

Exploring compressed sensing fMRI time series

Zach Stoebner

EECE 8396-01 S22

1 Introduction

Compressed sensing reconstructs signals by solving underdetermined linear systems under the conditions that the measurements are sparse in the domain and incoherent [1]. In engineering, if measurements are taken within an appropriate basis satisfying the restricted isometry property, i.e., the Gaussian, Bernoulli, or Fourier bases, then this prior structure makes full signal recovery possible [2].

Compressed sensing is challenging with fMRI because the temporal dynamics of hemodynamic signals are relatively slow compared to other fast-acquisition signals that historically benefit from compressed sensing [3]. Additionally, having fewer samples insinuates a loss of statistical power in subsequent analyses. Thankfully, fMRI signals boast two beneficial characteristics that are promising for compressed sensing: 1. they are linear time-invariant, and 2. they lie within, and can be transformed by, a Fourier basis [4].

1.1 Motivation

In medical imaging, it is often assumed that an image is sampled at the Nyquist rate, s.t., enough discrete measurements are taken to reconstruct a continuous whole ($M > N$) without loss of information. If high-fidelity reconstruction is possible sampling below the Nyquist rate, then MRI modalities would benefit since discerning a signal and quickly turning over an analysis reduces real costs. These potential gains beg the question: if an underdetermined linear system can be solved after sampling below the Nyquist rate, can we collect fewer samples and still recover a high-quality fMRI under a compressed sensing paradigm? *The purpose of this project is to explore approaches to compressed sensing that yield meaningful signal recoveries from heavy undersampling.*¹

2 Materials & methods

2.1 Dataset & Preprocessing

Since this project is exploratory, and also introductory, a proof-of-concept dataset was used to better interpret and understand compressed sensing in the context of fMRI at a small scale. Hence, the dataset used for this project were the voxel time series and associated task response function associated with the fMRI from Exercise 4, which, thankfully, was already corrected for slice-timing and motion. The temporal resolution (TR) was 2 seconds, there were 171 volumes in the image, and the spatial resolution was 3.75x3.75x3.8mm.

¹Code is available at: <https://github.com/zstoebns/CSfMRI-TS>.

The image was loaded with `nibabel` in Python and the stimulus file was loaded with `pandas`. In the same fashion as the exercise, the canonical hemodynamic response function (HRF), shown in Figure 5, was constructed from a double-gamma basis and convolved with the task response function (TRF) to generate an expected response function, shown in Figure 6. Next, nuisance regression using a general linear model (GLM) estimated beta coefficients for the expected response + linear and quadratic trends that fit the raw signal. Then, the beta map of the response coefficients for all voxels was used to select the most active voxel in a specified slice (slice=10). Compressed sensing was then explored using this voxel's time series.

2.2 Sensing voxel time series

Three recovery algorithms were investigated: embedded conic solver (ECOS) [5], orthant-wise limited-memory quasi-Newton (OWL-QN) [6], and block sparse Bayesian learning with bound optimization (BSBL-BO) [7]. Three separate signals were sensed at 10% undersampling intervals: raw signal (\mathbf{Y}), estimated signal from GLM fit ($\hat{\mathbf{Y}}$), and residual signal (\mathbf{Y}_r) from denoising the raw signal after nuisance regression. The discrete cosine transform (DCT) was chosen as the basis for compute speed and interpretability in the real domain. Figure 1 shows the three signals and their corresponding DCTs.

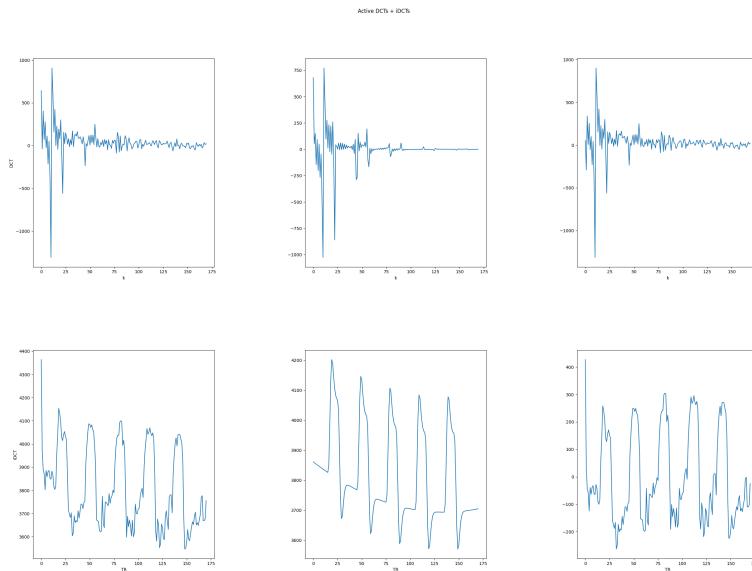


Figure 1: The DCTs of \mathbf{Y} , $\hat{\mathbf{Y}}$, and \mathbf{Y}_r (top) and their signals (bottom).

The compressed sensing problem was formulated as:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = \mathbf{Q}\mathbf{A}\mathbf{x}, \quad (1)$$

where \mathbf{Q} is the sampling identity matrix, \mathbf{A} is the measurement matrix defining the basis that transforms unknown full vector \mathbf{x} in the frequency domain to the sub-Nyquist observations \mathbf{y} in the time domain.

For recovery algorithms, the root mean squared error (RMSE) and peak signal-to-noise ratio (PSNR) curves were gathered to summarize trends across each degree of undersampling.

2.2.1 ECOS

ECOS falls along the line of traditional solvers for linear and quadratic programs, leveraging second-order cones, i.e., regions in the objective function's subspaces formed by the program's constraints, and finding optimal points within those cones. This algorithm is the default used by `cvxpy`, which was the library used to define and solve 1 at this baseline.

2.2.2 OWL-QN

OWL-QN is a variant of the L-BFGS algorithm and solves L1 regularization problems of the form: $f(\mathbf{x}) = g(\mathbf{x}) + C\|\mathbf{x}\|_1$ where g is a differentiable convex loss function and C is a constant. This algorithm exploits the inherent sparsity of the mode f and therefore is well-suited to this task. The L-BFGS solver distributed through `pylbfgs` was parametrized to run OWL-QN on this problem with a custom evaluation function, orthantwise constant of 5, and a Wolfe line search [8].

2.2.3 BSBL-BO

BSBL-BO was specially designed for compressed sensing signals with heavy amounts of non-sparsity but that are block sparse, i.e., equally-long time windows in the signal are intra-correlated which can be exploited for sensing. In particular, this method was shown to be far more effective at recovering action potential signals, which bear many similarities to fMRI signals, compared to convex optimization methods [9]. This method was performed using the BSBL-BO solver defined in `pybsbl` with $\lambda = 1$, $\gamma = -1$, initial $\lambda = 1e-3$, and $\epsilon = 1e-5$ for 100 iterations.

2.3 Nyquist rate

The Nyquist rate equals double the maximum non-zero frequency from the Fourier transform, which can be computed for both the HRF and TRF. Listing 1 shows the computation of the Nyquist rate computed from the Fourier transform and the frequency domain corresponding to the total number of samples taken at some TR.

```

1 import numpy as np
2
3 eps = np.finfo(float).eps
4
5 def nyquist_rate(ft, xf):
6     """
7         Computes Nyquist rate.
8         Parameters:
9             ft = Fourier transform sequence
10            xf = corresponding frequency domain
11        Returns:
12            scalar Nyquist rate
13        """
14    return 2 * xf[np.argwhere(np.abs(ft) > eps).max()]

```

Listing 1: Nyquist rate

3 Results

The Nyquist rate based on the HRF ($0.47 \text{ Hz} = 93\%$) and task response function ($0.49 \text{ Hz} = 98\%$) were slightly less than the sampling rate ($1/\text{TR} = 0.5 \text{ Hz}$) so all undersampling occurred below the Nyquist rate. Figures 2, 3, and 4 show the RMSE and PSNR scores at each level of undersampling for each of the three algorithms. Table 1 references the appendix figures showing examples of recovery results at 10%, 50%, and 90% for each signal for each of the algorithms.

3.1 ECOS

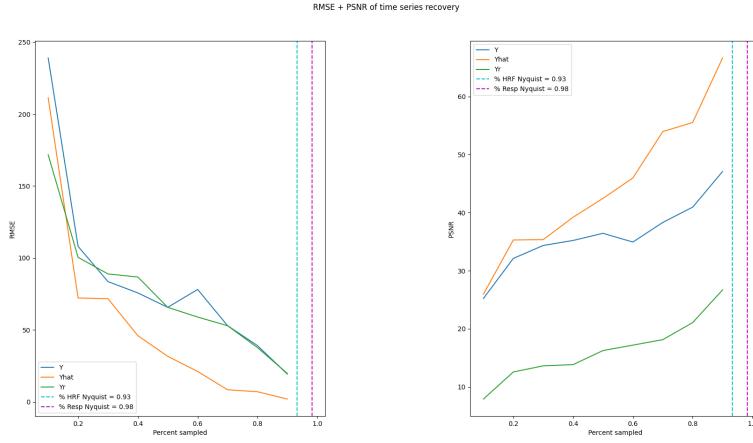


Figure 2: Summary metrics of RMSE (left) and PSNR (right) voxel time series recovery using L1 minimization in a pure convex optimization formulation solved with the ECOS algorithm.

3.2 OWL-QN

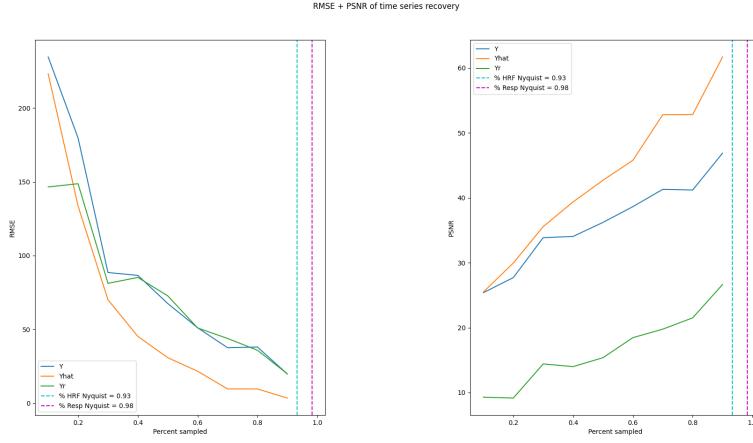


Figure 3: Summary metrics of RMSE (left) and PSNR (right) voxel time series recovery using L1 minimization in using the OWL-QN algorithm.

Table 1: Appendix figures with different levels of recovery for each algorithm.

| Algorithm | 0.1 | 0.5 | 0.9 |
|-----------|-----|-----|-----|
| ECOS | 7 | 8 | 9 |
| OWL-QN | 10 | 11 | 12 |
| BSBL-BO | 13 | 14 | 15 |

3.3 BSBL-BO

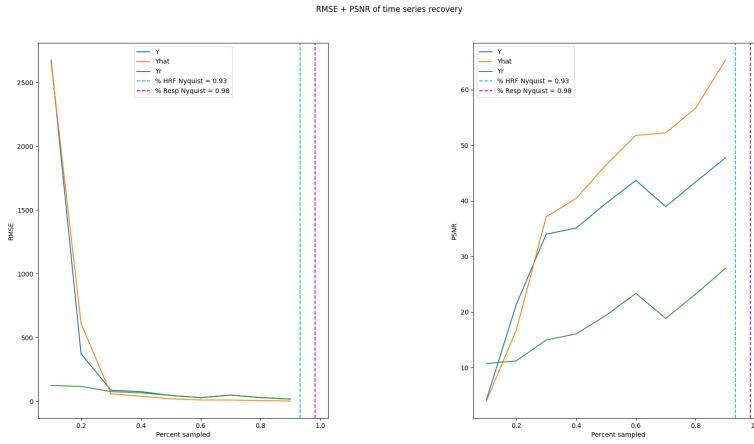


Figure 4: Summary metrics of RMSE (left) and PSNR (right) voxel time series recovery using L1 minimization in using the BSBL-BO algorithm.

4 Discussion

As expected, more samples yielded better recovery of the true signal for all three algorithms. ECOS and OWL-QN had similar scores for both RMSE and PSNR with linear increases in recovery performance with greater sampling. Such linear trends are undesirable for finding inflection points suitable for proposing viable subsamplings for speeding up acquisition. Hence, ECOS and OWL-QN are not promising methods for compressed sensing fMRI. All three algorithms had similar PSNR scores.

Surprisingly, the BSBL-BO method achieved low error, demonstrated by the strong elbow in the RMSE curve converging at around 30% subsampling. Comparing the 50% subsampled results shown in the Appendix, BSBL even sensed missing noise and outliers (e.g. the first time point), likely due to the Bayesian learning involved but even without overlapping blocks. This success suggests that there may be some utility in compressed sensing fMRI time series. However, these methods would require much more study to warrant any sort of practical deployment.

4.1 Challenges & limitations

The primary challenge of this work was understanding the methodologies for compressed sensing as a novice and then applying those in the context of fMRI, which is not well-suited to recovery in severe undersampling due to the high amount of noise present in fMRI time series and also relative non-sparsity. Compressed sensing is fairly esoteric and new, so literature and resources are sparse;

a library for general compressed sensing in signal processing would be extremely useful and save practitioners a lot of time, not needing to rebuild tricky implementations of established methods.

4.2 Future directions

Future directions for this work include a study on whole-volume sensing and application to a variety of datasets; like seed-based correlation maps, compressed sensing at varied subsamplings might serve as a useful empirical quality assurance measure since confounding image noise would likely thwart optimal recovery, even at the lowest amount of undersampling. Additionally, score-based networks that use Langevin annealing have achieved promising results in compressed sensing MRI [10] and deep residual networks using variable density spiral trajectories have even seen decent recovery of whole-volume fMRI [11]. Therefore, learning methods may yield superior recovery results compared to the conventional recovery algorithms discussed in this work.

References

- [1] E. J. Candès *et al.*, “Compressive sampling,” in *Proceedings of the international congress of mathematicians*, vol. 3, pp. 1433–1452, Citeseer, 2006.
- [2] D. Angelosante, G. B. Giannakis, and E. Grossi, “Compressed sensing of time-varying signals,” in *2009 16th International Conference on Digital Signal Processing*, pp. 1–8, IEEE, 2009.
- [3] X. Zong, J. Lee, A. J. Poplawsky, S.-G. Kim, and J. C. Ye, “Compressed sensing fmri using gradient-recalled echo and epi sequences,” *NeuroImage*, vol. 92, pp. 312–321, 2014.
- [4] O. Jeromin, M. S. Pattichis, and V. D. Calhoun, “Optimal compressed sensing reconstructions of fmri using 2d deterministic and stochastic sampling geometries,” *Biomedical engineering online*, vol. 11, no. 1, pp. 1–36, 2012.
- [5] A. Domahidi, E. Chu, and S. Boyd, “ECOS: An SOCP solver for embedded systems,” in *European Control Conference (ECC)*, pp. 3071–3076, 2013.
- [6] G. Andrew and J. Gao, “Scalable training of l 1-regularized log-linear models,” in *Proceedings of the 24th international conference on Machine learning*, pp. 33–40, 2007.
- [7] Z. Zhang and B. D. Rao, “Extension of sbl algorithms for the recovery of block sparse signals with intra-block correlation,” *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 2009–2015, 2013.
- [8] P. Wolfe, “Convergence conditions for ascent methods,” *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969.
- [9] H. Park and X. Liu, “Study on compressed sensing of action potential,” *arXiv preprint arXiv:2102.00284*, 2021.
- [10] A. Jalal, M. Arvinte, G. Daras, E. Price, A. G. Dimakis, and J. Tamir, “Robust compressed sensing mri with deep generative priors,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 14938–14954, 2021.
- [11] X. Li, T. Cao, Y. Tong, X. Ma, Z. Niu, and H. Guo, “Deep residual network for highly accelerated fmri reconstruction using variable density spiral trajectory,” *Neurocomputing*, vol. 398, pp. 338–346, 2020.

A Appendix

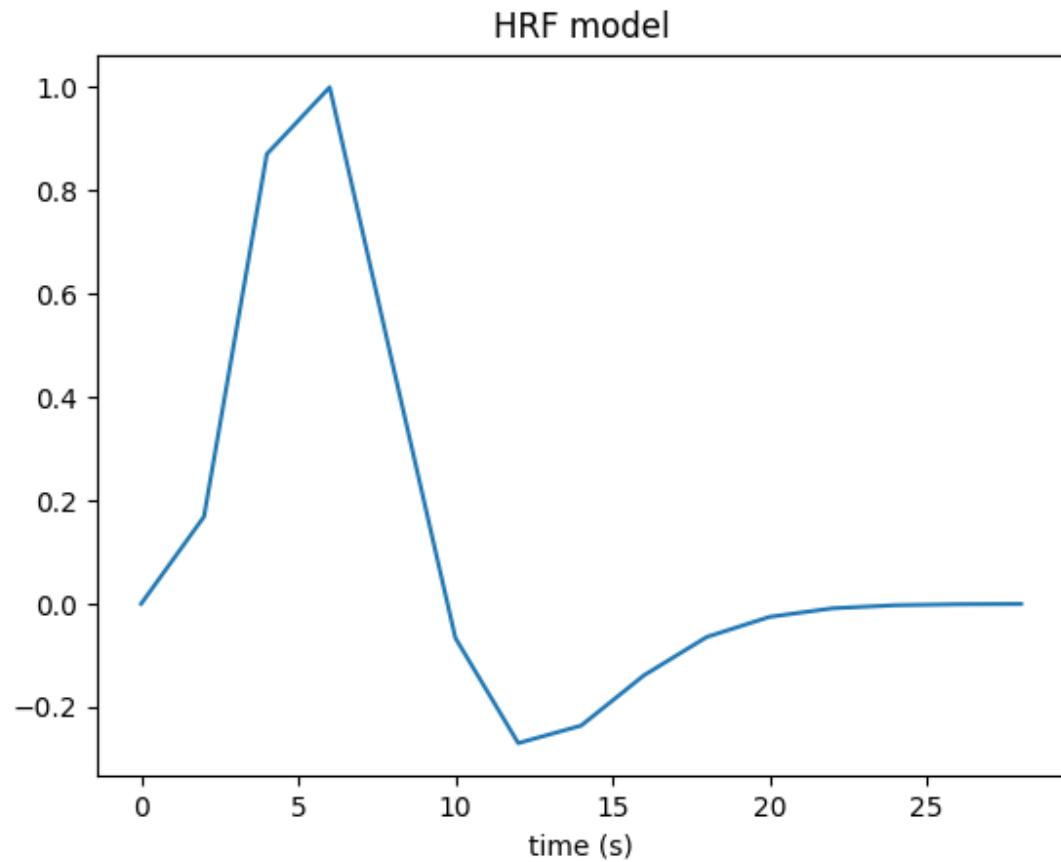


Figure 5: Canonical HRF used to construct the predicted response.

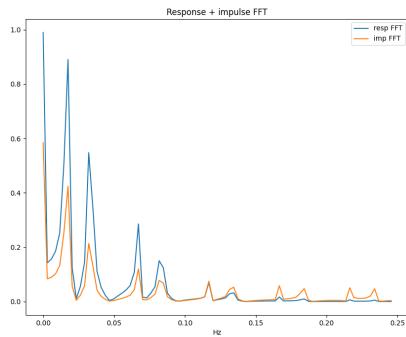
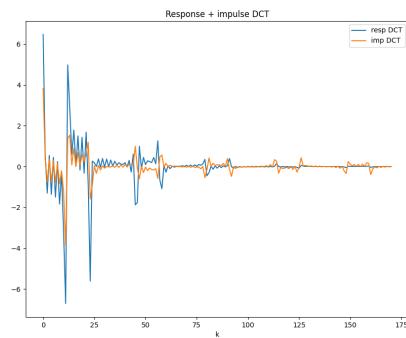
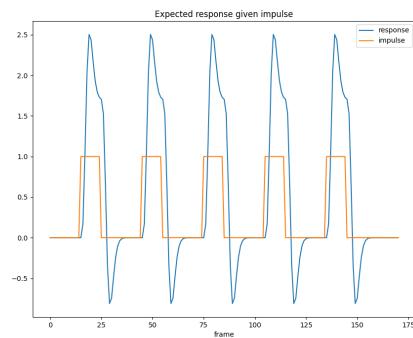


Figure 6: Expected response signals given the task impulse + their spectral transforms.

A.1 ECOS

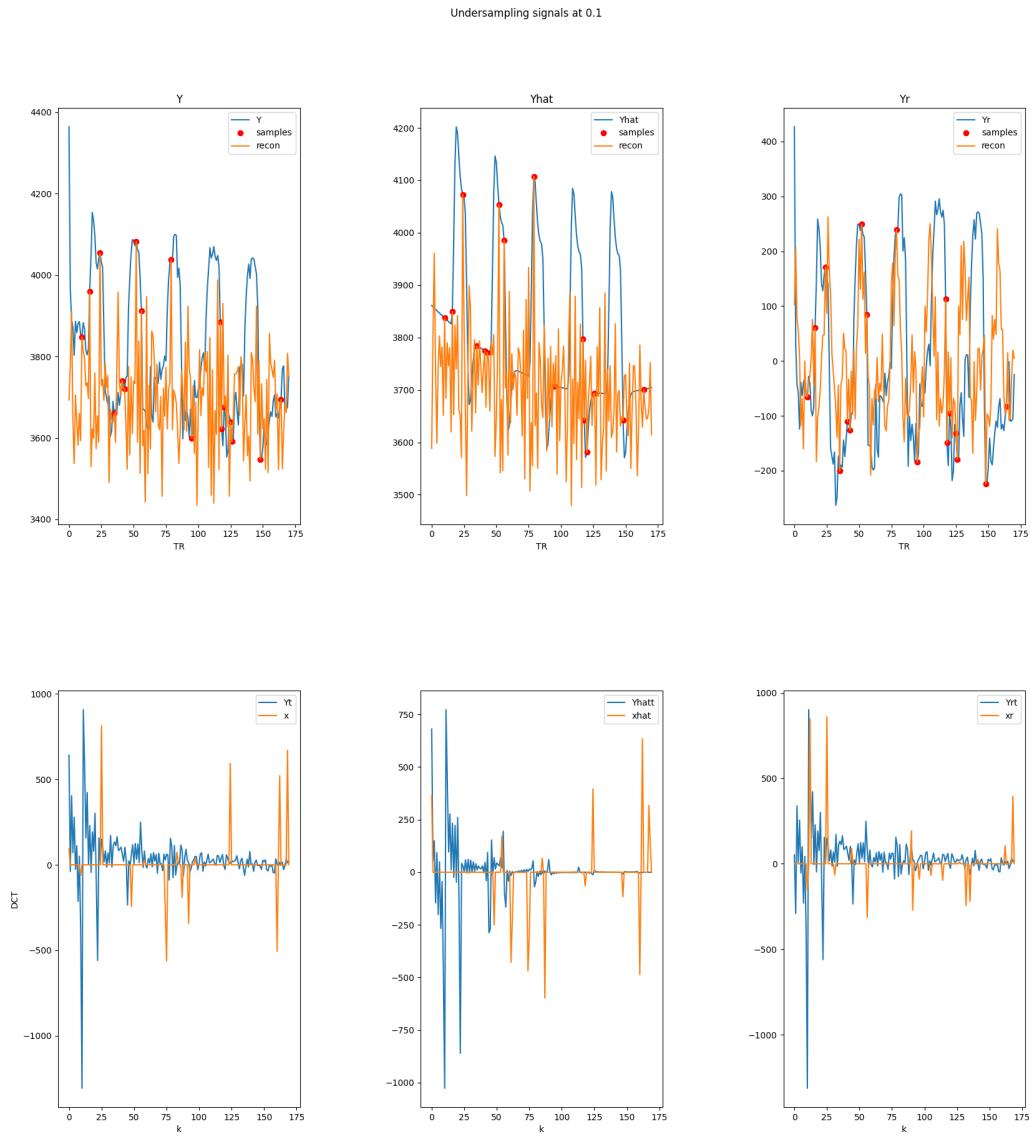


Figure 7: Signal sensing at 10% undersampling using pure convex optimization solved with the ECOS algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

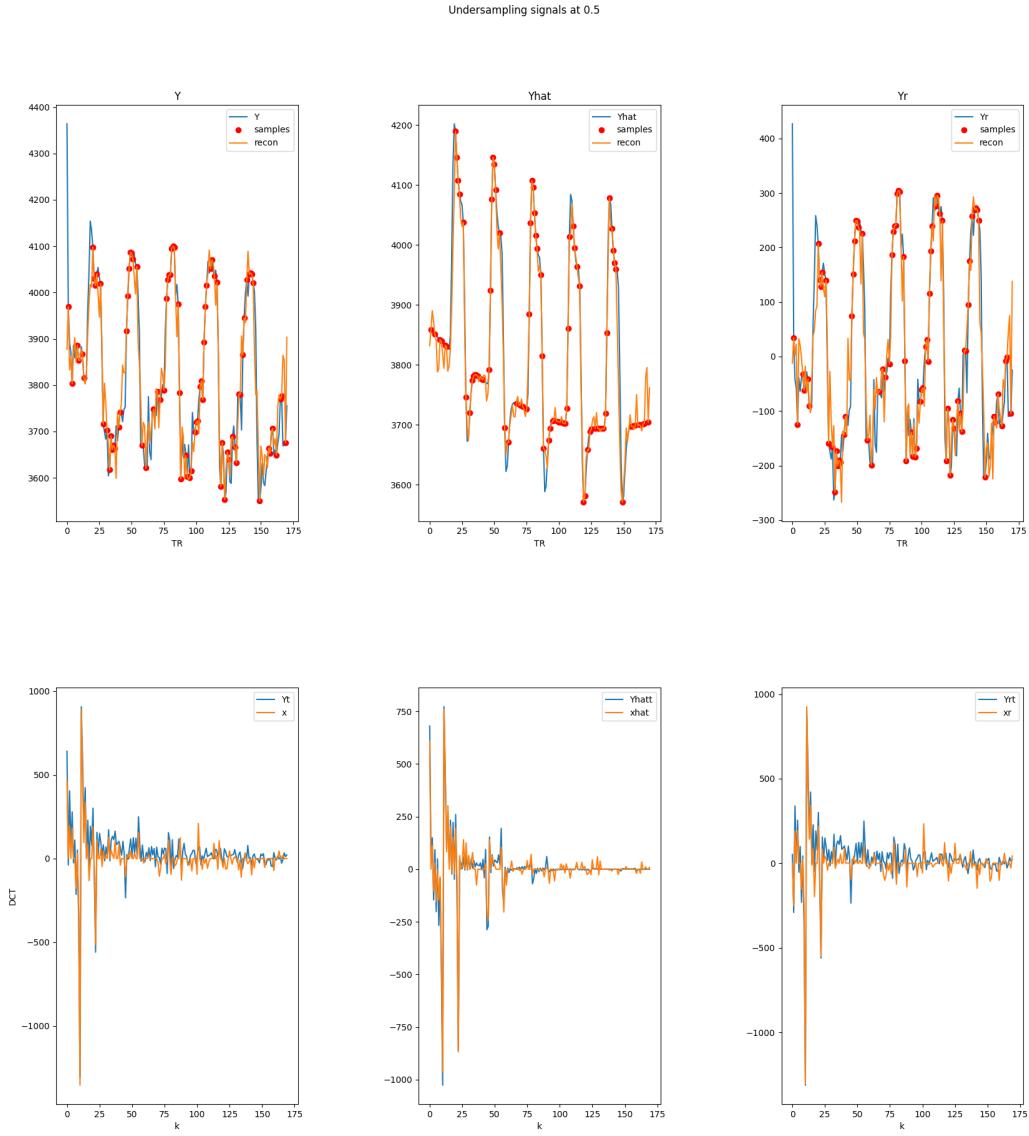


Figure 8: Signal sensing at 50% undersampling using pure convex optimization solved with the ECOS algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

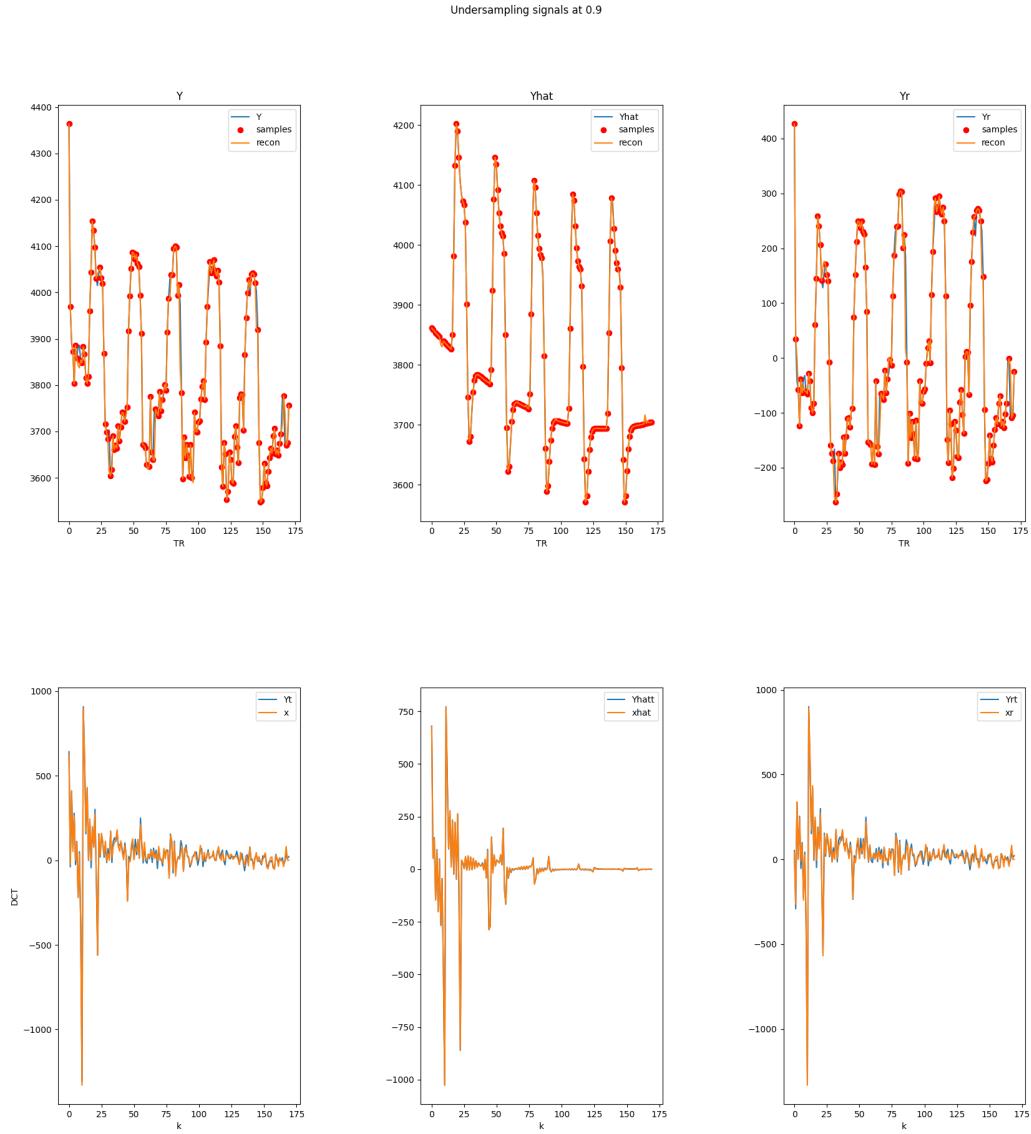


Figure 9: Signal sensing at 90% undersampling using pure convex optimization solved with the ECOS algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

A.2 OWL-QN

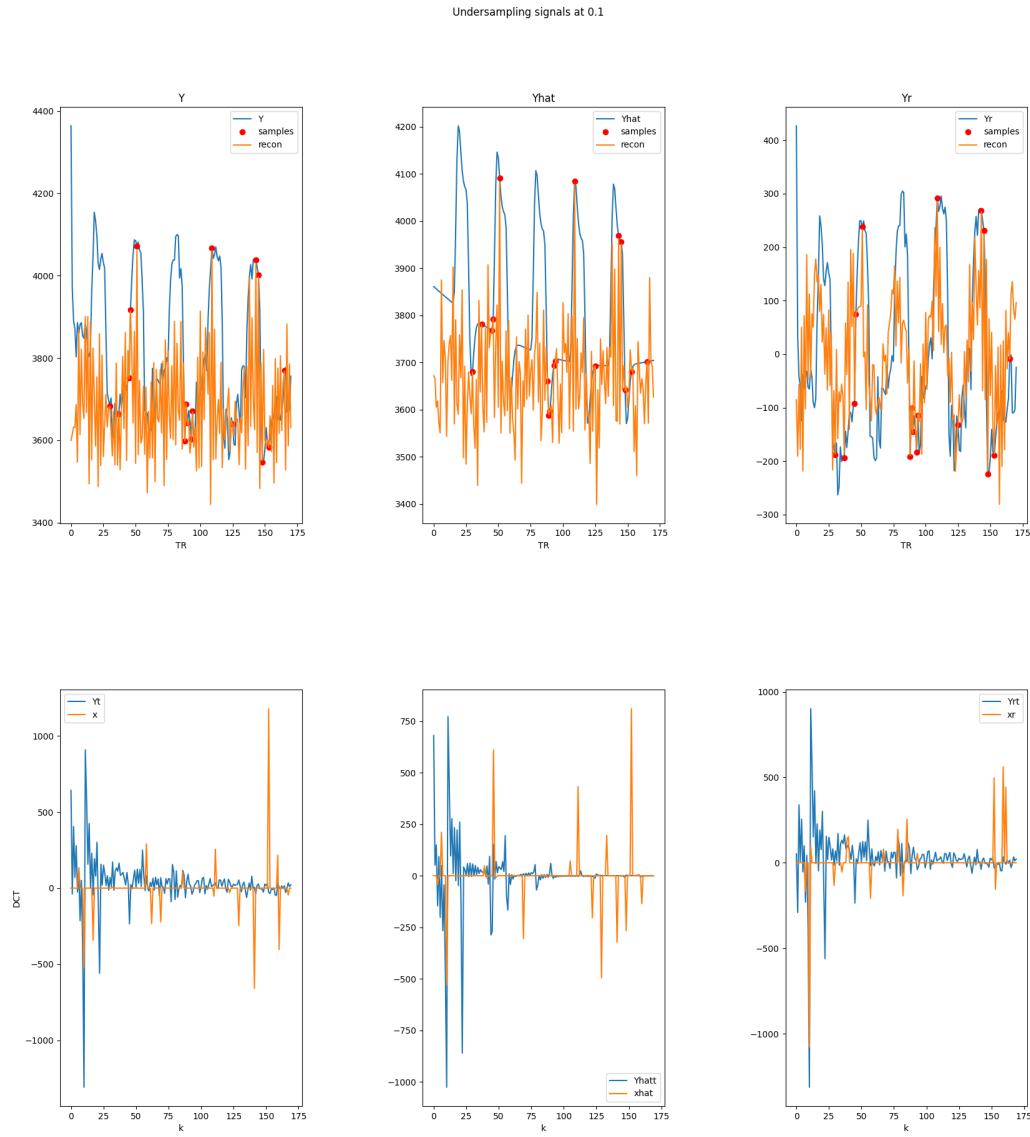


Figure 10: Signal sensing at 10% undersampling using the OWL-QN algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

Undersampling signals at 0.5

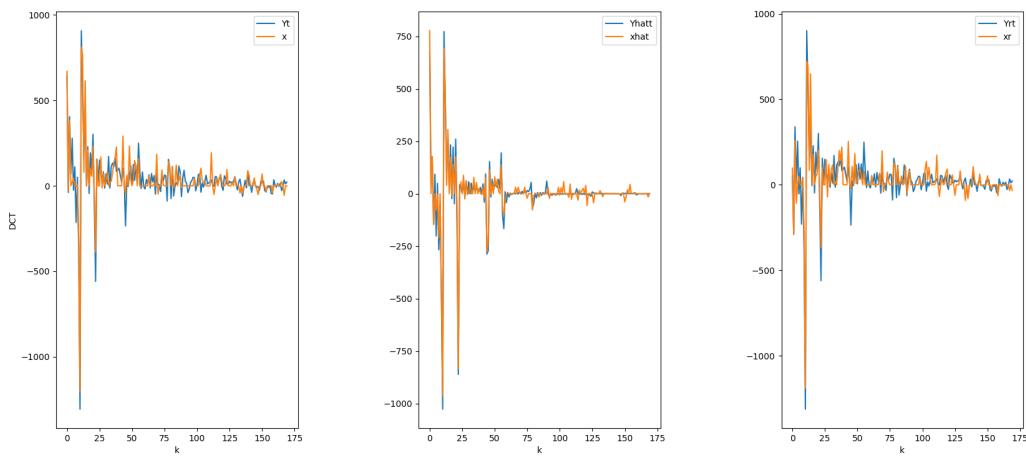
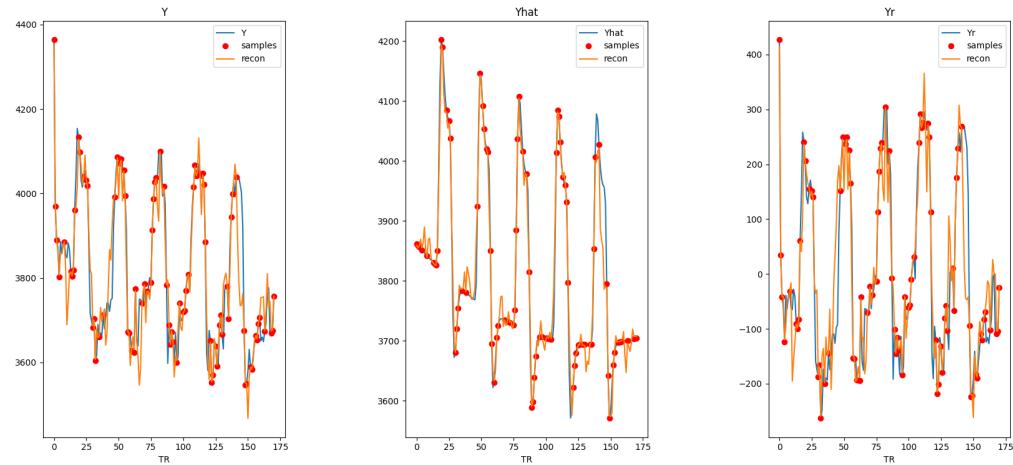


Figure 11: Signal sensing at 50% undersampling using the OWL-QN algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

Undersampling signals at 0.9

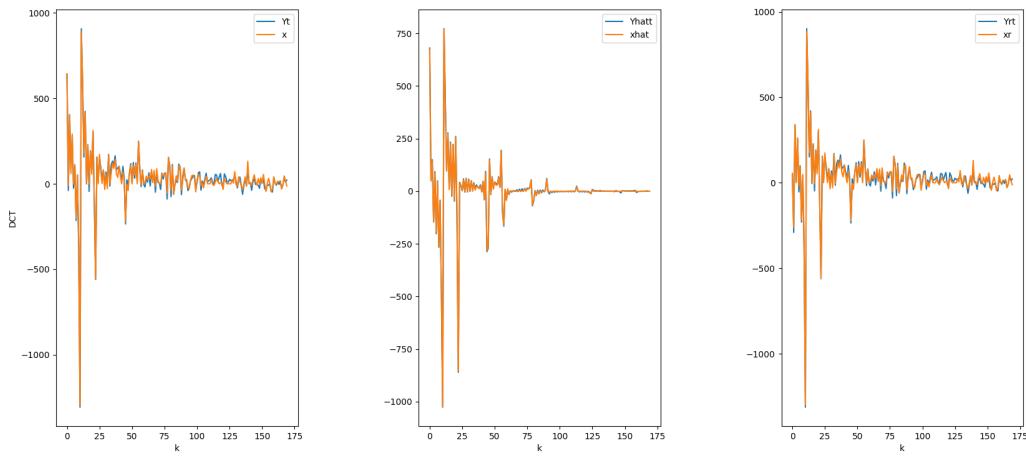
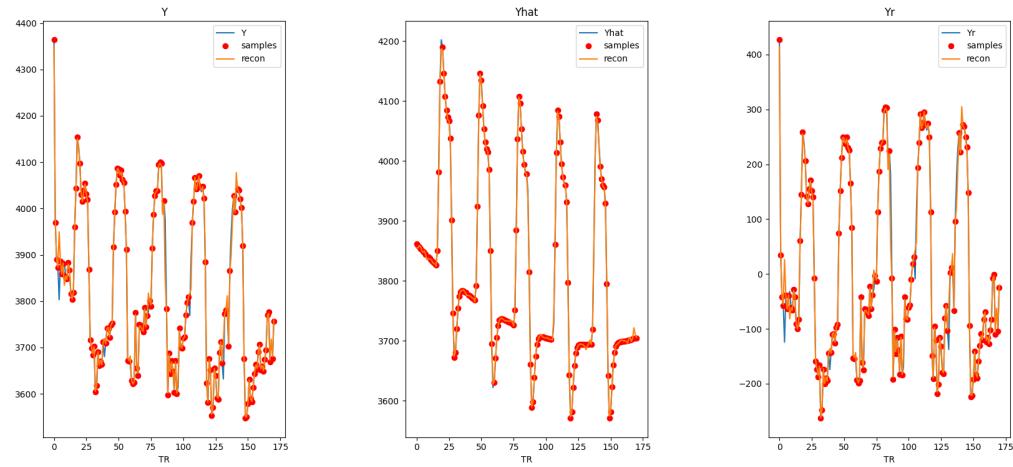


Figure 12: Signal sensing at 90% undersampling using the OWL-QN algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

A.3 BSBL-BO

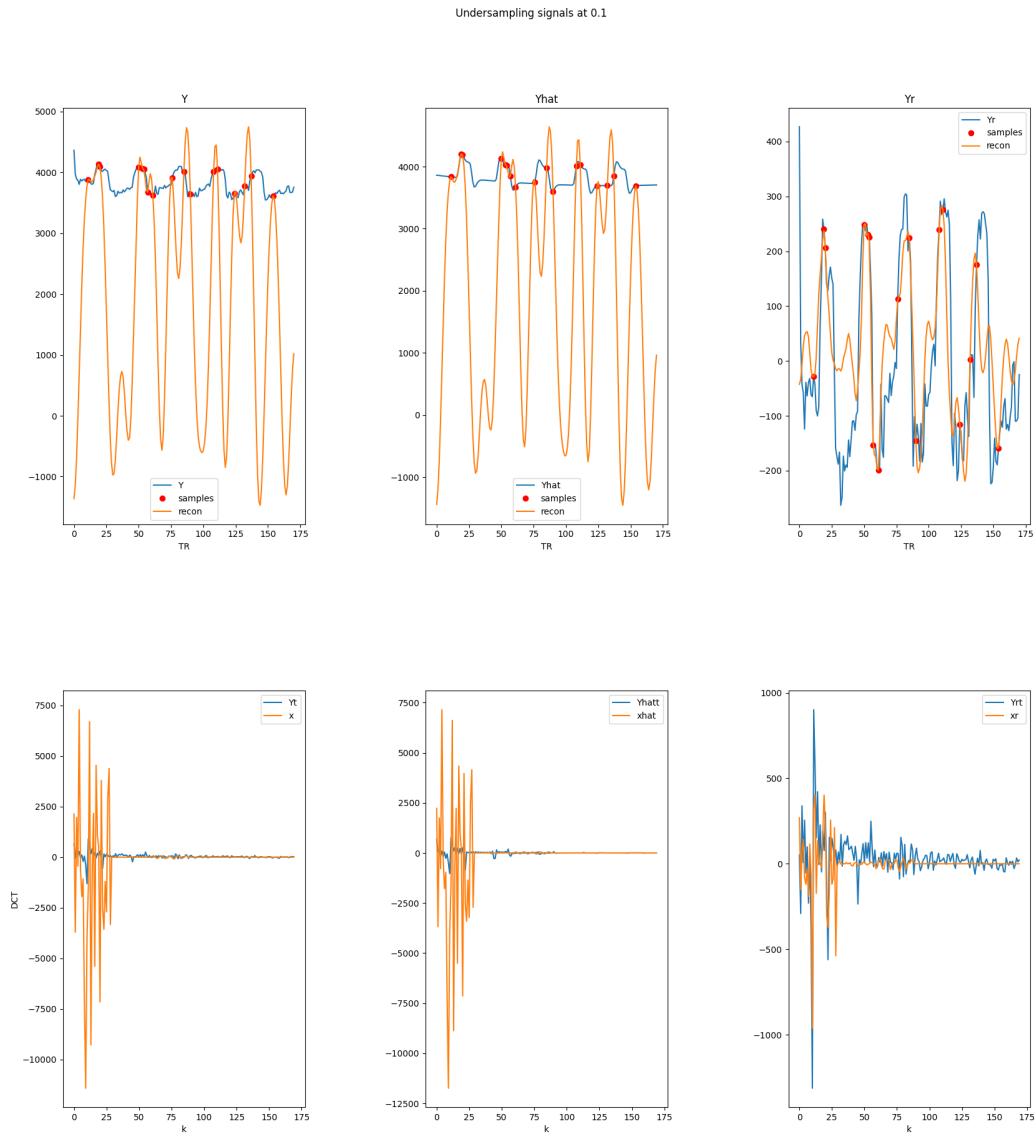


Figure 13: Signal sensing at 10% undersampling using the BSBL-BO algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

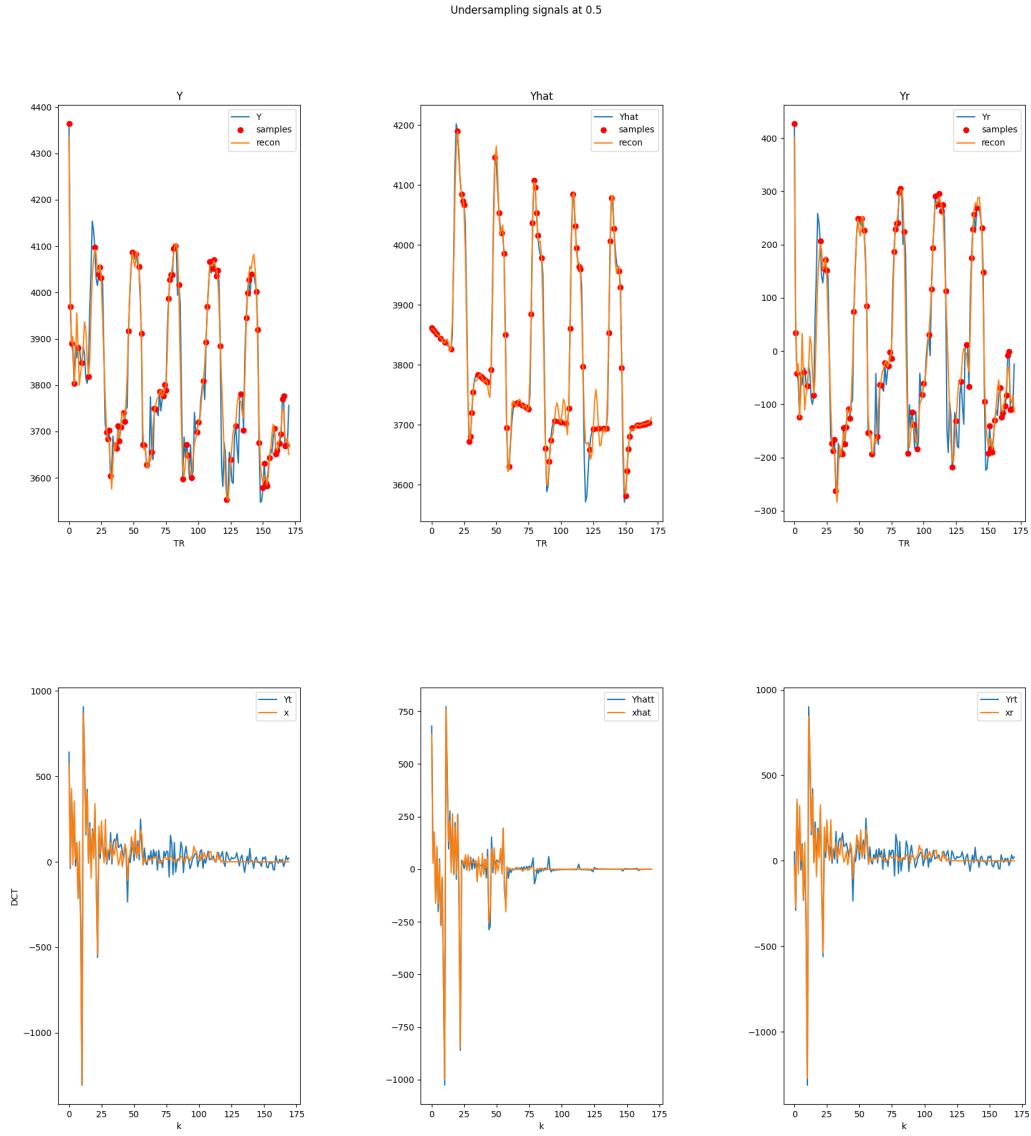


Figure 14: Signal sensing at 50% undersampling using the BSBL-BO algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.

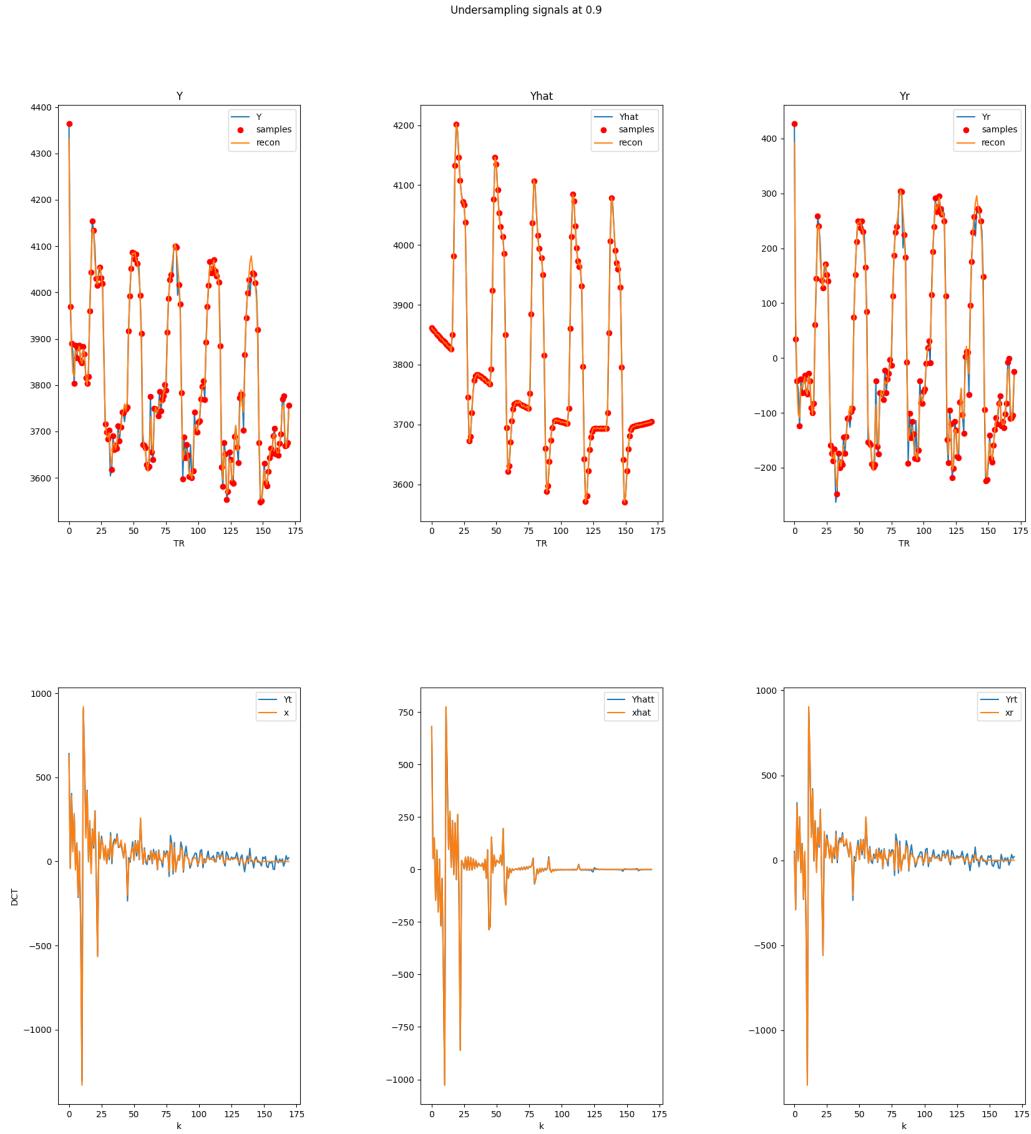


Figure 15: Signal sensing at 90% undersampling using the BSBL-BO algorithm. Y_i corresponds to time-domain signals whereas x_i corresponds to frequency-domain signals.