	x	$2x^2 - 1$...
U	...	0	1	$2x$	$4x^2 - 1$...
V	...	1	1	$2x - 1$	$4x^2 - 2x - 1$...
W	...	-1	1	$2x + 1$	$4x^2 + 2x - 1$...

$n \geq 0 \longrightarrow$

■ **Closed forms:** $\cos \theta = x$

$$T_n = \cos n\theta \quad U_n = \frac{\sin(n+1)\theta}{\sin \theta} \quad V_n = \frac{\cos(n+\frac{1}{2})\theta}{\cos \frac{1}{2}\theta} \quad W_n = \frac{\sin(n+\frac{1}{2})\theta}{\sin \frac{1}{2}\theta}$$