Complex Randomness-in-Structured Projections for Compressed Sensing (Technical Report)

Abdolreza Abdolhosseini Moghadam and Hayder Radha
Dep. of Electrical and Computer Engineering, Michigan State University, Email: {abdolhos,radha}@msu.edu

***Abstract*—We consider the problem of recovering a k-sparse signal (x) from a limited number of linear, noiseless compressive samples (y) using *random sparse projections* with some structure. Our approach is based on constructing sparse projections using strategies rooted in combinatorial design. Such sparse projection naturally divides the n-dimensional signal (x) into smaller sub-signals that exhibit some structured overlaps among them. In turn, each sub-signal is mapped onto a sample of the measurement vector (y). We show that when the number of measurement samples/sub-signals (m) is sufficiently large then this mapping leads to the isolation of a significant fraction of the k non-zero coefficients into different sub-signals (and hence measurement samples). We are able to recover the non-zero coefficients iteratively using a low-complexity algorithm that is reminiscent of well-known iterative channel decoding algorithms (e.g. LDPC). We prove that, using this framework, $k < m \leq 2k$ compressive samples, on average, are sufficient to recover a k-sparse signal perfectly, which is less than the sample requirements for the most complex compressed sensing frameworks. Meanwhile, the complexity of our decoding process is only $O\left(k \log(k) \log\left(\frac{n}{k}\right)\right)$ which is lower than the complexity of many well-known greedy solvers.**

***Index Terms*—Compressed Sensing, CRISP**

I. Introduction

The emerging field of Compressed Sensing (CS) [1]-[2] has shown that for many classes of signals we can sample at much lower rates than the Nyquist rate without losing any information. Consider a length n signal s which has a k sparse real-valued decomposition x in a known basis $\Psi_{n \times n}$: $s = \Psi x$. By k sparse decomposition we mean that x is non-zero only in k coordinates: $k = \|x\|_0 := \#\{i : x_i \neq 0\}$. The main idea of CS is that instead of sensing n samples ($\{s_i\}$), the encoder projects s into an incoherent frame $\phi_{m \times n}$ (with respect to the sparsifying domain Ψ) and senses m (where $k < m < n$) compressive samples y according to the projection (sensing) matrix $P = \phi\Psi$, i.e. $y = \phi s = Px$. Accordingly, the decoder recovers the sparse representation x (or equivalently s) by utilizing the compressive samples y and the projection matrix P. Generally the under-determined system of equations $y = Px$ has an infinite set of solutions. However, if certain properties outlined in [1]-[2] are met then it is possible to find a unique solution using the following optimization problem:

$$x = \arg\min \|\hat{x}\|_0 = \arg\min \|\hat{x}\|_1 : y = P\hat{x} \qquad (1)$$

In many CS applications (e.g. image sensing), the signal of interest has a large dimension. Therefore, designing low complexity recovery algorithms is of great importance. Further, recovering a signal from a minimum number of compressive samples is highly desirable. However under state-of-the-art CS solutions, the complexity of the solver has an inverse relationship to the number of compressive samples. For instance, greedy algorithms [3]-[5] have low complexities but require relatively large numbers of samples, on the other hand, convex relaxation (l_1 optimization) algorithms [6] recover the signal with fewer samples but they are extremely complex ($\Omega(n^3)$) relative to greedy algorithms.

In this paper, we propose Complex Randomness-in-Structured Projection (CRISP) matrices which enable a low complexity recovery for compressive sampling of k-sparse signals (or signals which have k-sparse representations). CRSIP is based on constructing sparse projections using strategies rooted in combinatorial design. A CRSIP projection naturally divides the n-dimensional signal (x) into smaller size sub-signals (or subsets). In section II, we show that dividing the signal coefficients (through the CRISP projection matrix) into random subsets with small structured overlaps results in the isolation of a significant fraction of non-zero coefficients into different compressive samples. Hence one can use a "divide-and-conquer" strategy to recover these isolated non-zero coefficients. After removing the effect of these isolated (recovered) coefficients from the compressive samples, one can continue this procedure iteratively until all coefficients are determined. Such strategy is reminiscent of well-known iterative decoding algorithms used in many signal processing and communication applications such as channel decoding (e.g., Low-Density-Parity-Check codes and its popular decoding algorithms that are based on belief propagation, message passing, etc. [12]). Our simulation results in section III verifies our analysis, and they show that under CRISP, $k < m \leq 2k$ complex valued samples are sufficient to recover a k sparse signal perfectly. We also show that the complexity of the CRISP solver is less than $O(k\log(k)\log(n/k))$ which is faster than Cosamp [4], an efficient greedy algorithm. In other words, on average, CRSIP requires less samples for perfect recovery, when compared to the most complex methods, meanwhile its complexity is lower than many fast greedy algorithms. Section IV concludes the paper.

II. Sparse Projection Matrices

Sparse projections provide an arguably unique framework of dividing the signal coefficients into smaller subsets; and hence, enabling a natural "divide-and-conquer" approach for the recovery process [7][8]. Traditional compressed sensing, which is based on popular dense projection matrices (e.g., random Gaussian or Fourier matrices), normally projects (all of) the n coefficients of the signal x (and hence all of the k non-zeroes of x) into each of the m observations of the vector y. Meanwhile a sparse-projection based method naturally projects different subsets $\chi = \{\chi_1, \chi_2, \ldots, \chi_m\}$ of the signal x into the observations $y = \{y_1, y_2, \ldots, y_m\}$. Here, χ is a superset where each of its members is a subset of x: $\forall i : \chi_i \subset x = \{x_1, \ldots, x_n\}$. For example, if the $i{-}th$ row of a sparse projection matrix P has zero entries everywhere except for non-zero entries in some locations, say $M_i = \{3, 8, 31\}$, then we have a corresponding subset $\chi_i = \{x_3, x_8, x_{31}\} = x_{M_i}$. Consequently, the $i - th$ compressive sample y_i is a function of the subset χ_i ($y_i = g_i(\chi_i)$), and y_i does not depend on the values of other elements of the signal x[1].

A. CRISP Projection Matrices

As before, let us denote the number of rows of the projection matrix P (corresponding to the number of sensed samples) by m and the number of columns of the projection matrix (or equivalently the signal length) by n. We form the final projection matrix P from a *base projection matrix* $p_{m\times n}$. Under the proposed CRISP framework, the base projection matrix $p_{m\times n}$ has binary entries (ones and zeros), and hence, this base matrix provides an underlying structure that can be used to populate the "ones" entries of $p_{m\times n}$ with random complex numbers (as explained later). This process of replacing the "ones" of $p_{m\times n}$ by complex numbers generates the final projection matrix P. Consequently, the underlying "structure" (and associated sparsity) of both the base matrix $p_{m\times n}$ and the final projection matrix P is the same. Below, we discuss the construction of both matrices.

Constructing the base projection matrix p can be achieved in many ways. One straightforward approach is to randomly select the entries of p where "ones" are placed. Such approach is illustrated by Algorithm 1. This simple approach does not impose any constraints on the underlying structure of p except for the parameter K, which is the number of "ones" that are placed randomly in each column of p. Note that K has to be small enough (relative to m and n) in order to have a sparse underlying structure for the (final) projection matrix P. Despite its simplicity, this random structure provides very good results as will be shown in the simulation section of the paper.

There are other methods that can be used for constructing a more constraining structure for the base matrix p. In other words, these methods can impose certain constraints on the underlying structure of the base projection matrix which leads (as explained above) to the same structure for P. Most notably, approaches that are inspired by channel coding methods for constructing parity check matrices, and in particular ones related to Low-Density-Parity-Check (LDPC) codes can be used for the binary base matrix p. More specifically, we focus here on combinatorial methods that are based on Balanced Incomplete Block Design (BIBD), which can be utilized to improve the performance of the CRSIP framework as we explain further below in this paper. Before proceeding, we briefly discuss the mapping of the binary base matrix p onto the final projection matrix P.

```
Input: $m$, $n$ and $K \geq 2$
Output: The base projection matrix $p$
for $i = 1$ to $n$ (all columns of $p$) do
    Select $K$ distinct random integers ($l = \{l_1, \ldots, l_K\}$)
    uniformly from $[m] = \{1, \ldots, m\}$
    For the $i - th$ column of $p$, set the entries in the rows
    indexed by $l$ to one and put zero in other indices:
        $p_{e,i} = \left\{ \begin{array}{ll} 1 & e \in l \\ 0 & e \in [m]/l \end{array} \right\}$    (2)
end
```

Algorithm 1: Generating the base projection matrix p.

In general, one can perform some form of matrix manipulation of p prior to replacing the "ones" of p onto some complex numbers. Further, one can have certain constraints on the complex numbers used in the projection matrix P. In this paper, we focus on applying two steps to the base matrix p, random permutation and employing random complex numbers. For example, the overall approach used when permuting (uniformly) the columns of p and substituting the non-zero entries of p by normalized random complex numbers yield the projection matrix P is shown below (Algorithm 2).

```
Input: The base projection matrix $p$
Output: The projection matrix $P$
Permute randomly the columns of $p$
for $i = 1$ to $m$ (all rows of $P$) do
    Let $M_i$ be the non-zero entries of the $i - th$ row of $p$
    and let $r = |M_i|$.
    Generate a row vector $f^{(i)}$ composed of $r$
    normalized random complex numbers.
    For the $i - th$ row of $P$, insert the random complex
    numbers $f^{(i)}$ in the column indices of $M_i$. That is:
    $P_{i,M_i} = f^{(i)}$ and $\forall l \notin M_i : P_{i,l} = 0$
end
```

Algorithm 2: Generating the projection matrix P from the base projection matrix p.

Now we say few words regarding the BIBD design procedure. As we eluded above, the incidence matrix of a Balanced Incomplete Block Design [9] represents a special case for the design of the base projection matrix p. Here, we briefly review a few key properties of such matrices. Let m, n, K and λ be positive integers such that $m > K \geq \lambda$. The incidence matrix of a (m, n, K, λ)-BIBD is a $m \times n$, matrix with the following properties:

[1] For brevity, we say that the $i - th$ measurement sample spans the indices of M_i.

- Each row is non-zero exactly in $r = \lambda(m-1)/(K-1)$ column indices. Also each of the $n = mr/K$ columns is non-zero exactly in K entries[2] and thus we have $\forall i, j \in [m] : r = |M_i| = |M_j|$.
- If we consider two distinct rows of this matrix, both are non-zero in exactly λ columns. That is: $\forall i, j \in [m], i \neq j : |M_i \cap M_j| = \lambda$.

Due to the special characteristics (listed above) of the proposed projection matrix, the following properties hold:

1) Since each column of our projection matrix is non-zero in K row indices, hence each signal coefficient x_i appears exactly in K random (different) equations/samples.
2) If the base projection matrix p is an incidence matrix of a BIBD then two distinct equations (compressive samples) have exactly λ common coefficients/unknowns.

In the next subsections we present the associated solver to this projection matrix and prove the validity of the solver.

B. CRISP Solver

We first highlight an important (yet arguably clear) proposition, which is critical for the successful working of the CRISP solver: if a subset of the coefficients has at most one non-zero coefficient, then only one complex valued random compressive sample is sufficient for the perfect recovery of that subset.

Proposition 1. Assume $u_{r\times 1}$ is a real-valued column vector and $f_{1\times r} = [\, e^{j\phi_1} \ldots e^{j\phi_r} \,]$ is a normalized row vector with (distinct) random complex entries, that is: $\forall i, j \in [r], i \neq j : \phi_i \in [0, \pi), \phi_i \neq \phi_j$. If u is non-zero in at most one index ($\|u\|_0 \leq 1$) then having $z = fu$ and f, one can recover u deterministically. Moreover having z and f, we can detect the event that u is non-zero in at least two indices. Specifically if $z = 0$ then $u = \mathbf{0}$. If the phase of z equals to the phase of $l - th$ entry of f ($\angle z = \angle f_l = \phi_l$) then u is non-zero only in the index of l and has the value of $u_l = z/f_l = \|z\|$. If the phase of z is not among the phases of f ($\{\phi_1, \ldots, \phi_r\}$), then u is non-zero in at least two indices.

Proof. Clearly if u is a zero vector then $z = 0$:

$$u = \mathbf{0} \Rightarrow z = 0 \tag{3}$$

Now assume $z = 0$. We show that with high probability $u = \mathbf{0}$. Note that if u is non-zero only in index of l ($\|u\|_0 = 1$ and $u_l \neq 0$) then $z = u_l e^{j\phi_l} \neq 0$. Now suppose u is non-zero in at least two indices ($\|u\|_0 \geq 2$). Hence z is a linear combination of at least two independent complex numbers. Suppose the values of all entries of u are bounded by M, i.e. $\forall u_i : |u_i| < M,\ M > 0$. Then the magnitude of z is a continuous random variable (at least) in the range of $[0, M)$. Thus the probability of the event that the magnitude of z takes *exactly* the value of zero, is zero. Hence with high probability:

$$u = \mathbf{0} \Leftarrow z = 0 \tag{4}$$

Combining (3) and (4) we have:

$$u = \mathbf{0} \Leftrightarrow z = 0 \tag{5}$$

Now assume u is non-zero only in the index of l. Hence: $z = u_l e^{j\phi_l}$. Note that in this case, the phase of z (i.e. $\angle z = \phi_l$) can be found in the ordered set of available phases in f ($\angle f = \{\phi_1, \ldots, \phi_r\}$). In other words, knowledge of the phase $\angle z = \phi_l$ translates into knowledge of the index l associated with the location of the (single) non-zero coefficient u_l. Consequently having z and f, we can recover u by:

$$\angle z = \angle f_l \Leftarrow u_i = \left\{ \begin{array}{cc} \|z\| & i = l \\ 0 & i \neq l \end{array} \right\} \tag{6}$$

Now we prove (by contradiction) that if the phase of z can be found in $\angle f$ then with high probability u is non-zero exactly in one index. Suppose u is non-zero in at least two indices and the phase of z can be found in $\angle f$. By the problem setup, the phase of z is a continuous random variable in the range of $[0, \pi)$. However, we know that the probability of the event that a continuous random variable ($\angle z$) attains the values of a finite set of numbers ($\angle f$), is zero and this is a contradiction. Thus:

$$\angle z = \angle f_l \Leftrightarrow u_i = \left\{ \begin{array}{cc} \|z\| & i = l \\ 0 & i \neq l \end{array} \right\} \tag{7}$$

Finally if $z \neq 0$ and the phase of z is not found in the ordered set $\angle f$, then u is non-zero at least in two indices. ■

The decoding scheme of the CRISP solver is iterative: we examine compressive samples to find some subsets of the signal spanning only one non-zero coefficient. This can be achieved by looking for the phases of the compressive samples ($\angle y = \{\angle y_1, \ldots, \angle y_m\}$) among the phases of the entries of the random vectors $f^{(\cdot)}$. For instance, assume that $\angle y_i$ equals to $\angle f_l^{(i)}$. Therefore using Proposition 1, one can determine the signal values in the $i - th$ subset (χ_i). Let x' be a $n \times 1$ vector that is zero in all indices except at the $l - th$ member of the $i - th$ index set M_i (i.e. index $M_{i,l}$) which equals to $y_i / f_l^{(i)}$. Subtracting the effect of the identified non-zero coefficient ($y' = Px'$) from the compressive samples leads to the following new under-determined system of equations: $y^{(1)} = y - y' = P(x - x') = Px^{(1)}$ where $x_e^{(1)} = x_e$ when $e \notin M_i$ and $x_e^{(1)} = 0$ when $e \in M_i$. Note that $\|x^{(1)}\|_0 = \|x\|_0 - 1$. Since there are exactly K distinct samples spanning the same non-zero coefficient, this reduction affects K more samples (equations). Specifically, if the base projection matrix p is the incidence matrix of a BIBD, then identifying the signal values in one subset reduces the number of unknowns (un-identified coefficients) in all equations $y_i = \sum_{e=1}^{r} f_e^{(i)} x_{M_{i,e}}$ by λ. Therefore the algorithm converges to the solution very fast. More specifically, consider a compressive sample y_t spanning two non-zeros including $x_{M_{i,l}}$: $y_t = \alpha x_{M_{i,l}} + \beta x_e,\ e \in \{1, \ldots, n\} - M_{i,l}$, where α and β are two distinct entries of $f^{(t)}$. Thus $y_t^{(1)} = y_t - y'_t = \beta x_e$. Consequently after this reduction, all compressive samples spanning two non-zeros such that one of these non-zeros is

[2] Note that the size of support of a signal (or sparsity measure) is denoted by k, while K is a parameter of a BIBD.

among the identified coefficients in the first iteration, turn into new samples spanning only one non-zero coefficient. Therefore in the next iteration, the phases of these compressive samples can be found among the available phases in the random vectors $f^{(\cdot)}$. This event results in identifying the signal values in new subsets (and consequently new coordinates).

Let ξ denotes the indices where the signal values are not determined yet at these indices. The proposed algorithm might be repeated until either all non-zeros are identified[3] or the number of un-identified coefficients $|\xi|$ equals to the rank of the projection matrix restricted to column indices ξ $(P_{,\xi})^4$. Otherwise our algorithm has failed. Algorithm 3 shows the CRISP decoding process. In the next subsection we present the analysis of the average behavior of CRISP. The analysis of the deviation (from the average behavior) is not in the scope of this report.

```
Input: $y$ and $P$
Output: the sparse vector $\hat{x}$ which $y = P\hat{x}$
Set $I = 0$, $y^{(0)} = y$, $\hat{x} = \mathbf{0}$ and $\xi = \{1, \dots, n\}$
repeat
    Find all indices $\beta = \{i : \ y_i^{(I)} \in \angle f^{(i)}\}$
    if $\beta = \emptyset$ then
        if $|\xi| \leq Rank(P_{,\xi})$ then
            $\hat{x_\xi} = \left(P^* P\right)^{-1} P^* y^{(I)}, \ \xi = \emptyset$
        end
        else
            Algorithm has failed
        end
    end
    for all $i \in \beta$ do
        Suppose $\angle y_i^{(I)} = \angle f_l^{(i)}$ then:
        $x'_e = \left\{ \begin{array}{cc} \|y_i^{(I)}\| & e = M_{i,l} \\ 0 & e \in \{1, \dots, n\} - M_{i,l} \end{array} \right\}$
        $\hat{x}_{M_{i,l}} = x'_{M_{i,l}}$
        $y^{(I)} = y^{(I)} - Px'$
    end
    $I = I + 1$, $\xi = \xi - M_i$
until all coefficients have been recovered ;
```

Algorithm 3: CRISP solver

C. The Average Behavior Analysis

In this subsection, we present the proof for our claim that CRISP on average needs only $k < m \leq 2k$ complex valued compressive samples for the perfect recovery of a signal, where k is the number of non-zero coefficients.

Lemma 1. *On average, CRISP requires $k < m \leq 2k$ compressive samples for the perfect recovery of a k sparse signal of length n. Moreover the complexity of such recovery is only $O\left(k \log(k) \log\left(\frac{n}{k}\right)\right)$.*

Proof. If non-zero coefficients are analogous to balls and compressive samples are analogous to bins then it is straightforward to see that the sensing and the decoding stages of the CRISP framework can be modeled as the following extension of "balls into bins" problem.

- **Sensing**: We have k sets of balls (corresponding to each non-zero coefficient) and m bins (corresponding to compressive samples). Each set of balls consists of K balls of the same color. For each set of balls, we randomly choose K different bins[5] and distribute the balls into those bins.
- **Recovery**: Whenever a bin has only one ball (or equivalently each time we observe that a sample spans only one non-zero coefficient), we remove that ball and all other $K - 1$ balls of the same color (we subtract the effect of that non-zero coefficient from the compressive samples).
- **Question**: Can we remove all balls if we follow the procedure outlined in the **Recovery** (or equivalently can we recover all non-zero coefficients)?

Let $C = m/k$ be the oversampling factor. Then the probability that a given sample y_j spans exactly i non-zeros (or equivalently the probability of the event that the $j-th$ bin has exactly i balls of distinct colors) can be calculated by:

$$\Pr\left(\|x_{M_j}\|_0 = i\right) = \Lambda_i = \frac{\binom{k}{i}\binom{m-1}{K-1}^i\binom{m-1}{K}^{k-i}}{\binom{m}{K}^k} \qquad (8)$$

Note that the enumerator is: choose i colors (distinct non-zero coefficients) among a total of k colors. Then take one ball from each of these i colors and throw them inside the $j-th$ bin. Now for each of these i colors, there are $K-1$ balls left to distribute among the remaining $m-1$ bins along with $k-i$ other sets of balls which have not been distributed yet as well. It is important to note that since the columns of CRISP projection matrix are permuted randomly, hence the probability that any given sample spans i non-zero coefficients, is constant for all samples. That is:

$$\forall l, j \in \{1, \dots, m\} : \Pr\left(\|x_{M_j}\|_0 = i\right) = \Pr\left(\|x_{M_l}\|_0 = i\right) \qquad (9)$$

Thus the expected number of samples, spanning only one non-zero is:

$$E\left(\#j : \|x_{M_j}\|_0 = 1\right) = m \Pr\left(\|x_{M_j}\|_0 = 1\right) \qquad (10)$$

$$= \frac{m\binom{k}{1}\binom{m-1}{K-1}\binom{m-1}{K}^{k-1}}{\binom{m}{K}^k} \qquad (11)$$

$$= \frac{mk(m-1)\frac{(m-1)!}{(K-1)!(m-K)!}\left(\frac{(m-1)!}{K!(m-K-1)!}\right)^{k-1}}{\left(\frac{m!}{K!(m-K)!}\right)^k} \qquad (12)$$

$$= \frac{kK(m-K)^{k-1}}{m^{k-1}} \qquad (13)$$

To simplify our analysis, let us consider the asymptotic case where k (and hence m and n) tends to infinity. Then the

[3] At this stage all updated compressive samples $y^{(\cdot)}$ are zero.

[4] Clearly at this stage, the number of unknowns equals to the number of equations and all remaining coefficients can be recovered.

[5] Since each column is non-zero in K random rows.

expected number of samples spanning only one non-zero would be:

$$\lim_{k\to\infty} E\left(\#j : \|x_{M_j}\|_0 = 1\right) = \lim_{k\to\infty} \frac{kK(m-K)^{k-1}}{m^{k-1}} \tag{14}$$

$$= \lim_{k\to\infty} kK\left(1-\frac{K}{m}\right)^{k-1} = \lim_{k\to\infty} kKe^{-\frac{K(k-1)}{m}} \tag{15}$$

$$= kKe^{-\frac{K}{C}} \tag{16}$$

Note that for large values of k and fixed values of $m = Ck$, the expected number of samples spanning only one non-zero is a linear function of k. These samples identify the locations of isolated non-zero coefficients which consequently lead to the recovery of the associated subsets of coefficients[6].

Here we should emphasize that not all of these isolated coefficients provide new information. For instance, it is possible that there are several samples spanning only one specific non-zero coefficient x_i. So it is important to exclude this redundant information from our analysis. The probability that one sample y_a spans only one non-zero coefficient x_i, given that there exists another compressive sample y_b ($b \neq a$) spanning the same coefficient x_i is:

$$\Pr\left(y_a = g_a(x_i) | y_b = g_b(x_i)\right) = \frac{\binom{k}{1}\binom{m-1}{1}\binom{m-2}{K-2}\binom{m-2}{K}^{k-1}}{\binom{m}{K}^{k}} \tag{17}$$

That is choose a non-zero coefficient x_i among k non-zeros and choose another sample y_a (except y_b) and let y_a be only function of x_i. Now randomly select $K-2$ other samples which would be functions of x_i and distribute the remaining $k-1$ non-zeros (counting their K repetitions) among remaining $m-2$ samples. Expanding (17) yields:

$$\mho = \Pr\left(y_a = g_a(x_i) | y_b = g_b(x_i)\right) = \tag{18}$$

$$\frac{k(m-1)\frac{(m-2)!}{(K-2)!(m-K)!}\left(\frac{(m-2)!}{K!(m-K-2)!}\right)^{k-1}}{\left(\frac{m!}{(m-K)!K!}\right)^{k}} = \tag{19}$$

$$\frac{kK(K-1)}{m}\left(\left(1-\frac{K}{m}\right)\left(1-\frac{K}{m-1}\right)\right)^{k-1} \tag{20}$$

Fix $m = Ck$, for large values of k (20) can be simplified to:

$$\lim_{k\to\infty} \mho = \frac{k(K^2-K)}{m}e^{-\frac{2K}{C}} \tag{21}$$

which is a linear function of sparsity measure k. Thus the expected number of samples which do not result to discovery of new coefficients (μ) is:

$$\mu = \frac{m}{K}\mho = \frac{m}{K}\frac{kK(K-1)}{m}e^{-\frac{2K}{C}} = k(K-1)e^{-\frac{2K}{C}} \tag{22}$$

Recall that (on average) $kKe^{\frac{K}{C}}$ samples span only one non-zero coefficient. Hence, the expected number of compressive samples spanning distinct non-zeros in the first iteration of the solver is:

$$\mu' = kKe^{\frac{K}{C}} - \mu = k\left(Ke^{-\frac{K}{C}} - (K-1)e^{-\frac{2K}{C}}\right) \tag{23}$$

In words, CRISP solver (on average) recovers μ' non-zero coefficients in the first iteration[7].

Since the columns of the CRISP projection matrix are permuted independently and uniformly, we can say that in the next iteration, there are $k^{(1)}$ non-zeros left to recover from m samples, where:

$$k^{(1)} = k - \mu' = kq,\ q = \left(1 - Ke^{-\frac{K}{C}} + (K-1)e^{-\frac{2K}{C}}\right) \tag{24}$$

Note that for fixed values of $C = m/k$, q is independent of k. It is straightforward to see that, (on average) in the $i-th$ iteration of the algorithm, there are: $k^{(i)} = kq^i$ non-zero coefficients which have not been recovered. Also note that even for $1 < C \leq 2$, we have $q < 1$. In other words, on average, $k < m \leq 2k$ is sufficient for the perfect recovery.

In the worst case, we might continue this process until there is only one non-zero left unidentified. To calculate the maximum number of required iterations for the perfect recovery, we should solve $kq^i \leq 1$ for i. Since $q < 1$, we need $O(\log(k))$ iterations to recover the signal. Note that on average, each row of our projection matrix has $r = nK/m$ non-zero entries[8]. In each iteration, for each of m samples, (on average) we need to search among r numbers to find a match between the phase of that compressive sample ($\angle y_i$) and the phases of the random complex vector $\angle f^{(i)}$. Without loss of generality assume that the phases of entries of $f^{(.)}$ are sorted ascending:

$$\forall i,j \in [r], l \in [m] : i < j \Rightarrow \angle f_i^{(l)} < \angle f_j^{(l)} \tag{25}$$

then the complexity of the solver on average would be $O\left(m\log(r)\log(k)\right) = O\left(k\log(k)\log\left(\frac{n}{k}\right)\right)$ which is less than the complexity of Cosamp [4]. ■

Finally it is crucial to highlight the following note: the results derived in this section are not in contradiction of the optimality of Robust Uncertainty Principles of [1]. Although *on average* we need $k < m \leq 2k$ complex valued samples for

[6] Intuitively, we expect that by increasing the oversampling factor $C = m/k$, all non-zero coefficients (counting their repetitions in K samples) appear in different samples. Equation (16) is indeed consistent with our intuition:

$$\lim_{C\to\infty} kKe^{-\frac{K}{C}} = kK$$

On the other hand if we design our projection matrix more dense by increasing the value of K (recall that each coefficient appears in K different samples), then in the extreme case each sample would be a function of all coefficients. This suggests that there would be no subsets of coefficients spanning only one non-zero. This is also consistent with our result in equation (16) as:

$$\lim_{K\to\infty} kKe^{-\frac{K}{C}} = 0$$

[7] Note that for large values of $C = m/k$, we have:

$$\lim_{C\to\infty} \mu' = k$$

i.e., if the number of samples is much greater than the number of non-zeros then k distinct non-zeros would be recovered in the first iteration of CRISP which is intuitive.

[8] If the base projection matrix is an incidence of a BIBD, then each row has exactly $r = nK/m$ non-zero entries.

the perfect recovery, however as we decrease the oversampling factor of $C = m/k$, then q tends to one. Thus less and less non-zero coefficients would be recovered in each iteration of the CRISP solver. Hence small deviation from this average behavior would lead to failure of the algorithm in extreme cases.

III. Simulation Results

We have tested the proposed CRISP on a large number of standard CS signals. Specifically we compared the performance of CRISP (in terms of quality of recovered signal as a function of the number of compressive samples and also the required time for decoding) versus the popular Gaussian random projection matrix and dominant Basis Pursuit (BP) and OMP solvers. These standard test signals from SparseLab [13] are sparse in DCT and Wavelet domains but not in the time domain. Thus the sparsifying transforms Ψ in these simulations are DCT/Wavelet transform matrices.

In Fig.1a, we show the results of an extreme case simulation. Not only the signal is very short $n = 10$, it is not sparse as well $k = 6$ (sparsity ratio of $k/n = 60\%$). Under traditional CS, it is impossible to recover such signals [11], however the CRISP solver is able to recover the signal perfectly with only $m = 7$ samples. We performed our simulations under various configurations of signal length and sparsity ratio (Fig. 1b- Fig. 1d and Fig. 2). In all these scenarios, $k < m \leq 2k$ complex samples was enough to achieve perfect reconstruction while BP and OMP failed to recover the signal given the same number of compressive samples. Moreover the required time for CRISP decoder is shorter than many fast greedy algorithms. In all simulations where the sparsity measure k is not very small, the required time for the signal recovery is less than the required time for BP and OMP, thus verifying our claim in practice.

Here we should highlight an important note: due to the strong structures and rich properties of BIBDs, we expect that a sensing matrix generated based on an incidence matrix of a BIBD boosts CRISP performances in terms of the solver complexity and also the sample requirements for the perfect recovery. For instance, in Fig. 2b, we have used a Hamming matrix of weight two (all m tuples with $K = 2$ non-entries in each column) as the base projection matrix p. Hence p is an incidence matrix of a $(m, m(m-1)/2, 2, 1)$-BIBD. In that simulation, we observed that if the incidence of a $(150, 11175, 2, 1)$-BIBD was used as the base projection matrix, then $m = 1.5k$ samples was enough for the perfect recovery. On the other hand, if we have used Algorithm 1 to generate the base projection, then $m = 2k$ samples was required for the perfect recovery and also the decoding time would increase as well. However we should recall that: although there are numerous methods for generating a BIBD with some given parameters [9], it is possible that there are no BIBD designs that exist in some configurations of (m, n, K, λ).

IV. Conclusion

In this paper, we showed that if we sense one complex valued compressive sample from random subsets of a sparse signal, such that any two subsets intersect in a few indices, then on average we only need $k < m \leq 2k$ complex valued compressive samples to recover a signal from these samples. Moreover the complexity of the recovering procedure is on the order of (or less than) the complexity of fast greedy algorithms.

Acknowledgment

This work was supported in part by NSF.

References

[1] Emmanuel Cands, Justin Romberg, and Terence Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. IEEE Trans. on Information Theory, 52(2) pp. 489 - 509, February 2006.

[2] David Donoho and Yaakov Tsaig, “Extensions of compressed sensing”. (Signal Processing, 86(3), pp. 533-548, March 2006).

[3] David L. Donoho, Yaakov Tsaig, Iddo Drori, and Jean-Luc Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit”. Preprint, 2007.

[4] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. Preprint, 2008.

[5] Y. C. Pati R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition”, Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers, Nov. 1993.

[6] S. Chen, , D. Donoho, and M. Saunders. “Atomic decomposition by basis pursuit”. Technical Report 479, Department of Statistics, Stanford University, May 1995.

[7] Radu Berinde and Piotr Indyk, “Sparse recovery using sparse random matrices”, Preprint, 2008.

[8] Shriram Sarvotham, Dror Baron, and Richard Baraniuk, “Sudocodes - Fast measurement and reconstruction of sparse signals”. Proc. IEEE Int. Symposium on Information Theory (ISIT), Seattle,Washington, July 2006.

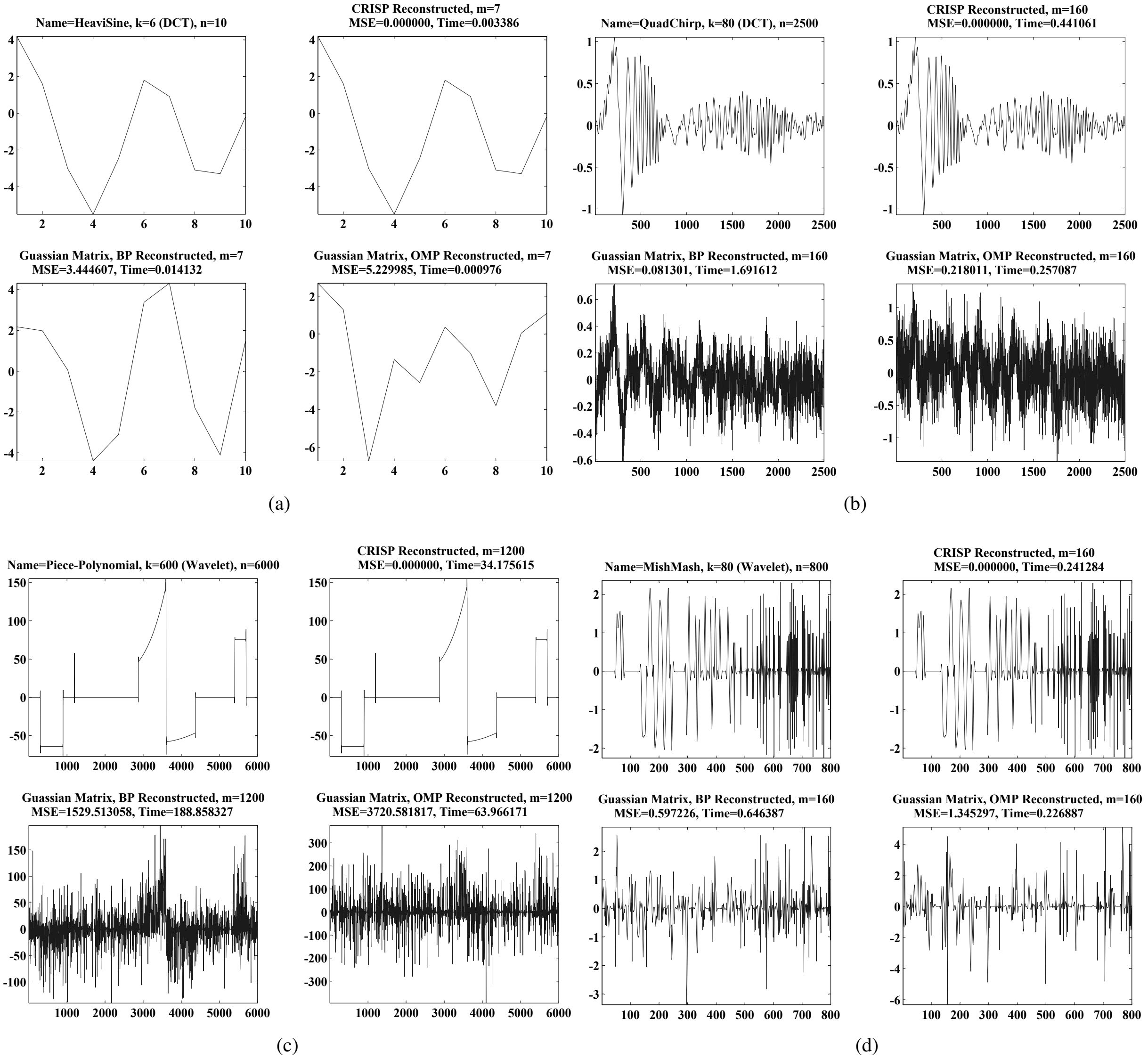
[9] “Combinatorial Designs: Construction and Analysis” by Douglas R. Stinson, Springer-Verlag, New York inc, 2004.

[10] http://www.egr.msu.edu/waves/people/nima/Crisp-icassp2010-tech.pdf

[11] A.M. Bruckstein, D.L. Donoho, and M. Elad, “From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images”, SIAM Review, Vol. 51, No. 1, Pages 34-81, February 2009.

[12] M.Luby, “LT Codes”, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[13] http://sparselab.stanford.edu



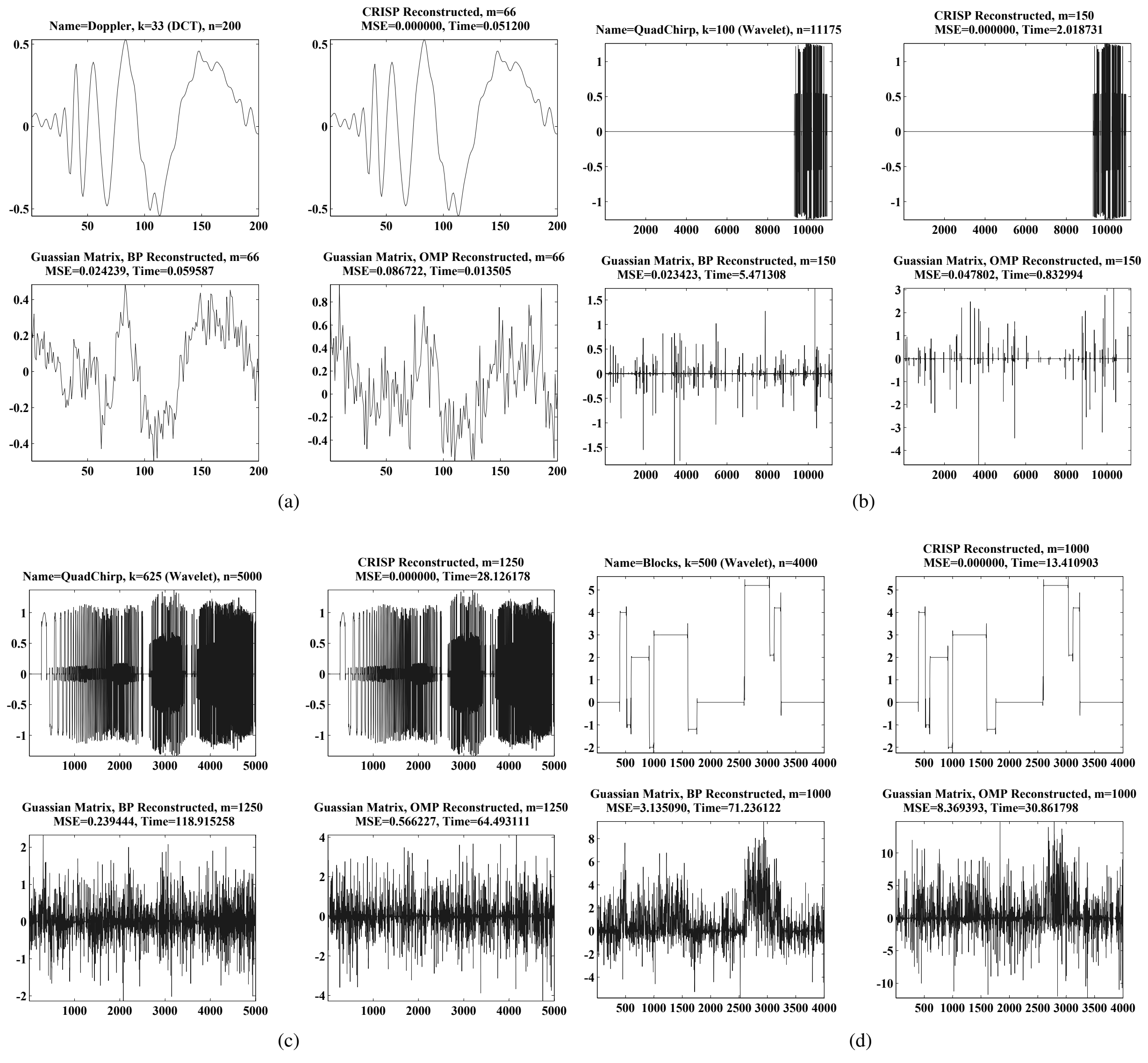
(a)

(b)

(c)

(d)

Fig. 1: m, n and k represent number of samples, signal length and signal sparsity respectively.



(a)

(b)

(c)

(d)

Fig. 2: Simulation Results (continued): m, n and k represent number of samples, signal length and signal sparsity respectively.