II. Linear Inverse Problems and Least-Squares Signal Processing

In this chapter of the course, we will focus on problems of the form:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$
$$(\text{vector in } \mathbb{R}^M) = (M \times N \text{ matrix})(\text{vector in } \mathbb{R}^N).$$

We are given $\boldsymbol{A}$, we observe $\boldsymbol{y}$ and want to find (or estimate) $\boldsymbol{x}$.

This is call a **linear inverse problem**, and is one of the most important and fundamental concepts in all of engineering, science, and applied mathematics.

You can think of the entries of $\boldsymbol{y}$ as containing different indirect **observations** or **measurements** of the unknown vector $\boldsymbol{x}$:

$$\boldsymbol{y} = \begin{bmatrix} \langle \boldsymbol{x}, \boldsymbol{a}_1 \rangle \\ \langle \boldsymbol{x}, \boldsymbol{a}_2 \rangle \\ \vdots \\ \langle \boldsymbol{x}, \boldsymbol{a}_M \rangle \end{bmatrix},$$

where $\boldsymbol{a}_m^{\mathrm{T}}$ (or $\boldsymbol{a}_m^{\mathrm{H}}$ if the entries are complex) is the mth row of $\boldsymbol{A}$. Each entry of $\boldsymbol{y}$ is a different **linear functional** of $\boldsymbol{x}$, an inner product of a known vector $\boldsymbol{a}_m$ against the unknown vector $\boldsymbol{x}$.

We will be interested in cases where $M > N$ (more observations than unknowns), $M = N$ (exactly as many observations as unknowns), and $M < N$ (fewer observations than unknowns). If an exact solution does not exist (which in general it does not), we want a principled way to do the best we can. We also want everything we do to be **stable** in the presence of noise in the observations.

The first thing we will do is see some examples of how systems of equations of this type arise in typical signal processing problems.

Discretizing inverse problems

In many real-world applications, the signal or image we are measuring is a function of a continuous variable (or variables for images). Of course, if we are going to reconstruct the signal/image on a computer, our answer will ultimately be discrete. In this section, we discuss a general way to discretize linear inverse problems using a basis representation.

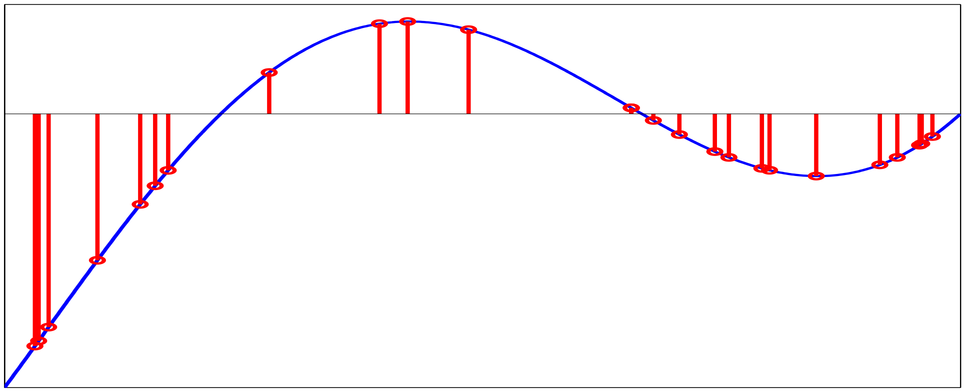
We will start with three concrete examples: reconstruction from non-uniform samples, deconvolution of a continuous-time signal, and the 2D tomography problem ("reconstruction from projections"). From these three examples, it should be clear how the essential framework can be generalized.

Example: Reconstruction from non-uniform samples

We have seen methods for recovering continuous-time signals from samples that are uniformly spaced in time; these have all reduced (in one way or another) to some kind of interpolation between the samples. When the samples are non-uniform, the problem requires a different approach.

We observe M samples of a continuous-time signal $f(t)$ at *non-uniform* locations $t_1, \ldots, t_M$,

$$y[m] = f(t_m), \quad m = 1, \ldots, M.$$



We will not be able to do much without some kind of assumption on the signal $f(t)$ — if we only assume that it is finite energy, then there are of course an infinite number of ways we could "connect the dots" between the samples. Our operating assumption will be that $f(t)$ lies in (or at least close to) an N-dimensional subspace spanned by a basis $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N$. For example, if $f(t)$ is non-zero on an interval $[0, T]$, and we take

$$\psi_n(t) = \begin{cases} 1, & n = 1, \\ \cos(2\pi(n-1)t/T), & n = 2, \ldots, (N+1)/2, \\ \sin(2\pi(n-(N+1)/2)t/T), & n = (N+3)/2, \ldots, N \end{cases}$$

for odd N, the basis functions $\{\boldsymbol{\psi}_n\}$ correspond to a partial Fourier series expansion, and our linear model is something akin to $f(t)$ being bandlimited. This is one example of a basis which might be used; the proper choice involved carefully assessing prior information we have about the signal being samples, and translating that information into a basis model.

To set this up the signal recovery as a linear inverse problem, we expand $f(t)$ using an N-dimensional basis $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N$:

$$f(t) = \sum_{n=1}^{N} x[n]\psi_n(t),$$

where the $x[n]$ are the expansion coefficients for $f(t)$ — again, knowing the $x[n]$ is the same as knowing $f(t)$. The samples $y[m]$ can then be written as a linear combination of samples (at the same locations) of each of the basis functions:

$$y[m] \;=\; f(t_m) \;=\; \sum_{n=1}^{N} x[n]\psi_n(t_m).$$

We can rewrite the expression above as the matrix-vector product:

$$\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[M] \end{bmatrix} = \begin{bmatrix} \psi_1(t_1) & \psi_2(t_1) & \cdots & \psi_N(t_1) \\ \psi_1(t_2) & \psi_2(t_2) & \cdots & \psi_N(t_2) \\ \vdots & & & \vdots \\ \psi_1(t_M) & \psi_2(t_M) & \cdots & \psi_N(t_M) \end{bmatrix} \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[N] \end{bmatrix}$$

By stacking up the observed samples values into $\boldsymbol{y} \in \mathbb{R}^M$ and the unknown basis expansion coefficients into $\boldsymbol{x} \in \mathbb{R}^N$, we can rewrite the expression above as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x},$$

where the $M \times N$ matrix $\boldsymbol{A}$ has entries

$$A[m, n] = \psi_n(t_m).$$

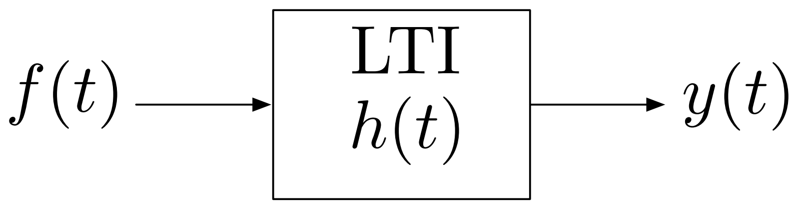
To recover the input $f(t)$, we solve the system of equations above to get $\hat{\boldsymbol{x}}$, and then synthesize

$$\hat{f}(t) = \sum_{n=1}^{N} \hat{x}[n]\, \psi_n(t).$$

The coefficients are captured in a finite-dimensional vector, but they specify a function of a continuous variable.

In this example and in those that follow, the reconstruction of course depends on the basis that was chosen. If the $\{\boldsymbol{\psi}_n\}_n$ are chosen improperly, so that the true underlying signal is not close to their span, the reconstruction will in general not be accurate.

Example: Deconvolution



Problem: given the output $y(t)$ of a linear time-invariant (LTI) system with known impulse response $h(t)$, determine the input $f(t)$.

Applications in which this problems arises are manifold:

- image deblurring
- seismology
- channel equalization in digital communications
- $\cdots$

We observe (samples of) a continuous-time function $y(t)$ which is the result of a *linear operator* applied to another continuous-time signal $f(t)$:

$$y(t) = \int_{-\infty}^{\infty} h(t-\tau) f(\tau) \, \mathrm{d}\tau.$$

We can turn this into a discrete matrix problem using a basis expansions for $f(t)$.

Suppose, for example, that $f(t)$ is *time-limited*, in that we know $f(t)$ is zero outside of $[0, T]$. Let $\{\psi_n\}$ be a basis for $L_2([0, T])$. (We have seen many different examples of bases for signals time-limited to an interval.) Then any $f(t)$ can be written as

$$f(t) = \sum_n x[n]\, \psi_n(t),$$

and recovering the expansion coefficients $x[n]$ is the same as recovering $f(t)$. We can re-write the integral equation above as

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} h(t-\tau) \sum_n x[n]\, \psi_n(\tau)\ \mathrm{d}\tau \\ &= \sum_n x[n] \left(\int_{-\infty}^{\infty} h(t-\tau)\, \psi_n(\tau)\ \mathrm{d}\tau \right). \end{aligned}$$

In practice, we will in general not observe the continuous-time signal $y(t)$, but rather observe a finite set of **samples** of $y(t)$. Suppose we observe M samples at times $t = t_1, t_2, \ldots, t_M$; for what we are doing here it does not matter whether the samples are equally spaced or not. Now we can write our M observations of $f(t)$ as

$$\begin{aligned} y[m] := y(t_m) &= \sum_n x[n] \left(\int_{-\infty}^{\infty} h(t_m-\tau)\, \psi_n(\tau)\ \mathrm{d}\tau \right) \\ &= \sum_n A[m,n]\, x[n] \end{aligned}$$

where

$$A[m,n] = \int_{-\infty}^{\infty} h(t_m-\tau)\, \psi_n(\tau)\ \mathrm{d}\tau = \langle \boldsymbol{h}_m, \boldsymbol{\psi}_n \rangle,$$

where $\boldsymbol{h}_m$ is the function

$$h_m(t) = h(t_m - t).$$

Also in practice, we will only be able to recover a finite number of the expansion coefficients $x[n]$. Say we settle for recovering $x[n]$, $n = 1, \ldots N$. Then the M samples of $y(t)$ can be written as different linear combinations of the N expansion coefficients:

$$y[m] = \sum_{n=1}^{N} A[m,n]\, x[n], \quad m = 1, \ldots, M.$$

If we collect all of the $A[m,n]$ into a $M \times N$ matrix $\boldsymbol{A}$, all of the observations $y[m]$ into the vector $\boldsymbol{y} \in \mathbb{R}^M$, and all of the unknown expansion coefficients $x[n]$ into the vector $\boldsymbol{x} \in \mathbb{R}^N$, we can write the deconvolution problem as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

Again, the solution $\hat{\boldsymbol{x}}$ we get by solving the system of equations above is a vector in $\mathbb{R}^N$, but it specifies as continuous-time signal which we synthesize using

$$\hat{f}(t) = \sum_{n=1}^{N} \hat{x}[n]\, \psi_n(t).$$

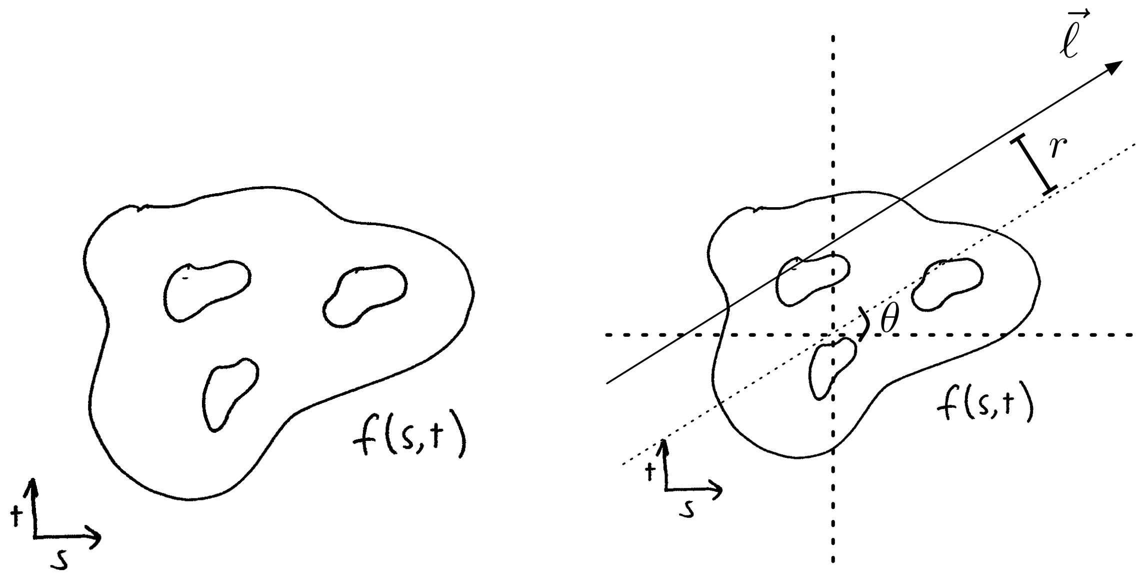
Example: Tomographic reconstruction

There are many situations where we would like to look inside of an object without have to cut it open. This is particularly true in medical imaging applications, where we would like to get a picture of a person's internal tissue structure in a non-invasive manner.

One method to learn about the interior of an object while only taking measurements on the exterior is **tomography**. When you get a CAT scan on your head, X-rays of a known intensity are emitted on one side of your head, and the intensity is measured as it exits the other side of your head. This is done at many different angles and orientations. The idea is that each of these measurements tells us about the *net absorption* of all the tissues along a narrow path. Below, we will see that a collection of such measurements can be untangled to form a coherent picture of the internal tissue structure.

The Radon transform

In the 2D tomographic reconstruction problem, the image $f(s,t)$ we wish to acquire is sampled using line integrals. We can parameterize a line $\vec{\ell}$ using an offset r and an angle θ as shown below:



The line $\vec{\ell}$ is the set of points obeying a linear constraint:

$$\vec{\ell} = \{(s,t) \ : \ -s\sin\theta + t\cos\theta = r\}$$

The integral of $f(s,t)$ along $\vec{\ell}$ is given by

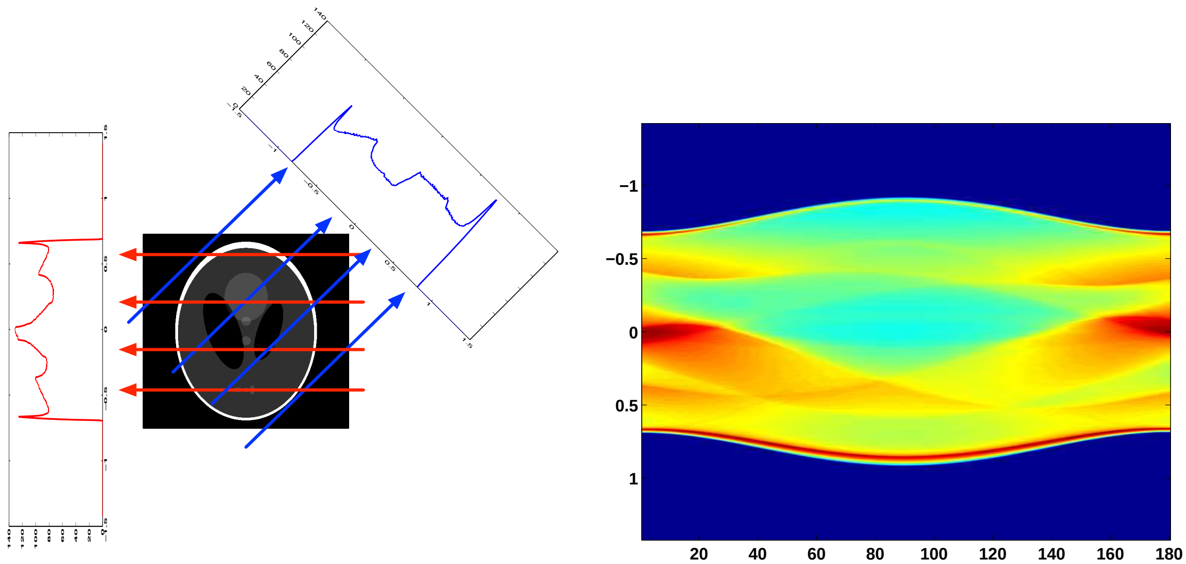
$$R_{r,\theta}[\boldsymbol{f}] = \begin{cases} \int f\left(s, \frac{r+s\sin\theta}{\cos\theta}\right)\,\mathrm{d}s & |\theta| \leq \pi/4 \\ \int f\left(\frac{t\cos\theta - r}{\sin\theta}, t\right)\,\mathrm{d}t & \pi/4 < |\theta| \leq \pi/2 \end{cases}.$$

Of course, these expressions are equal to one another except when $\theta = 0, \pi/2$. Note also that the measurements are unique only over a range of π, as $R_{r,\theta+\pi}[\boldsymbol{f}] = R_{-r,\theta}[\boldsymbol{f}]$. It is sometimes convenient to write the line integral as a 2D integral of $f(s,t)$ against a *delta ridge*:

$$R_{r,\theta}[\boldsymbol{f}] = \int\int f(s,t)\delta(-s\sin\theta + t\cos\theta - r)\ \mathrm{d}s\ \mathrm{d}t, \qquad (1)$$

where $\delta(\cdot)$ is the Dirac delta function.

The collection of all such line integrals $\{R_{r,\theta}[\boldsymbol{f}],\ \theta \in [0,\pi],\ r \in \mathbb{R}\}$ is called the **Radon transform** of $f(s,t)$. The radon transform is itself a continuous function of two variables. The figure below show an illustrative example: on the left, we see $R_{r,\theta}$ of a test image as a function of r for two different fixed values of θ. On the right is the collection $R_{r,\theta}$ as a function of both r and θ.



Left: $R_{r,\pi/4}[\boldsymbol{f}]$ *and* $R_{r,\pi}[\boldsymbol{f}]$ *as a function of* r*, where* $f(s,t)$ *is the Shepp-Logan phantom. Right: The Radon transform of the phantom. The rows are indexed by* r *and the columns by* θ *(in degrees).*

Reconstruction from a discrete set of line integrals

Given measurements $y[m]$, $m = 1,\ldots,M$ corresponding to line integrals at different different offsets r_m and angles θ_m (i.e. a finite set of samples of the Radon transform), which have possibly been corrupted by noise, we would like to estimate the underlying image $f(s,t)$. If the measurements are dense in (r,θ) space, the natural approach to this problem is to use filtered backprojection. Our focus here will be setting this problem up as finite linear inverse problem

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \text{noise}, \quad \boldsymbol{y} \in \mathbb{R}^M, \boldsymbol{x} \in \mathbb{R}^N$$

so that it can be attacked with the general set of tools for solving such problems (e.g. least-squares).

We start by choosing a finite-dimensional space $\mathcal{V}$ in which to perform the reconstruction that comes equipped with a set of N basis vectors $\{\psi_\gamma(s,t)\}$. We will use the general index $\gamma \in \Gamma$ where Γ is a set of size N as, depending on the basis, it may be convenient to index the basis in different ways (i.e. by integers, pairs of integers over the same range, pairs of integers over different ranges, etc.).

For example, if $f(s,t)$ is non-zero only for $(s,t) \in [0,1]^2$, we might take our reconstruction space $\mathcal{V}$ to be the set of all "pixellated" images — images that are piecewise-constant on squares of side length $1/n$ for some integer n. A natural basis for this space is the set of indicator functions on these squares:

$$\psi_{j,k}(s,t) = \begin{cases} 1 & s \in [j/n, (j+1)/n],\ t \in [k/n, (k+1)/n] \\ 0 & \text{otherwise} \end{cases}$$

Using our general index notation, we can write any $f(s,t) \in \mathcal{V}$ as

$$f(s,t) = \sum_{\gamma \in \Gamma} x[\gamma] \psi_\gamma(s,t),$$

where $\Gamma = \{(j,k) \ : \ j,k = 0,1,\ldots,n-1\}$ with size $N = n^2$, and the $x[\gamma] \in \mathbb{R}$ are the basis expansion coefficients, which are, in this case, the pixel values. (Another natural basis for V would be the two-dimensional Haar basis we encountered earlier). The point is that knowing the discrete set of coefficients $x[\gamma]$ for all $\gamma \in \Gamma$ is the same as knowing the continuous-space function $f(s,t)$.

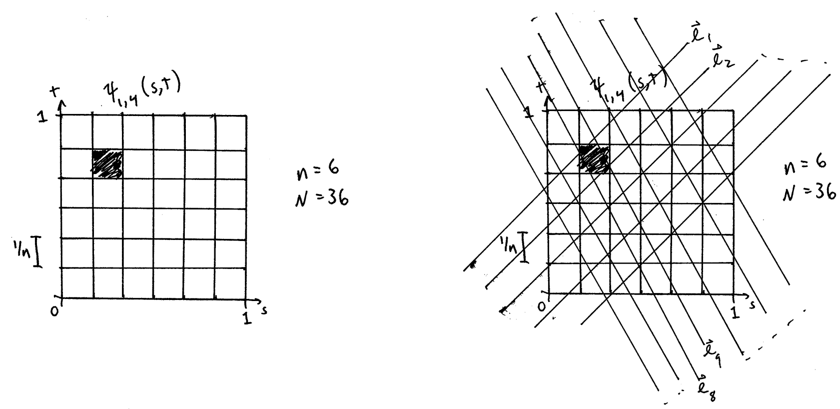
We can also write the measurements of an $f(s,t) \in \mathcal{V}$ in terms of the basis functions:

$$\begin{aligned}
y[m] &= R_{r_m,\theta_m}\left[\sum_{\gamma\in\Gamma} x[\gamma]\psi_\gamma(s,t)\right] \\
&= \sum_{\gamma\in\Gamma} x[\gamma] R_{r_m,\theta_m}\left[\psi_\gamma(s,t)\right] \qquad \text{(since } R_{r,\theta}[\cdot] \text{ is linear)} \\
&= \sum_{\gamma\in\Gamma} A[m,\gamma]x[\gamma] \qquad \text{where } A[m,\gamma] = R_{r_m,\theta_m}\left[\psi_\gamma(s,t)\right],
\end{aligned}$$

which can be written in more compact form as

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{2}$$

The entries of the $M \times N$ matrix $\boldsymbol{A}$ contain the results of each of the M measurements functionals $R_{r_m,\theta_m}[\cdot]$ applied to each of the N basis functions $\psi_\gamma(s,t)$, the N-vector $\boldsymbol{x}$ contains the expansion coefficients for $f(s,t)$ in the basis $\{\boldsymbol{\psi}_\gamma\}$, and $\boldsymbol{y}$ contains the M measurements. This is illustrated in the figure below. As we can see, not too many of the $\vec{\ell}_m$ pass through a given pixel, meaning that the matrix $\boldsymbol{A}$ will be very sparsely populated.



Left: A sketch of one of the basis functions $\psi_\gamma(s,t)$ from the discussion above. Right: The entries of $\boldsymbol{A}$ in the column indexed by γ will be the result of measuring the basis function $\psi_\gamma(s,t)$: $A[m,\gamma] = R_{r_m,\theta_m}[\psi_\gamma]$.

Of course, the true underlying image will in general not lie in the chosen finite-dimensional subspace $\mathcal{V}$. This means that even when there is no measurement noise, there will still be some inherent error in our calculations. But solving (2) will in some sense find the member of $\mathcal{V}$ that best explains the measurements that have been observed. If the true image can be closely approximated by a member of $\mathcal{V}$, then we will not lose much through this discretization. A major consideration in choosing the space $\mathcal{V}$ is how well we can use it to approximate images we expect to encounter.

General linear operators

The discretization technique above can be very naturally generalized to different kinds of measurement operators that map signals of a continuous variable(s) into $\mathbb{R}^M$. All we need is that the **functionals** that map the continuous-time signal $f(t)$ to the measurements $\{y[m]\}$ are **linear**.

Let's make this a little more precise mathematically. The word "functional" means a mapping from a function to a number — in this case, it is from the underlying signal $f(t)$ to a measurement $y[m]$ — we will use[1] $\mathscr{L}_m(\cdot) : L_2(\mathbb{R}) \rightarrow \mathbb{R}$ to denote these functionals. A **linear functional** simply means that

$$\mathscr{L}_m(\alpha \boldsymbol{f} + \beta \boldsymbol{g}) = \alpha \mathscr{L}_m(\boldsymbol{f}) + \beta \mathscr{L}_m(\boldsymbol{g}), \quad \text{for all} \quad \boldsymbol{f}, \boldsymbol{g} \in L_2(\mathbb{R}).$$

Examples of linear functionals include things we have seen above. For instance, sampling at a known location t_m,

$$\mathscr{L}_m(\boldsymbol{f}) = f(t_m)$$

[1]The input space doesn't necessarily need to be $L_2(\mathbb{R})$; it can be any linear space of continuous-time signals.

and integrating against a known function $\phi_m(t)$

$$\mathscr{L}_m(\boldsymbol{f}) = \int_{-\infty}^{\infty} f(t)\phi_m(t)\ \mathrm{d}t,$$

are both linear functionals. Note that the convolution example above falls into the latter category with $\phi_m(t) = h(t - t_m)$.

If we have a finite basis decomposition, then discretization works exactly as it did in the special cases above:

$$y[m] = \mathscr{L}_m(\boldsymbol{f}) = \mathscr{L}_m\left(\sum_{n=1}^{N} x[n]\boldsymbol{\psi}_n\right) = \sum_{n=1}^{N} \mathscr{L}_m(\boldsymbol{\psi}_n)x[n],$$

and so we have the matrix equation

$$\begin{bmatrix} y[1] \\ y[2] \\ \vdots \\ y[M] \end{bmatrix} = \begin{bmatrix} \mathscr{L}_1(\boldsymbol{\psi}_1) & \mathscr{L}_1(\boldsymbol{\psi}_2) & \cdots & \mathscr{L}_1(\boldsymbol{\psi}_N) \\ \mathscr{L}_2(\boldsymbol{\psi}_1) & \mathscr{L}_2(\boldsymbol{\psi}_2) & \cdots & \mathscr{L}_2(\boldsymbol{\psi}_N) \\ \vdots & & & \vdots \\ \mathscr{L}_M(\boldsymbol{\psi}_1) & \mathscr{L}_M(\boldsymbol{\psi}_2) & \cdots & \mathscr{L}_M(\boldsymbol{\psi}_N) \end{bmatrix} \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[N] \end{bmatrix}$$

or $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. Notice that since the entries of $\boldsymbol{A}$ do not depend on $\boldsymbol{f}$, they can be pre-computed.