# Computational Sensing and Imaging

Imperial College London

Liyang DOU

## Contents

# Introduction

**Computational sensing** Co-design of hardware and software (algrithms) $\rightarrow$ enables sensing beyond conventional barriers

- **Hardware and Computational Tools**: imaging toolkit, algorithmic toolkit
- **Plenoptic Models**: spatially coded imaging, temporally coded imaging, light field imaging, multi-spectral imaging
- **Case Studies**: single-pixel imaging, 3D ToF imaging, lifetime imaging, HDR imaging, tomography

### Common Fourier Transform Pairs

| **Time Domain** $f(t)$ | **Frequency Domain** $\hat{f}(\omega)$ |
|---|---|
| $\delta(t)$ | $1$ |
| $1$ | $2\pi\delta(\omega)$ |
| $u(t)$ | $\pi\delta(\omega) + \frac{1}{j\omega}$ |
| $\mathrm{rect}(t)$ | $\mathrm{sinc}\left(\frac{\omega}{2}\right)$ |
| $\mathrm{sinc}(t)$ | $\mathrm{rect}\left(\frac{\omega}{2\pi}\right)$ |
| $e^{j\omega_0 t}$ | $2\pi\delta(\omega-\omega_0)$ |
| $\cos(\omega_0 t)$ | $\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$ |
| $\sin(\omega_0 t)$ | $\frac{\pi}{j}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$ |
| $e^{-at}u(t),\ a>0$ | $\frac{1}{j\omega+a}$ |
| $t^n e^{-at}u(t)$ | $\frac{n!}{(j\omega+a)^{n+1}}$ |

### Key Fourier Transform Properties

| **Time Domain** | **Frequency Domain** |
|---|---|
| $f(t-t_0)$ | $e^{-j\omega t_0}\hat{f}(\omega)$ |
| $f(t)e^{j\omega_0 t}$ | $\hat{f}(\omega-\omega_0)$ |
| $f(at)$ | $\frac{1}{\lvert a\rvert}\hat{f}\left(\frac{\omega}{a}\right)$ |
| $f(-t)$ | $\hat{f}(-\omega)$ |
| $\frac{d^n f(t)}{dt^n}$ | $(j\omega)^n\hat{f}(\omega)$ |
| $t^n f(t)$ | $j^n\frac{d^n\hat{f}(\omega)}{d\omega^n}$ |
| $f(t)*g(t)$ | $\hat{f}(\omega)\cdot\hat{g}(\omega)$ |
| $f(t)\cdot g(t)$ | $\frac{1}{2\pi}(\hat{f}*\hat{g})(\omega)$ |

# 1 Sampling & Quantisation

From Analogue to Digital, we are concerned with two key questions:

- **Theory:** Can a continuous function be fully represented by a discrete sequence?
- **Practice:** How is this discrete representation realised in hardware?

**Question 1: Nyquist–Shannon Sampling Theorem** If a function $f(t)$ contains no frequencies higher than $\Omega$ (rad/sec), then it can be completely reconstructed from its samples spaced:

$$T_{\mathrm{Nyq}} = \frac{\pi}{\Omega} \quad \text{seconds apart.}$$

This is the Nyquist rate. For multidimensional signals, bandlimitedness must be satisfied in *each direction* to prevent information loss. This extends naturally to 2D imaging and 3D voxel acquisition when signals are separable.

**Question 2: Sampling in Practice** To digitise a signal, two key steps are involved:

- **Sampling:** Discretisation of *time* (ADC step 1)
- **Quantisation:** Discretisation of *amplitudes* (ADC step 2)

**Pulse-Code Modulation (PCM)** is based on a **mid-rise quantiser**. For a given bit budget $b$ and dynamic range $\pm\beta$, the amplitude $a$ is quantised as:

$$Q_b^\beta(a) = \begin{cases} \beta & a > \beta \\ \left(\left\lfloor \frac{a}{\delta_\beta} \right\rfloor + \frac{1}{2}\right)\delta_\beta & -\beta < a < \beta \\ -\beta & a < -\beta \end{cases}, \quad \delta_\beta = \frac{2\beta}{2^b - 1}$$

- $\delta_\beta$: bin width (resolution)
- $\lfloor\cdot\rfloor + \frac{1}{2}$: aligns value to the *centre* of the quantisation bin
- Mid-rise: 0 lies between two bins (i.e., not centred in any bin)

Quantisation Error is the difference between the original signal $f[k]$ and its quantised version $\tilde{f}[k]$:

$$e_q[k] = f[k] - \tilde{f}[k]$$

Typically bounded by $\pm\delta_\beta/2$, but accumulates visually (e.g., banding artefacts in images).

There is a trade-off between dynamic range $2\beta$ and resolution $\delta_\beta$. For $b$ bits:

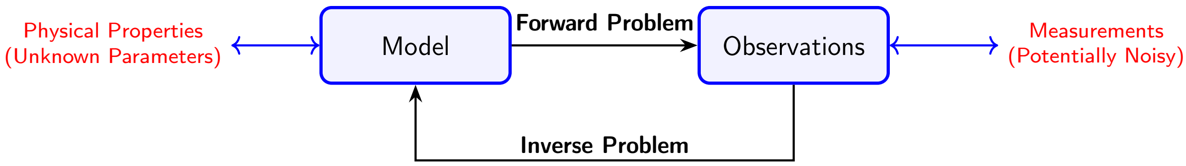
$$\delta_\beta = \frac{2\beta}{2^b - 1}$$

which means higher $b \Rightarrow$ more bins $\Rightarrow$ finer resolution. For fixed $b$, larger dynamic range leads to worse resolution, and vice versa.

**4 Sampling Techniques** Various sampling techniques are proposed to navigate the trade-off between resolution and dynamic range:

1. **Point-wise Sampling** (traditional method)
   Measure instantaneous values of a signal at uniformly spaced time instants using a Dirac-like impulse. This is the default assumption in most theoretical discussions.
2. **Average Sampling**
   Instead of sampling with a delta function, use a local averaging kernel (e.g., boxcar window). This smooths out high-frequency noise, improving robustness to noise while introducing blur due to convolution with a wide kernel.
3. **Unlimited Sampling**
   Not required for the exam. Just be aware that it exists as a theoretical method to bypass amplitude saturation using modulo non-linearity.
4. **One-Bit Sampling (Exam Focus, in Section 7)**
   Quantise the signal using only 1 bit: retain only the sign of the difference between the input and the running integral of the quantised output. This technique enables signal recovery with extremely low quantisation (1-bit) by relying on **temporal oversampling**.

# 2 Modelling: Forward vs Inverse Problems




**Linear Problem Modelling** Many problems of interest involve a linear forward operator $h$. If $h$ is translation-invariant we have:

$$g(t) = (f * h)(t) \quad \text{or} \quad \mathbf{g} = \mathbf{H}\mathbf{f} \text{ (vector matrix form)}$$

where $f(t)$ is the original signal, and $g(t)$ is the measurements. The inverse problem is about recovering $f(t)$ given $g(t), h(t)$, which can be represented in Fourier domain:

$$g(t) = (f * h)(t) \quad \rightarrow \quad \hat{g}(\omega) = \hat{f}(\omega)\hat{h}(\omega) \quad \rightarrow \quad \hat{h}(\omega) = \frac{\hat{g}(\omega)}{\hat{h}(\omega)}$$

**Common Inverse Problems (Not Examined)** Typical problems in computational imaging include:

- **Deconvolution:** Remove the influence of the measurement device (e.g., diffraction, motion blur).
- **Denoising:** Recover signal from noisy observations.
- **Superresolution:** Fuse multiple low-resolution images at sub-pixel shifts to reconstruct a high-resolution image.
- **Frequency Estimation:** Recover frequencies from sampled signals (e.g., EEG) modeled as sums of complex exponentials.
- **Phase Estimation:** Estimate phase shifts due to time-of-flight (ToF) propagation delay.
- **Phase Retrieval:** Recover signal or structure from magnitude-only Fourier measurements.

# 3 Linear Systems and Linear Algebra

## 3.1 Linear Systems

### Definitions

- **Linear system:** Let $u$ and $v$ be the inputs to the linear system $\mathcal{L}$. $\mathcal{L}$ is linear if for all $a, b \in \mathbb{C}$,

$$\mathcal{L}[au + bv] = a\mathcal{L}[u] + b\mathcal{L}[v]$$

- **LTI system:** A system $\mathcal{L}$ is *linear time-invariant* if:

$$g(t) = \mathcal{L}[f](t) \quad \Rightarrow \quad g_\tau(t) = \mathcal{L}[f_\tau](t)$$

  where $f_\tau(t) = f(t - \tau)$ is the time-shifted signal.
- **Impulse response:** The response of an LTI system $\mathcal{L}$ to the Dirac impulse $\delta(t)$ is

$$g(t) = \mathcal{L}[\delta](t)$$

- **Convolution Representation (Exam Focus):** The output of an LTI system to input $f$ is the convolution with the impulse response $g$:

$$\mathcal{L}[f](t) = \int f(\tau)\, \mathcal{L}[\delta_\tau(t)]\, d\tau = (f * g)(t)$$

  Some useful properties of the convolution operator:

- Commutativity: $(f * g)(t) = (g * f)(t)$
- Associativity: $(f * g * h)(t) = (h * f * g)(t) = (g * h * f)(t)$
- Differentiation property: $\frac{d}{dt}(f * g)(t) = \left(\frac{df}{dt} * g\right)(t) = \left(f * \frac{dg}{dt}\right)(t)$
- Dirac convolution property: $f * \delta(t - \tau) = f(t - \tau)$

- **Causality:** A system is causal if its output depends only on past and current inputs:

$$\mathcal{L}[f](t) \text{ does not depend on } f(t'),\ t' > t.$$

- **BIBO stablility:** A system is BIBO stable if for bounded input, the output remains bounded. That is, if $\max_\tau |f(\tau)| < \infty$, then

$$|\mathcal{L}[f](t)| \leq \int |f(\tau)||g(t-\tau)|\, d\tau \leq \max_\tau |f(\tau)| \int |g(t)|\, dt,$$

which holds whenever the impulse response $g(t)$ is absolutely integrable.

- **Eigenfunction:** A function $e(t)$ is an eigenfunction and $\lambda$ is the corresponding eigenvalue of $\mathcal{L}$ if

$$\mathcal{L}[e](t) = \lambda e(t)$$

**Convolution Input-Output Relationship Proof (Exam Focus)** Let an LTI system be fully characterised by its impulse response $g(t)$. Given an input $f(t)$, the output is defined as

$$y(t) = \mathcal{L}[f](t)$$

Using the properties of linearity and time invariance:

1. For an impulse input $\delta(t - \tau)$, the output is $g(t - \tau)$.
2. Any signal $f(t)$ can be represented as:

$$f(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t - \tau)\, d\tau$$

3. Applying the system operator:

$$\mathcal{L}[f](t) = \mathcal{L}\left\{\int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)\, d\tau\right\} \overset{\text{linearity}}{=} \int_{-\infty}^{\infty} f(\tau)\mathcal{L}[\delta(t-\tau)]\, d\tau$$

$$\overset{\text{time invariance}}{=} \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\, d\tau = (f * g)(t)$$

Hence, the output of an LTI system is the convolution of the input signal $f(t)$ and the impulse response $g(t)$. This derivation relies critically on both linearity and time-invariance:

- If the system is **non-linear**, superposition fails.
- If the system is **time-variant**, the response to $\delta(t - \tau)$ is not $g(t - \tau)$, so the expression fails.

**Exponential Eigenfunction Property (Exam Focus)** Let the input be $f(t) = e^{j\omega t}$, where $\omega \in \mathbb{R}$ is a fixed frequency. Then when it passes a LTI system with an impulse response $g(t)$:

$$h(t) = (g * f)(t) = \int_{-\infty}^{\infty} g(\tau)e^{j\omega(t-\tau)}\, d\tau = e^{j\omega t} \underbrace{\int_{-\infty}^{\infty} g(\tau)e^{-j\omega\tau}\, d\tau}_{G(\omega)} = G(\omega) \cdot e^{j\omega t}$$

Hence, $\mathcal{L}[f_\omega](t) = \lambda_\omega f_\omega(t)$, i.e. $e^{j\omega t}$ is an eigenfunction of the LTI system, with eigenvalue $\lambda_\omega = G(\omega)$.

In the same way, real-valued sinusoidal signals are also preserved in form:

- For $f(t) = \cos(\omega t) = \Re\{e^{j\omega t}\}$, output is $h(t) = \Re\{G(\omega)e^{j\omega t}\} = |G(\omega)| \cos(\omega t + \angle G(\omega))$
- For $f(t) = \sin(\omega t) = \Im\{e^{j\omega t}\}$, output is $h(t) = \Im\{G(\omega)e^{j\omega t}\} = |G(\omega)| \sin(\omega t + \angle G(\omega))$

These results follow from Euler's formula and the linearity of the system.

## 3.2 Linear Algebra

### Notations and Definitions

- **Scalars** are single values, e.g., $m \in \mathbb{Z}$, $x_1 \in \mathbb{R}$, $z \in \mathbb{C}$.
- **Vectors** $\mathbf{x} \in \mathbb{C}^N$ are ordered arrays of scalars. Elements: $x_1, \ldots, x_N$.
- **Matrices** $\mathbf{A} \in \mathbb{C}^{M \times N}$ are 2D arrays formed by stacking vectors:

$$\mathbf{A} = [\mathbf{a}_1 \; \mathbf{a}_2 \; \cdots \; \mathbf{a}_N] \quad \text{with} \quad \mathbf{a}_n \in \mathbb{C}^M$$

- **Rank** of a matrix $\mathbf{A}$: number of linearly independent columns (or rows). A full-rank square matrix is invertible.

### Norms

- $p$**-norm**:

$$\|\mathbf{x}\|_p = \left( \sum_{k=1}^{n} |x_k|^p \right)^{1/p} \quad \text{(vector)} \qquad \|f\|_p = \left( \int |f(z)|^p \, dz \right)^{1/p} \quad \text{(function)}$$

- **Euclidean norm (2-norm)**:

$$\|\mathbf{x}\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_N|^2}$$

- **Sum of squares / Energy**:

$$\|\mathbf{x}\|_2^2 = \sum_n |x_n|^2 = \mathbf{x}^\top \mathbf{x}$$

- **Assumption**: All vectors and matrices are real-valued unless otherwise stated. For complex-valued cases, use conjugate transpose: $\mathbf{x}^\top \to \mathbf{x}^H$.

| **Type** | **Function** | **Gradient w.r.t. x** |
|---|---|---|
| Linear | $\mathbf{a}^T\mathbf{x}$ or $\mathbf{x}^T\mathbf{a}$ | $\mathbf{a}$ |
| Linear | $\mathbf{A}\mathbf{x}$ | $\mathbf{A}$ |
| Linear | $\mathbf{x}^T\mathbf{A}$ | $\mathbf{A}^T$ |
| Quadratic | $\mathbf{x}^T\mathbf{x}$ | $2\mathbf{x}$ |
| Quadratic | $\mathbf{x}^T\mathbf{A}\mathbf{x}$, symmetric $\mathbf{A}$ | $2\mathbf{A}\mathbf{x}$ |
| Quadratic | $\mathbf{x}^T\mathbf{A}\mathbf{x}$, general $\mathbf{A}$ | $(\mathbf{A} + \mathbf{A}^T)\mathbf{x}$ |
| Least Squares | $\|\mathbf{x}\|^2 = \mathbf{x}^T\mathbf{x}$ | $2\mathbf{x}$ |
| Least Squares | $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ | $2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$ |
| Least Squares | $(\mathbf{A}\mathbf{x} - \mathbf{b})^T(\mathbf{A}\mathbf{x} - \mathbf{b})$ | $2\mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b})$ |

# 4 Optimisation

## 4.1 Inverse Problems and Least Squares

We consider inverse problems of the form:

$$\mathbf{H}\mathbf{x} = \mathbf{y}$$

where $\mathbf{H}$ models a physical system, $\mathbf{x}$ is the linearised and discretised real-world signal, and $\mathbf{y}$ represents recorded measurements. Solving for $\mathbf{x}$ constitutes the inverse problem, which is typically ill-posed or under-determined.

**Least Squares Formulation** The classical approach formulates the problem as:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$$

which corresponds to minimising the mean squared error (MSE) of estimation. This has a closed-form solution when $\mathbf{H}$ has full column rank.

Depending on the structure of the matrix $\mathbf{H} \in \mathbb{R}^{M\times N}$, the least squares problem falls into one of the following categories. In weighted least squares, $\mathbf{W} \in \mathbb{R}^{M\times M}$ is a symmetric positive-definite diagonal matrix encoding per-measurement confidence weights, i.e. $\sum_n w_n r_n^2$ where $\mathbf{r} = \mathbf{y} - \mathbf{H}\mathbf{x}$.

Regularised LS here denotes $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2$, and weighted regularisation denotes $\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{A}\mathbf{x}\|_2^2$ (note the difference).

| **Case (Condition)** | **Solution Expression** | **Weighted Solution** |
|---|---|---|
| Full Rank (Square, $M = N$) | $\mathbf{x} = \mathbf{H}^{-1}\mathbf{y}$ | $\mathbf{x} = \mathbf{H}^{-1}\mathbf{y}$ |
| Over-determined (Tall, $M > N$) | $\mathbf{x} = (\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{y}$ | $\mathbf{x} = (\mathbf{H}^\top\mathbf{W}\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{W}\mathbf{y}$ |
| Under-determined (Fat, $M < N$) | $\mathbf{x} = \mathbf{H}^\top(\mathbf{H}\mathbf{H}^\top)^{-1}\mathbf{y}$ | $\mathbf{x} = \mathbf{W}^{-1}\mathbf{H}^\top(\mathbf{H}\mathbf{W}^{-1}\mathbf{H}^\top)^{-1}\mathbf{y}$ |
| Regularised LS (Tikhonov) | $\mathbf{x} = (\mathbf{H}^\top\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^\top\mathbf{y}$ | – |
| Weighted Regularisation | $\mathbf{x} = (\mathbf{H}^\top\mathbf{H} + \lambda\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{H}^\top\mathbf{y}$ | – |

**Regularisation** When the problem is ill-posed or under-determined, regularisation is introduced to avoid the problem of rank deficiency:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\mathcal{R}(\mathbf{x})$$

where $\mathcal{R}(\mathbf{x})$ is a regulariser that encodes prior knowledge about $\mathbf{x}$.

Typical regularisers include:

- $\|\mathbf{x}\|_1$: **LASSO** — promotes sparsity via the $\ell_1$-norm; widely used in compressed sensing and feature selection.
- $\|\mathbf{x}\|_2^2$: **Ridge** or **Tikhonov** regularisation — penalises large entries, assuming Gaussian noise.
- $\|\mathbf{D}\mathbf{x}\|_1$: **Total Variation (TV)** — promotes sparsity in gradient domain; $\mathbf{D}$ is the finite-difference operator.
- $\|\mathbf{x}\|_0$: **Cardinality penalty** — counts the number of non-zero elements in $\mathbf{x}$; non-convex and NP-hard, approximated by greedy methods such as OMP.

## 4.2 Constrained Least Squares (CLS)

We now consider the **Constrained Least Squares (CLS, Exam Focus)** problem, where the goal is to minimise the squared error under a linear equality constraint:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.t. } \mathbf{C}\mathbf{x} = \mathbf{b}$$

where $\mathbf{H} \in \mathbb{R}^{M\times N}$, $\mathbf{x} \in \mathbb{R}^N$, and $\mathbf{y} \in \mathbb{R}^M$. The matrix $\mathbf{C} \in \mathbb{R}^{p\times N}$ represents the constraint, and $\mathbf{b} \in \mathbb{R}^p$ is the associated constraint value.

This problem arises when we want the solution to not only fit the measurements approximately (via least squares), but also satisfy some exact physical or structural constraints.

**Solution** To solve the problem, we introduce a Lagrange multiplier $\boldsymbol{\mu} \in \mathbb{R}^p$, and define the Lagrangian:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\mu}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \boldsymbol{\mu}^\top(\mathbf{C}\mathbf{x} - \mathbf{b})$$

We need to eliminate the $\boldsymbol{\mu}$ and represent $\mathbf{x}$ using $\mathbf{H}, \mathbf{y}, \mathbf{C}, \mathbf{b}$.

Take gradients and apply first-order optimality (KKT conditions):

- Stationarity w.r.t. $\mathbf{x}$:

$$\nabla_{\mathbf{x}}\mathcal{L} = -2\mathbf{H}^\top(\mathbf{y} - \mathbf{H}\mathbf{x}) + \mathbf{C}^\top\boldsymbol{\mu} = \mathbf{0}$$
$$\Rightarrow \mathbf{x} = (\mathbf{H}^\top\mathbf{H})^{-1}\left(\mathbf{H}^\top\mathbf{y} - \tfrac{1}{2}\mathbf{C}^\top\boldsymbol{\mu}\right)$$

- Constraint:

$$\mathbf{C}\mathbf{x} = \mathbf{b}$$

- Substitute $\mathbf{x}$ into the constraint:

$$\mathbf{C}(\mathbf{H}^\top\mathbf{H})^{-1}\left(\mathbf{H}^\top\mathbf{y} - \tfrac{1}{2}\mathbf{C}^\top\boldsymbol{\mu}\right) = \mathbf{b}$$

$$\Rightarrow \boldsymbol{\mu} = 2\left(\mathbf{C}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{C}^\top\right)^{-1}\left(\mathbf{C}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{y} - \mathbf{b}\right)$$

- Plug back to obtain closed-form solution:

$$\boxed{\mathbf{x} = (\mathbf{H}^\top\mathbf{H})^{-1}\left[\mathbf{H}^\top\mathbf{y} - \mathbf{C}^\top\left(\mathbf{C}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{C}^\top\right)^{-1}\left(\mathbf{C}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{H}^\top\mathbf{y} - \mathbf{b}\right)\right]}$$

**Special Cases**

- If $\mathbf{H} = \mathbf{I}$ and $\mathbf{b} = \mathbf{0}$, then

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 \quad \text{s.t. } \mathbf{C}\mathbf{x} = \mathbf{0} \quad \Rightarrow \quad \mathbf{x} = \mathbf{y} - \mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}\mathbf{y}$$

- If $\mathbf{y} = \mathbf{0}$, then

$$\min_{\mathbf{x}} \|\mathbf{H}\mathbf{x}\|_2^2 \quad \text{s.t. } \mathbf{C}\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = (\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{C}^\top(\mathbf{C}(\mathbf{H}^\top\mathbf{H})^{-1}\mathbf{C}^\top)^{-1}\mathbf{b}$$

- If $\mathbf{y} = \mathbf{0}$ and $\mathbf{H} = \mathbf{I}$, then

$$\min_{\mathbf{x}} \|\mathbf{x}\|_2^2 \quad \text{s.t. } \mathbf{C}\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = \mathbf{C}^\top(\mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{b}$$

## 4.3 LASSO and Soft Thresholding

**Sparsity and LASSO Regularisation** In many inverse problems, we seek sparse solutions, where most entries are zero. This is particularly useful in compressed sensing, feature selection, and signal denoising. A common approach is to use $\ell_1$-regularisation, leading to the **LASSO** (Least Absolute Shrinkage and Selection Operator) formulation:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

The $\ell_1$-term encourages many components of $\mathbf{x}$ to become exactly zero, promoting sparsity. Geometrically, the $\ell_1$-ball has sharp corners (compared to $\ell_2$-ball), making the optimum more likely to lie on an axis, i.e., have zero coefficients.

**Soft-Thresholding and Closed-Form LASSO Solution (Exam Focus)** We consider the simplified case where $\mathbf{H} = \mathbf{I}$, i.e., identity operator. The LASSO problem becomes:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 = \sum_{n=0}^{N-1}\left((y_n - x_n)^2 + \lambda|x_n|\right)$$

This objective is separable across the components of $\mathbf{x}$, and can be solved coordinate-wise:

$$\min_{x_n} (y_n - x_n)^2 + \lambda|x_n|$$

We analyse three regions of $x_n$:

- If $x_n > 0 \Rightarrow |x_n| = x_n$, objective becomes: $(y_n - x_n)^2 + \lambda x_n$

- If $x_n < 0 \Rightarrow |x_n| = -x_n$, objective becomes: $(y_n - x_n)^2 - \lambda x_n$
- If $x_n = 0$, objective is non-smooth and needs direct comparison.

**Take derivatives in smooth regions:**

$$\frac{d}{dx_n} = -2(y_n - x_n) + \lambda \cdot \operatorname{sgn}(x_n) = 0 \quad \Rightarrow \quad x_n = y_n - \frac{\lambda}{2} \cdot \operatorname{sgn}(x_n)$$

**Combine into soft-thresholding rule:**

$$\begin{aligned}\operatorname{soft}_\lambda(y_n) &= \begin{cases} y_n - \frac{\lambda}{2} & \text{if } y_n > \frac{\lambda}{2} \\ y_n + \frac{\lambda}{2} & \text{if } y_n < -\frac{\lambda}{2} \\ 0 & \text{otherwise} \end{cases} \\ &= \left(y_n - \tfrac{\lambda}{2} \cdot \operatorname{sgn}(y_n)\right) \cdot \mathbb{1}_{|y_n| > \lambda/2}(y_n) \\ &= \max\left(|y_n| - \tfrac{\lambda}{2}, 0\right) \cdot \operatorname{sgn}(y_n)\end{aligned}$$

Hence, the optimal update is:

$$\mathbf{x}^\star = \operatorname{soft}_\lambda(\mathbf{y})$$

Each coordinate is "shrunk" toward zero by $\frac{\lambda}{2}$, and small coefficients are zeroed out. This shrinking behaviour is what makes $\ell_1$-regularisation an effective sparsity prior.

**Remarks**

- The solution is non-linear but piecewise continuous.
- For general $\mathbf{H} \neq \mathbf{I}$, closed-form does not exist — we need iterative methods like ISTA or ADMM.
- This soft-thresholding behaviour explains why LASSO prefers sparse solutions: small coefficients are suppressed, while large ones are retained with shrinkage.

**Sparse Recovery** In real-world scenarios, sometimes we need to recover a **sparse signal** $\mathbf{x} \in \mathbb{R}^N$, i.e., a signal with only a few non-zero components:

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} \end{bmatrix}^\top, \quad \text{where } x_i \neq 0 \text{ for } i \in \mathcal{I}, \quad x_i = 0 \text{ otherwise}$$

The observed measurements $\mathbf{y}$ are related to $\mathbf{x}$ via a linear process:

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad \mathbf{A} \in \mathbb{R}^{M \times N}, \quad M \ll N$$

This is an underdetermined system. To recover $\mathbf{x}$ from $\mathbf{y}$, typical sparse recovery algorithms include:

- **Orthogonal Matching Pursuit (OMP)**: Greedy method that iteratively selects the basis most correlated with the residual.
- **Basis Pursuit (BP)**: Convex relaxation replacing $\ell_0$-norm with $\ell_1$-norm; solved via linear programming.
- **Soft-Thresholding**: Closed-form solution for the LASSO objective when $\mathbf{A} = \mathbf{I}$, applied element-wise to shrink small values.

# 5 High-Resolution Frequency Estimation & Prony's Method

**Problem Formulation** The high-resolution frequency estimation problem seeks to recover sinusoidal components from noisy observations. The signal model is:

$$y_n = \sum_{k=0}^{K-1} \alpha_k e^{j\omega_k n}, \quad n = 0, 1, \ldots, N-1$$

where:

- $y_n$: Noisy sinusoidal measurements
- $\alpha_k$: Amplitude of the $k$-th complex exponential
- $\omega_k \in [-\pi, \pi]$: Frequency of the $k$-th component

Here, $n$ indexes the discrete time instants at which samples are collected. The signal $y_n$ can be interpreted as a linear combination of $K$ damped or undamped complex exponentials, each corresponding to a specific oscillatory frequency. **Goal:** Estimate the parameters $\{\alpha_k, \omega_k\}_{k=1}^{K}$ from $\{y_n\}_{n=0}^{N-1}$.

### Solution Approaches

- **Prony's Method (Exam Focus)**: Models $y_n$ as the output of an all-pole system, fits an annihilating filter, and recovers $\omega_k$ via root finding.
- **Matrix Pencil Method**: Uses shifted Hankel matrices and their eigenstructure to estimate signal poles robustly.

**Prony's Method** Let $z_k = e^{j\omega_k}$. Then the original signal model can be rewritten as:

$$y_n = \sum_{k=0}^{K-1} \alpha_k z_k^n, \quad n = 0, 1, \ldots, N-1$$

**Step 1: Filter Construction** Instead of estimating the parameters directly, we first construct a filter that annihilates the exponential components in $y_n$. This is done by designing a filter whose zeros coincide with $\{z_k\}$, i.e., each $z_k$ is a root of $\hat{h}(z)$. Let $h_n$ for $n = 0, 1, \ldots, K$ be the filter with z-transform:

$$\hat{h}(z) = \sum_{n=0}^{K} h_n z^{-n} = \prod_{k=0}^{K-1} (1 - z_k z^{-1})$$

Note that $h_0 = 1$. Hence:

$$\begin{aligned} h_n * y_n &= \sum_{m=0}^{K} h_m y_{n-m} = \sum_{m=0}^{K} h_m \sum_{k=0}^{K-1} \alpha_k z_k^{n-m} \\ &= \sum_{k=0}^{K-1} \alpha_k z_k^n \left( \sum_{m=0}^{K} h_m z_k^{-m} \right) = \sum_{k=0}^{K-1} \alpha_k z_k^n \hat{h}(z_k) = 0, \quad n \in [K, N-1] \end{aligned}$$

It can be posed as $N - K$ equations:

$$\begin{cases} y_K h_0 + y_{K-1} h_1 + \cdots + y_0 h_K = 0 \\ y_{K+1} h_0 + y_K h_1 + \cdots + y_1 h_K = 0 \\ \qquad \vdots \\ y_{N-1} h_0 + y_{N-2} h_1 + \cdots + y_{N-1-K} h_K = 0 \end{cases}$$

which has an unique solution as long as $N - K \geq K \Rightarrow N \geq 2K$. It can also be written in matrix form as follows, which is a Toeplitz matrix system $\mathbf{T}(\mathbf{y}) \cdot \mathbf{h} = \mathbf{0}$ and $\mathbf{T}(\mathbf{y})$ is at most of rank-$K$.

$$\underbrace{\begin{bmatrix} y_K & y_{K-1} & \cdots & y_0 \\ y_{K+1} & y_K & \cdots & y_1 \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-1} & y_{N-2} & \cdots & y_{N-1-K} \end{bmatrix}}_{=\mathbf{T}(\mathbf{y})} \underbrace{\begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_K \end{bmatrix}}_{=\mathbf{h}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

**Step 2: Constraint Minimisation** To solve annihilating filter $\mathbf{h}$, we pose the following constrained minimisation problem:

$$\min_{\mathbf{h}} \|\mathbf{T}(\mathbf{y})\mathbf{h}\|_2^2, \quad \text{subject to } \|\mathbf{h}\|_2^2 = 1$$

Following the standard solution of CLS problem (Section 4.2), we define the Lagrangian:

$$\mathcal{L}(\mathbf{h}, \mu) = \frac{1}{2}\|\mathbf{T}(\mathbf{y})\mathbf{h}\|_2^2 + \frac{\mu}{2}\left(\|\mathbf{h}\|_2^2 - 1\right)$$

Take derivatives w.r.t. $\mathbf{h}$:

$$\frac{\partial}{\partial \mathbf{h}}\mathcal{L}(\mathbf{h}, \mu) = \mathbf{T}(\mathbf{y})^{\mathrm{H}}\mathbf{T}(\mathbf{y})\mathbf{h} + \mu\mathbf{h}$$

Setting derivatives to zero:

$$\mathbf{T}(\mathbf{y})^{\mathrm{H}}\mathbf{T}(\mathbf{y})\mathbf{h} + \mu\mathbf{h} = 0 \quad \Rightarrow \quad \mathbf{T}(\mathbf{y})^{\mathrm{H}}\mathbf{T}(\mathbf{y})\mathbf{h} = -\mu\mathbf{h}$$

The optimal solution $\mathbf{h}$ is the eigenvector of $\mathbf{T}(\mathbf{y})^{\mathrm{H}}\mathbf{T}(\mathbf{y})$ associated with eigenvalue $-\mu$.
Substituting into the original cost function:

$$\|\mathbf{T}(\mathbf{y})\mathbf{h}\|_2^2 = \mathbf{h}^{\mathrm{H}}\mathbf{T}(\mathbf{y})^{\mathrm{H}}\mathbf{T}(\mathbf{y})\mathbf{h} = \mathbf{h}^{\mathrm{H}}(-\mu\mathbf{h}) = -\mu\|\mathbf{h}\|_2^2 = -\mu$$

Hence, minimising the objective corresponds to choosing the eigenvector with **smallest eigenvalue**.

**Step 3: Recovering Frequencies and Amplitudes** Once $\mathbf{h}$ is found, its z-transform $\hat{h}(z)$ has roots $z_k = e^{j\omega_k}$, from which the **frequencies** $\omega_k$ can be recovered. Furthermore, from the signal model:

$$y_n = \sum_{k=0}^{K-1} \alpha_k z_k^n, \quad n = 0, 1, \ldots, N-1$$

we can reorganise the measurements in matrix form $\mathbf{Z}\boldsymbol{\alpha} = \mathbf{y}$ where:

$$\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_0 & z_1 & \cdots & z_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ z_0^{N-1} & z_1^{N-1} & \cdots & z_{K-1}^{N-1} \end{bmatrix}}_{\mathbf{Z}} \underbrace{\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{K-1} \end{bmatrix}}_{\boldsymbol{\alpha}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{K-1} \end{bmatrix}}_{\mathbf{y}}$$

Since $\mathbf{Z}$ is a tall matrix, the **amplitudes** $\boldsymbol{\alpha}$ can be estimated via least-squares, which gives rise to the final parameters of interest $\{\alpha_k, \omega_k\}_{k=1}^{K}$:

$$\boldsymbol{\alpha} = (\mathbf{Z}^H\mathbf{Z})^{-1}\mathbf{Z}^H\mathbf{y}$$

# 6 Time-of-Flight (Multi-Depth) Imaging

## 6.1 Optical Function (Plenoptic Function)

**(Exam Focus)** The optical (plenoptic) function describes the total light rays in a scene, parameterised by seven variables:

$$P(x, y, z, \theta, \phi, \lambda, t) = P(\mathbf{r}, \theta, \phi, \lambda, t)$$

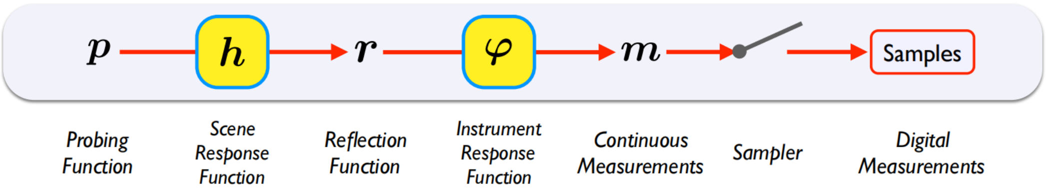
- $\mathbf{r} = (x, y, z)$: Spatial coordinates – where the light is observed.
- $(\theta, \phi)$: Orientation angles (azimuth, elevation) – direction of the light ray.
- $\lambda$: Wavelength – colour or spectral component.
- $t$: Time – temporal variation of light.

In practical imaging systems, the full plenoptic function is marginalised to yield lower-dimensional representations, such as the 2D image $P(x, y)$ captured by a camera sensor.

Traditional imaging systems often assume a steady scene and ignore the temporal dimension $t$. However, recent advances in **Time-Resolved Imaging (TRI)** leverage the time-of-flight of light to capture the dynamic temporal structure of a scene, enabling applications such as depth imaging, transient light transport, and non-line-of-sight vision.

## 6.2 Coded Time-of-Flight Imaging

**(Exam Focus)** We consider an active depth sensing system where a modulated optical signal is emitted and the reflected signal is measured to infer the distance to objects. This framework models the measurement process using a series of convolution operations.




**Imaging Pipeline** The coded Time-of-Flight (ToF) imaging pipeline is driven by a periodic probing signal $p(t)$, which is emitted into the scene. The light interacts with multiple reflective surfaces in the scene, each contributing to a time-delayed and scaled version of the probe. The returned signal $r(t)$ is thus the convolution of the probe with the scene response function $h(t)$.

To measure time delays (and thus infer depth), the system applies a matched filter to the received signal, typically a time-reversed copy of the probe $\bar{p}(t) = p(-t)$. This forms a convolution $r * \bar{p}(t)$, which highlights time alignment between the received signal and the probe.

This filtering process produces the measured signal $m(t)$, which can be viewed as the convolution of the scene response (SRF) $h(t)$ with the instrument response function (IRF) $\phi(t) = p * \bar{p}(t)$. In this context, $\phi(t)$ acts as a low-pass filter because the autocorrelation of a band-limited probe smooths and limits high-frequency content from the scene, suppressing noise and temporal aliasing.

- $p(t)$: Probing signal, periodic in time, i.e., $p(t) = p(t + \Delta)$
- $h(t) = \sum_k \rho_k \delta(t - t_k) = \sum_k \rho_k \delta\left(t - \frac{2d_k}{c}\right)$: Scene response function (SRF), consisting of multiple reflectors at depth $d_k$ with reflectivity $\rho_k$
- $r(t) = (h * p)(t)$: Reflected signal from the scene
- $\bar{p}(t) = p(-t)$: Matched filter (time-reversed probe)
- $\phi(t) = p * \bar{p}(t)$: Instrument response function (IRF), i.e., autocorrelation of the probe; its peaks reveal the time-of-flight delays corresponding to multiple reflector depths
- $m(t) = (r \otimes p)(t) = \sum r(\tau)p(\tau + t)d\tau = (r * \bar{p})(t) = (h * p * \bar{p})(t) = (h * \phi)(t)$: Lock-in measurements as filtered scene response; $\otimes$ stands for cross correlation; after sampling, $m(t)$ becomes $m[n]$

**Depth Recovery** In the multi-depth case, depth recovery becomes a sparse signal recovery problem: identify the time shifts and amplitudes of each component in $m(t)$, i.e., recovering a sparse signal convolved with a known kernel using a limited number of noisy samples.

| | **Single Reflector** | **Multiple Reflectors** |
|---|---|---|
| **Scene Response** | $h(t) = \rho_0 \delta(t - t_0)$ | $h(t) = \sum_k \rho_k \delta\left(t - \frac{2d_k}{c}\right)$ |
| **Measurement** | $m(t) = \rho_0 \phi(t - t_0)$ | $m(t) = \sum_k \rho_k \phi\left(t - \frac{2d_k}{c}\right) = \phi * h(t)$ |
| **Depth Estimation** | $\tilde{t}_0 = \arg\max_t m(t) \Rightarrow d = \frac{c \cdot \tilde{t}_0}{2}$ | Identify via sparse recovery:<br>• Sparse signal: $h(t)$<br>• Kernel: $\phi(t)$<br>• Observation: $m(t) = \phi * h(t)$ |

Taking Fourier transforms of the measurement:

$$\hat{m}(\omega) = \hat{\phi}(\omega) \cdot \hat{h}(\omega) = |\hat{p}(\omega)|^2 \cdot \hat{h}(\omega)$$

This shows that the spectrum of the measurement equals the spectrum of the scene response modulated by the energy of the probing function. Since $|\hat{p}(\omega)|^2$ typically concentrates in low frequencies, it acts as a low-pass filter on $\hat{h}(\omega)$, attenuating high-frequency components of the scene response.

## 6.3 Estimating Transient Components with Prony's Method

To recover multi-depth scene information from time-resolved measurements, we model the observed signal as a convolution between a sparse scene response $h(t)$ and a known system kernel $\phi(t)$.

| **SRF** | **IRF** | **Measurements** |
|---|---|---|
| $h(t) = \sum_{k=0}^{K-1} \Gamma_k \delta(t - t_k)$ | $\phi(t) = p * \bar{p}(t)$ | $m(t) = (h * \phi)(t) = \sum_{k=0}^{K-1} \Gamma_k \phi(t - t_k)$ |

where $\Gamma_k$ is the reflectivity and $t_k = 2d_k/c$ is the round-trip time delay from depth $d_k$.

**Fourier Domain Representation** Assume the probing function is approximately bandlimited, i.e.,

$$\phi(t) \approx \sum_{|m| \leq M_0} \hat{\phi}_m e^{jm\omega_0 t}$$

Then the measurements can be interpreted as the scene response filtered by the low-pass kernel $\phi(t)$. The measurement signal can thus be expressed as a superposition of a finite number of frequency components:

$$\begin{aligned}
m(t) &= \sum_{k=0}^{K-1} \Gamma_k \phi(t - t_k) \\
&= \sum_{k=0}^{K-1} \Gamma_k \sum_{|m| \leq M_0} \hat{\phi}_m e^{jm\omega_0 (t - t_k)} \\
&= \sum_{|m| \leq M_0} \hat{\phi}_m \left( \sum_{k=0}^{K-1} \Gamma_k e^{-jm\omega_0 t_k} \right) e^{jm\omega_0 t} \\
&= \sum_{|m| \leq M_0} \hat{\phi}_m y_m e^{jm\omega_0 t}
\end{aligned}$$

where

$$y_m = \sum_{k=0}^{K-1} \Gamma_k e^{-jm\omega_0 t_k}$$

This separates the known kernel $\hat{\phi}_m$ from the unknown scene-dependent term $y_m$, which is a sum of exponentials. Since $m(t)$ is bandlimited, each $y_m$ can be recovered by projecting $m(t)$ onto the corresponding complex exponential and dividing by $\hat{\phi}_m$, i.e.:

$$y_m = \frac{1}{\hat{\phi}_m} \cdot \int m(t) e^{-jm\omega_0 t} \, dt$$

**Parameter Recovery via Prony's Method** The goal is to recover $\Gamma_k$ and $t_k$ from the frequency-domain coefficients $y_m$, which takes the form:

$$y_m = \sum_{k=0}^{K-1} \Gamma_k e^{-jm\omega_0 t_k}$$

This is a classical problem in spectral estimation. Methods such as Prony's method, Matrix Pencil, and ESPRIT can be applied to estimate the unknown delays $t_k$ and amplitudes $\Gamma_k$ from the finite set of measurements $\{y_m\}_{m=-M_0}^{M_0}$.

By converting the convolutional model into the frequency domain and isolating the exponential terms, we reduce the multi-depth imaging problem to sparse frequency estimation. Prony's method enables super-resolution recovery of closely spaced reflectors beyond the Fourier resolution limit.

# 7 1-bit Time-Resolved Imaging

**Introduction** 1-bit time-resolved imaging is introduced to address the limitations of conventional high-speed imaging systems, which often require expensive, power-hungry hardware to achieve fine temporal resolution.

By using simple on-chip comparators instead of full analog-to-digital converters (ADCs), 1-bit sensors drastically reduce hardware complexity, power consumption, and data bandwidth. Despite the extreme quantisation, these systems can still recover depth and transient scene information by leveraging the sparsity of natural scenes and advanced signal processing techniques. This enables fast, low-cost, and scalable 3D imaging solutions, especially suited for high-resolution or resource-constrained applications.

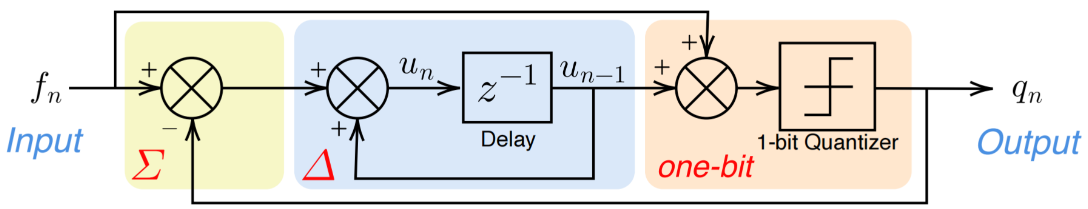
The imaging pipeline adopted in this chapter is identical to that of Section 6.2, involving:

- the probing function $p(t)$,
- the scene response $h(t)$,
- the reflected signal $r(t)$,
- the instrument response $\phi(t)$, and
- the continuous measurements $m(t)$.

However, the key difference lies in the final step. Traditional time-of-flight imaging focuses on interpreting the measured signal $m(t)$, while this chapter is concerned with its **digitisation**.

- Instead of the conventional **Shannon-Nyquist approach**: $m(t) \to y[n] = m(nT)$
- We investigate **alternative quantisation strategies**: $m(t) \to y[n] \in \{-1, 1\}$

**Quantisation Rule** The 1-bit delta-sigma quantisation can be described in 2 equivalent formulations:




- **Recursive (State-Space) Form:** The quantised output $q[n] \in \{-1, +1\}$ and accumulator $u[n]$ evolve as follows. The accumulator stores the quantisation error, enabling high-pass noise shaping.

$$q[n] = \text{sgn}(u[n-1] + f[n])$$
$$u[n] = u[n-1] + f[n] - q[n]$$

- **Difference (Convolution) Form:** Alternatively, the rule can be written as:

$$q[n] = f[n] - (u[n] - u[n-1]) = f[n] - (u * v)[n]$$
$$v[n] = \delta[n] - \delta[n-1]$$

where $v[n]$ is a discrete-time finite-difference filter (high-pass).

| **Time Domain** | **Fourier Domain** |
|---|---|
| $q[n] = f[n] - (u * v)[n]$ | $\hat{Q}(\omega) = \hat{F}(\omega) - \hat{U}(\omega)\hat{V}(\omega)$ |

Note that $\hat{V}(\omega) = 1 - e^{-j\omega}$, which has a high-pass feature. By filtering the quantised sequence with a difference filter $v[n]$, we are able to move the noise component $\hat{U}(\omega)\hat{V}(\omega)$ concentrates away from $\omega = 0$.

**Recovery for Bandlimited Functions** Assume $f(t) \in PW_\Omega$, i.e., a $\mu\Omega$-bandlimited signal sampled at rate $\mu$ times Nyquist. Then, one can recover $f(t)$ by **interpolating** with a reconstruction kernel $\varphi(t)$:

$$\tilde{f}(t) = \frac{1}{\mu} \sum_{n=-\infty}^{\infty} q[n] \cdot \varphi\left(\frac{t}{T} - \frac{n}{\mu}\right)$$

where $\varphi(t)$ is a reconstruction kernel with support $\Omega$, such as a sinc or exponential-Gaussian. This recovery is valid for smooth inputs and yields small reconstruction error if $\mu \gg 1$.

For higher-order filtering $v^{[L]}[n] = (v * \cdots * v)[n]$, the reconstruction error decays as:

$$\left| f(t) - \tilde{f}(t) \right| \leq \frac{1}{\mu^L} \|\varphi^{(L)}\|_1 \cdot \max_n |u[n]|$$

Hence, larger oversampling rate ($\mu$) and larger noise shaping order ($L$) reduce error.

**Model in Time-Resolved Imaging** Let $m_{\mathbf{r}}(t)$ denote the time-resolved measurements at pixel $\mathbf{r}$.

$$\textbf{SRF:} \quad h_{\mathbf{r}}(t, t') = \sum_{k=0}^{K-1} \Gamma_{\mathbf{r}}[k]\, \delta(t - t' - \tau_{\mathbf{r}}[k])$$

$$\textbf{Measurements:} \quad m_{\mathbf{r}}(t) = (h_{\mathbf{r}} * \phi_{\mathbf{r}})(t) = \sum_{k=0}^{K-1} \Gamma_{\mathbf{r}}[k] \cdot \phi_{\mathbf{r}}(t - \tau_{\mathbf{r}}[k])$$

Here, $t'$ denotes the illumination time (i.e., when the probing pulse is transmitted), while $t$ is the observation time (i.e., when the echo is received). The SRF $h_{\mathbf{r}}(t, t')$ captures how an input impulse at time $t'$ contributes to the received signal at time $t$ via delays $\tau_{\mathbf{r}}[k]$, corresponding to round-trip travel times from different reflective surfaces.

Let $s_{\mathbf{r}}(t) = h_{\mathbf{r}}(t, 0)$. Discretising and applying 1-bit sampling:

$$\begin{aligned} q_{\mathbf{r}}[n] &= m_{\mathbf{r}}[n] - (u_{\mathbf{r}} * v)[n] \\ &= (\phi_{\mathbf{r}} * s_{\mathbf{r}})(t) - (u_{\mathbf{r}} * v)[n] \\ \hat{Q}_{\mathbf{r}}(\omega) &= \hat{\phi}_{\mathbf{r}}(\omega)\hat{s}_{\mathbf{r}}(\omega) - \hat{U}_{\mathbf{r}}(\omega)\hat{V}(\omega) \\ &= \underbrace{\hat{\phi}_{\mathbf{r}}(\omega) \sum_{k=0}^{K-1} \Gamma_{\mathbf{r}}[k] e^{-j\omega\tau_{\mathbf{r}}[k]}}_{\text{low-pass}} - \underbrace{\hat{U}_{\mathbf{r}}(\omega)(1 - e^{-j\omega})}_{\text{high-pass}} \end{aligned}$$

**Parameter Estimation** To extract the reflectivity–delay pairs $\{\Gamma_{\mathbf{r}}[k], \tau_{\mathbf{r}}[k]\}$ from the 1-bit measurement $q_{\mathbf{r}}[n]$, we operate in the Fourier domain. The recovery proceeds as follows:

- **Given:**
  - 1-bit samples $q_{\mathbf{r}}[n]$ at a pixel location $\mathbf{r}$ $\quad (\Rightarrow \hat{Q}_{\mathbf{r}}(\omega))$
  - 1-bit samples of the calibrated probing pulse $q_{\mathbf{r},\phi}[n]$ $\quad (\Rightarrow \hat{Q}_{\mathbf{r},\phi}(\omega) \approx \hat{\phi}_{\mathbf{r}}(\omega))$
- **Deconvolution:**
  - Choose an appropriate frequency support $M_0$ for the probing pulse
  - Compute the deconvolved signal:

$$\hat{s}_{\mathbf{r}}(m\omega_0) = \frac{\hat{Q}_{\mathbf{r}}(m\omega_0)}{\hat{Q}_{\mathbf{r},\phi}(m\omega_0)} \approx \sum_{k=0}^{K-1} \Gamma_r[k] e^{-jm\omega_0\tau_r[k]}, \quad |m| \leq M_0$$

- **Sine-Fitting:**
  - Solve the following nonlinear least-squares problem:

$$\min_{\{\Gamma_{\mathbf{r}}[k], \tau_{\mathbf{r}}[k]\}} \left\| \hat{s}_{\mathbf{r}}(m\omega_0) - \sum_{k=0}^{K-1} \Gamma_{\mathbf{r}}[k] e^{-jm\omega_0\tau_{\mathbf{r}}[k]} \right\|^2$$

  - This is a classical problem in spectral estimation, and can be solved via Prony's method, ESPRIT, or the Matrix Pencil method (Bhandari et al., ICASSP 2013).