Sparse Representation using Stepwise Tikhonov Regularization with Offline Computations

Ramón A. Delgado and Richard H. Middleton, *Fellow, IEEE,*

***Abstract*—This letter describes a novel algorithm for sparse reconstruction. The method uses offline computations to reduce the computational burden of online execution. The approach relies on the recently proposed Stepwise Tikhonov Regularization method to implement forward selection procedures such as OLS, OMP and STIR. Numerical simulations show the efficacy of the proposed approach which is competitive against state-of-the-art implementation of OLS and OMP.**

I. Introduction

Sparse representations have been extensively studied [1]–[8], and used in the last decades [9]–[13]. The main concept is based on optimization problems dealing with the ℓ_0 pseudo-norm, $\|\mathbf{x}\|_0$, that counts the number of non-zero elements in a vector $\mathbf{x}$. This letter focuses on the following least squares problem subject to a ℓ_0-norm constraint.

$$\mathcal{P}_0: \quad \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \ \|\mathbf{y}-\mathbf{A}\mathbf{x}\|^2 \ \text{ subject to } \ \|\mathbf{x}\|_0 \leq \gamma$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{x} \in \mathbb{R}^n$. Problem $\mathcal{P}_0$ is challenging due to its combinatorial nature [14], and in most applications the computation of global solutions for $\mathcal{P}_0$ is considered impractical.

A popular heuristic for $\mathcal{P}_0$ is the ℓ_1-norm. This heuristic induces sparsity by replacing the ℓ_0-norm term, with a ℓ_1-norm term, see e.g. [15]–[17]. Greedy algorithms are typically used to find a sub-optimal solution of $\mathcal{P}_0$. These algorithms are initialized with a trivial solution, such as $\hat{\mathbf{x}} = 0$, and then $\|\hat{\mathbf{x}}\|_0$ is gradually increased until a satisfactory solution is found. See [18]–[20] for a survey on greedy algorithms.

For problem $\mathcal{P}_0$, well known greedy algorithms include Orthogonal Least Squares (OLS) [21] and Orthogonal Matching Pursuit (OMP) [22]. OLS at each step, selects the element that provides the largest reduction of the least squares cost, while OMP selects the element having the largest correlation with the current residual. A state-of-the-art implementation of OLS, called accelerated OLS (AOLS) is described in [23]. Several state-of-the-art implementations of OMP can be found in [24].

Recently, a greedy approach to solve $\mathcal{P}_0$ has been introduced in [25]. The approach exploits connections between Tikhonov regularization and sparse representation to propose a greedy algorithm called Stepwise Tikhonov Regularization (STIR). The STIR criterion selects the element that will have the largest absolute value at the next iteration.

R.A. Delgado and R.H. Middleton are with the School of Electrical Engineering and Computing, University of Newcastle, Australia.
E-mail: {Ramon.Delgado, Richard.Middleton}@newcastle.edu.au.
Manuscript received April 19, 2005; revised August 26, 2015.

Sufficient conditions to recover sparse and compressible signals using OLS and OMP are given in [26]. Analogous sufficient conditions for STIR have been reported in [25].

This letter proposes implementation improvements for the STIR algorithm:

1) An efficient implementation of the STIR algorithm.
2) Methods to reuse calculations that do not depend on the vector $\mathbf{y}$. This is done across several instances of $\mathcal{P}_0$.
3) Implementations of the OLS and OMP procedures based on the STIR framework that, under certain circumstances, outperforms state-of-the-art implementations of OLS and OMP.

The second item is noteworthy because significant computational savings can be obtained when the matrix $\mathbf{A}$ does not change across several instances of the problem $\mathcal{P}_0$. Most sparse representation algorithms make little use of this factor. Often, computations are only reused for the Gramian $\mathbf{A}^\top\mathbf{A}$ and the normalization of the columns of $\mathbf{A}$, see e.g. [24].

In this letter, it assumed that the same matrix $\mathbf{A}$ will be used over several instances of problem $\mathcal{P}_0$. Thus, some calculations involving matrix $\mathbf{A}$ can be reused. Such calculations are referred to being done *offline*. On the other hand, it is assumed that a new vector $\mathbf{y}$ is used for each instance of problem $\mathcal{P}_0$. Thus, calculations depending on the vector $\mathbf{y}$ cannot be reused and need to be computed *online*. Online computations include the algorithm initialization, the computations during iterations and finally, if needed, the computations after the iterations.

The letter is organized as follows: Section II revisits Stepwise Tikhonov Regularization. In Section III the proposed implementation of STIR is described. Section IV numerically compares the proposed approach with other methods. Finally, conclusions are drawn in Section V.

Matrices are denoted by bold uppercase letters and vectors are denoted with bold lower-case letters. $A_{i,j}$ denotes the element in i-th row and the j-th column of a matrix $\mathbf{A}$. Similarly, v_i denotes the i-th element of the vector $\mathbf{v}$. $\mathbf{A}_{j-col}$ denotes the j-th column of $\mathbf{A}$. The vector $\mathbf{e}_\ell$ denotes the ℓ-th column of the identity matrix.

II. Stepwise Tikhonov Regularization

This section reviews the Stepwise Tikhonov Regularization (STIR) method introduced in [25].

The STIR method, at each iteration, solves a least squares problem with a Tikhonov regularization to induce zeros in the solution. Let $\mathcal{S}^{(k)}$ be the indices of the elements in the model at the beginning of iteration k, and a corresponding diagonal matrix $\mathbf{W}^{(k)}$. The diagonal elements of $\mathbf{W}^{(k)}$ are defined such

Peer-reviewed author's copy of:
R.A. Delgado and R.H. Middleton. **Sparse Representation using Stepwise Tikhonov Regularization with Offline Computations**. In *IEEE Signal Processing Letters*, 2019.
Available at https://doi.org/10.1109/LSP.2019.2907370

that $W_{i,i}^{(k)} = 1$ if $i \notin \mathcal{S}^{(k)}$ and $W_{j,j}^{(k)} = 0$ if $j \in \mathcal{S}^{(k)}$. Next, consider the following least squares problem with a Tikhonov regularization

$$\mathcal{P}_{tikho}^{k}: \quad \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \ \|\mathbf{y}-\mathbf{A}\mathbf{x}\|^2 + \rho \mathbf{x}^\top \mathbf{W}^{(k)} \mathbf{x}$$

where $\rho > 0$ is a regularization constant and $\mathbf{W}^{(k)}$ is the regularization matrix.

The following Lemma describes the solution of $\mathcal{P}_{tikho}^{k}$.

Lemma 1. *Given $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{y} \in \mathbb{R}^m$, $\rho > 0$ and a positive semi-definite matrix $\mathbf{W}^{(k)}$. Suppose that $\left(\mathbf{A}^\top\mathbf{A} + \rho\mathbf{W}^{(k)}\right)$ is positive definite. Then, the optimal solution of $\mathcal{P}_{tikho}^{k}$ is given by*

$$\hat{\mathbf{x}}^{(k)} = \frac{1}{\rho}\left(\frac{1}{\rho}\mathbf{A}^\top\mathbf{A} + \mathbf{W}^{(k)}\right)^{-1}\mathbf{A}^\top\mathbf{y} \tag{1}$$

Note that $\mathcal{P}_{tikho}^{k}$ has a closed-form solution (1). The STIR method exploits closed-form solutions when $\mathbf{W}^{(k)}$ is updated.

Let $\mathbf{W}^{(k)}$ be a diagonal matrix describing the current elements in the model, $\mathcal{S}^{(k)}$, as used in Tikhonov regularization. Consider that the element with index ℓ is added to the model, i.e. $\mathcal{S}^{(k+1)} = \mathcal{S}^{(k)} \cup \{\ell\}$ and $\mathbf{W}^{(k+1)}$ is given by

$$\mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \mathbf{e}_\ell\mathbf{e}_\ell^\top \tag{2}$$

Define the following auxiliary matrix

$$\mathbf{Z}^{(k)} \coloneqq \left(\frac{1}{\rho}\mathbf{A}^\top\mathbf{A} + \mathbf{W}^{(k)}\right)^{-1} \tag{3}$$

and the cost of $\mathcal{P}_{tikho}^{k}$ is represented as

$$g^k(\mathbf{x}) \coloneqq \|\mathbf{y}-\mathbf{A}\mathbf{x}\|^2 + \rho\, \mathbf{x}^\top\mathbf{W}^{(k)}\mathbf{x} \tag{4}$$

The following Theorem describes the relationship between the optimal solutions of $\mathcal{P}_{tikho}^{k}$ and $\mathcal{P}_{tikho}^{k+1}$.

Theorem 1 ([25]). *Consider $\mathbf{W}^{(k+1)}$ as in* (2) *and suppose that the assumptions in Lemma 1 hold for $\mathbf{W}^{(k)}$ and $\mathbf{W}^{(k+1)}$. Thus, the following relationships hold*

$$\hat{\mathbf{x}}^{(k+1)} = \hat{\mathbf{x}}^{(k)} + \frac{\hat{x}_\ell^{(k)}}{1-Z_{\ell,\ell}^{(k)}}\mathbf{Z}_{\ell-col}^{(k)} \tag{5}$$

$$\mathbf{Z}^{(k+1)} = \mathbf{Z}^{(k)} + \frac{1}{1-Z_{\ell,\ell}^{(k)}}\mathbf{Z}_{\ell-col}^{(k)}(\mathbf{Z}_{\ell-col}^{(k)})^\top \tag{6}$$

and

$$g^k(\hat{\mathbf{x}}^{(k)}) - g^{k+1}(\hat{\mathbf{x}}^{(k+1)}) = \rho\frac{(\hat{x}_\ell^{(k)})^2}{1-Z_{\ell,\ell}^{(k)}} \tag{7}$$

The main ingredients in the STIR method are described in Theorem 1. The closed-form expressions in Theorem 1 allow the STIR method to get the optimal solution of $\mathcal{P}_{tikho}^{k+1}$ using a computationally inexpensive update from the solution of $\mathcal{P}_{tikho}^{k}$. Additionally, from (5) it can be deduced that

$$\hat{x}_\ell^{(k+1)} = \frac{\hat{x}_\ell^{(k)}}{1-Z_{\ell,\ell}^{(k)}} \tag{8}$$

Equation (8) shows the value of $\hat{x}_\ell^{(k+1)}$ when ℓ is included in the model at the k-th iteration. This is used by the STIR method to implement a forward stepwise selection procedure. This procedure selects the candidate element that will have the largest absolute value at the next iteration, by using

$$v_i^{(k+1,i)} = \frac{\hat{x}_i^{(k)}}{1-Z_{i,i}^{(k)}} \tag{9}$$

Thus, the STIR method proposed in [25] uses the following selection criterion

$$\ell = \arg\max_{i\notin\mathcal{S}^{(k)}} \left|v_i^{(k+1,i)}\right| \tag{10}$$

III. Proposed approach

This section describes a fast implementation of the STIR method [25] and how the variables in the STIR method can be used to implement the forward stepwise selection criteria used by OLS and OMP.

A. Efficient STIR algorithm

The efficient STIR implementation described below is based on the fact that the updates required by the STIR method (given by (5), (7) and (9)) only depend on the diagonal elements of $\mathbf{Z}^{(k)}$, the current estimate $\hat{\mathbf{x}}^{(k)}$ and the column of $\mathbf{Z}^{(k)}$ corresponding to the selected element, i.e. $\mathbf{Z}_{\ell-col}^{(k)}$.

Denote as i_j the index of the element that has been included in the model at the j-th iteration, then the ℓ-th column of $\mathbf{Z}^{(k+1)}$ for $k > 1$ can be computed as follows

$$\mathbf{Z}_{\ell-col}^{(k)} = \mathbf{Z}_{\ell-col}^{(0)} + \sum_{j=1}^{k-1}\mathbf{Z}_{i_j-col}^{(j)}\frac{Z_{i_j,\ell}^{(j)}}{1-Z_{i_j,i_j}^{(j)}} \tag{11}$$

and $\mathbf{Z}_{\ell-col}^{(1)} = \mathbf{Z}_{\ell-col}^{(0)}$. Equation (11) depends on $\mathbf{Z}_{i_j-col}^{(j)}$ for $j = 1,\ldots,k-1$ and $\mathbf{Z}^{(0)}$. Additionally, to compute $v_i^{(k+1,i)}$ it is necessary to keep track of the diagonal values of $\mathbf{Z}^{(k)}$. These diagonal elements are updated as follows

$$Z_{i,i}^{(k+1)} = Z_{i,i}^{(k)} + \frac{\left(Z_{i,\ell}^{(k)}\right)^2}{1-Z_{\ell,\ell}^{(k)}} \tag{12}$$

for $i = 1,\ldots,n$ and where $\ell = i_k$.

This letter proposes to use (11) and (12) to compute $\mathbf{Z}_{\ell-col}^{(k)}$ and the diagonal elements of $\mathbf{Z}^{(k)}$ for $k > 0$, without computing irrelevant entries of $\mathbf{Z}^{(k)}$ as used in (6). Notice that $\mathbf{Z}^{(0)}$ needs to be computed as a full matrix to initialize the updates in (11) and (12). The initial solution, $\hat{\mathbf{x}}^{(0)}$ also needs to be computed as follows

$$\hat{\mathbf{x}}^{(0)} = \mathbf{H}\mathbf{y} \tag{13}$$

where $\mathbf{H} = \frac{1}{\rho}\mathbf{Z}^{(0)}\mathbf{A}^\top$. For the case when matrix $\mathbf{A}$ does not change over multiple instances of $\mathcal{P}_0$, matrices $\mathbf{Z}^{(0)}$ and $\mathbf{H}$ can be computed offline.

Algorithm 1 (see Table) describes the proposed algorithm. In some applications, step 2 in algorithm 1 can be done offline, significantly reducing the online execution time. In Algorithm 1, expressions for $\mathbf{H}$ and $\mathbf{Z}^{(0)}$ have been derived using Woodbury matrix identity, see e.g. [27].

Algorithm 1 STIR algorithm with offline computations.

1) Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, ρ, $\mathbf{y} \in \mathbb{R}^m$ and γ.
2) Offline computations:
   - Set $\mathbf{H} = \mathbf{A}^\top(\rho\mathbf{I} + \mathbf{A}\mathbf{A}^\top)^{-1}$
   - Set $\mathbf{Z}^{(0)} = \mathbf{I} - \mathbf{H}\mathbf{A}$
   - Set $\mathbf{d}^{(0)} = \text{diag}\{\mathbf{Z}^{(0)}\}$
3) Initialize:
   - Set $\hat{\mathbf{x}}^{(0)} = \mathbf{H}\mathbf{y}$
   - Set $\mathcal{S}^{(0)}$ as an empty set.
4) For $k = 0 \dots \gamma - 1$ compute:
   a) $\ell = \arg\max_{i\notin\mathcal{S}^{(k)}} \frac{|\hat{x}_i^{(k)}|}{(1-d_i^{(k)})}$
   b) $\mathcal{S}^{(k+1)} = \mathcal{S}^{(k)} \cup \{\ell\}$
   c) if $k > 0$ then
   $$\mathbf{z}^{(k)} = \mathbf{Z}_{\ell-col}^{(0)} + \sum_{j=0}^{k-1} q^{(j)} z_\ell^{(j)} \mathbf{z}^{(j)}$$
   else
   $$\mathbf{z}^{(k)} = \mathbf{Z}_{\ell-col}^{(0)}$$
   end if
   d) $q^{(k)} = 1/(1 - d_\ell^{(k)})$
   e) $\hat{\mathbf{x}}^{(k+1)} = \hat{\mathbf{x}}^{(k)} + q^{(k)}\hat{x}_\ell^{(k)}\mathbf{z}^{(k)}$
   f) for $i = 1, \dots, n$
   $$d_i^{(k+1)} = d_i^{(k)} + q^{(k)}\left(z_i^{(k)}\right)^2$$
5) Return $\hat{\mathbf{x}}^{(\gamma)}$

B. Complexity analysis

The analysis of the computational complexity of the proposed STIR algorithm can be decomposed into its offline and online components.

The offline part is dominated by the computation of $\mathbf{H}$ and $\mathbf{Z}^{(0)}$ that require $\mathcal{O}(m^2n)$ for the computation of $\mathbf{A}\mathbf{A}^\top$, $\mathcal{O}(m^3)$ operations for the subsequent matrix inversion and $\mathcal{O}(mn^2)$ to compute $\mathbf{H}\mathbf{A}$. Therefore, the offline calculations have a computational complexity of $\mathcal{O}(mn^2 + m^2n + m^3)$.

The computational complexity of the online computations is dominated by the computation of $\hat{\mathbf{x}}^{(0)}$ requiring $\mathcal{O}(nm)$ operations and by the computation of $\mathbf{z}^{(k)}$ in each of the γ iterations of the algorithm. The cost of computing $\mathbf{z}^{(k)}$ at iteration k is $\mathcal{O}(nk)$, and the sum over γ iterations is $\sum_{k=1}^{\gamma} nk = \frac{n(\gamma^2-\gamma)}{2}$ which gives a computational complexity of $\mathcal{O}(n\gamma^2)$. Thus, the online part of the proposed STIR algorithm has a complexity of $\mathcal{O}(nm + n\gamma^2)$. The overall complexity of the proposed algorithm is $\mathcal{O}(mn^2 + m^2n + m^3 + n\gamma^2)$.

To give a comparison with other methods, Table I summarize the computational complexity of the offline and online components of several algorithms in the literature. The accelerated OLS (AOLS) proposed in [23] for the case when only one element is selected at each iteration. The efficient OMP implementations studied in [24], including OMP implement using (i) Cholesky factorization, denoted as CHOL-OMP; (ii) the matrix inversion formula, denoted as MIL-OMP; and (iii) QR decomposition, denoted as QR-OMP.

TABLE I: Offline and online computational complexity for several greedy algorithms.

Algorithm	Complexity of Offline operations	Complexity of Online operations
STIR	$\mathcal{O}\left(mn^2 + m^2n + m^3\right)$	$\mathcal{O}(mn + n\gamma^2)$
AOLS	-	$\mathcal{O}(mn\gamma + n\gamma^2)$
CHOL-OMP	$\mathcal{O}(mn^2)$	$\mathcal{O}(mn + n\gamma^2 + \gamma^3)$
MIL-OMP	$\mathcal{O}(mn^2)$	$\mathcal{O}(mn + n\gamma^2 + m\gamma^2)$
QR-OMP	$\mathcal{O}(mn^2)$	$\mathcal{O}(mn\gamma + m\gamma^2)$

Table I shows that the *online* complexity of the proposed STIR algorithm competes with state-of-the-art implementations of other greedy algorithms and it is better than the online complexity of AOLS. However, because of the high complexity of the offline operations, the overall complexity of the proposed STIR algorithm is higher than the complexity of AOLS. Similarly, the overall complexity of STIR is slightly higher than the complexity of the OMP implementations.

C. STIR-based implementations of OLS and OMP

Here we describe how the STIR method can implement the forward stepwise selection criteria used by OLS and OMP.

A variant of Algorithm 1 may implement the OLS criterion. To do so, it is necessary to compute the cost reduction obtained by selecting a candidate element. In the STIR framework, such a reduction in cost is given by $g^k(\hat{\mathbf{x}}^{(k)}) - g^{k+1}(\hat{\mathbf{x}}^{(k+1)})$ described in (7). Thus, an OLS-STIR algorithm would use the following selection criterion in step 4.a of Algorithm 1.

$$\ell = \arg\max_{i\notin\mathcal{S}^{(k)}} \frac{(\hat{x}_i^{(k)})^2}{(1-d_i^{(k)})} \tag{14}$$

Notice that the OLS-STIR selection criterion (14) uses the same variables that the STIR selection criterion and can be implemented by slightly changing step 4.a of Algorithm 1.

To implement the OMP criterion in the STIR framework, it is necessary to compute the correlation between the current residual and the columns of the matrix $\mathbf{A}$. Assuming that the columns of $\mathbf{A}$ are normalized, i.e. $\|\mathbf{A}_{j-col}\| = 1$ for $j = 1, \dots, n$, the correlation is given by

$$\mathbf{c}^{(k)} = \mathbf{A}^\top\left(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}^{(k)}\right) \tag{15}$$

The following Theorem expresses $\mathbf{c}^{(k)}$ as a function of the variables in the STIR algorithm.

Theorem 2. *Let* $\mathbf{c}^{(k)} \in \mathbb{R}^n$ *given by* (15) *with* $\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^n$ *being a solution of* $\mathcal{P}_{tikho}^k$ *where the assumptions in Lemma 1 hold. Then,* $\mathbf{c}^{(k)}$ *can be expressed as*

$$c_i^{(k)} = \begin{cases} \rho\hat{x}_i^{(k)} & i \notin \mathcal{S}^{(k)} \\ 0 & i \in \mathcal{S}^{(k)} \end{cases} \tag{16}$$

Proof. Straightforward after considering (1), (3) and that $(1/\rho)\mathbf{A}^\top\mathbf{A}\mathbf{Z}^{(k)} = \mathbf{I} - \mathbf{W}^{(k)}\mathbf{Z}^{(k)}$. □

Then, an OMP-STIR algorithm would use the following selection criterion in step 4.a of Algorithm 1.

$$\ell = \arg\max_{i\notin\mathcal{S}^{(k)}} |\hat{x}_i^{(k)}| \tag{17}$$

Remark 1. *The OMP criterion only requires information about the current iteration, namely the current residual. In contrast, the OLS and STIR selection criteria require knowledge about the next iteration. OLS requires knowledge about the cost while STIR requires knowledge about the estimate.*

IV. Numerical simulations

This section presents a numerical study of the proposed methods. The proposed OLS-STIR, OMP-STIR and STIR algorithms are used with and without offline computations. These methods are compared against the existing methods AOLS, CoSaMP [28], and the OMP implementations detailed in [24], namely, CHOL-OMP, QR-OMP, and MIL-OMP. For CoSaMP, 10 iterations are used.

For matrix $\mathbf{A}$, the number of rows is $m = 500$ and the number of columns, n, is pick from the set $\{500, 600, \ldots, 2000\}$. For each value of n, a $\mathbf{A} \in \mathbb{R}^{m \times n}$ matrix is constructed, with each matrix entry being normally distributed random variables with zero mean and unitary variance. The columns of $\mathbf{A}$ are normalized such that $\|\mathbf{A}_{j-col}\| = 1$ for $j = 1, \ldots, n$. For each generated $\mathbf{A}$, 100 Monte Carlo simulations are generated using different vectors $\mathbf{y}$. The resulting $\mathcal{P}_0$ problem is solved with all the methods using $\gamma = 200$ and the execution time is measured. The simulations are performed using Matlab on a laptop with a 2.6GHz quad core processor with 32GB of RAM.

Fig. 1a shows the average execution time for several greedy algorithms. The AOLS running time is the largest and the execution time for the proposed STIR and OLS-STIR methods are considerably smaller than AOLS and CoSaMP. The STIR and OLS-STIR methods using offline computations have a significantly smaller average execution time than other methods.

Fig. 1b shows the average *online* execution time for several OMP implementations. Here, offline computations are not considered in the reported execution time. The proposed OMP-STIR method outperforms other OMP implementation, but the CHOL-OMP algorithm has a similar average execution time.

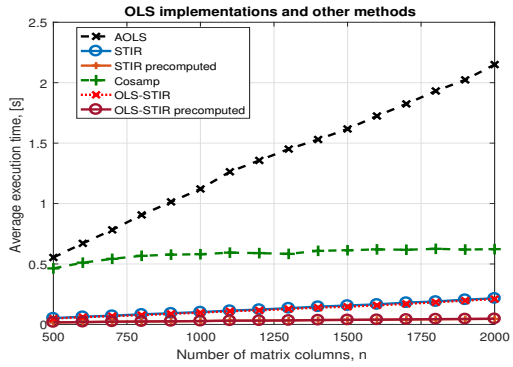
Next, a medium-scale example is tested, with $\gamma = 1000$, $m = 10000$ and the number of columns n is selected from the set $\{10000, 15000, \ldots, 40000\}$.

Fig. 1c shows the execution time of the *offline* part of the OMP implementations for the medium scale example. The OMP-STIR algorithm has the largest execution time for the offline part. This is consistent with the computational complexity results in Table I.

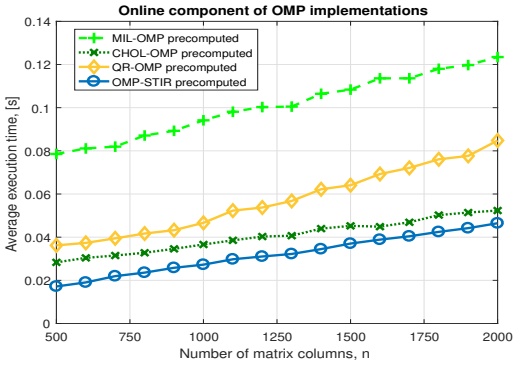
Fig. 1d shows the execution time of the *online* part of the OMP implementations. The proposed OMP-STIR algorithm outperforms all other methods.

V. Conclusion

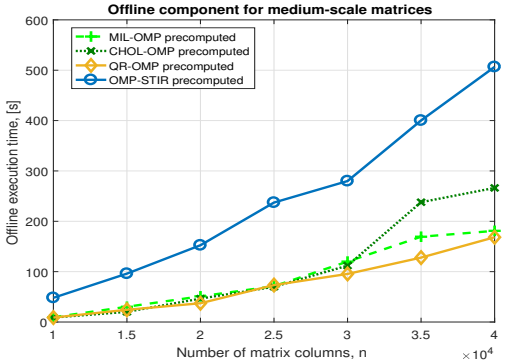
This letter proposed a general algorithmic framework to implement the greedy strategies of OLS, OMP, and STIR. The proposed approach uses offline computations to reduce the online execution time. Numerical simulations demonstrate that the proposed approach outperforms existing methods implementing OLS and STIR in terms of online execution time. The proposed OMP-STIR implementation competes with other state-of-the-art implementations of OMP.



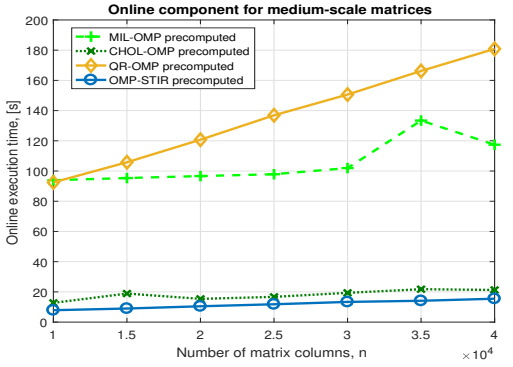
(a) Average execution time for AOLS, CoSaMP and the proposed STIR and OLS-STIR methods



(b) Average online execution time for the OMP implementations.



(c) Medium scale example: execution time for the *offline* component of the OMP implementations.



(d) Medium scale example: execution time for the *online* component of the OMP implementations.

Fig. 1: Comparison of the execution time of the proposed algorithms and other methods.

REFERENCES

[1] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[2] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 310–316, Apr. 2010.

[3] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[4] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, Jun. 2010.

[5] A. d'Aspremont, "A semidefinite representation for some minimum cardinality problems," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5. IEEE, 2003, pp. 4985–4990.

[6] O. P. Burdakov, C. Kanzow, and A. Schwartz, "Mathematical programs with cardinality constraints: Reformulation by complementarity-type conditions and a regularization method," *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 397–425, Jan. 2016.

[7] R. A. Delgado, J. C. Agüero, and G. C. Goodwin, "A novel representation of rank constraints for real matrices," *Linear Algebra and its Applications*, vol. 496, pp. 452–462, May 2016.

[8] M. Nikolova, "Relationship between the optimal solutions of least squares regularized with ℓ_0-norm and constrained by k-sparsity," *Applied and Computational Harmonic Analysis*, vol. 41, no. 1, pp. 237–265, Jul. 2016.

[9] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[10] R. P. Aguilera, G. Urrutia, R. A. Delgado, D. Dolz, and J. C. Aguero, "Quadratic model predictive control including input cardinality constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3068–3075, Jun. 2017.

[11] B. Cassidy, C. Rae, and V. Solo, "Brain activity: connectivity, sparsity, and mutual information," *IEEE Transactions on Medical Imaging*, vol. 34, no. 4, pp. 846–860, Apr. 2015.

[12] R. Arastoo, M. Bahavarnia, M. V. Kothare, and N. Motee, "Closed-loop feedback sparsification under parametric uncertainties," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, Dec. 2016.

[13] D. Carrasco, A. Mattews, G. Goodwin, R. Delgado, and A. Medioli, "Design of MDIs for Type 1 diabetes treatment via rolling horizon cardinality-constrained optimisation," in *20th IFAC World Congress (IFAC 2017)*, Toulouse, France, 2017.

[14] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.

[15] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[16] E. J. Candès and Y. Plan, "Near-ideal model selection by ℓ_1 minimization," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2145–2177, Oct. 2009.

[17] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[18] T. Blumensath, M. E. Davies, and G. Rilling, "Greedy algorithms for compressed sensing," in *Compressed Sensing*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge University Press, 2012, pp. 348–393.

[19] E. C. Marques, N. Maciel, L. A. B. Naviner, H. Cai, and J. Yang, "A review of sparse recovery algorithms," *IEEE Access*, pp. 1–1, 2018.

[20] Y. Arjoune, N. Kaabouch, H. E. Ghazi, and A. Tamtaoui, "Compressive sensing: performance comparison of sparse recovery algorithms," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, Jan. 2017.

[21] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *Neural Networks, IEEE Transactions on*, vol. 2, no. 2, pp. 302–309, Mar 1991.

[22] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, Nov 1993, pp. 40–44 vol.1.

[23] A. Hashemi and H. Vikalo, "Accelerated orthogonal least-squares for large-scale sparse reconstruction," *Digital Signal Processing*, vol. 82, pp. 91–105, Nov. 2018.

[24] B. L. Sturm and M. G. Christensen, "Comparison of orthogonal matching pursuit implementations," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug. 2012, pp. 220–224.

[25] R. Delgado, Z. Chen, and R. H. Middleton, "Stepwise Tikhonov regularisation: application to the prediction of HIV-1 drug resistance," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018.

[26] C. Herzet, A. Drémeau, and C. Soussen, "Relaxed recovery conditions for OMP/OLS by exploiting both coherence and decay," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 459–470, 2016.

[27] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.

[28] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009.