TSBB06 Multi-dimensional Signal Processing

Lecture 2A

Recap, introduction and overview

Recap from *Mathematical Toolbox*…

- Definition of a vector space
- Subspace
- **Linear combination**
- Linear dependency/independency
- Linear span
- Basis
- Coordinates
- **Scalar product**
- Norm (from scalar product)
- Orthogonality
- ON (ortho-normal)

$$\mathbb{R}^n$$

Recap from *Mathematical Toolbox*…

- Matrices **M** with $n \times m$ elements in $\mathbb{R}$ represent linear transformations $\mathbb{R}^m \rightarrow \mathbb{R}^n$
- Range, null space, and rank of matrix **M**
- Determinant, trace
- Transpose, denoted M^T
- Special cases: symmetric matrix, anti-symmetric matrix, orthogonal matrix, $O(n)$, $SO(n)$, $so(n)$
- Eigenvalues of a symmetric matrix

Can be generalized to $\mathbb{C}^n$

- See *Mathematical Toolbox…*, Section 3.10
- **Special scalar product** in $\mathbb{C}^n$
- Complex transpose, denoted $\mathbf{M}^*$
- Hermitian matrix, anti-Hermitian matrix
- Unitary matrix, $U(n)$, $SU(n)$, $su(n)$

Scalar product

In a general vector space with a scalar product:

- The scalar product must satisfy
  1. $\langle \cdot | \cdot \rangle : V \times V \rightarrow F$
  2. $\langle \mathbf{a} + \mathbf{b} | \mathbf{c} \rangle = \langle \mathbf{a} | \mathbf{c} \rangle + \langle \mathbf{b} | \mathbf{c} \rangle$
  3. $\langle \alpha \mathbf{a} | \mathbf{b} \rangle = \alpha \langle \mathbf{a} | \mathbf{b} \rangle$
  4. $\langle \mathbf{a} | \mathbf{b} \rangle = \langle \mathbf{b} | \mathbf{a} \rangle^*$
  5. $\langle \mathbf{a} | \mathbf{a} \rangle \geq 0, \text{ with } = 0 \text{ iff } \mathbf{a} = \mathbf{0}$

  for vectors **a**, **b**, **c** and scalar α
- Here, * denotes complex conjugation

Examples of scalar product spaces

- $V = \mathbb{R}^N$ = the set of all ordered *N*-tuples of real numbers form a vector space
- We represent such ordered *N*-tuples as columns of *N* real numbers
- Given an $N \times N$ symmetric matrix $\mathbf{G}_0$:
- A scalar product between vectors **a** and **b** is given as $\langle \mathbf{a} | \mathbf{b} \rangle = \mathbf{b}^T \mathbf{G}_0 \mathbf{a}$

Examples of scalar product spaces

- We can choose $\mathbf{G}_0$ = $\mathbf{I}$ = the $N \times N$ identity matrix
  - Gives the "standard" scalar product (dot product) $\mathbf{b}^*\mathbf{a}$
- However, any $\mathbf{G}_0$ that is
  - Symmetric
  - positive definite (what is that?)

  also gives a scalar product that satisfies properties 1. – 5. (please check!)
- In fact, a $\mathbf{G}_0$ that is symmetric and positive definite is both sufficient and necessary in this case.

- **<u>Consequently</u>:** there are many ways to define a scalar product space based on $\mathbb{R}^N$

Examples of scalar product spaces

- Similarly, the set of all ordered *N*-tuples of complex numbers, $\mathbb{C}^N$, forms a vector space
- We represent such ordered *N*-tuples as columns of *N* complex numbers
- A scalar product between vectors **a** and **b** is given as $\langle \mathbf{a} | \mathbf{b} \rangle = \mathbf{b}^* \mathbf{G}_0 \mathbf{a}$
  - where $\mathbf{G}_0$ is an $N \times N$ *Hermitian* matrix
  - where $\mathbf{G}_0$ also is positive definite

Examples of scalar product spaces

- The set of $N \times M$ real matrices is a vector space (how?)
- As a scalar product between matrices **A** and **B** in this vector space can, for example, be defined as

$$\langle \mathbf{A} | \mathbf{B} \rangle = \text{trace}(\mathbf{B}^T \mathbf{A})$$

Frobenius scalar product

Examples of scalar product spaces

- More interesting cases of vector spaces are generated by considering specific sets of functions
- This has to be done with some care to assure that
  - the sum of two functions is again in the set
    - convergence of Cauchy sequences
  - a scalar product exists

Examples function spaces

Examples:

- Polynomials
  or more often: polynomials up to order p
- Functions that are periodic with period T
- Functions that are continuously differentiable d times
- Functions that are zero outside some region Ω
- Square integrable functions (L^2) (what is that?)

Examples of scalar product spaces of functions

- As scalar product between two functions we often use

$$\langle f|g\rangle = \int f(x)g^*(x)dx$$

- The range of integration can be chosen as
  - For polynomials: an arbitrary finite interval
  - For T-periodic functions: x_0 to x_0 + T
  - Compactly supported functions: Ω
  - L^2: $-\infty$ to $+\infty$

Examples of scalar product for function spaces

- In some applications we may also want to consider scalar products defined as

$$\langle f | g \rangle = \int f(x) w(x) g^*(x) dx$$

- In this case we assume that $w(x) > 0$ (why?)

Henceforth

- We will only consider scalar-product spaces
- They are either of type $\mathbb{R}^N$, $\mathbb{C}^N$, or some function space
- They will often be finite dimensional
- Occasionally, we will venture into the infinite dimensional case
  - With intuitive reasoning, rather than based on rigor mathematics
- Functions can be of a single or multiple variables
- We will often consider the case that the scalar product space and a basis is already given
  - **In general, the basis is not orthogonal!**

Coordinates

- Given that we have a scalar product space V with a basis $\mathbf{b}_k$, $k = 1, \ldots N$
- And some vector $\mathbf{v} \in V$
- How do we determine the coordinates c_k of $\mathbf{v}$ relative the basis?

This is one of the main themes in this part of the course!!

$$\mathbf{v} = \sum_{k=1}^{N} c_k \mathbf{b}_k$$

Signal processing

Typical operations in signal processing

- Convolution (filtering)
- Transformation (e.g. Fourier transform)
- Sampling
  - From continuous-time signal to discrete-time signal
- Reconstruction
  - From discrete-time signal to continuous-time signal

Signal spaces

- Signals are functions of time and/or spatial position
  - Audio (time)
  - An image (2D spatial)
  - Video (time+2D spatial) - aliased!!
  - MRI volume (3D spatial)
  - ...
- We are also interested in other types of functions, e.g., filters or analysing functions that, from a practical point of view, are not proper signals.
  - These, too, are here referred to as “signals”

Continuous and discrete signals

- The *domain* of the signal function (of time and/or position) is either
  - a continuous set $\mathbb{R}^D$ (”continuous signals”)
  - a discrete set $\mathbb{Z}^D$ (”discrete signals”)

The set of all integers

- *D* is the dimension of the domain
  - For example, *D*=2 for an image
  - A.k.a. *outer dimension* of the signal

Infinite or truncated domain

- Furthermore, the domain of the signal can either be regarded as infinite or truncated
- When we make an analysis of an audio signal, we often assume that it has well-defined values for $t \in [-\infty, \infty]$
- In practice, both an audio signal, and more typically an image, have well-defined values only for a truncated domain, e.g., $t \in [t_1, t_2]$

The codomain of the signal

- The signal function maps its domain to a set called the *codomain*
- For many signals, the codomain can be assumed to be the set of real numbers, $\mathbb{R}$
  - Audio
  - Gray-scale images
  - MRI density volumes

The codomain of the signal

- For some signals, the codomain is rather a set/vector of real numbers:
  - Colour images
  - ...
- In the case that the codomain is a vector space, its dimension is the *inner dimension* of the signal

Digitised and truncated codomain

- In many practical cases, $\mathbb{R}$ may not be the appropriate codomain if the signal values are both sampled and digitised
  - Example: pixel values from a digital camera are represented by integers
- For certain signals, we know that signal values are always positive
  - Example: Image intensities are always positive

The signal vector

- Despite the various types of signals that may emerge from practical problems, we choose to describe them here as *vectors*, elements of a well-defined *vector space*.

Convolution

- The convolution between functions *f* and *g* of a discrete and 1-dimensional variable:

$$h[n] = \sum_k f[n-k]\, g[k]$$

where summation is made over some subset of $\mathbb{Z}$

  - In general, summation over the infinite set $\mathbb{Z}$

- Example: *f* is a signal, *g* is the filter, *h* is the filtered signal. **All signals are assumed real**

Convolution

- Algebraically, the convolution operation has two faces, two interpretations:
  - It can be seen as a *linear combination* of vectors
  - It can be seen as computing *scalar products*

- Which face we want to see depends on the application, both are valid
  - But usually only one fits your particular problem

Convolution as a linear combination

- We can see $h[n]$ as a vector in the signal space V

$$h[n] = \sum_k f_k[n] c_k$$

- with $f_k[n] = f[n - k]$ and $c_k = g[k]$

Convolution as a linear combination

- This view of the convolution operation implies that *h* is a **linear combination** of shifted versions of the signal *f* together with the elements of the filter *g*
- Convolution is a commutative operation:

  $$g = f * g = g * f$$

  $\Rightarrow$ we can also describe *h* as shifted versions of the filter *g* in a linear combination with the elements of the signal *f*
- If *g* has finite support (a FIR-filter), then *h* is a ”proper” (finite) linear combination

Convolution as scalar products

- Alternatively, we can see each element in *h* as the result of a **scalar product** between two function vectors, one derived from the input signal and one derived from the filter.

$$h[n] = \sum_k f[n-k]g[k]$$

Convolution as scalar products

- For a fixed n_0, we get the scalar $h[n_0]$ as

$$h[n_0] = \sum_k f[k]\, p[k]$$

where

$$p[k] = g[n_0 - k]$$

g is mirrored around n_0

Convolution as scalar products

- Formally we can write this as

$$h[n_0] = \langle f | p \rangle$$

where *p* and *g* are function vectors

- Notice that the function *p* depends on n_0

Convolution as scalar products

- Alternatively, we can write $h[n_0]$ as

$$h[n_0] = \sum_k f[k]\, g[n_0 - k]$$

or

$$h[n_0] = \langle g[n_0 - k] | f[k] \rangle$$

This is a sloppy, but convenient, way of writing $h[n_0]$

Convolution as scalar products

- Since convolution is commutative ($f * g = g * f$), we now have 4 different ways of expressing $h[n_0]$:

$$h[n_0] = \langle f[k] | g[n_0 - k] \rangle$$
$$h[n_0] = \langle g[n_0 - k] | f[k] \rangle$$
$$h[n_0] = \langle g[k] | f[n_0 - k] \rangle$$
$$h[n_0] = \langle f[n_0 - k] | g[k] \rangle$$

In all cases, the summation in the scalar product is made over variable *k*

Convolution as scalar products

- In the summation over *k*, we can either go from $-\infty$ to ∞, or vice versa
- This gives four more possibilities:

$$h[n_0] = \langle f[-k] | g[n_0 + k] \rangle$$
$$h[n_0] = \langle g[n_0 + k] | f[-k] \rangle$$
$$h[n_0] = \langle g[-k] | f[n_0 + k] \rangle$$
$$h[n_0] = \langle f[n_0 + k] | g[-k] \rangle$$

Convolution: summary

- We have seen that the convolution $f * g$ operation produces
  - a function *h* that can be seen as
    - shifted versions of f linearly combined with g
    - shifted versions of g linearly combined with f

  - Function values $h[n_0]$ that can be seen as (at least) 8 different types of scalar products between functions derived from f, g, and n_0

Generalization

This idea generalizes directly to

- Complex valued functions (how?)
- Functions of 2 or more variables (how?)
- Functions of continuous variables (how?)

Fourier transform

- There are different (and equivalent) ways of defining the Fourier transform
- For example, we can use the following definition:

$$F(\mathbf{u}) = \int_{\mathbb{R}^D} f(\mathbf{x})\, e^{-i\mathbf{x}^T\mathbf{u}}\, d\mathbf{x}$$

$$f(\mathbf{x}) = \frac{1}{(2\pi)^D} \int_{\mathbb{R}^D} F(\mathbf{u})\, e^{i\mathbf{x}^T\mathbf{u}}\, d\mathbf{u}$$

Integration is made of the entire variable domain $\mathbb{R}^D$

The inverse Fourier transform

- The expression of the inverse Fourier transform suggests that the function *f* (not the function value *f*(**x**)!) can be interpreted as the functions

$$\frac{e^{i\mathbf{x}^T\mathbf{u}}}{(2\pi)^D}$$

linearly combined with *F*(**u**)

This is a function of **x**, but it is also indexed (or enumerated) by the variable **u**

The Fourier transform

- We know that any suitable *f* (which?) can be transformed to *F*, and then inversely transformed back again
- This suggests that the functions

$$\frac{e^{i\mathbf{x}^T\mathbf{u}}}{(2\pi)^D}$$

form a basis for this function set and that *F*(**u**) are the coordinates of *f* in this basis

The Fourier transform

- This idea is intuitively correct, but is complicated to show formally since we then have to consider sets of vectors that are not only infinite, but also uncountable
  - since they are indexed by the variable $\mathbf{u} \in \mathbb{R}^D$

- We will return to the Fourier transform when we have defined dual bases

Sampling

The *sampling theorem* states that

- If $f(x)$ is band-limited to the interval $[-\pi, \pi]$

  $F(u) = 0$ outside the interval

- It can be sampled at integer values of x:

  sample k: $s[k] = f(k), \;\; k \in \mathbb{Z}$

- such that $f(x)$ can be reconstructed as

$$f(x) = \sum_{k=-\infty}^{\infty} s[k] \operatorname{sinc}(x-k) \qquad \operatorname{sinc}(x) = \frac{\sin \pi x}{\pi x}$$

Sampling

- The reconstruction formula suggests that *f*(*x*) can be written as a set of integer shifted sinc-functions sinc(*x*-*k*) **linearly combined** with *s*[*k*]
- Furthermore, the functions sinc(*x*-*k*) appear to form a basis for the set of $[-\pi, \pi]$ band-limited functions ($k \in \mathbb{Z}$)
  - At least if they are linearly independent!
- The sample values *s*[*k*] must then be the coordinates of *f* in this basis

Sampling

- The sample values *s*(*k*) can be written as

$$s[k] = f(k) = \int_{-\infty}^{\infty} f(x)\, \delta(x-k)\, dx$$

which intuitively can be written as

$$s[k] = \langle f(x) | \delta(x-k) \rangle$$

Sampling

- This expression, however, is formally not correct since the *sampling functions* $\delta(x-k)$ are not band limited (why?)
- However, what about using other sampling functions?
- For example: How can $f(x)$ be reconstructed if we instead sample it with rectangular functions?

Conclusion

- In order to derive such a generalisation of the sampling theorem we need to better understand how coordinates and bases are related in the general case:
  - For general ($\mathbf{G}_0 \neq \mathbf{I}$) scalar product spaces
  - For general (non-orthogonal) bases

- We need dual bases!

What you should know includes

- Generalizations of the scalar product
- Examples of vectors spaces, other than $\mathbb{R}^n$
- Convolution described either as scalar products or as a linear combination
- Inverse Fourier transform as computing linear combinations
- Sampling defined as computing scalar products
- Reconstruction from sampling by linear combinations (of sinc-functions)