Imperial College London

Approximate Strang-Fix: Sparse Sampling with any Acquisition Device

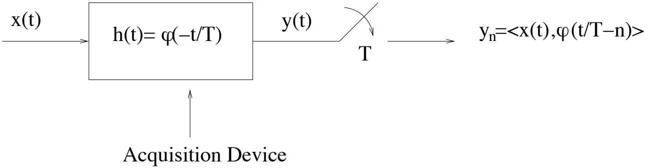
Pier Luigi Dragotti

June 10, 2013[1]

[1]This research is supported by European Research Council ERC, project 277800 (RecoSamp)

Problem Statement

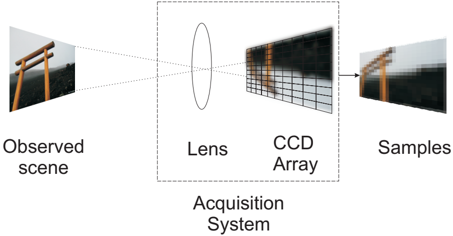
You are given a class of functions. You have a sampling device. Given the measurements $y_n = \langle x(t), \varphi(t/T - n)\rangle$, you want to reconstruct $x(t)$.



Natural questions:

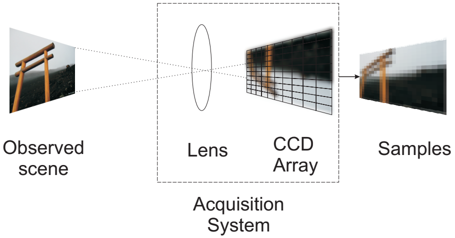
- When is there a one-to-one mapping between $x(t)$ and y_n?
- What signals can be sampled and what kernels $\varphi(t)$ can be used?
- What reconstruction algorithm?

Problem Statement



- The low-quality lens blurs the images.
- The images are sampled by the CCD array. images.

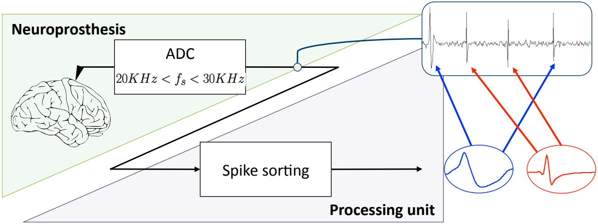
Problem Statement



- The world is analogue (audio, images, sound, brain), but computation is digital
- If you like sparsity, you need 'analogue' sparsity models
- The sampling kernel is the bridge between these two worlds

Motivation: Sampling Everywhere

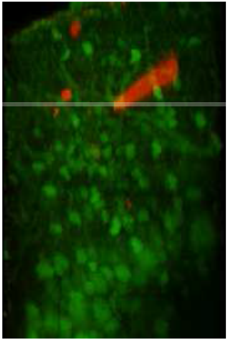
Applications in Neuroscience

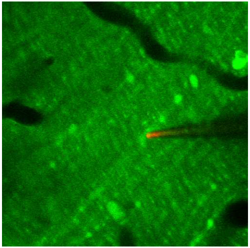


Imperial College
London

erc

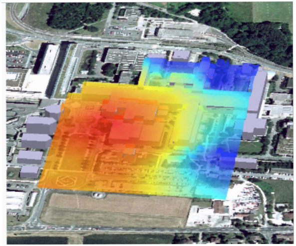
Neural Activity Detection





Motivation: Sampling Everywhere

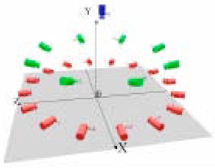
Sensor networks

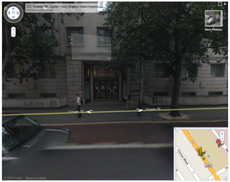


- The source (phenomenon) is distributed in space and time.
- The phenomenon is sampled in space (finite number of sensors) and time.

Motivation: Free Viewpoint Video

Multiple cameras are used to record a scene or an event. Users can freely choose an arbitrary viewpoint for 3D viewing.



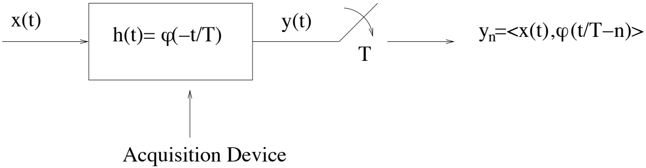


- This is a multi-dimensional sampling and interpolation problem.

Outline

- Sampling Kernels and Strang-Fix Conditions
- From Samples to Signals
  - Traditional FRI Sampling
  - e-MOMS (Maximum Order Minimum Support Kernels)
  - Applications in Image Super-Resolution
- Approximate Strang-Fix
- Sparse Sampling with any Kernel
- Application in Neuroscience
- Conclusions and Outlook

Sampling Kernels



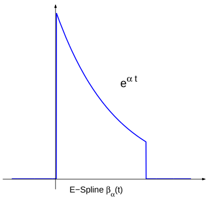
- Given by nature
  - Diffusion equation, Green function. Ex: sensor networks.
- Given by the set-up
  - Designed by somebody else. Ex: Hubble telescope, digital cameras.
- Given by design
  - Pick the best kernel. Ex: engineered systems.

Sampling Kernels

Any kernel $\varphi(t)$ that can reproduce exponentials:

$$\sum_n c_{m,n}\varphi(t-n) = e^{\alpha_m t}, \qquad \alpha_m = \alpha_0 + m\lambda \text{ and } m = 0, 1, ..., L.$$

This includes any composite kernel of the form $\gamma(t) * \beta_{\vec{\alpha}}(t)$ where $\beta_{\vec{\alpha}}(t) = \beta_{\alpha_0}(t) * \beta_{\alpha_1}(t) * ... * \beta_{\alpha_L}(t)$ and $\beta_{\alpha_i}(t)$ is an Exponential Spline of first order [UnserB:05].

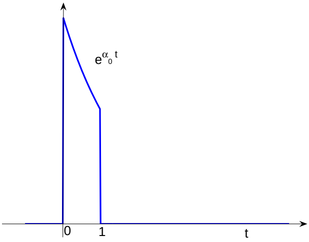


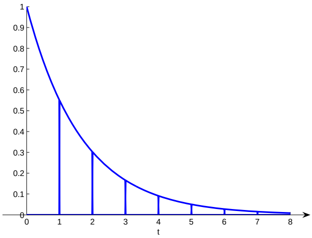
$$\beta_\alpha(t) \Leftrightarrow \hat{\beta}(\omega) = \frac{1 - e^{\alpha - j\omega}}{j\omega - \alpha}$$

Notice:

- α can be complex.
- E-Spline is of compact support.
- E-Spline reduces to the classical polynomial spline when $\alpha = 0$.

Exponential Reproducing Kernels





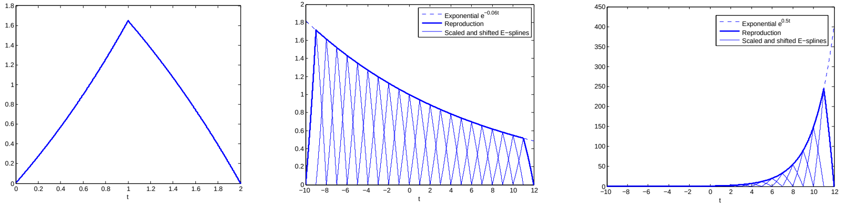
The E-spline of first order $\beta_{\alpha_0}(t)$ reproduces the exponential $e^{\alpha_0 t}$:

$$\sum_n c_{0,n}\beta_{\alpha_0}(t-n) = e^{\alpha_0 t}.$$

In this case $c_{0,n} = e^{\alpha_0 n}$. In general, $c_{m,n} = c_{m,0}e^{\alpha_m n}$.

Imperial College London

Exponential Reproducing Kernels



Here the E-spline is of second order and reproduces the exponential $e^{\alpha_0 t}$, $e^{\alpha_1 t}$: with $\alpha_0 = -0.06$ and $\alpha_1 = 0.5$.

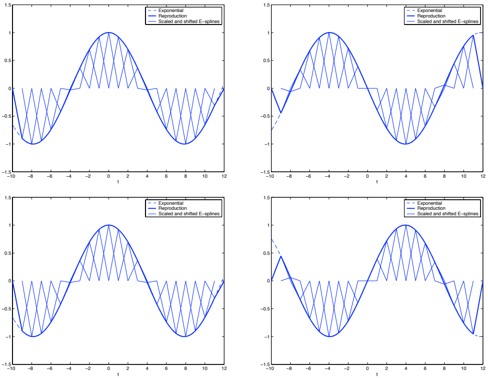
Exponential Reproducing Kernels

- The exponent α of the E-splines can be complex. This means $\beta_\alpha(t)$ can be a complex function.
- However if pairs of exponents are chosen to be complex conjugate then the spline stays real.
- Example:

$$\beta_{\alpha_0+j\omega_0}(t) * \beta_{\alpha_0-j\omega_0}(t) = \begin{cases} \frac{\sin \omega_0 t}{\omega_0} e^{\alpha_0 t} & 0 \leq t < 1 \\ -\frac{\sin \omega_0 (t-2)}{\omega_0} e^{\alpha_0 t} & 1 \leq t < 2 \\ 0 & \text{Otherwise} \end{cases}$$

When $\alpha_0 = 0$ (i.e., purely imaginary exponents), the spline is called trigonometric spline.

Exponential Reproducing Kernels



Here $\vec{\alpha} = (-j\omega_0, j\omega_0)$ and $\omega_0 = 0.2$. $\sum_n c_{n,m}\beta_{\vec{\alpha}}(t-n) = e^{jm\omega_0} \qquad m = -1, 1.$

Notice: $\beta_{\vec{\alpha}}(t)$ is a real function, but the coefficients $c_{m,n}$ are complex.

Generalised Strang-Fix Conditions

A function $\varphi(t)$ can reproduce the exponential:

$$e^{\alpha_m t} = \sum_n c_{m,n}\varphi(t-n)$$

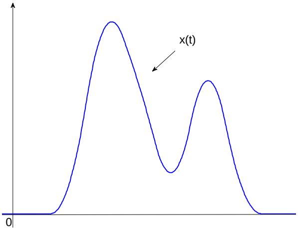
if and only if

$$\hat{\varphi}(\alpha_m) \neq 0 \text{ and } \hat{\varphi}(\alpha_m + j2\pi l) = 0 \qquad l \in \mathbb{Z} \setminus \{0\}$$

where $\hat{\varphi}(s)$ is the bilateral Laplace transform of $\varphi(t)$.

Also note that $c_{m,n} = c_{m,0}e^{\alpha_m n}$ with $c_{m,0} = \hat{\varphi}(\alpha_m)^{-1}$.

From Samples to Signals



- Consider any $x(t)$ with $t \in [0, N)$ and sampling period $T = 1$.
- The sampling kernel $\varphi(t)$ satisfies

$$\sum_n c_{m,n}\varphi(t-n) = e^{\alpha_m t} \qquad m = 1, ..., L,$$

- We want to retrieve $x(t)$, from the samples $y_n = \langle x(t), \varphi(t-n)\rangle$, $n = 0, 1, ..., N-1$.

From Samples to Signals

We have that

$$
\begin{aligned}
s_m &= \sum_{n=0}^{N-1} c_{m,n} y_n \\
&= \langle x(t), \sum_{n=0}^{N-1} c_{m,n}\varphi(t-n)\rangle \\
&= \int_{-\infty}^{\infty} x(t) e^{\alpha_m t} dt, \qquad m = 1, \dots, L.
\end{aligned}
$$

- s_m is the bilateral Laplace transform of $x(t)$ evaluated at α_m.
- When $\alpha_m = j\omega_m$ then $s_m = \hat{x}(j\omega_m)$ where $\hat{x}(j\omega)$ is the Fourier transform of $x(t)$.

From Samples to Signals

- Consider signals which are completely specified by a finite number of free parameters
- This is an 'analogue' sparsity model
- For classes of parametrically sparse signals there is a one-to-one mapping between samples and signal:

$$x(t) \Leftrightarrow \hat{x}(j\omega_m) \qquad m = 1, 2, ..., L$$

- The number d of degrees of freedom of the signal must satisfy $d \leq L$

Sampling Streams of Diracs

- Assume $x(t)$ is a stream of K Diracs on the interval of size N: $x(t)=\sum_{k=0}^{K-1}x_k\delta(t-t_k),\ t_k\in[0,N)$.
- We restrict $\alpha_m=\alpha_0+m\lambda \quad m=1,...,L$ and $L\geq 2K$.
- We have N samples: $y_n=\langle x(t),\varphi(t-n)\rangle,\ n=0,1,...N-1$:
- We obtain

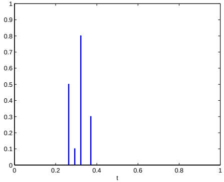
$$
\begin{aligned}
s_m &= \sum_{n=0}^{N-1} c_{m,n}y_n \\
&= \int_{-\infty}^{\infty} x(t)e^{\alpha_m t}dt, \\
&= \sum_{k=0}^{K-1} x_k e^{\alpha_m t_k} \\
&= \sum_{k=0}^{K-1} \hat{x}_k e^{\lambda m t_k} = \sum_{k=0}^{K-1} \hat{x}_k u_k^m, \qquad m=1,...,L.
\end{aligned}
$$

Prony's Method

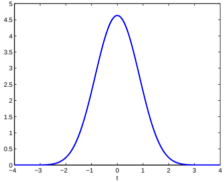
- The quantity
$$s_m = \sum_{k=0}^{K-1} \hat{x}_k u_k^m, \qquad m = 1, ..., L$$
is a sum of exponentials.
- Retrieving the locations u_k and the amplitudes $\hat{x}_k$ from $\{s_m\}_{m=1}^{L}$ is a classical problem in spectral estimation and was first solved by Gaspard de Prony in 1795.
- Given the pairs $\{u_k, \hat{x}_k\}$, then $t_k = (\ln u_k)/\lambda$ and $x_k = \hat{x}_k / e^{\alpha_0 t_k}$.

Imperial College London

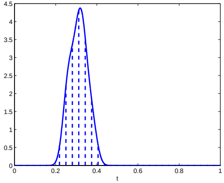
Sampling Streams of Diracs: Numerical Example



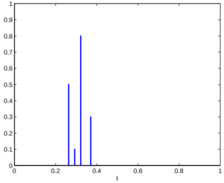
(a) Original Signal



(b) Sampling Kernel ($\beta_7(t)$)



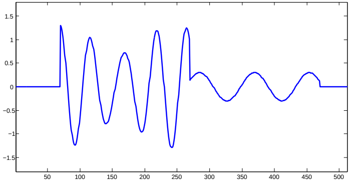
(c) Samples



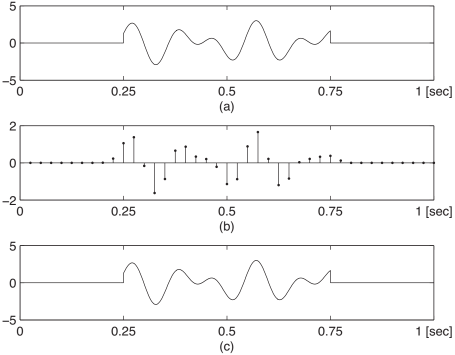
(d) Reconstructed Signal

Sparse Sampling: Extensions

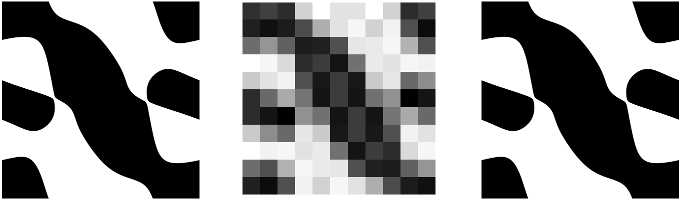
Using variations of Prony's method other signals can be sampled such as for example piecewise sinusoidal signals [BerentDragotti:10].



Numerical Example



Sampling 2-D domains



The curve is implicitly defined through the equation [PanBluDragotti:11]:

$$f(x,y)=\sum_{k=1}^{K}\sum_{i=1}^{I}b_{k,i}e^{-j2\pi xk/M}e^{-j2\pi yi/N}=0.$$

The coefficients $b_{k,i}$ are the only free parameters in the model.

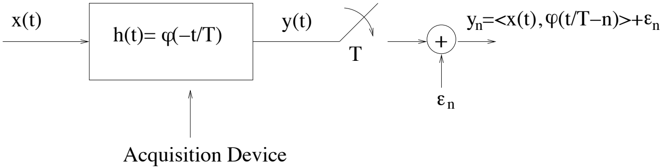
Sampling 2-D domains

samples

interpolation

inter+ curve constraint

Robust Sparse Sampling



- The measurements are noisy
- The noise is additive and i.i.d. Gaussian
- Many robust versions of Prony's method exist (e.g., Cadzow, matrix pencil)

Robust Sparse Sampling: Best Kernel

The exponential reproducing kernel has the following form

$$\varphi(t) = \gamma(t) * \beta_{\vec{\alpha}}(t).$$

How should we choose $\gamma(t)$ and α_m, $m = 1, ..., L$ so as to minimize the effect of noise?
Let $Y = (y_0, y_1, ..., y_{N-1})^T$ and $S = (s_1, s_2, ..., s_L)^T$, in the noiseless case:

$$S = \mathbf{C}Y.$$

When additive noise is present

$$\hat{S} = \mathbf{C}Y + \mathbf{C}\epsilon.$$

Here $\mathbf{C}$ is the $L \times N$ matrix of the exponential reproducing coefficients $c_{m,n} = c_{m,0}e^{\alpha_m n}$.

Robust Sparse Sampling: Best Kernel (cont'd)

- We want a well-conditioned **C**.
- Since $c_{m,n} = c_{m,0} e^{\alpha_m n}$:

$$\mathbf{C} = \begin{pmatrix} c_{1,0} & 0 & \cdots & 0 \\ 0 & c_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{L,0} \end{pmatrix} \begin{pmatrix} 1 & e^{\alpha_1} & \cdots & e^{\alpha_1(N-1)} \\ 1 & e^{\alpha_2} & \cdots & e^{\alpha_2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\alpha_L} & \cdots & e^{\alpha_L(N-1)} \end{pmatrix}$$

- Stability requires α_m to be purely imaginary, specifically, $\alpha_m = j\omega_m = j2\pi(m-1)/L,\ m = 1, 2, ..., L$
- and $|c_{m,0}| = 1,\ m = 1, 2, ..., L.$

Robust Sparse Sampling: Best Kernel (cont'd)

- Since $c_{m,0} = \hat{\varphi}(j\omega_m)$, $|c_{m,0}| = 1$ is achieved by imposing $|\hat{\gamma}(j\omega_m)\hat{\beta}_{\vec{\alpha}}(j\omega_m)| = 1$, $m = 1, ..., L$.
- We pick the kernel with the shortest support:

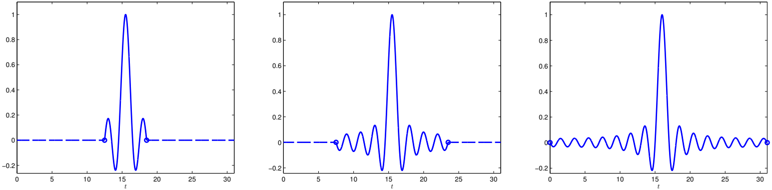
$$\varphi(t) = \sum_{\ell=0}^{L-1} d_\ell \beta_{\vec{\alpha}}^{(\ell)}(t),$$

- In frequency:

$$\hat{\varphi}(j\omega) = \hat{\beta}_{\vec{\alpha}}(j\omega) \sum_{\ell=0}^{L-1} d_\ell (j\omega)^\ell,$$

- Therefore $\hat{\gamma}(j\omega) = \sum_{\ell=0}^{L-1} d_\ell (j\omega)^\ell$. Thus the coefficients d_ℓ are chosen so that the polynomial $\hat{\gamma}(j\omega)$ interpolates the points $(j\omega_m, |\hat{\beta}_{\vec{\alpha}}(j\omega_m)|^{-1})$.

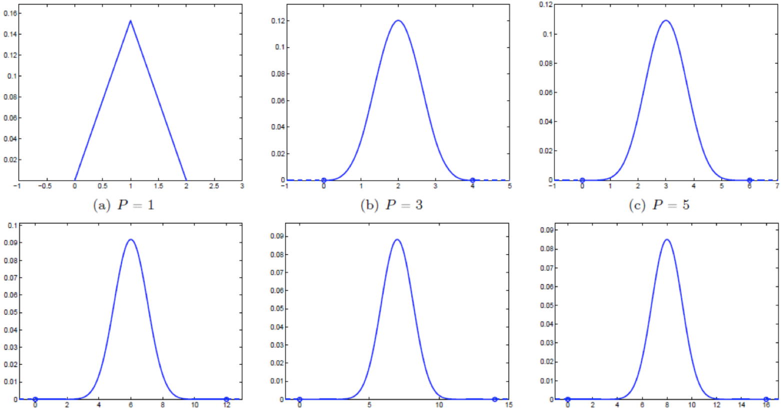
Examples of Best Kernels



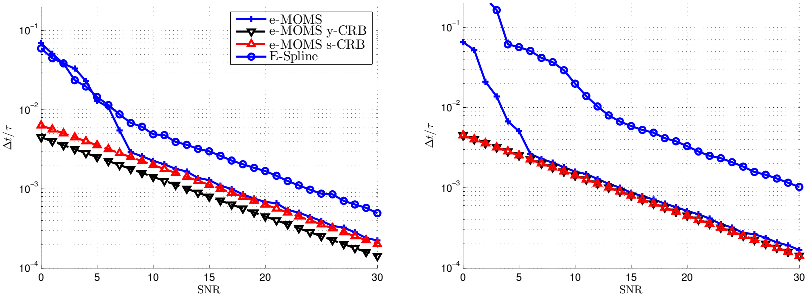
L=6   L=16   L=31

- We call these kernels Exponential MOMS (e-MOMS), where MOMS stands for Maximum Order Minimum Support [Uriguen-Dragotti-Blu-11-13].
- They correspond to one period of the Dirichlet function
- SoS kernels [Eldar et al.-11] are a sub-set of eMOMS.

Examples of E-Splines Kernels

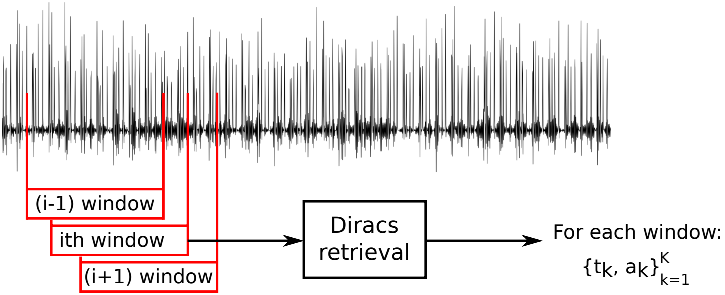

(a) $P = 1$ (b) $P = 3$ (c) $P = 5$
(d) $P = 11$ (e) $P = 13$ (f) $P = 15$

e-MOMS vs E-splines



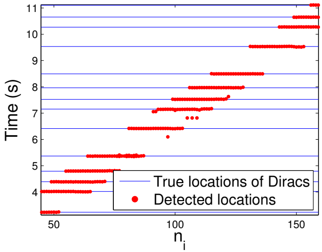
$K = 2$ and we measure the error in the retrieval of the location of the Diracs.

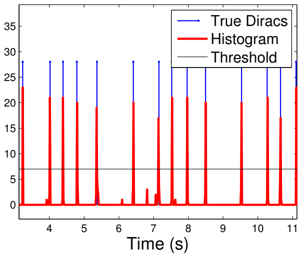
Retrieving 1000 Diracs with e-MOMS



- Retrieve Diracs using a sliding window
- Locations of true Diracs are consistent across windows [Onativia-Uriguen-Dragotti-13]

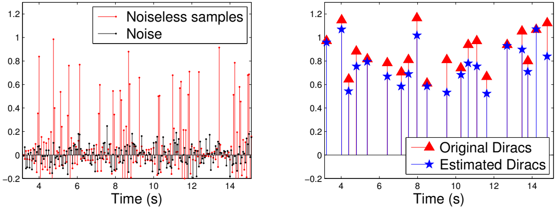
Retrieving 1000 Diracs with e-MOMS





Imperial College London

Retrieving 1000 Diracs with e-MOMS

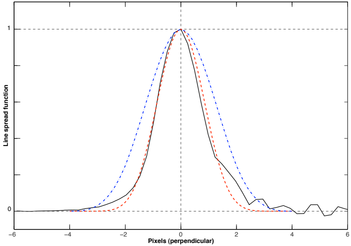


(a) y_n samples **(b)** Reconstructed stream

- $K = 1000$ Diracs in an interval of 630 seconds, $N = 10^5$ samples, $T = 0.06$ and $SNR = 10$dB
- 9997 Diracs retrieved with an error $\epsilon < T/2$
- Average accuracy $\Delta t = 0.005$, execution time 105 seconds.

Application: Image Super-Resolution [Baboulaz-D-09]

(a)Original (2014 $\times$ 3039)



(b) Point Spread function

Application: Image Super-Resolution

Acquisition with Nikon D70

(a)Original (2014×3040)

(b) ROI (128×128)

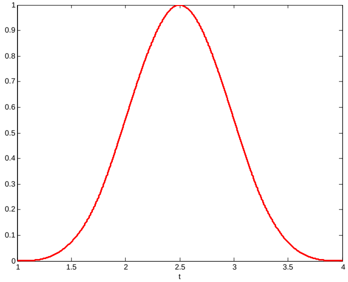
(b) Super-res (1024×1024)

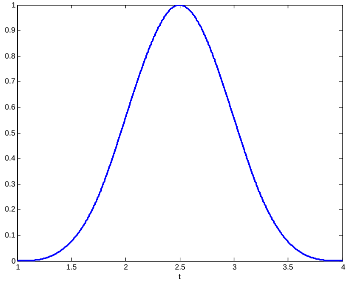
Application: Image Super-Resolution

(a)Original (48 $\times$ 48)

(b) Super-res (480 $\times$ 480)

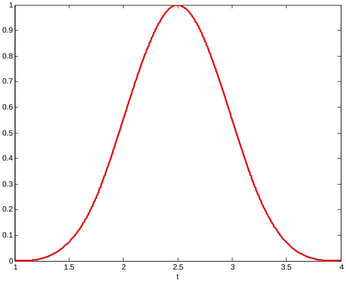
Spot the Difference



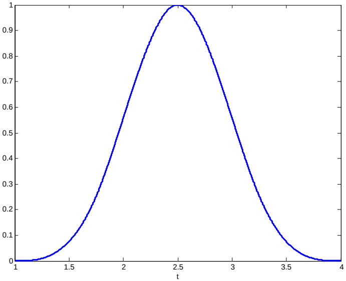


Imperial College London

Spot the Difference

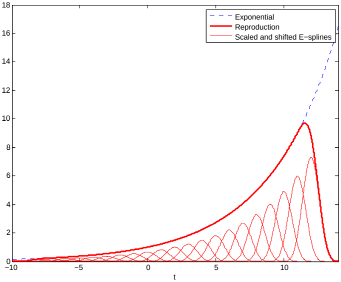


$$\alpha_m = \{-0.4, -0.2, 0.2, 0.4\}$$

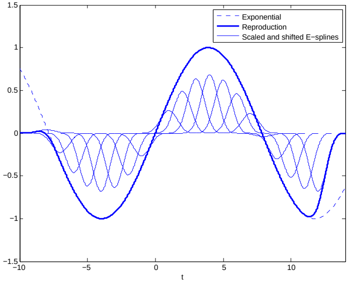


$$\alpha_m = \{-j0.4, -j0.2, j0.2, j0.4\}$$

Spot the Difference



'Explodential' Spline ;-)'



Exponential Spline

Approximate Strang-Fix

- Assume $\varphi(t)$ cannot reproduce exponentials, we want to find the coefficients $c_n = c_0 e^{\alpha t}$ such that:

$$\sum_{n\in\mathbb{Z}} c_n \varphi(t-n) \approx \mathrm{e}^{\alpha t}.$$

Approximate Strang-Fix

- Assume $\varphi(t)$ cannot reproduce exponentials, we want to find the coefficients $c_n = c_0 e^{\alpha t}$ such that:

$$\sum_{n \in \mathbb{Z}} c_n \varphi(t-n) \approxeq \mathrm{e}^{\alpha t}.$$

- Approximation error

$$\varepsilon(t) = f(t) - e^{\alpha t} = \mathrm{e}^{\alpha t}\left[1 - c_0 \sum_{l \in \mathbb{Z}} \hat{\varphi}(\alpha + j2\pi l)\mathrm{e}^{j2\pi l t}\right].$$

Approximate Strang-Fix

- Assume $\varphi(t)$ cannot reproduce exponentials, we want to find the coefficients $c_n = c_0 e^{\alpha t}$ such that:

$$\sum_{n\in\mathbb{Z}} c_n \varphi(t-n) \approxeq \mathrm{e}^{\alpha t}.$$

- Approximation error

$$\varepsilon(t) = f(t) - e^{\alpha t} = \mathrm{e}^{\alpha t}\left[1 - c_0 \sum_{l\in\mathbb{Z}} \hat{\varphi}(\alpha + j2\pi l)\mathrm{e}^{j2\pi l t}\right].$$

- *Least-squares approximation* ($e^{\alpha t}$ orthogonal to *span*$\{\varphi(t-n)\}_{n\in\mathbb{Z}}$)

$$c_n = \frac{\hat{\varphi}(-\alpha)}{\hat{a}_\varphi(\mathrm{e}^{\alpha})}\mathrm{e}^{\alpha n},$$

where $\hat{a}_\varphi(\mathrm{e}^{\alpha}) = \sum_{l\in\mathbb{Z}} a_\varphi[l]\mathrm{e}^{-\alpha l}$ is the z-transform of $a_\varphi[l] = \langle \varphi(t-l), \varphi(t)\rangle$, evaluated at $z = \mathrm{e}^{\alpha}$.

Approximate Strang-Fix

- Assume $\varphi(t)$ cannot reproduce exponentials, we want to find the coefficients $c_n = c_0 e^{\alpha t}$ such that:

$$\sum_{n\in\mathbb{Z}} c_n \varphi(t-n) \approxeq \mathrm{e}^{\alpha t}.$$

- Approximation error

$$\varepsilon(t) = f(t) - e^{\alpha t} = \mathrm{e}^{\alpha t}\left[1 - c_0 \sum_{l\in\mathbb{Z}} \hat{\varphi}(\alpha + j2\pi l)\mathrm{e}^{j2\pi l t}\right].$$

Approximate Strang-Fix

- Assume $\varphi(t)$ cannot reproduce exponentials, we want to find the coefficients $c_n = c_0 e^{\alpha t}$ such that:

$$\sum_{n\in\mathbb{Z}} c_n \varphi(t-n) \approxeq \mathrm{e}^{\alpha t}.$$

- Approximation error

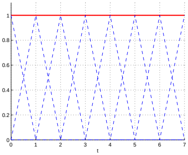
$$\varepsilon(t) = f(t) - e^{\alpha t} = \mathrm{e}^{\alpha t}\left[1 - c_0 \sum_{l\in\mathbb{Z}} \hat{\varphi}(\alpha + j2\pi l)\mathrm{e}^{j2\pi l t}\right].$$

- *Constant Least-squares approximation*

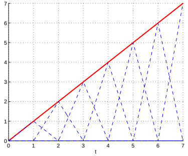
$$c_0 = \hat{\varphi}(\alpha)^{-1} \Rightarrow c_n = \hat{\varphi}(\alpha)^{-1} e^{\alpha n}$$

- **Advantage**: only need to know the Laplace transform of $\varphi(t)$ at α.

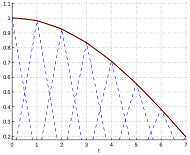
Approximate Strang-Fix- Example with Linear Splines



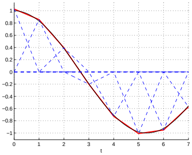
Repr. of 1 (exact)



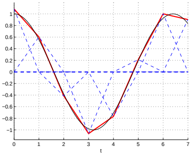
Repr. of t (exact)



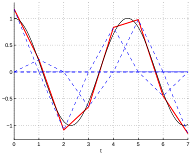
Approx. of $\mathrm{Re}\{e^{-j\frac{\pi}{16}t}\}$



Approx. of $\mathrm{Re}\{e^{-j\frac{3\pi}{16}t}\}$



Approx. of $\mathrm{Re}\{e^{-j\frac{5\pi}{16}t}\}$



Approx. of $\mathrm{Re}\{e^{-j\frac{7\pi}{16}t}\}$

Approximate FRI recovery

- Assume the signal to retrieve is a stream of K Diracs.
- Reproduce approximately α_m $m = 1, 2, ..., L$
- Obtain

$$s_m = \sum_{n=0}^{N-1} c_{m,n} y_n = \sum_{k=0}^{K-1} x_k u_k^m - \underbrace{\sum_{k=0}^{K-1} a_k \varepsilon_m \left(\frac{t_k}{T} \right)}_{\zeta_m}$$

- Treat the error as noise and retrieve the Diracs using robust FRI reconstruction
- Note that given a first estimate of the Diracs, we can estimate $\varepsilon_m \left(\frac{t_k}{T} \right)$ and repeat the estimation.

Approximate FRI recovery- Choice of α_m

- We want a well-conditioned $\mathbf{C}$.
- Since $c_{m,n} = c_{m,0} e^{\alpha_m n}$:

$$\mathbf{C} = \begin{pmatrix} c_{1,0} & 0 & \cdots & 0 \\ 0 & c_{2,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c_{L,0} \end{pmatrix} \begin{pmatrix} 1 & e^{\alpha_1} & \cdots & e^{\alpha_1(N-1)} \\ 1 & e^{\alpha_2} & \cdots & e^{\alpha_2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\alpha_L} & \cdots & e^{\alpha_L(N-1)} \end{pmatrix}$$

- Stability requires α_m to be purely imaginary: $\alpha_m = j\omega_m$

Approximate FRI recovery- Choice of α_m

- We want a well-conditioned **C**.
- Since $c_{m,n} = c_{m,0} e^{\alpha_m n}$:

$$\mathbf{C} = \begin{pmatrix} \hat{\varphi}^{-1}(\alpha_1) & 0 & \cdots & 0 \\ 0 & \hat{\varphi}^{-1}(\alpha_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\varphi}^{-1}(\alpha_L) \end{pmatrix} \begin{pmatrix} 1 & e^{\alpha_1} & \cdots & e^{\alpha_1(N-1)} \\ 1 & e^{\alpha_2} & \cdots & e^{\alpha_2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\alpha_L} & \cdots & e^{\alpha_L(N-1)} \end{pmatrix}$$

- Stability requires α_m to be purely imaginary: $\alpha_m = j\omega_m$
- Typically, $\varphi(t)$ low-pass filter $\Rightarrow$ pick $j\omega_m$ close to the origin

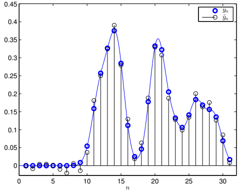
Approximate FRI recovery- Choice of α_m

- We want a well-conditioned **C**.
- Since $c_{m,n} = c_{m,0} e^{\alpha_m n}$:

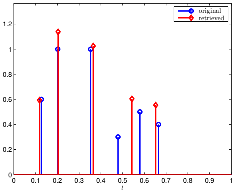
$$\mathbf{C} = \begin{pmatrix} \hat{\varphi}^{-1}(\alpha_1) & 0 & \cdots & 0 \\ 0 & \hat{\varphi}^{-1}(\alpha_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\varphi}^{-1}(\alpha_L) \end{pmatrix} \begin{pmatrix} 1 & e^{\alpha_1} & \cdots & e^{\alpha_1(N-1)} \\ 1 & e^{\alpha_2} & \cdots & e^{\alpha_2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{\alpha_L} & \cdots & e^{\alpha_L(N-1)} \end{pmatrix}$$

- Stability requires α_m to be purely imaginary: $\alpha_m = j\omega_m$
- Typically, $\varphi(t)$ low-pass filter $\Rightarrow$ pick $j\omega_m$ close to the origin
- Choose $L \sim N$ so that **C** square

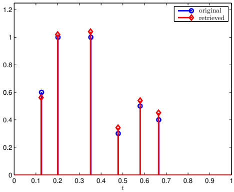
Approximate FRI recovery: Numerical Example



(a) y_n and $\tilde{y}_n$



(b) Default FRI retrieval

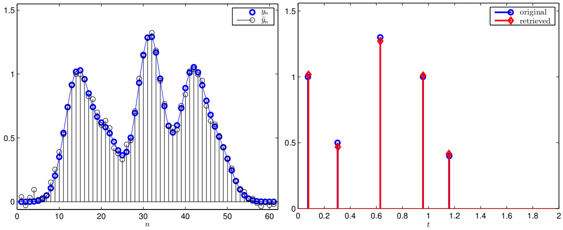


(c) Approx. FRI retrieval

Estimation of $K = 6$ Diracs with the B-Spline kernel of order $L = 16$, $N = 31$. (b) Default polynomial recovery. (c) Approximate recovery with $\alpha_m = j\frac{\pi}{1.5(P+1)}(2m - P)$, $m = 0, \ldots, P$ where $P + 1 = 21$, SNR=25dB.

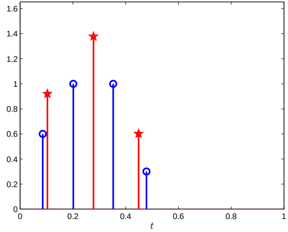
Approximate FRI recovery: Numerical Example

Gaussian Kernel

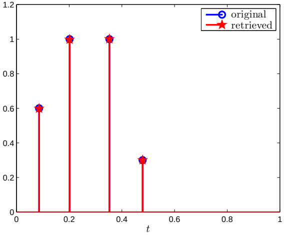


Approximate FRI with the Gaussian kernel. $K = 5$, $N = 61$, SNR=25dB. Recovery using the approximate method with $\alpha_m = j\frac{\pi}{3.5(P+1)}(2m - P)$, $m = 0, \ldots, P$ where $P + 1 = 21$.

Universal FRI recovery



(a) Default

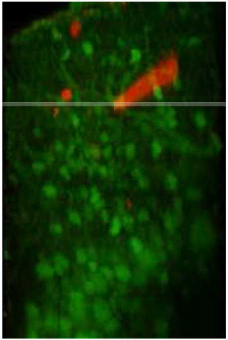


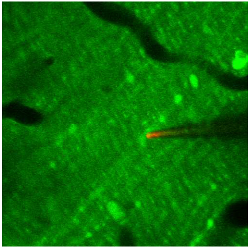
(b) Approximation

Reconstruction of $K = 4$ Diracs using the default strategy, part (a), and the approximate framework, part (b). Sampling Kernel: B-spline of order $P = 5$.

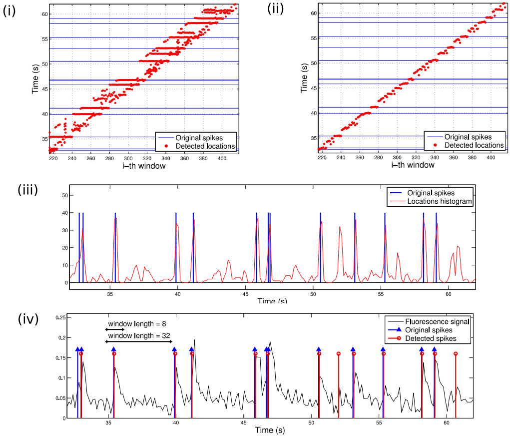
Imperial College London

Neural Activity Detection

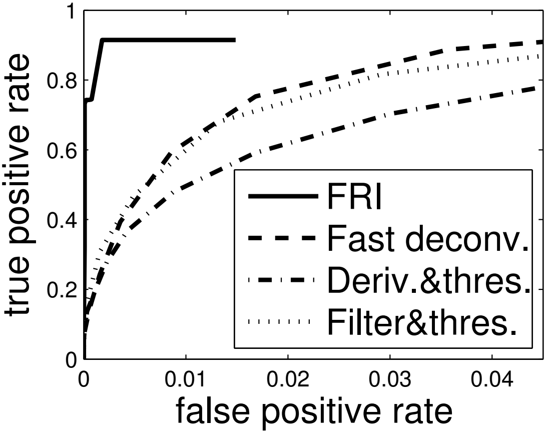




Calcium Transient Detection

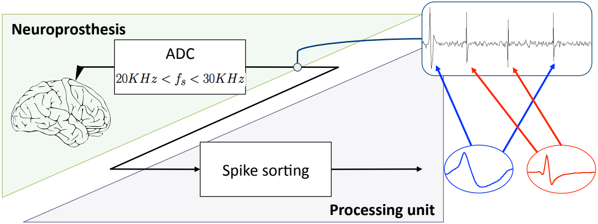


Calcium Transient Detection



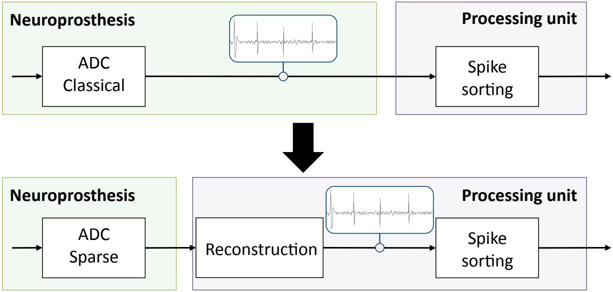
Application in Neuroscience

Applications in Neuroscience

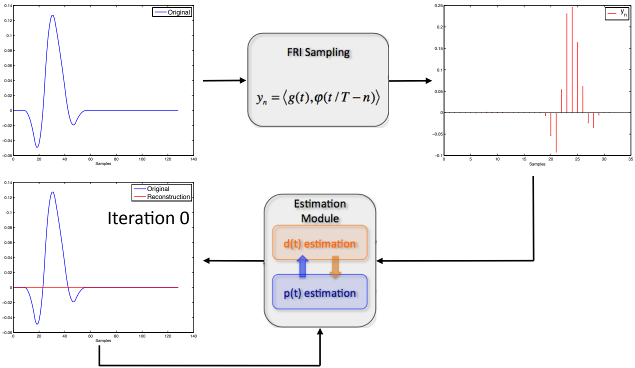


Application in Neuroscience

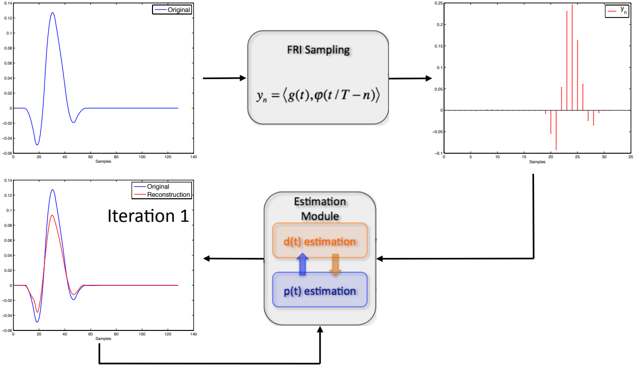
Insight: Sample at lower rate and reconstruct the signal outside the implant



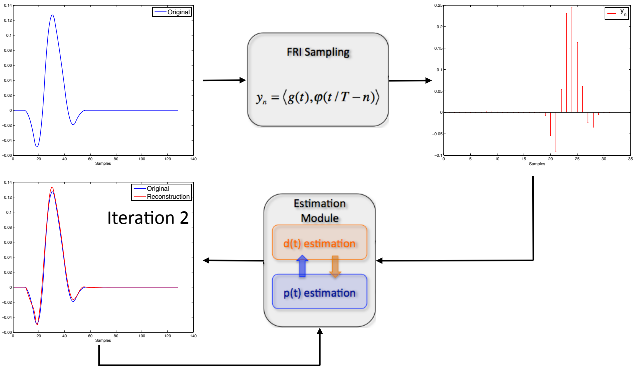
Stream of Pulses with unknown Shape



Stream of Pulses with unknown Shape



Stream of Pulses with unknown Shape



Application in Neuroscience

- Classical Sampling (C) $f_s = 24KHz$
- Sparse Sampling (F) $f_s = 5.8KHz$

- Two recordings of 1000 spikes from 3 different neurons.
  - Classical sampling: $f_s = 24KHz$ (C)
  - FRI sampling: $f_s = 5.8KHz$ (F)
- The classical sampling signal and the reconstruction from FRI sampling are fed to a spike sorting algorithm.

		Missed spikes		False positives		Misclassified spikes		Unclassified spikes		Success Rate	
Spike set	Noise s.d.	24K C	5.8K F	24K C	5.8K F	24K C	5.8K F	24K C	5.8K F	24K C	5.8K F
Easy (1)	0.05	111	135	0	2	22	21	30	20	83.7	82.2
	0.1	93	91	6	9	29	34	9	4	86.3	86.2
	0.15	143	129	7	21	50	56	1	2	79.9	79.2
	0.2	248	216	1	18	37	44	1	2	71.3	72
Difficult (2)	0.05	140	149	0	0	17	7	70	71	77.3	77.3
	0.1	101	80	0	16	418	199	0	16	48.1	69.9
	0.15	115	86	1	20	346	454	0	0	53.8	44
	0.2	160	108	3	19	441	420	0	0	39.6	45.3
(Av.)	0.125	138.88	124.25	2.24	13.13	170	154.38	13.88	14.38	67.5	69.51

Conclusions

Sampling signals using sparsity models:

- New framework that allows the sampling and reconstruction of infinite-dimensional continuous-time signals at a rate smaller than Nyquist rate.
- It is a non-linear problem
- Different possible algorithms with various degrees of efficiency and robustness
- Approximate Strang-Fix method: universal and robust to noise

Outlook:

- Promising applications in neuroscience
- Applications to the inversion of physical fields from sensors' measurements

Still many open questions from theory to practice!

References

On sampling

- J. Uriguen, T. Blu, and P.L. Dragotti 'FRI Sampling with Arbitrary Kernels', IEEE Trans. on Signal Processing, December 2012 (submitted)
- T. Blu, P.L. Dragotti, M. Vetterli, P. Marziliano and L. Coulot 'Sparse Sampling of Signal Innovations: Theory, Algorithms and Performance Bounds,' IEEE Signal Processing Magazine, vol. 25(2), pp. 31-40, March 2008
- P.L. Dragotti, M. Vetterli and T. Blu, 'Sampling Moments and Reconstructing Signals of Finite Rate of Innovation: Shannon meets Strang-Fix', IEEE Trans. on Signal Processing, vol.55 (5), pp.1741-1757, May 2007.
- J.Berent and P.L. Dragotti, and T. Blu,'Sampling Piecewise Sinusoidal Signals with Finite Rate of Innovation Methods,' IEEE Transactions on Signal Processing, Vol. 58(2),pp. 613-625, February 2010.
- J. Uriguen, P.L. Dragotti and T. Blu,'On the Exponential Reproducing Kernels for Sampling Signals with Finite Rate of Innovation' in Proc. of Sampling Theory and Application Conference, Singapore, May 2011.
- H. Pan, T. Blu, and P.L. Dragotti, 'Sampling Curves with Finite Rate of Innovation' in Proc. of Sampling Theory and Application Conference, Singapore, May 2011.

References (cont'd)

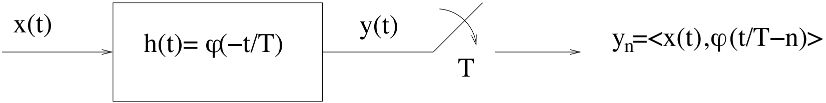
On Image Super-Resolution

- L. Baboulaz and P.L. Dragotti, 'Exact Feature Extraction using Finite Rate of Innovation Principles with an Application to Image Super-Resolution', IEEE Trans. on Image Processing, vol.18(2), pp. 281-298, February 2009.

On Application in Neuroscience

- J. Caballero, J.A. Uriguen, S. Schultz and P.L. Dragotti, Spike Sorting at Sub-Nyquist Rates, in Proc. of IEEE International Conf. on Acoustic, Speech and Signal Processing (ICASSP), Kyoto, Japan, April 2012.
- Jon Onativia, Simon R. Schultz, and Pier Luigi Dragotti, A Finite Rate of Innovation algorithm for fast and accurate spike detection from two-photon calcium imaging, Journal of Neural Engineering, June 2013 (to appear).

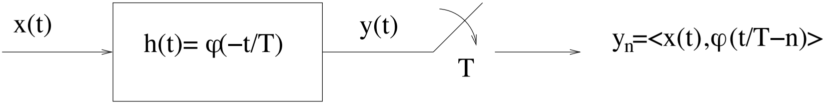
Structural-Sparsity vs Sparse Samples

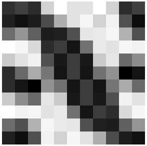


Structural Sparse

Imperial College London

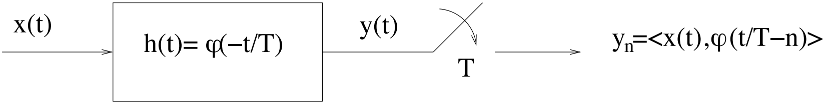
Structural-Sparsity vs Sparse Samples



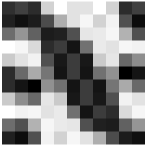


Non-sparse samples

Structural-Sparsity vs Sparse Samples



Structural Sparse



Non-sparse samples