Reconstruction from Modulo Measurements

Viraj Shah and Chinmay Hegde

ECpE Department, Iowa State University

1 Measurement Model

We consider the problem of recovery of a signal from its modulo measurements obtained through compressive sensing. Simply put, we aim to recover $\mathbf{x}^* \in \mathbb{R}^n$ from the modulo measurements $y_i = \mod(\langle \mathbf{a_i} \cdot \mathbf{x}^* \rangle, R)$ for $i = \{1, 2, ..., m\}$, where $\mod$ is modulo operation with respect to R. Typically, $m < n$. For simplicity, we assume that the modulo function operates only within two periods, one on the either side of the origin, as shown in the Fig. 1. We construct $\mathbf{A} = [\mathbf{a_1}\ \mathbf{a_2}\ ...\ \mathbf{a_m}]^T$ with i.i.d. Gaussian entries. The primary assumption in our model is that the natural signal $\mathbf{x}^*$ is $s-$sparse in a chosen basis.

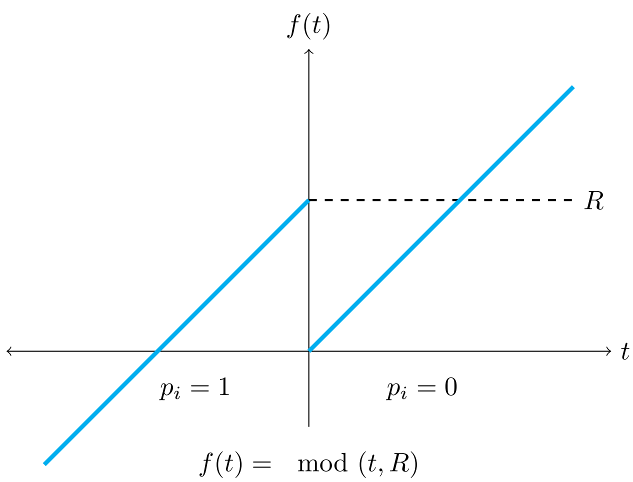

Figure 1: *Modified modulo function for the given problem*

We can write the modified equation for the modulo operation under consideration as:

$$f(t) = \mod(t, R) = t + \left(\frac{1 - \operatorname{sgn}(t)}{2}\right) R,$$

where $\operatorname{sgn}(t)$ is a signum function.

For the measurement model of the given problem, We define the corrected linear measurements as:

$$y_{c,i} = \langle \mathbf{a_i} \cdot \mathbf{x}^* \rangle.$$

We also define the bin-index p_i^* as,

$$p_i^* = \frac{1 - \operatorname{sgn}(\langle \mathbf{a_i} \cdot \mathbf{x}^* \rangle)}{2}.$$

Thus,

$$y_i = \langle \mathbf{a_i} \cdot \mathbf{x}^* \rangle + p_i^* R = y_{c,i} + p_i^* R.$$

It is evident that if we can recover $\mathbf{p}^*$ successfully, we can calculate the correct compressed measurements $\langle \mathbf{a_i} \cdot \mathbf{x}^* \rangle$ and use them to reconstruct $\mathbf{x}^*$ with any sparse recovery algorithm such as CoSaMP.

2 Reconstruction Algorithm

In this section, we describe our AltMin based approach to recover $\mathbf{x}^*$ and $\mathbf{p}^*$, given $\mathbf{y}, \mathbf{A}, s, R$. We call our algorithm MoRAM - Modulo Reconstruction using Alternative Minimization. Our approach comprises of two steps: (i) estimate initialization step, and (ii) Descent step through alternative minimization.

2.1 Initialization

Similar to many non-convex techniques, MoRAM also requires an initial estimate $\mathbf{x^0}$ that is close to the true signal $\mathbf{x}^*$. The basic idea is to calculate the significant indices (or the support of $\mathbf{x}^*$, $S = support(\mathbf{x}^*)$) using the suitable biased estimators, and then calculate the initial estimate using a first order biased estimator M only for those significant indices contained in the support of $\mathbf{x}^*$. This initialization procedure is quite simple, and requires the tuning of only one parameter, the sparsity (s).

For support estimation, we use measurements y_i to construct a biased estimator L, for which the marginal L_{jj} corresponding to the j^{th} element is given by:

$$L_{jj} = \frac{1}{m} \sum_{i=1}^{m} y_i^2 a_{ij}^2, \quad \text{for} \quad j \in 1, ..., n.$$

Note that the expectation $\mathbb{E}[\mathbf{L}]$ is given by,

$$\mathbb{E}[L_{jj}] = 4x_j^{*2} + 2R^2(1 - c_1) - 2R\|\mathbf{x}^*\|c_2 \quad \text{ where, } c_1, c_2 \in \mathbb{R}$$

indicating that a clear separation exists for values of expectation for $j \in S$ and $j \in S^c$, because x_j is zero for $j \in S^c$ and non-zero otherwise. Therefore, we can form the approximation of the support, $\widehat{S}$ by collecting the indices of the higher magnitude elements of $\mathbb{E}[L]$. With the support known, we can throw away the columns of $\mathbf{A}$ corresponding to $j \in S^c$ for the next steps. This would make the computation faster.

Next step is to obtain the initial estimate using the first order estimator $\mathbf{M}$, defined as:

$$M_{jj} = \frac{1}{m} \sum_{i=1}^{m} y_i a_{ij}, \quad \text{for} \quad j \in S.$$

To calculate $\mathbf{x^0}$, we use the fact that

$$\mathbb{E}[\mathbf{M}] = \left(1 - \sqrt{\frac{2}{\pi}} \frac{R}{2}\right) \mathbf{x}^*.$$

Given enough number of samples, the sample mean of the above estimator lie very close to the expectation value. Thus, we can calculate the initial estimate $\mathbf{x^0}$ as:

$$\mathbf{x_j^0} = \begin{cases} \frac{1}{m} \sum_{i=1, j \in S}^{m} y_i a_{ij} \quad , & \text{if } j \in S \\ 0, & j \in S^c. \end{cases} \tag{1}$$

However, it should be noted that the quality of the initial estimate is a direct function of the number of measurements (m).

Algorithm 1 MoRAM

Inputs: $\mathbf{y}$, $\mathbf{A}$, s, R
Output: $\widehat{x}$
$m, n \leftarrow \text{size}(\mathbf{A})$
$s_p \leftarrow 0.1 \cdot m$
Initialization
$\mathbf{x^0} \leftarrow \text{Oracle}(\mathbf{y}, \mathbf{A})$
Alternative Minimization
for $l = 0 : \text{N}$ **do**
    $\mathbf{p^t} \leftarrow \frac{\mathbf{1} - \text{sgn}(\langle \mathbf{A} \cdot \mathbf{x^t} \rangle)}{2}$
    $\mathbf{x^{t+1}} \leftarrow \arg\min_{[\mathbf{x}\ \mathbf{d}]^T \in \mathcal{M}_{s+s_p}} \| \begin{bmatrix} \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \end{bmatrix} - \mathbf{y} \|_2^2 = \text{CoSaMP}(\frac{1}{\sqrt{m}} \begin{bmatrix} \mathbf{A} & \mathbf{I} \end{bmatrix}, \frac{1}{\sqrt{m}} \mathbf{y}, s + s_p, [\mathbf{x^t}\ \ \mathbf{p^t}]^T)$
end for

2.2 Alternative Minimization

Using Eq. 1, we calculate the initial estimate of the signal $\mathbf{x^0}$ which is relatively close to the true vector $\mathbf{x}^*$. Starting with $\mathbf{x^0}$, we calculate the estimates $\mathbf{p}$ and $\mathbf{x}$ in alternating fashion to converge to the original signal $\mathbf{x}^*$. At each iteration of our Alternative Minimization, we use the current estimate of the signal $\mathbf{x^t}$ to get the value of the bin-index vector $\mathbf{p^t}$ as following:

$$\mathbf{p^t} = \frac{\mathbf{1} - \text{sgn}(\langle \mathbf{A} \cdot \mathbf{x^t} \rangle)}{2}. \tag{2}$$

Given $\mathbf{x^0}$ is close to $\mathbf{x}^*$, $\mathbf{p^0}$ would also be close to $\mathbf{p}^*$. Ideal way is to calculate the correct compressed measurements $\mathbf{y_c^t}$ using $\mathbf{p^t}$, and use $\mathbf{y_c}$ with CoSaMP to calculate the next estimate $\mathbf{x_{t+1}}$. Thus,

$$\begin{gathered}\mathbf{y_c^t} = \langle \mathbf{A}\mathbf{x_t} \rangle = \mathbf{y} - \mathbf{p^t} R, \\ \mathbf{x^{t+1}} = \underset{\mathbf{x} \in \mathcal{M}_s}{\arg\min} \|\mathbf{A}\mathbf{x} - \mathbf{y_c^t}\|_2^2, \\ \implies \mathbf{x^{t+1}} = \text{CoSaMP}(\frac{1}{\sqrt{m}}\mathbf{A}, \frac{1}{\sqrt{m}}\mathbf{y_c^t}, s, \mathbf{x_t}). \end{gathered} \tag{3}$$

However, it should be noted that even the small error $\mathbf{d} = \mathbf{p^t} - \mathbf{p}^*$ would reflect heavily in the calculation of $\mathbf{y_c^t}$, as each incorrect bin-index would add a noise of the magnitude R in $\mathbf{y_c^t}$. Experiments suggest that the CoSaMP is not robust enough to cope up with such large errors in $\mathbf{y_c^t}$. To tackle this issue, we augmented the sparse recovery problem using the fact that the nature of error $\mathbf{d_p^t}$ is sparse; and each erroneous element of $\mathbf{p}$ adds a noise of the magnitude R in $\mathbf{y_c^t}$. We take the sparsity of $\mathbf{d^t}$ to be $s_p = 0.1 \times m$, suggesting that at most the 10% of the total elements are classified with wrong bin-indices.

The augmented optimization problem becomes,

$$\begin{gathered}\mathbf{x^{t+1}} = \underset{[\mathbf{x}\ \mathbf{d}]^T \in \mathcal{M}_{s+s_p}}{\arg\min} \| \begin{bmatrix} \mathbf{A} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{d} \end{bmatrix} - \mathbf{y} \|_2^2, \\ \implies \mathbf{x^{t+1}} = \text{CoSaMP}(\frac{1}{\sqrt{m}} \begin{bmatrix} \mathbf{A} & \mathbf{I} \end{bmatrix}, \frac{1}{\sqrt{m}} \mathbf{y}, s + s_p, [\mathbf{x^t}\ \ \mathbf{p^t}]^T). \end{gathered} \tag{4}$$

We call the step in Eq. 4 a Robust CoSaMP.

We repeat the steps of bin index calculation (as in Eq. 2) and sparse recovery (as in Eq. 3 or Eq. 4) alternatively for N iterations. While the sparse recovery with robust CoSaMP (Eq. 4) improves the reconstruction performance for large values of R by making the sparse recovery step less susceptible to the errors, CoSaMP can also used in its original form (as in Eq. 3) for lower values of R.

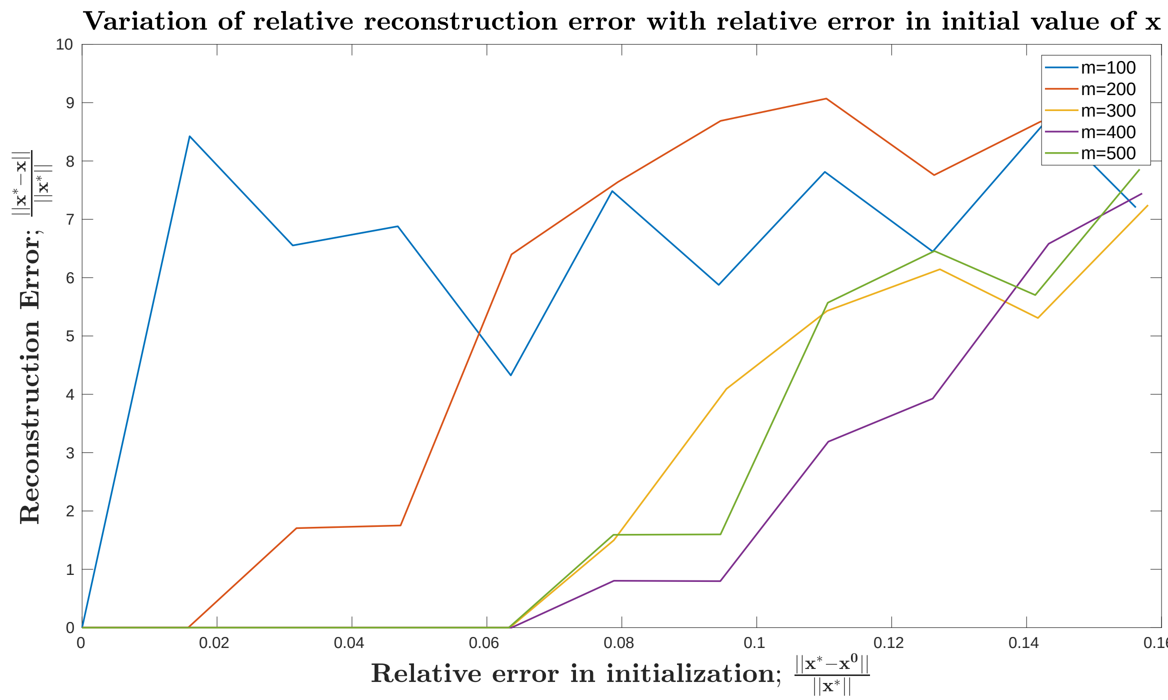

Figure 2

Thus, we can have two variants of the MoRAM algorithm: (i)MoRAM with CoSaMP, and (ii) MoRAM with robust CoSaMP.

2.3 Experiments

2.3.1 Analyzing sensitivity towards the initial estimate

For this experiment, we use a synthetic signal generated randomly with $n = 1000$ and $s = 20$. Our aim is to analyze the sensitivity of our algorithm towards the initial estimate $\mathbf{x^0}$. For that, we compute the initial estimate $\mathbf{x^0}$ by adding a Gaussian noise to the original signal. In Fig. 2, we plot the variation of the relative reconstruction error ($\frac{\|\mathbf{x^*}-\mathbf{x^N}\|}{\|\mathbf{x^*}\|}$) with the relative error in initial estimate ($\frac{\|\mathbf{x^*}-\mathbf{x^0}\|}{\|\mathbf{x^*}\|}$). We plot the similar curves for different values of number of measurements m.

2.3.2 Performance of our algorithm for signal reconstruction

We perform experiments on a synthetic signal generated randomly with $n = 1000$ and $s = 5$. We compute the initial estimate $\mathbf{x^0}$ using first order estimator method described in 2.1. we plot the variation of the relative reconstruction error ($\frac{\|\mathbf{x^*}-\mathbf{x^N}\|}{\|\mathbf{x^*}\|}$) with number of measurements m for both the variants of sparse recovery algorithm as described in 2.2.

It is important to note that unlike the absolute value function, the modulo function described in Fig. 1 is not scale-invariant. The modulo function works over the quantities $y_{c,i} = \langle \mathbf{a_i} \cdot \mathbf{x^*} \rangle, i = 1, .., m$; and it is defined over the parameter R; thus depending on the magnitudes of $y_{c,i}$ and R relative to each other, the behavior of the measurement model and the reconstruction algorithm would be altered. For instance, if the value of R is too small compared to the range of the $y_{c,i}$, the modulo operation would hardly have any effect on the measurements, leaving $\mathbf{y_c} \approx \mathbf{y}$. To

Fixed: $R = 1, n = 1000, s = 5$			
CoSaMP		**robust CoSaMP**	
$\|\mathbf{x}^*\| = 1$	$\|\mathbf{x}^*\| = 0.5$	$\|\mathbf{x}^*\| = 1$	$\|\mathbf{x}^*\| = 0.5$
Figure 3	Figure 4	Figure 5	Figure 6

Table 1: The Results

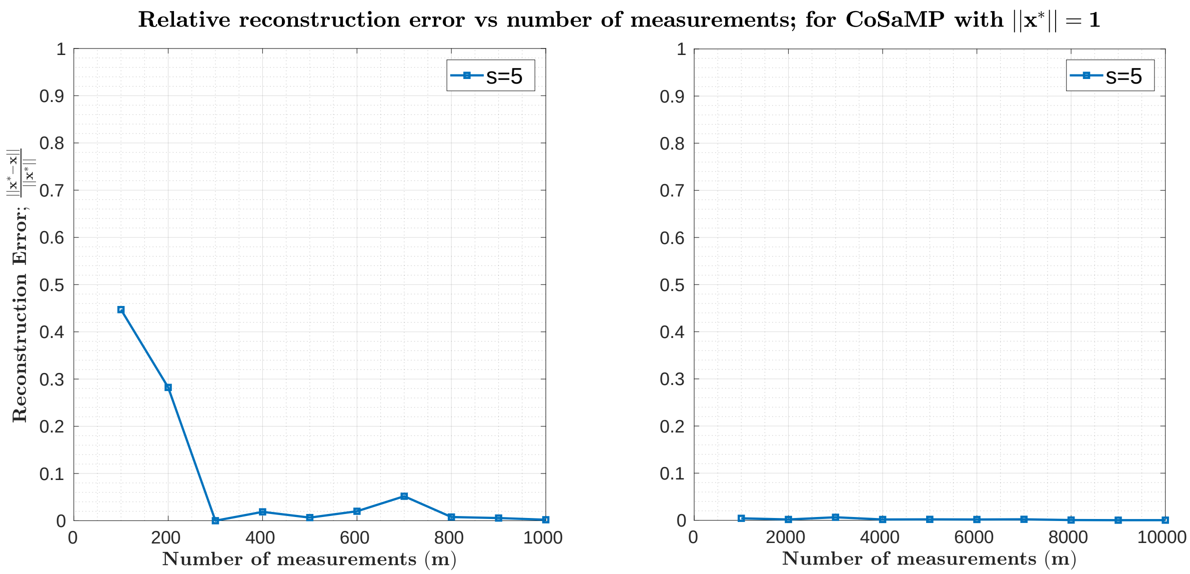

Figure 3

analyze such variations, we fix the $R = 1$ in our experiments, while varying the signal strength to vary the magnitudes of $y_{c,i}$. We measure the signal strength by the norm of the original signal ($\|\mathbf{x}^*\| = 1$).

Another important factor affecting the reconstruction is the quality of the initial estimate ($\mathbf{x^0}$) obtained through first order estimation. As described in 2.1, the quality of the initial estimate is a direct function of number of measurements (m). As we set m higher, the initial estimate $\mathbf{x^0}$ would move closer to the original signal $\mathbf{x}^*$. For our experiments, we consider two ranges of m: $m \in [100, 1000]$ and $m \in [1000, 10000]$.

In the Table 1, we provide experimental results for each of the combination above.

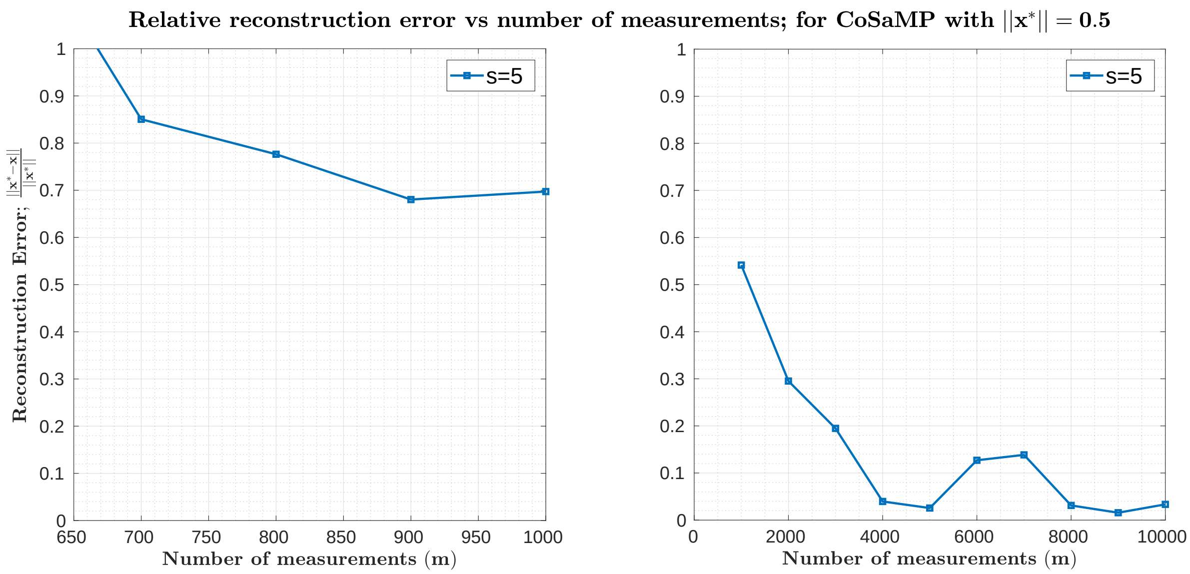

Figure 4

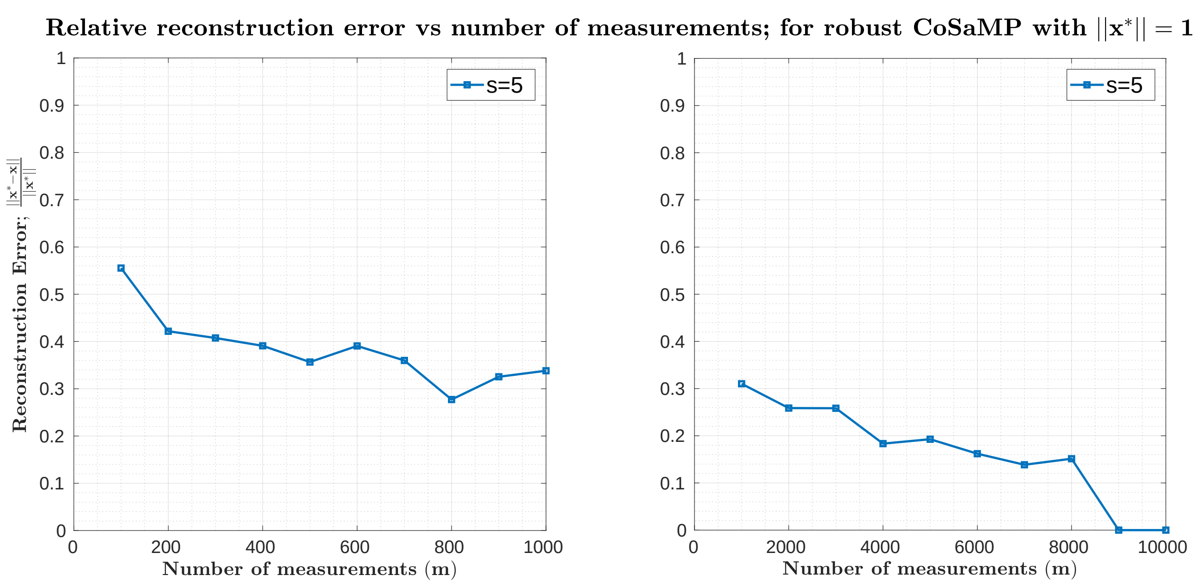

Figure 5

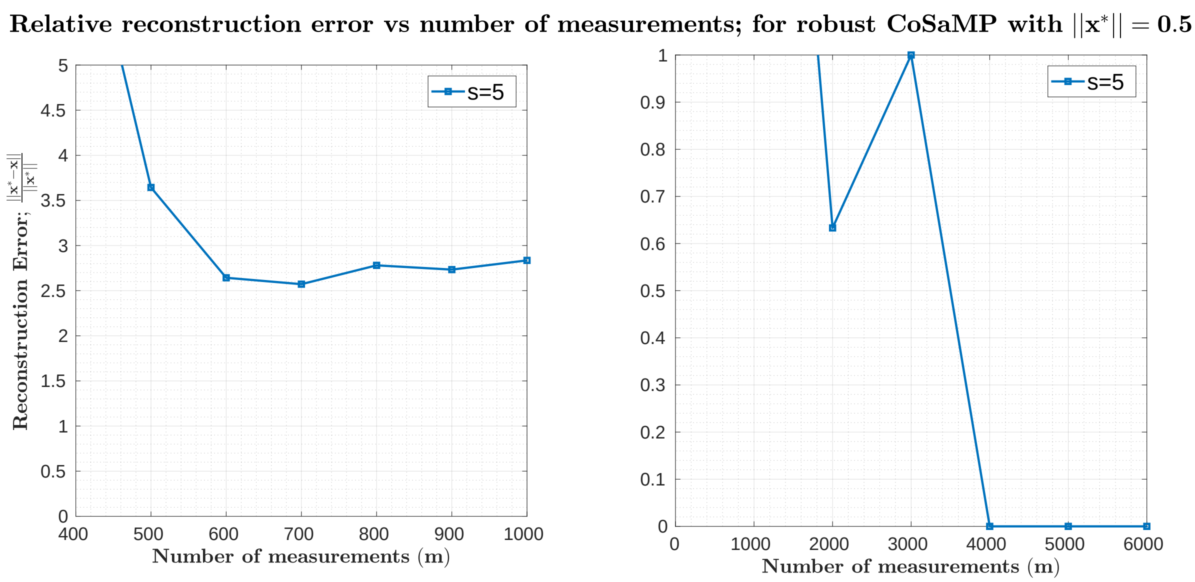
Relative reconstruction error vs number of measurements; for robust CoSaMP with $||\mathbf{x}^*|| = 0.5$
Reconstruction Error; $\frac{||x^* - x||}{||x^*||}$
Number of measurements (m)
s=5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
400
500
600
700
800
900
1000
Number of measurements (m)
s=5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1000
2000
3000
4000
5000
6000

Figure 6