EXPERIMENT WD   Demonstrator: Zhao, Yang

Recover Sparse Signals from Under-Sampled Observations

$$y = Ax$$

y: observation vector

x: unknown data vector

Assume that $\|A_{:i}\|_2 := \left(\sum_{j=1}^{m} A_{ji}^2\right)^{\frac{1}{2}} = 1$

$A \in \mathbb{R}^{m\times n}$

Ex.1

$$x = A^{-1}y$$

A^{-1} is not exist? $\rightarrow$ A^{+}, persudo invert A

If column of A is linear independent, $A^T \cdot A$ is invertable

$$A^{+} = (A^TA)^{-1} \cdot A^T$$

If row of A is linear independent, $A \cdot A^T$ is invertable

$$A^{+} = A \cdot (A \cdot A^T)^{-1}$$

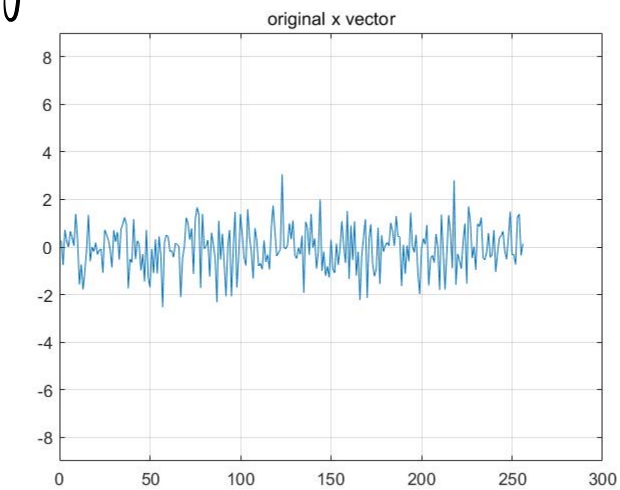
$$\boxed{y = A \cdot x \rightarrow \underbrace{A^{+}y}_{\text{pinv}(A)\cdot y} = A^{+}A \cdot x = x}$$

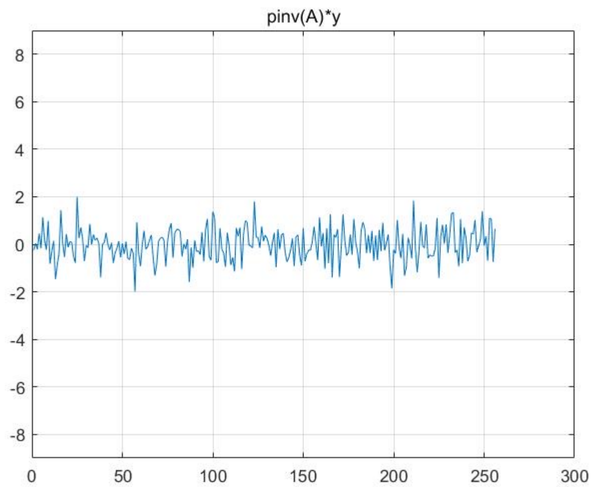
$x = A\backslash y$, in matlab, the operation is to find the solution of $y = Ax$.

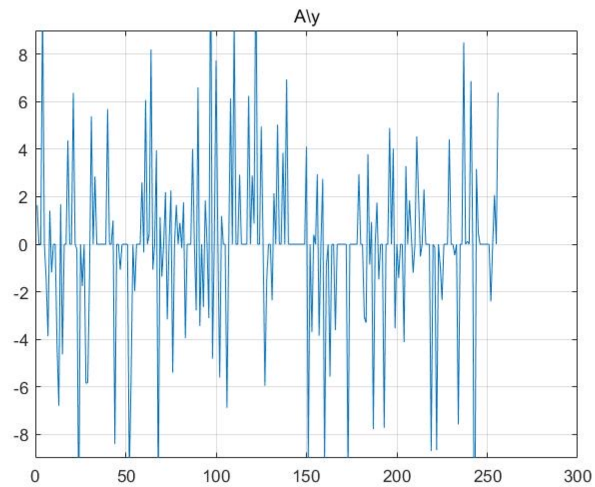
Compare the result $\hat{x}_1 = \text{pinv}(A) \cdot y$

and $\hat{x}_2 = A\backslash y$

The result of original x, $\hat{x}_1$ and $\hat{x}_2$ is shown below







The two results are quiet different with each other and the original x vector.

EX.2

By observing the three algorithms, all of them need the calculation of support$(H_s(\cdot))$.
Hence, firstly develop the function of support$(H_s(\cdot))$

compare $|x|$ can be achieved by comparing x^2

In MATLAB, the function $\boxed{\text{maxk}(x, k)}$ can find the maximum k values in vector x and the indexs of the k values.

```
function supportH(x, S)
[v, result] = maxk(x.^2, S)
```

Using the seed function to control the r.v.

The Orthogonal Matching Pursuit (OMP) Algorithm.

$x = 0, \quad S = \phi, \quad y_r = y$

$S = S \cup \text{supp}(H_1(A^T y_r))$

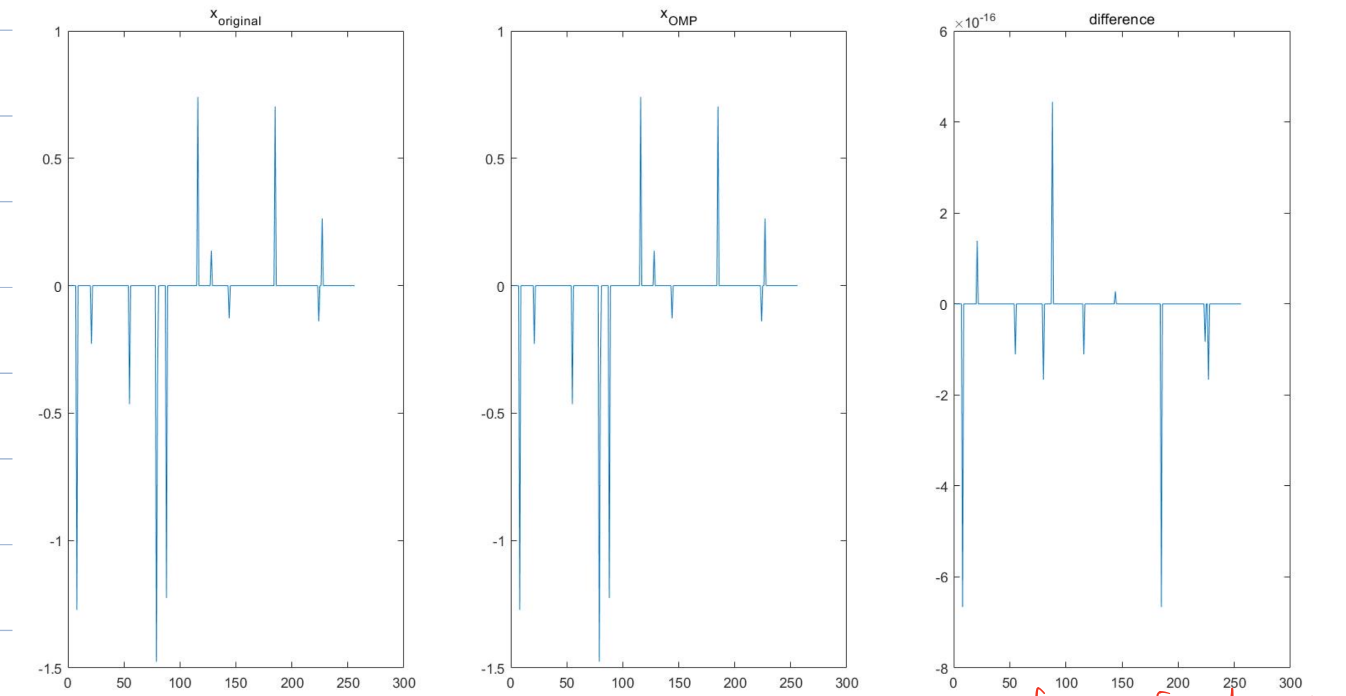
$\hat{x}_S = A_S^{\dagger} y, \quad \hat{x}_{S^c} = 0$

$y_r = y - A\hat{x}$

$err = \|y_r\|$

End criteria:

$\boxed{err \geq err\_previous}$



This figure illustrates the comparision of original x and the result of OMP, as well as the difference.

It's obvious that the difference is very small.

Actually, the error $\frac{\|x_{omp} - x\|}{\|x\|} = 4.163 \times 10^{-16}$

The Subspace Pursuit Algorithm

$S = supp(H_s(A^T y))$

$y_r = resid(y, A_S)$

$err = \|y_r\|$

End criteria:

$\boxed{err \geq err\_previous.}$

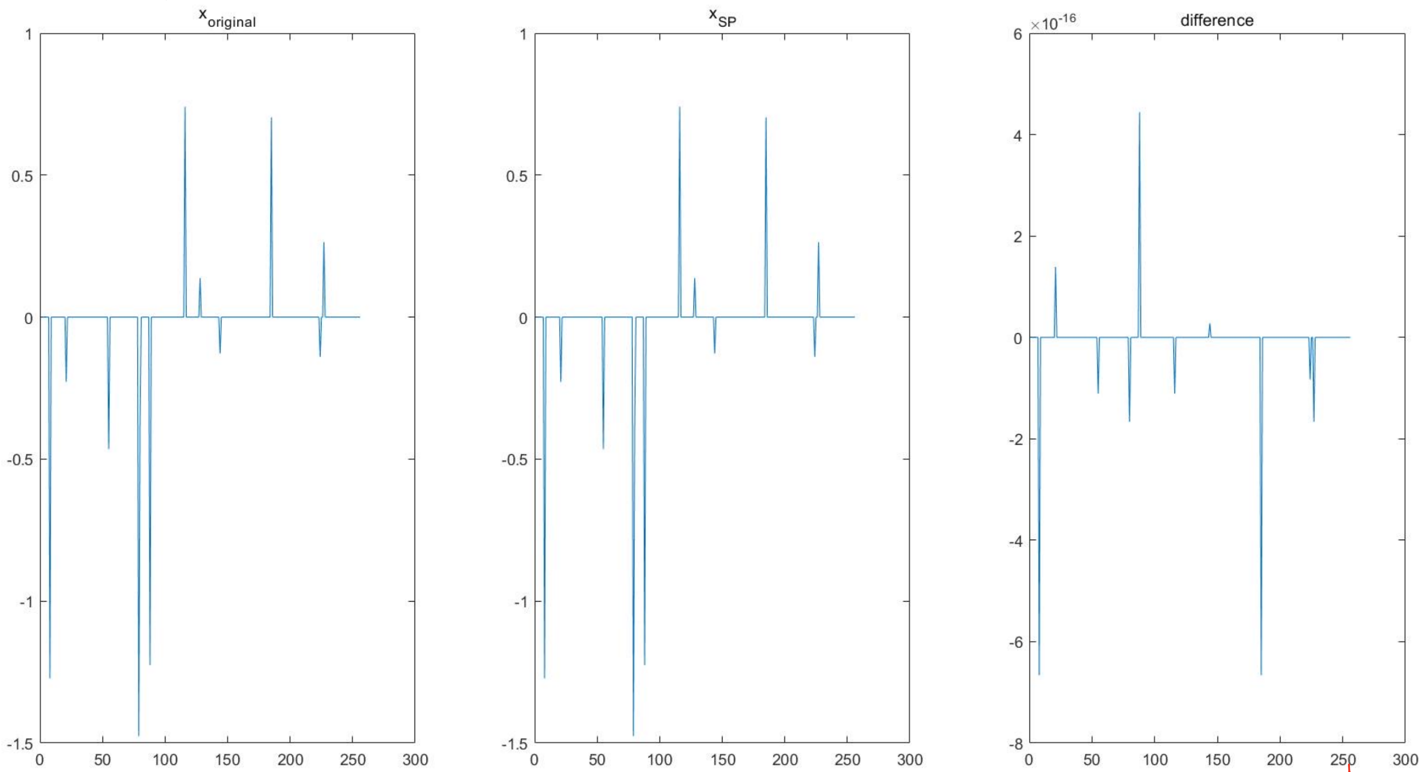
$\tilde{S} = S \cup supp(H_s(A^T y_r))$

$b_{\tilde{S}} = A^{+}_{\tilde{S}} y, \quad b_{\tilde{S}^c} = 0$

$S = supp(H_s(b))$

$\hat{x} = A^{+}_{S} y, \quad \hat{x}_{S^c} = 0$

$y_r = y - A\hat{x}$



The figure shows the comparison between original x and the $\hat{x}$ obtained by SP method.

It is easy to notify that the difference is also very small. the error $\frac{\|x_{lsp} - x\|}{\|x\|} = 4.163 \times 10^{-16}$, which has the same performance when $S=12$

The Iterative HardthreshOlding Algorithm

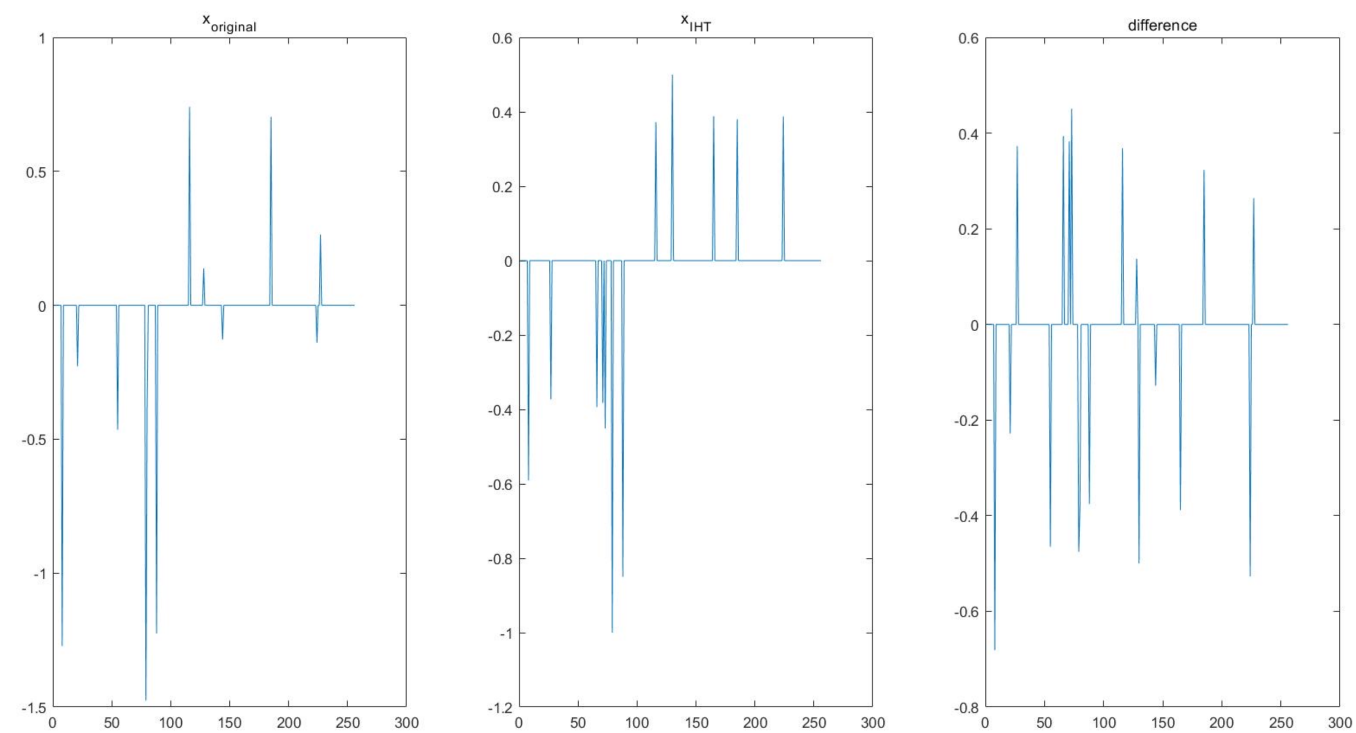
$\hat{x} = 0$

$\hat{x} = H_s(\hat{x} + A^T(y - A\hat{x}))$

err = $\|y - r\|$

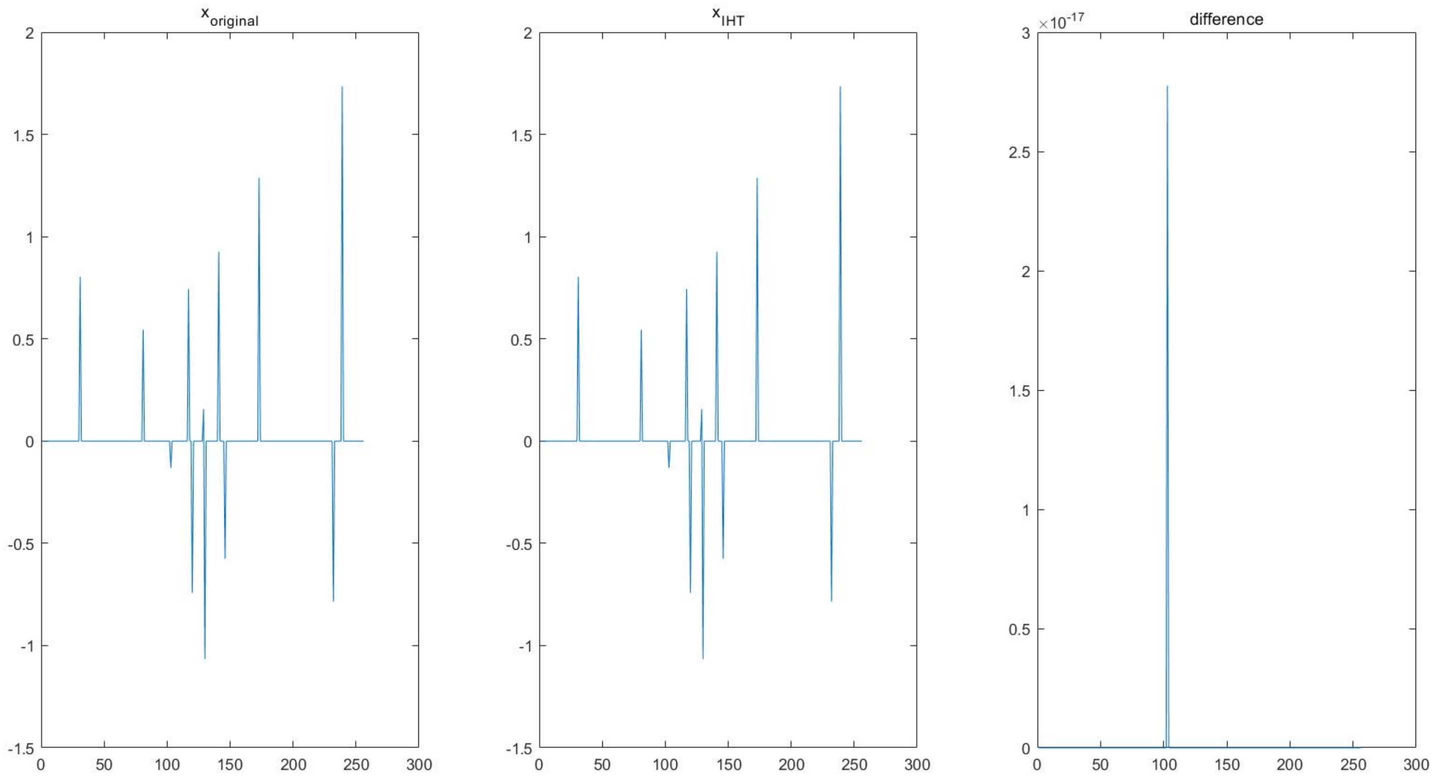
End criteria:

$\boxed{err \geq err\_previous.}$



This figure shows that the performance of IHT is not good. And the error is 0.6707

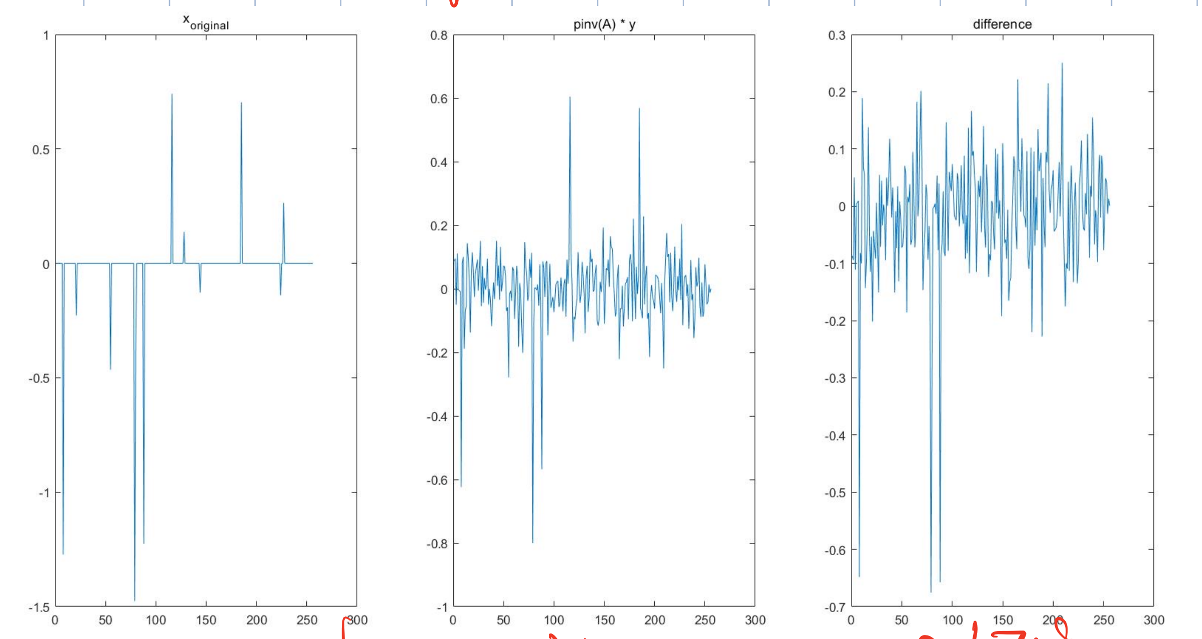
I also try the IHT Algorithm for other random variables by chaning the random seed



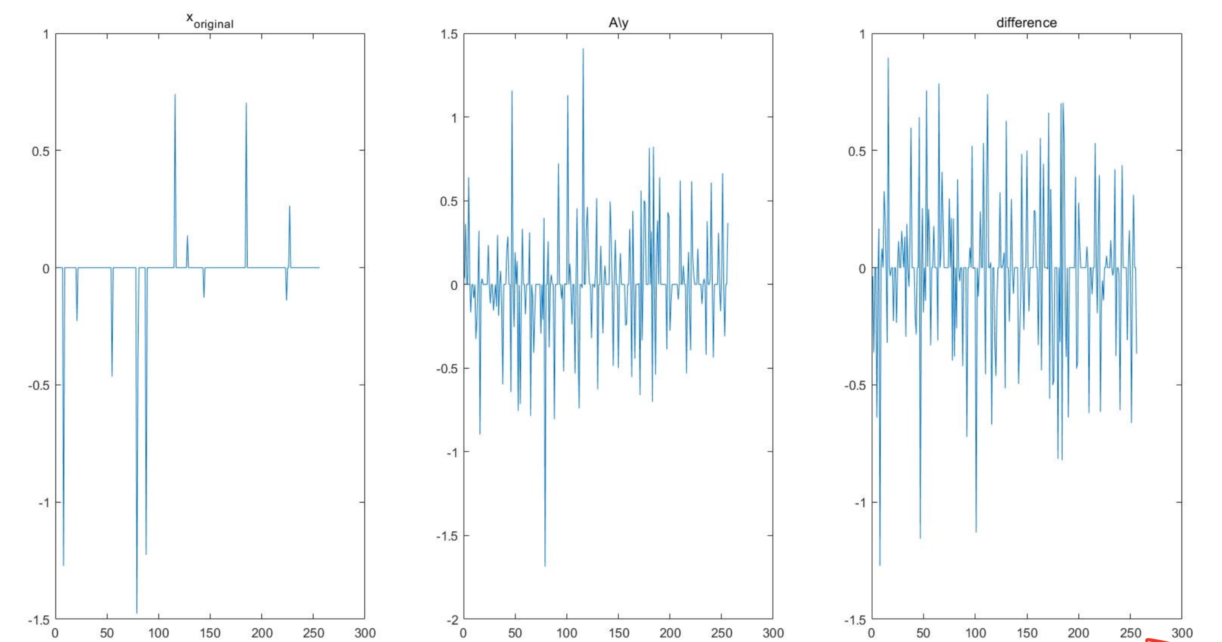
In some cases, the IHT is capable of recovering very good result. The error in this figure is only 8.08×10^{-18}

pinv(A)·y & A\y

Both are ~~not~~ not good methods!



the error of pinv(A)·y is 0.6719



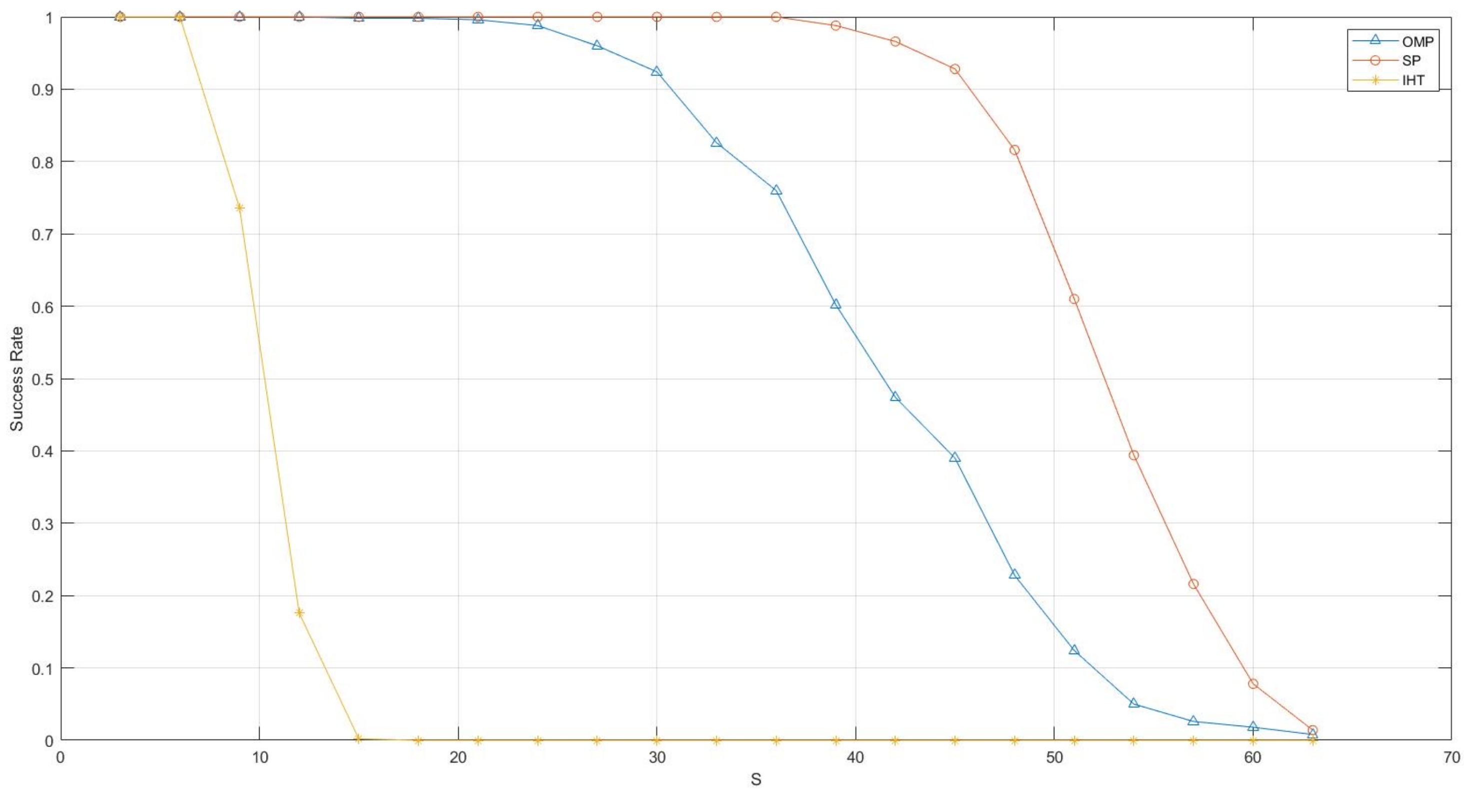
the error of A\y is 1.8543. Worst!

Ex.3

Comparision of Three Algorithm

$S = 3:3:63$, total 21 numbers

run 100 independent tests for each S



In this figure, with the increasement of S, the performance of IHT is dramaticlly decreased. And the success rate decreased to zero when $S = 15$. It is obviuy that SP has the best performance with the increasement of S, while the success rates of both SP and OMP drop to zero at $S = 63$.