Algebraic Signal Processing Theory

Markus Püschel

Electrical and Computer Engineering

Carnegie Mellon University

Collaborators:

José Moura (ECE, CMU)
Martin Rötteler (NEC, Princeton)
Jelena Kovacevic (BME, CMU)

This work was funded by NSF
under awards SYS-9988296 **and** *SYS-310941*

Preliminaries

- **Algebra (as used in this talk) is the theory of groups, rings, and fields**
- **The scope of the algebraic theory is linear signal processing (SP)**
- **In this talk we focus on the discrete case (infinite and finite signals)**
- **Background papers:**
  - *Basic theory (main paper):*
    Püschel and Moura, "Algebraic Theory of Signal Processing," submitted
  - *Fast algorithms:*
    Püschel and Moura, "Algebraic Theory of Signal Processing: 1-D Cooley-Tukey Type Algorithms," submitted
    Püschel and Moura (SIAM J. Comp 03) and earlier work (Egner and Püschel)
  - *New lattice transforms:*
    Püschel and Rötteler (ICASSP '04, DSP '04, ICASSP '05, ICIP '05)
  - *Sampling:*
    Kovacevic and Püschel (ICASSP '06)

Organization

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

The Basic Idea

- SP is built around the key concepts:
  signals, filters (convolution), z-transform, spectrum, Fourier transform

	infinite time	finite time	infinite space	finite space	other models	generic case
	z-transform	finite z-transform ?	C-transform ?	finite C-transform ?	next neighbor time-variant spatial hexagonal/quincunx lattice … and others	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		$\mathcal{F}$

derivation

Algebraic theory: All are instantiations of the same theory

The Basic Idea (cont'd)

- Key concept in the algebraic theory:

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

algebra of filters — signal module — associated "z-transform"

- Infinite and finite time and infinite and finite space are signal models
- But many others are possible
- Once the signal model is defined, all other concepts follow

Signal model →
- Signal
- Filter
- "z-transform"
- Spectrum
- Fourier transform
- Frequency response
- …

Why Algebraic Theory?

- **Identifies the filtering (convolution), "z-transform," spectrum, etc., that goes with the DCTs/DSTs and other existing transforms**

- **Explains boundary conditions for finite signal models**
  - E.g., why periodic for DFT and symmetric for the DCTs.

- **New signal models beyond time**
  - Space
  - Space in higher dimension (nonseparable hexagonal lattice, quincunx lattice)

- **A comprehensive theory of fast transform algorithms**
  - Current state: Hundreds of publications, but …
  - Algebraic theory: Concise derivation, classification, reason for existence, many new fast algorithms found for DCTs/DSTs and new lattice transforms

What we are Not Trying to do

- Restate existing knowledge in a more complicated way
- Do math for the math's sake
- Provide a theory that is purely "descriptive," i.e., cannot be applied

- The algebraic theory is "operational:"
  - Enables the derivation of new signal models
  - Enables the derivation of new fast algorithms for existing and new transforms

Organization

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

The Algebraic Structure of Signal and Filter Space

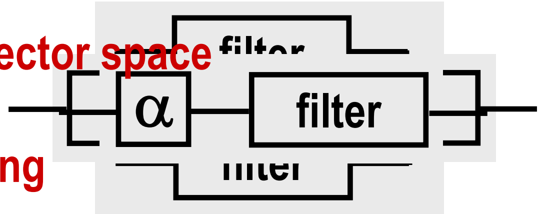
- **Signal space, available operations:**
  - signal + signal = signal
  - α • signal = signal

vector space

- **Filter space, available operations:**
  - filter + filter = filter
  - α • filter = filter
  - filter • filter = filter

vector space

ring



- **Filters operate on signals:**
  - filter • signal = signal

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

(Algebraic) Signal Model

- **Signals arise as sequences of numbers**

$$(s_n)_{n \in I} \in \mathbb{C} \times \mathbb{C} \times \cdots = \mathbb{C}^I$$

  - To obtain a notion of filtering, Fourier transform, etc., one needs to assign module and algebra

- **Example: infinite discrete time:** $(s_n)_{n \in \mathbb{Z}}$

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

$$\mathcal{M} = \{\sum s_n z^{-n}\}, \quad \mathcal{A} = \{\sum h_k z^{-k}\}$$

} **signal model**

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

$\mathcal{A}$	**algebra of filters**
$\mathcal{M}$	**an $\mathcal{A}$-module of signals**
Φ	**linear mapping $\mathbb{C}^I \to \mathcal{M}$**

Algebras Occurring in SP: Shift-Invariance

- **What is the shift?**
  - A special filter x (=z^{-1}) = 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
$\Leftrightarrow$ $\mathcal{A}$ is commutative

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

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

Example: Finite Time Model and DFT

- **Finite signals:** $(s_0, \dots, s_{n-1}) \quad \dim(\mathcal{M}), \dim(\mathcal{A}) < \infty$

- **Signal model:** $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n - 1)$

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

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

$$\Phi: \ (s_0, \dots, s_{n-1}) \mapsto s(x) = \sum s_i x^i \in \mathcal{M}$$ Finite z-transform

- **Spectrum and Fourier transform from Chinese remainder theorem**

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

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

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**

Organization

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

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

Operation of filters on signals is automatically defined
(the linear extension of the shift operation)

Time and Space (cont'd)

Chebyshev polynomials

- **Time: we are done**

$$\mathcal{A} = \{\sum h_k x^k\}$$

$$\mathcal{M} = \{\sum s_n x^n\}$$

$$\Phi : \ (s_n)_{n\in\mathbb{Z}} \mapsto \sum_{n\in\mathbb{Z}} s_n x^n$$ z-transform

- **Space:**

$$\mathcal{A} = \{\sum h_k T_k\}$$

$$\mathcal{M} = \{\sum s_n C_n\}$$

but: $\cdots \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \cdots$

$C_{-1} \quad C_0 \quad C_1 \quad C_2$

each a linear combination of C_n, $n \geq 0$ ← | → linearly independent

Signal model only for right-sided sequences:

$$\Phi : \ (s_n)_{n\geq 0} \mapsto \sum_{n\geq 0} s_n C_n$$ C-transform

Left Signal Extension

Chebyshev polynomials

- **Infinite space model:**

$$\mathcal{A} = \{\sum_{k\geq 0} h_k T_k\}$$

$$\mathcal{M} = \{\sum_{n\geq 0} s_n C_n\}$$

$$\Phi : \ (s_n)_{n\geq 0} \mapsto \sum_{n\geq 0} s_n C_n$$

$$\ldots \quad C_{-1} \quad C_0 \quad C_1 \quad C_2 \quad \ldots$$

left signal extension depends on choice of C ← | → linearly independent

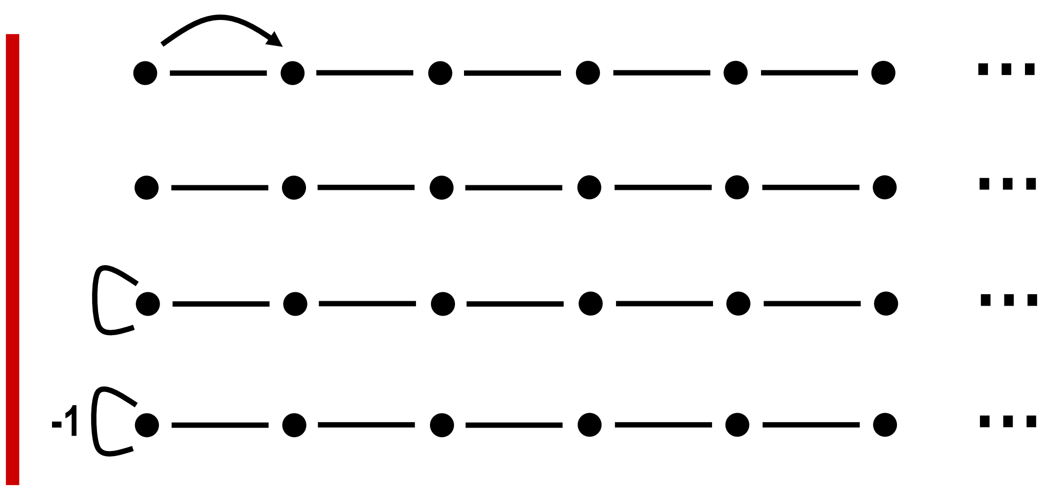
- **Simplest signal extension: monomial** $C_{-n} = aC_k$

- **Monomial if and only if** $C \in \{T, U, V, W\}$

Visualization

- Infinite discrete time (z-transform)

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



$T_{-n} = T_n$

$U_{-n} = -U_{n-2}$

$V_{-n} = V_{n-1}$

$W_{-n} = -W_{n-1}$

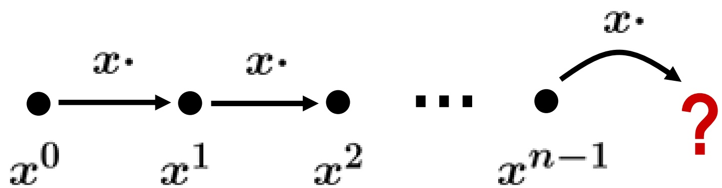
left boundary

Organization

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

Derivation: Finite Time Model

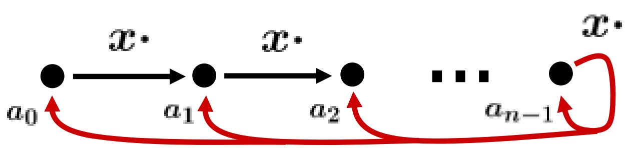
n time marks



$$\{\sum_{i=0}^{n-1} s_i x^i\}$$

- not closed under shift
- no module

- **Solution: Right boundary condition**



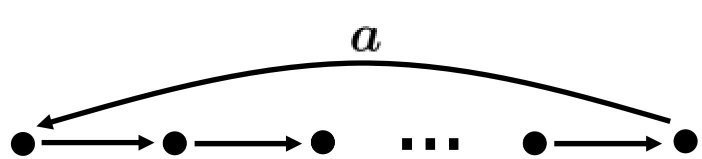
$$x^n = a_{n-1}x^{n-1} + \dots + a_0x^0$$

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

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

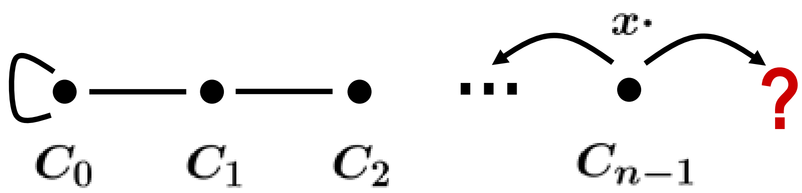
- **Monomial signal extension:** $p(x) = x^n - a, \quad a \neq 0$ periodic
  (a = 1: finite z-transform)

- **Visualization:**



Derivation: Finite Space Model

n space marks



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

$$\{\sum_{i=0}^{n-1} s_i C_i\}$$

- not closed under shift
- no module

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

$$\begin{aligned} C_n &= C_{n-2} \\ C_n &= 0 \\ C_n &= C_{n-1} \\ C_n &= -C_{n-1} \end{aligned}$$

- 16 finite space models $\Longleftrightarrow$ 16 DCTs/DSTs as Fourier transforms

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-8** 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

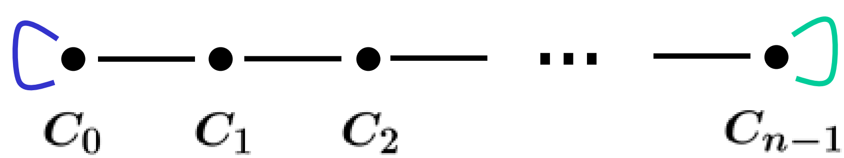
- **Example: Signal model for DCT, type 2:**

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

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

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

- **Visualization:**



1D Trigonometric Transforms

Time (complex): complex finite z-transform — Section VI-B

Φ	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{F} = \mathcal{P}_{b,\alpha}$	other $\mathcal{F}$
$\mathbf{s} \mapsto \sum s_k x^k$	$\mathbb{C}[x]/(x^n - a)$	regular	$\text{DFT}_n \cdot D$	—
	$\mathbb{C}[x]/(x^n - 1)$	regular	$\text{DFT}_n = \text{DFT-1}_n$	DFT-2_n
	$\mathbb{C}[x]/(x^n + 1)$	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 $\mathcal{F}$
$\mathbf{s} \mapsto \sum s_k x^k$	$\mathbb{R}[x]/(x^n - 1)$	regular	n.a.	$\text{RDFT}_n = \text{RDFT-1}_n$
	$\mathbb{R}[x]/(x^n - 1)$	regular	n.a.	RDFT-2_n
	$\mathbb{R}[x]/(x^n - 1)$	regular	n.a.	$\text{DHT}_n = \text{DHT-1}_n$ (DWT-1_n)
	$\mathbb{R}[x]/(x^n - 1)$	regular	n.a.	DHT-2_n (DWT-2_n)
	$\mathbb{R}[x]/(x^n + 1)$	regular	n.a.	RDFT-3_n
	$\mathbb{R}[x]/(x^n + 1)$	regular	n.a.	RDFT-4_n
	$\mathbb{R}[x]/(x^n + 1)$	regular	n.a.	DHT-3_n (DWT-3_n)
	$\mathbb{R}[x]/(x^n + 1)$	regular	n.a.	DHT-4_n (DWT-4_n)

Space (complex/real): finite C-transform (C = T,U,V,W) — Sections VIII-B, IX-B, XI-B

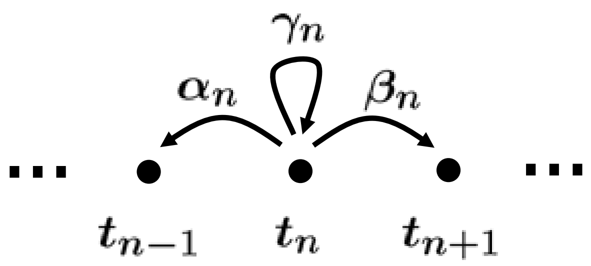
Φ	$\mathcal{M}$	$\mathcal{A}$	$\mathcal{F} = \mathcal{P}_{b,\alpha}$	other $\mathcal{F}$
$\mathbf{s} \mapsto \sum s_k T_k$	$\mathbb{C}[x]/(x^2 - 1)U_{n-2}$	regular	$\text{DCT-1}_n = \overline{\text{DCT-1}}_n$	—
	$\mathbb{C}[x]/T_n$	regular	$\text{DCT-3}_n = \overline{\text{DCT-3}}_n$	—
	$\mathbb{C}[x]/(x - 1)W_{n-1}$	regular	$\text{DCT-5}_n = \overline{\text{DCT-5}}_n$	—
	$\mathbb{C}[x]/(x + 1)V_{n-1}$	regular	$\text{DCT-7}_n = \overline{\text{DCT-7}}_n$	—
	$\mathbb{C}[x]/(T_n - \cos r\pi)$	regular	$\text{DCT-3}_n(r) = \overline{\text{DCT-3}}_n(r)$	—
$\mathbf{s} \mapsto \sum s_k U_k$	$\mathbb{C}[x]/T_n$	regular	$\overline{\text{DST-3}}_n$	DST-3_n
	$\mathbb{C}[x]/U_n$	regular	$\overline{\text{DST-1}}_n$	DST-1_n
	$\mathbb{C}[x]/V_n$	regular	$\overline{\text{DCT-7}}_n$	DCT-7_n
	$\mathbb{C}[x]/W_n$	regular	$\overline{\text{DST-5}}_n$	DST-5_n
	$\mathbb{C}[x]/(T_n - \cos r\pi)$	regular	$\overline{\text{DST-3}}(r)_n$	$\text{DST-3}(r)_n$
$\mathbf{s} \mapsto \sum s_k V_k$	$\mathbb{C}[x]/(x - 1)W_{n-1}$	regular	$\overline{\text{DCT-6}}_n$	DCT-6_n
	$\mathbb{C}[x]/V_n$	regular	$\overline{\text{DCT-8}}_n$	DCT-8_n
	$\mathbb{C}[x]/(x - 1)U_{n-1}$	regular	$\overline{\text{DCT-2}}_n$	DCT-2_n
	$\mathbb{C}[x]/T_n$	regular	$\overline{\text{DCT-4}}_n$	DCT-4_n
	$\mathbb{C}[x]/(T_n - \cos r\pi)$	regular	$\overline{\text{DCT-4}}(r)_n$	$\text{DCT-4}(r)_n$
$\mathbf{s} \mapsto \sum s_k W_k$	$\mathbb{C}[x]/(x + 1)V_{n-1}$	regular	$\overline{\text{DST-8}}_n$	DST-8_n
	$\mathbb{C}[x]/W_n$	regular	$\overline{\text{DST-6}}_n$	DST-6_n
	$\mathbb{C}[x]/T_n$	regular	$\overline{\text{DCT-4}}_n$	DCT-4_n
	$\mathbb{C}[x]/(x + 1)U_{n-1}$	regular	$\overline{\text{DST-2}}_n$	DST-2_n
	$\mathbb{C}[x]/(T_n - \cos r\pi)$	regular	$\overline{\text{DST-4}}(r)_n$	$\text{DST-4}(r)_n$
$\mathbf{s} \mapsto \sum s_k x^k$	$\mathbb{C}[x]/(x^n - 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	$\text{RDFT}_n = \text{RDFT-1}_n$
	$\mathbb{C}[x]/(x^n - 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	RDFT-2_n
	$\mathbb{C}[x]/(x^n - 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	$\text{DHT}_n = \text{DHT-1}_n$
	$\mathbb{C}[x]/(x^n - 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	DHT-2_n
	$\mathbb{C}[x]/(x^n + 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	RDFT-3_n
	$\mathbb{C}[x]/(x^n + 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	RDFT-4_n
	$\mathbb{C}[x]/(x^n + 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	DHT-3_n
	$\mathbb{C}[x]/(x^n + 1)$	$\langle (x^{-1} + x)/2 \rangle$	n.a.	DHT-4_n

- 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," submitted

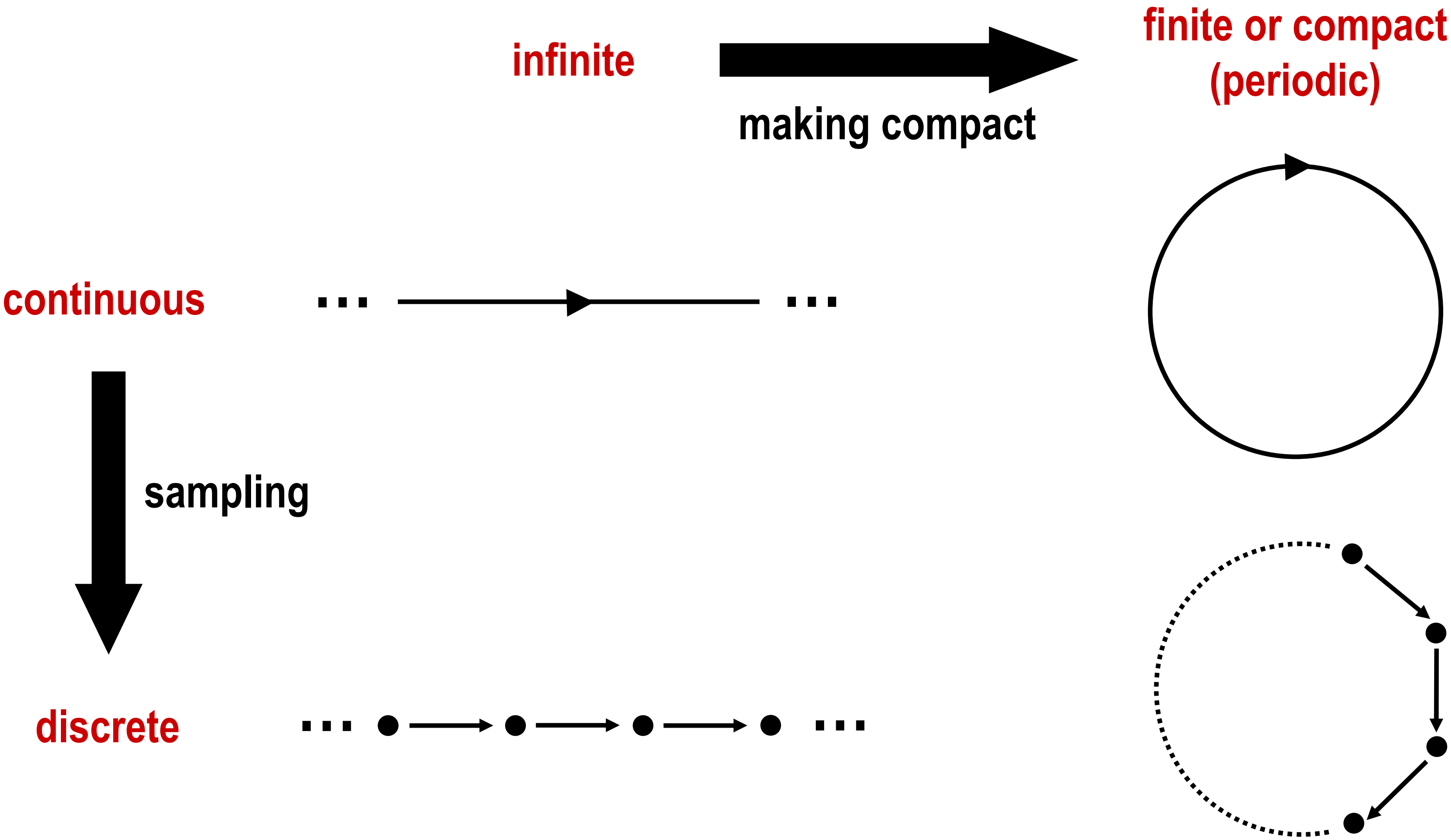
More Exotic 1-D Model

- Generic next neighbor shift



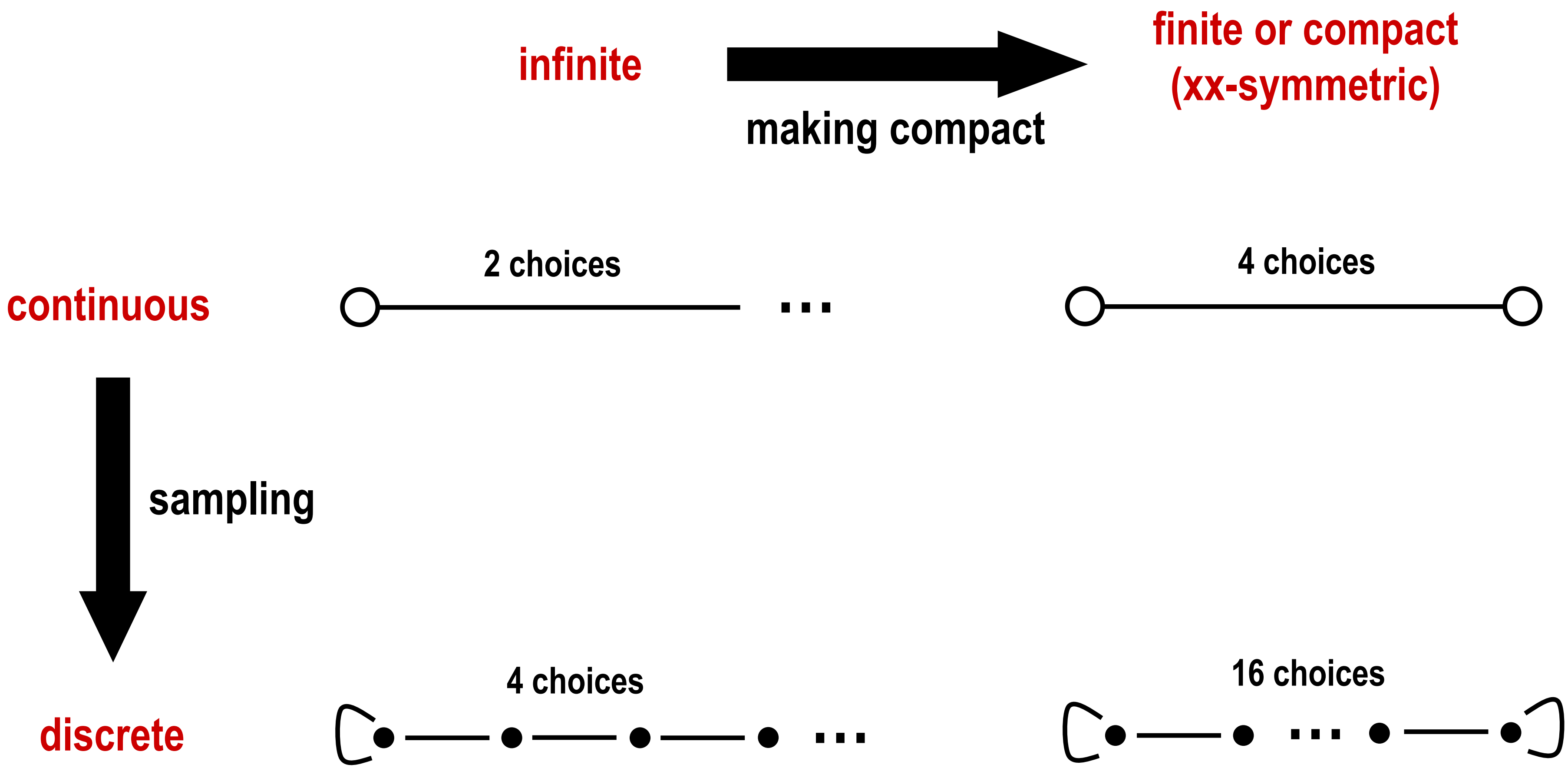
- Space variant but shift invariant
- Same procedure yields infinite and finite models
- Connects to orthogonal polynomials
- Applications?

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



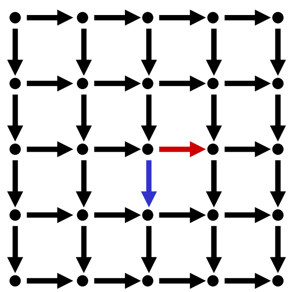
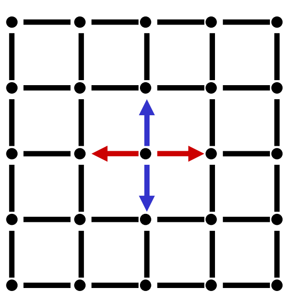
$$h \star s = \int s(\tau) h(t-\tau) d\tau$$

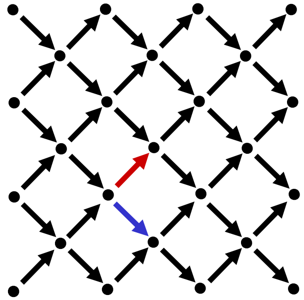
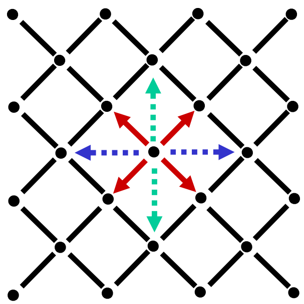
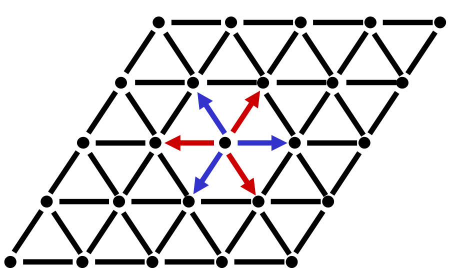
Top-Down: 1-D Space (Undirected) Models



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

Finite Signal Models in Two Dimensions

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

 time, nonseparable	$\mathbb{C}[u,v]/\langle u^n - 1, u^{\frac{n}{2}} - v^{\frac{n}{2}}\rangle$ **time shifts: u, v**	$\mathrm{DDQT}_{n\times\frac{n}{2}}$ **ICASSP ’05 (see also Mersereau)**
 space, nonseparable	$\mathbb{C}[u,v,w]/\langle T_{n/2}(u), T_{n/2}(v), 4w^2-(u+1)(v+1)\rangle$ **space shifts: u, v, w**	$\mathrm{DQT}_{n\times\frac{n}{2}}$ **ICIP ‘05**
 space, nonseparable	$\mathbb{C}[x,y]/\langle C_n(x,y), \overline{C}_n(x,y)\rangle$ **space shifts: u, v** $u\cdot C_{i,j} = \frac{1}{3}(C_{i,j+1} + C_{i-1,j} + C_{i+1,j-1})$ $v\cdot C_{i,j} = \frac{1}{3}(C_{i-1,j+1} + C_{i,j-1} + C_{i+1,j})$	$\mathrm{DTT}_{n\times n}$ **ICASSP ‘04**

Organization

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

DCT, type III

Algorithm derivation

II. THE ODD-FACTOR ALGORITHM

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

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

where sequence length N is an arbitrarily composite integer expressed by

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

mutually prime). The IDCT can be decomposed into

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

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

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

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

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

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

$$= \sum_{k=1}^{N/q-1} \left\{ X(k) + \sum_{i=1}^{(q-1)/2} (-1)^i \left[X\left(\frac{2iN}{q}+k\right) + X\left(\frac{2iN}{q}-k\right)\right]\right\} \cos \frac{\pi(2n+1)k}{2(N/q)} + \sum_{i=0}^{(q-1)/2} (-1)^k X\left(\frac{2iN}{q}\right)$$

$$= \sum_{k=0}^{N/q-1} U(k) \cos \frac{\pi(2n+1)k}{2(N/q)}. \qquad (6)$$

It is noted that input $x[(2i+1)N/q]$ is excluded from (6). By defining $S_i(k) = X(2iN/q+k) + X(2iN/q-k)$ and $T_i(k) = X(2iN/q+k) - X(2iN/q-k)$, where $i = 1, \cdots, (q-1)/2$, we have

$$U(k) = \begin{cases} X(k) + \sum_{i=1}^{(q-1)/2} (-1)^i S_i(k) & k = 1, \cdots, N/q-1 \\ \sum_{i=0}^{(q-1)/2} (-1)^i X\left(\frac{2iN}{q}\right) & k = 0. \end{cases} \qquad (7)$$

Therefore, (6) can be computed by a length-N/q IDCT. By combining (4) and (5), we form two new sequences defined by

$$F(n,m) = \frac{x(qn+m) + x(qn+q-m-1)}{2} = \sum_{k=0}^{N-1} X(k) \cos \frac{\pi(q-1-2m)k}{2N} \cdot \cos \frac{\pi(2n+1)k}{2(N/q)} \qquad (8)$$

$$G(n,m) = \frac{x(qn+m) - x(qn+q-m-1)}{2} = \sum_{k=0}^{N-1} X(k) \sin \frac{\pi(q-1-2m)k}{2N} \cdot \sin \frac{\pi(2n+1)k}{2(N/q)}. \qquad (9)$$

If we define $\alpha = \pi(q-1-2m)$ for simplicity, (8) can be further decomposed into

$$F(n,m) = \sum_{k=1}^{N/q-1} \left\{ \sum_{i=1}^{(q-1)/2} X\left(\frac{2iN}{q}+k\right) \cdot \cos \frac{\alpha(2iN/q+k)}{2N} \cos \frac{\pi(2n+1)(2iN/q+k)}{2(N/q)} + \sum_{i=1}^{(q-1)/2} X\left(\frac{2iN}{q}-k\right) \cdot \cos \frac{\alpha(2iN/q-k)}{2N} \cos \frac{\pi(2n+1)(2iN/q-k)}{2(N/q)} \right\}$$

$$+ \sum_{k=1}^{N/q-1} X(k) \cos \frac{\alpha k}{2N} \cos \frac{\pi(2n+1)k}{2(N/q)} + \sum_{i=0}^{(q-1)/2} X\left(\frac{2iN}{q}\right) \cos \frac{i\alpha}{q} \cdot \cos \frac{\pi(2n+1)(2iN/q)}{2(N/q)}$$

$$= \sum_{k=1}^{N/q-1} \sum_{i=1}^{(q-1)/2} (-1)^i \left\{ S_i(k) \cos \frac{\pi\alpha_i}{q} \cos \frac{\alpha k}{2N} - T_i(k) \sin \frac{\alpha i}{q} \sin \frac{\alpha k}{2N} \right\} \cos \frac{\pi(2n+1)k}{2(N/q)}$$

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

$$= \sum_{k=0}^{N/q-1} V(k,m) \cos \frac{\pi(2n+1)k}{2(N/q)} \qquad (10)$$

which is a length-N/q IDCT whose input sequence is calculated by

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 FDCT 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 lengths for $N = 5^m$ 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

Typical derivation
(More than hundred such papers)

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

$$V(k,m) = \begin{cases} X(k) \cos \frac{\alpha k}{2N} + \sum_{i=1}^{(q-1)/2} (-1)^i \left\{ S_i(k) \cos \frac{\alpha i}{q} \cos \frac{\alpha k}{2N} - T_i(k) \sin \frac{\alpha i}{q} \sin \frac{\alpha k}{2N} \right\} & k = 1, \cdots, N/q-1 \\ \sum_{i=0}^{(q-1)/2} (-1)^i X\left(\frac{2iN}{q}\right) \cos \frac{\alpha i}{q} & k = 0. \end{cases} \qquad (11)$$

$$W(k,m) = \begin{cases} \left\{ X(k) + \sum_{i=1}^{(q-1)/2} (-1)^i S_i(k) \cos \frac{\alpha i}{q} \right\} \sin \frac{\alpha k}{2N} + \left\{ \sum_{i=1}^{(q-1)/2} (-1)^i T_i(k) \sin \frac{\alpha i}{q} \right\} \cos \frac{\alpha k}{2N} & k = 1, \cdots, N/q-1 \\ \sum_{i=1}^{(q-1)/2} (-1)^i X\left[\frac{2i-1}{q}\right] \sin \frac{\alpha(2i-1)}{2q} & k = N/q. \end{cases} \qquad (13)$$

G. Bi "Fast Algorithms for the Type-III DCT of Composite Sequence Lengths" IEEE Trans. SP 47(7) 1999

Fast Algorithms: Cooley-Tukey FFT

Signal model: Finite z-transform $\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/(x^n - 1)$

$$\Phi : (s_0, \ldots, s_{n-1}) \mapsto s(x) = \sum s_i x^i \in \mathcal{M}$$

Fourier transform

$$\mathrm{DFT}_n : \mathbb{C}[x]/(x^n - 1) \rightarrow \mathbb{C}[x]/(x - \omega_n^0) \oplus \ldots \oplus \mathbb{C}[x]/(x - \omega_n^{n-1})$$

DFT

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

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

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

Cooley-Tukey FFT

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

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

Cooley-Tukey FFT Type Algorithms

assume p decomposes

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

$$\mathbb{C}[x]/p(x)$$

$\mathcal{F}$

coarse decomposition $(\mathcal{F}' \otimes I)B$

$$\mathbb{C}[x]/(r(x)-\beta_1) \oplus \ldots \oplus \mathbb{C}[x]/(r(x)-\beta_n)$$

complete decomposition $P(\bigoplus \mathcal{F}_i)$

$$\mathbb{C}[x]/(x-\alpha_1) \oplus \ldots \oplus \mathbb{C}[x]/(x-\alpha_n)$$

Example:

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

$$\mathrm{DFT}_n = L^n_m(I_k \otimes \mathrm{DFT}_m)T^n_m(\mathrm{DFT}_k \otimes I_m)$$

Application to DCTs/DSTs

- Decomposition properties of Chebyshev polynomials

$$T_{km} = T_k(T_m)$$

- Induced Cooley-Tukey type algorithms (most not known before)

$$\mathrm{DTT}_n(r) = K_m^n(\bigoplus \mathrm{DTT}_m(r_i))(\overline{\mathrm{DST\text{-}3}}_k(r) \otimes I_m)B_{n,k}$$
$$\mathrm{DTT}_n(r) = K_m^n(\bigoplus \mathrm{DTT}_m(r_i))(\mathrm{DCT\text{-}3}_k(r) \otimes I_m)B_{n,k}$$

DCT/DST 3/4

DCT/DST 1/2

DCT/DST 5-8

Algebraic Theory of Algorithms (Beyond DFT)

- **General Cooley-Tukey type algorithms**
  - many new algorithms for DCTs/DSTs, RDFT, DHT, DQT, DTT, …
- **General prime-factor type algorithms**
- **General Rader type algorithms**
- **Explains and easily derives practically all existing algorithms and relationships between transforms**
- **Formulates general principle that accounts for all algorithms**

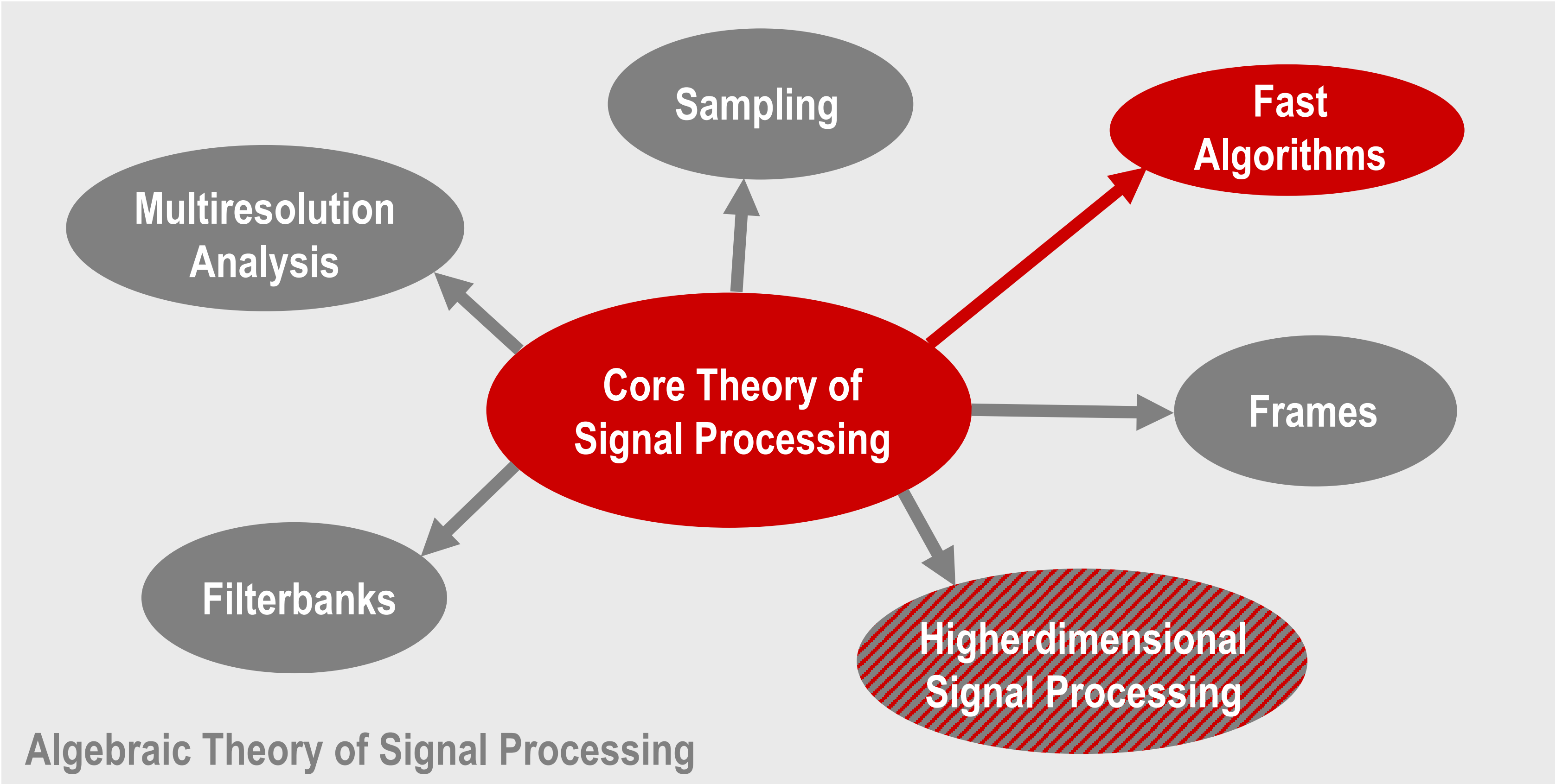
Organization

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

Related Work on Algebraic Methods in SP

- **Algebraic systems theory (Kalman, Basile/Marro, Wonham/Morse, Willems/Mitter, Fuhrmann, Fliess, …)**
  - Focuses on infinite discrete time; different type of questions
- **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)

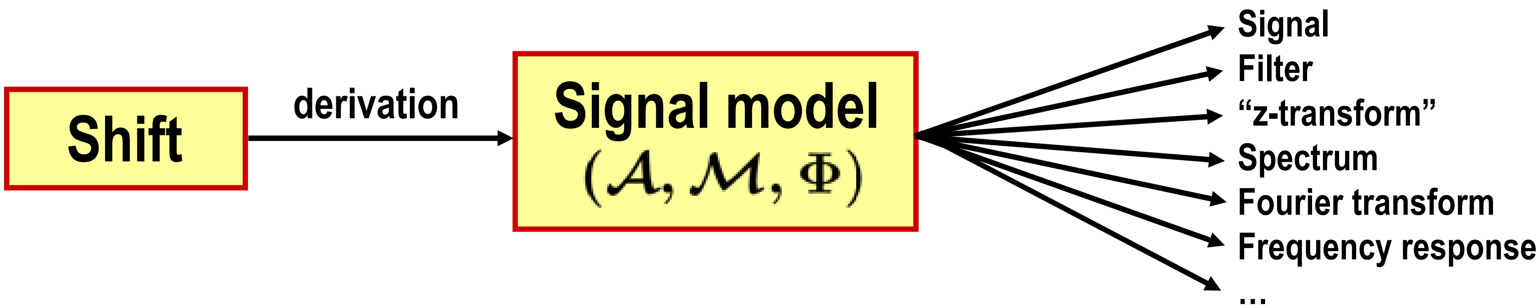
Future Work



Collaborators: José Moura, Jelena Kovacevic, Martin Rötteler

Algebraic Theory of Signal Processing: Conclusions

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



- **General (axiomatic) approach to linear SP**
- **Finite SP, understanding existing transforms**
- **First new applications:**
  - New SP methods (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:** $C_0 = 1,\ C_1 = ax + b$ choice

$$C_{n+1} = 2xC_n - C_{n-1} \quad \Leftrightarrow \quad xC_n = \tfrac{1}{2}(C_{n+1} + C_{n-1})$$

- **Special cases:**

	...	-1	$n = 0$	1	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	$2x + 1$	$4x^2 + 2x - 1$	...

$n \geq 0$ ⟶

symmetry

$$T_{-n} = T_n$$
$$U_{-n} = -U_{n-2}$$
$$V_{-n} = V_{n-1}$$
$$W_{-n} = -W_{n-1}$$

- **Closed forms:** $\cos\theta = x$

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

The General Fourier Transform $\mathcal{F}$

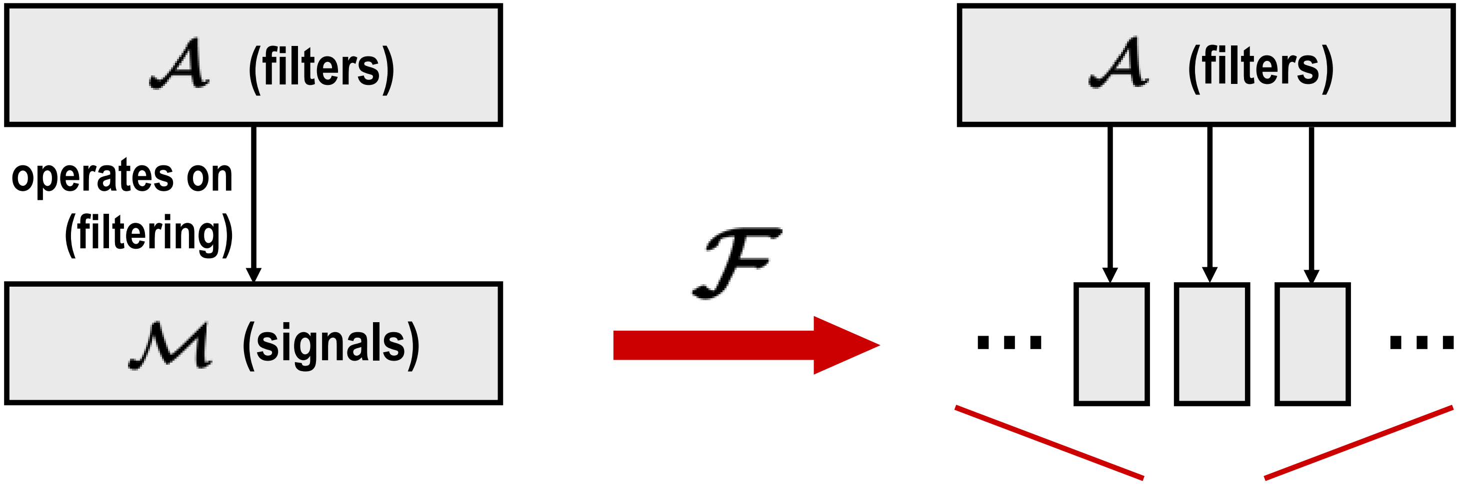
- Infinite discrete time:

$$\mathcal{F}: \sum s_n z^{-n} \mapsto \sum s_n e^{-j\omega n}, \quad \omega \in [-\pi, \pi)$$

projection onto

$$\mathcal{M}_\omega = \langle \sum e^{j\omega n} z^{-n} \rangle$$ eigenspace for all filters

- Given any signal model $(\mathcal{A}, \mathcal{M}, \Phi)$



spectrum = set of smallest invariant subspaces $\mathcal{M}_\omega, \ \omega \in W$

Finite Shift-Invariant Signal Models

- **Finite signals:** $(s_0, \ldots, s_{n-1})$ $\quad \dim(\mathcal{M}), \dim(\mathcal{A}) < \infty$

- **Which finite-dimensional algebras are commutative?**
  Answer: Polynomial algebras (focus on one variable)

$$\mathbb{C}[x]/p(x) = \{h(x) = \sum h_k x^k \mid \deg(h) < \deg(p)\}$$

Signal model:

$$\mathcal{A} = \mathcal{M} = \mathbb{C}[x]/p(x), \quad \Phi: \ (s_0, \ldots, s_{n-1}) \mapsto \sum s_i p_i(x)$$

Filtering (convolution): multiplication modulo $p(x)$

$$h(x) \cdot s(x) \ \mathrm{mod}\ p(x), \quad h(x) \in \mathcal{A}, \ s(x) \in \mathcal{M}$$