Sparsity-Promoting Methods in Remote Control Systems

Masaaki Nagahara
Graduate School of Informatics, Kyoto University,
Yoshida Honmachi Sakyo-ku, Kyoto, 606-8501, JAPAN
nagahara@ieee.org

***Abstract*—In remote control systems, efficient representation of control signals is one of the crucial issues because of bandwidth limitation of the communication channel, such as a wireless communication link between the controller and the controlled object. Recently, a new method based on the technique of compressive sampling has been proposed, in which control signals are sparsely representation based on convex relaxation. There exist however so many sparsity-promoting methods for compressive sampling. In this study, we perform a comparative study of sparsity-promoting methods and reveal their advantages and disadvantages by simulation in view of remote control over rate-limited networks.**

***Index Terms*—remote control, networked control, compressive sampling, compressed sensing, sparse representation**

I. Introduction

In *remote control* [1], the controlled objects are located away from controllers, and control signals are to be transmitted through rate-limited channels such as wireless channels [2] or the Internet [3]. To send control signals through such communication channels, efficient signal compression or representation is essential. For this, an approach has been proposed in [4], [5] for *sparsifying* control signals using the *compressive sampling* technique [6], [7], [8] for remote control systems.

The aim of compressive sampling (aka compressed sensing) is to reconstruct signals from much less information by assuming that the original signal is sparse [9], [10], [11]. The core idea used in this area is to introduce a sparsity index in the optimization. To be more specific, the sparsity index of a vector $\boldsymbol{v}$ is defined by the amount of nonzero elements in $\boldsymbol{v}$ and is usually denoted by $\|\boldsymbol{v}\|_0$, called the "ℓ^0 norm." The compressed sensing problem is then formulated by ℓ^0-norm optimization. By ℓ^0-norm optimization, one obtain a sparse vector contains many 0-valued elements, and can be highly compressed by only coding a few nonzeros and their locations. A well-known example of this sparsity-inducing compression is JPEG [12].

The associated optimization problem is however hard to solve, since it is a combinatorial one [13]. One approach to the combinatorial optimization is an iterative greedy algorithm called *Orthogonal Matching Pursuit* (OMP) [14] and its extension, *Compressive Sampling Matching Pursuit* (CoSaMP) [15]. Another tractable approach for ℓ^0-optimization is convex relaxation to use the ℓ^1 norm instead of the highly nonconvex ℓ^0 norm. This is called *Basis Pursuit Denoising* (BPDN) in signal processing, and can be also effectively solved via fast algorithms such as NESTA (Nestrov's Algorithm) based on Nesterov's method. [19], and FISTA (Fast IST Algorithm) [16] based on iterative shrinkage-thresholding [17], [18].

In this study, we perform a comparative study of the algorithms, OMP, CoSaMP, NESTA, and FISTA, for remote control systems. We reveal their advantages and disadvantages by simulation in view of remote control over rate-limited networks.

This research was supported in part under the JSPS Grant-in-Aid for Scientific Research (C) No. 24560543.

Notation

For vector $\boldsymbol{v} = [v_1, \ldots, v_n]^\top \in \mathbb{R}^n$, the ℓ^1 and ℓ^2 norms are respectively defined by

$$\|\boldsymbol{v}\|_1 := \sum_{i=1}^{n} |v_i|, \quad \|\boldsymbol{v}\|_2 := \left(\sum_{i=1}^{n} v_i^2\right)^{1/2}.$$

The set of indices of nonzero elements in $\boldsymbol{v}$ is denoted by $\mathrm{supp}(\boldsymbol{v}) := \{j : v_j \neq 0\}$, and the "$\ell^0$" norm is defined by $\|\boldsymbol{v}\|_0 := |\mathrm{supp}(\boldsymbol{v})|$, where $|\mathrm{supp}(\boldsymbol{v})|$ is the number of members in the set $\mathrm{supp}(\boldsymbol{v})$. For a matrix Φ, its norm is defined by

$$\|\Phi\| := \max_{\|\boldsymbol{v}\|_2 \neq 0} \frac{\|\Phi \boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2} = \sigma_{\max}(\Phi),$$

where $\sigma_{\max}(\Phi)$ is the maximum singular value of Φ. We denote by $L^2[0,T]$ the set of all square integrable functions on $[0,T]$ $(T > 0)$, endowed with the inner product

$$\langle f, g \rangle := \int_0^T f(t) g(t) \mathrm{d}t, \quad f, g \in L^2[0,T]$$

and the L^2 norm $\|f\|_2 := \langle f, f \rangle^{1/2}$.

II. Control Problem

A. Plant model

In this study, we consider a control problem of a linear system P on a finite time interval (or horizon) $[0,T]$, modeled by the following state-space equations:

$$P : \begin{cases} \dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \\ y(t) = \boldsymbol{c}^\top \boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{R}^\nu, \ t \in [0,T], \end{cases} \tag{1}$$

where $A \in \mathbb{R}^{\nu \times \nu}$ and $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^{\nu \times 1}$. The initial state $\boldsymbol{x}_0 \in \mathbb{R}^\nu$ is assumed to be given. We also assume that the system is stable.

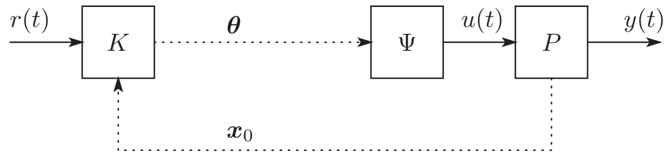

Fig. 1. Remote control system

B. Tracking problem

For the system (1), we here consider the *tracking problem*; given a reference signal $r(t)$ over the time interval $[0,T]$ and the state-space observation $\boldsymbol{x}(0)$ at the initial time $t=0$, find a control signal $u(t)$ over $[0,T]$ that minimizes the tracking error

$$E(u) := \|y-r\|_2^2 = \|Pu-r\|_2^2.$$

In addition, we assume *remote control* where the controller, denoted by K, that generates the control signal is located away from the controlled plant P. Fig. 1 shows a block diagram of the remote control system where the dotted lines indicate rate-limited networks such as wireless networks. In this system, the control signal $u(t)$ over $[0,T]$ should be represented in a finite dimensional vector, denoted by $\boldsymbol{\theta}$, which is transmitted through the rate-limited network. The received vector is transformed to the control signal $u(t)$, $t\in[0,T]$ by the *decoder* Ψ defined by

$$u = \Psi\boldsymbol{\theta} = \sum_{m=-M}^{M} \theta_m \psi_m,$$

where ψ_m is the m-th Fourier basis function on $[0,T]$, that is,

$$\psi_m(t) := \frac{1}{\sqrt{T}}\exp(\mathrm{j}\omega_m t),\ t\in[0,T],\ \omega_m := \frac{2\pi m}{T}.$$

Note that the control signal u is band-limited up to the frequency $\omega_M = 2\pi M/T$. Let V_M denote the set of such signals, that is,

$$V_M := \mathrm{span}\{\psi_m : m=-M,\dots,M\} \subset L^2[0,T].$$

Then we have $u\in V_M$. We also assume that the reference signal r is in V_M.

Under the above assumptions, the output $y(t)$ can be rewritten in terms of the Fourier coefficients θ_i as

$$y(t) = \boldsymbol{c}^\top \exp(tA)\boldsymbol{x}_0 + \sum_{m=-M}^{M} \theta_m \langle \kappa(t,\cdot), \psi_m\rangle,$$

where

$$\kappa(t,\tau) := \begin{cases} \boldsymbol{c}^\top \exp\left[(t-\tau)A\right]\boldsymbol{b}, & \text{if } 0\le\tau<t\le T,\\ 0, & \text{otherwise.}\end{cases}$$

Sampling the continuous-time signal $y(t)$ with sampling time $h := T/(N-1)$, where $N := 2M+1$, the dimension of the signal subspace V_M, we have

$$y(t_n) = \boldsymbol{c}^\top \exp(t_n A)\boldsymbol{x}_0 + \sum_{m=-M}^{M} \theta_m \langle \phi_n, \psi_m\rangle,$$
$$t_n := (n-1)h,\quad \phi_n := \kappa(t_n,\cdot),\quad n=1,2,\dots,N.$$

By this, the *sampled* tracking error

$$E_{\mathrm{d}}(u) = h\sum_{n=1}^{N} |y(t_n)-r(t_n)|^2$$

is rewritten as

$$E_{\mathrm{d}}(\boldsymbol{\theta}) = h\,\|G\boldsymbol{\theta} - H\boldsymbol{x}_0 - \boldsymbol{r}\|_2^2,$$

where

$$G := \begin{bmatrix} \langle\phi_1,\psi_{-M}\rangle & \dots & \langle\phi_1,\psi_M\rangle\\ \langle\phi_2,\psi_{-M}\rangle & \dots & \langle\phi_2,\psi_M\rangle\\ \vdots & \ddots & \vdots\\ \langle\phi_N,\psi_{-M}\rangle & \dots & \langle\phi_N,\psi_M\rangle \end{bmatrix} \in \mathbb{C}^{N\times N}, \quad (2)$$

$$\boldsymbol{r} := \begin{bmatrix} r(t_1)\\ r(t_2)\\ \vdots\\ r(t_N)\end{bmatrix} \in \mathbb{R}^N, \quad H := \begin{bmatrix} \boldsymbol{c}^\top \exp(t_1 A)\\ \boldsymbol{c}^\top \exp(t_2 A)\\ \vdots\\ \boldsymbol{c}^\top \exp(t_N A)\end{bmatrix} \in \mathbb{R}^{N\times\nu}.$$

C. Sparse control vector

Since the network is rate-limited, the vector $\boldsymbol{\theta}$ should be compressed (or represented) in as small data size as possible. For this purpose, we have proposed in [4] to use the method of *compressive sampling*.

Let U be a random "decimation" matrix of the form

$$U = \begin{bmatrix} \boldsymbol{e}_{i(1)}\\ \boldsymbol{e}_{i(2)}\\ \vdots\\ \boldsymbol{e}_{i(K)}\end{bmatrix} \in \{0,1\}^{K\times N},$$

where $i(1) < i(2) < \dots < i(K)$ are the random variables of the uniform distribution on $\{1,2,\dots,N\}$, and

$$\boldsymbol{e}_i := [0,\dots,0,\overset{\overset{i}{\vee}}{1},0,\dots,0], \quad i=1,2,\dots,N.$$

This is a model of low rate random sampling of a signal on $[0,T]$. We define the random sampling instants by

$$t_{i(k)} = i(k)\cdot h = i(k)\cdot\frac{T}{N-1}, \quad k=1,2,\dots,K<N.$$

By using the matrix U, random sampling of $y(t)$ on $[0,T]$ is given by:

$$\boldsymbol{y} = \begin{bmatrix} y\left(t_{i(1)}\right)\\ y\left(t_{i(2)}\right)\\ \vdots\\ y\left(t_{i(K)}\right)\end{bmatrix} = UG\boldsymbol{\theta} + UH\boldsymbol{x}_0.$$

The associated optimization problem is formulated by

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^N} \|\boldsymbol{\theta}\|_0 \ \text{ subject to } \ \|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2^2 \le \epsilon, \quad (3)$$

where $\Phi := UG$ and $\boldsymbol{\alpha} = U(\boldsymbol{r} - H\boldsymbol{x}_0)$. The objective here is to minimize $\|\boldsymbol{\theta}\|_0$ to obtain a sparse control vector with an ℓ^2 norm constraint for tracking performance. The optimization (3) is a combinatorial one, which is hard to solve in a reasonable computational time. For this, we adopt computationally tractable algorithms in the next section.

Algorithm 1 OMP for sparse control vector $\boldsymbol{\theta}$

Require: $\boldsymbol{\alpha} \in \mathbb{R}^K$ {observed vector}
Ensure: $\boldsymbol{\theta}$ {sparse control vector}
  $\boldsymbol{\theta}[0] := \mathbf{0}$, $\boldsymbol{r}[0] := \boldsymbol{\alpha} - \Phi\boldsymbol{\theta}[0]$, $\mathcal{S}[0] := \mathrm{supp}\{\boldsymbol{\theta}[0]\} = \emptyset$.
  $k := 0$.
  while $\|\boldsymbol{r}[k]\|_2 > \epsilon$ **do**
    for $j = 1$ **to** N **do**
      $z_j := \dfrac{\boldsymbol{\phi}_j^\top \boldsymbol{r}[k]}{\|\boldsymbol{\phi}_j\|_2^2} = \arg\min_{z \in \mathbb{R}} \|\boldsymbol{\phi}_j z - \boldsymbol{r}[k]\|_2^2$.
      $e_j := \|\boldsymbol{\phi}_j z_j - \boldsymbol{r}[k]\|_2^2$.
    end for
    Find a minimizer $j_0 \notin \mathcal{S}[k]$ such that $e_{j_0} \leq e_j$, for all $j \notin \mathcal{S}[k]$.
    $\mathcal{S}[k+1] := \mathcal{S}[k] \cup \{j_0\}$
    $\boldsymbol{\theta}[k+1] := \arg\min_{\mathrm{supp}\{\boldsymbol{\theta}\} = \mathcal{S}[k+1]} \|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2^2$.
    $\boldsymbol{r}[k+1] := \boldsymbol{\alpha} - \Phi\boldsymbol{\theta}[k+1]$.
    $k := k+1$.
  end while
  return $\boldsymbol{\theta} = \boldsymbol{\theta}[k]$.

Algorithm 2 NESTA for sparse control vector $\boldsymbol{\theta}$

Require: $\boldsymbol{\alpha} \in \mathbb{R}^K$ {observed vector}
Ensure: $\boldsymbol{\theta}$ {sparse control vector}
  $\boldsymbol{\theta}[0] := \mathbf{0}$, $\boldsymbol{w}[0] := \mathrm{sat}_\delta(\boldsymbol{\theta}[0])$, $\boldsymbol{v}[0] := \frac{1}{2}\boldsymbol{w}[0]$.
  $k := 1$.
  repeat
    $\boldsymbol{q}[k] := \boldsymbol{\theta}[k] - \delta\boldsymbol{w}[k]$.
    $\lambda_k := \max\left\{0, \frac{1}{\delta\epsilon}\|\boldsymbol{\alpha} - \Phi\boldsymbol{q}[k]\|_2 - \frac{1}{\delta}\right\}$.
    Solve the following linear equation for $\boldsymbol{\eta}[k]$:

$$(I + \delta\lambda_k\Phi^\top\Phi)\boldsymbol{\eta}[k] = \delta\lambda_k\Phi^\top\boldsymbol{\alpha} + \boldsymbol{q}[k].$$

    Solve the following linear equation for $\boldsymbol{\zeta}[k]$:

$$(I + \delta\lambda_k\Phi^\top\Phi)\boldsymbol{\zeta}[k] = \delta\lambda_k\Phi^\top\boldsymbol{\alpha} + \boldsymbol{\theta}[0] + \boldsymbol{q}[k].$$

    $\boldsymbol{\theta}[k+1] := \frac{2}{k+4}\boldsymbol{\zeta}[k] + \frac{k+2}{k+4}\boldsymbol{\eta}[k]$.
    $\boldsymbol{w}[k+1] := \mathrm{sat}_\delta(\boldsymbol{\theta}[k+1])$.
    $\boldsymbol{v}[k+1] := \frac{1}{2k+4}\boldsymbol{w}[k+1] + v[k]$.
    $k := k+1$.
  until $|f_\delta(\boldsymbol{\theta}[k-1]) - f_\delta(\boldsymbol{\theta}[k-2])| \leq \mathtt{EPS}$ **or** $k \geq \mathtt{MAXITER}$.
  return $\boldsymbol{\theta} = \boldsymbol{\theta}[k-1]$.

III. Sparsity-Promoting Methods

There are many algorithms to obtain a numerical solution of (3): greedy pursuit, bayesian framework, nonconvex optimization, and brute force [20]. In this article, we adopt greedy pursuit and convex optimization, in particular.

A. Greedy pursuit

Among many methods of greedy pursuit, the orthogonal matching pursuit (OMP) [14] is the most popular. **Algorithm 1** shows the algorithm of OMP. If the sparsity of the solution, $\|\boldsymbol{\theta}\|_0$ is previously known, then one can adopt a faster method called CoSaMP (Compressive Sampling Matching Pursuit) [15]. The CoSaMP algorithm takes advantage of RIP (Restricted Isometry Property) of sparse signals. For the algorithm, see [15].

B. Convex optimization

Another method to solve the optimization (3) is convex relaxation. For the relaxation, we replace the term $\|\boldsymbol{\theta}\|_0$ by the ℓ^1 norm $\|\boldsymbol{\theta}\|_1$, and solve the following convex optimization:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^N} \|\boldsymbol{\theta}\|_1 \quad \text{subject to} \quad \|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2^2 \leq \epsilon. \tag{4}$$

The solution of this optimization can be obtained by a standard method of convex programming, such as interior point method. However, there exist much faster method called NESTA (Nesterov's Algorithm) [19]. The main idea of the NESTA algorithm is to replace the non-smooth $\|\boldsymbol{\theta}\|_1$ term by a smooth function called Huber's function defined as

$$f_\delta(\boldsymbol{\theta}) = \sum_{i=1}^N |\theta_i|_\delta, \quad |\theta_i|_\delta := \begin{cases} \frac{1}{2\mu}\theta_i^2, & \text{if } |\theta_i| < \mu, \\ |\theta_i| - \frac{\mu}{2}, & \text{otherwise.} \end{cases}$$

By using Huber's function, the optimization (4) is approximated as

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^N} f_\delta(\boldsymbol{\theta}) \text{ subject to } \|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2 \leq \epsilon. \tag{5}$$

Note that for any $\boldsymbol{\theta} \in \mathbb{R}^N$, $f_\delta(\boldsymbol{\theta}) \to \|\boldsymbol{\theta}\|_1$ as $\delta \to 0$. Applying the Nesterov's method to the optimization problem (5), we obtain the NESTA algorithm shown in **Algorithm 2**. In this algorithm, $\mathrm{sat}_\delta(\boldsymbol{\theta})$ is a saturation function defined by

$$[\mathrm{sat}_\delta(\boldsymbol{\theta})]_i := \begin{cases} \frac{1}{\mu}\theta_i, & \text{if } |\theta_i| < \mu, \\ \mathrm{sgn}(\theta_i), & \text{otherwise,} \end{cases}$$

where $[\cdot]_i$ is the i-th element of a vector, and

$$\mathrm{sgn}(x) := \begin{cases} 1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0. \end{cases}$$

Note that the function $\mathrm{sat}_\delta(\boldsymbol{\theta})$ is the gradient of $f_\delta(\boldsymbol{\theta})$, i.e.,

$$\mathrm{sat}_\delta(\boldsymbol{\theta}) = \nabla f_\delta(\boldsymbol{\theta}).$$

It is known that the sequence $\{\boldsymbol{\theta}[0], \boldsymbol{\theta}[1], \dots\}$ converges to the solution of (5) for any initial vector $\boldsymbol{\theta}[0]$ at the convergence rate $O(1/k^2)$.

We can also adopt another method for the optimization (4) by considering the following Lagrange form:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^N} F(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}} \|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2^2 + \mu\|\boldsymbol{\theta}\|_1. \tag{6}$$

For this optimization problem, FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) is known to be a fast algorithm [16]. **Algorithm 3** shows the FISTA algorithm. In this algorithm, $\mathrm{shrink}_\mu(\boldsymbol{\theta})$ is a shrinkage function defined by

$$[\mathrm{shrink}_\mu(\boldsymbol{\theta})]_i := \mathrm{sgn}(\eta_i)(|\eta_i| - \lambda/c)_+,$$

Algorithm 3 FISTA for sparse control vector $\boldsymbol{\theta}$

Require: $\boldsymbol{\alpha} \in \mathbb{R}^K$ {observed vector}
Ensure: $\boldsymbol{\theta}$ {sparse control vector}
  $\boldsymbol{\theta}[0] := \mathbf{0}$, $\tilde{\boldsymbol{\theta}}[1] := \mathbf{0}$, $\beta[1] := 0$.
  $k := 1$.
  repeat
    $\boldsymbol{\theta}[k] := \mathrm{shrink}_\mu\bigl(\frac{1}{c}\Phi^\top(\boldsymbol{\alpha} - \Phi\tilde{\boldsymbol{\theta}}[k]) + \tilde{\boldsymbol{\theta}}[k]\bigr)$.
    $\beta[k+1] := \frac{1}{2} + \sqrt{\frac{1}{4} + \beta[k]^2}$.
    $\tilde{\boldsymbol{\theta}}[k+1] := \boldsymbol{\theta}[k] + \frac{\beta[k]-1}{\beta[k+1]}\bigl(\boldsymbol{\theta}[k] - \boldsymbol{\theta}[k-1]\bigr)$.
    $k := k+1$.
  until $|F(\boldsymbol{\theta}[k-1]) - F(\boldsymbol{\theta}[k-2])| \leq \texttt{EPS}$ **or** $k \geq \texttt{MAXITER}$.
  return $\boldsymbol{\theta} = \boldsymbol{\theta}[k-1]$.

where $(x)_+ := \max\{x, 0\}$ for $x \in \mathbb{R}$. If the parameter c is chosen such that $c \geq \|\Phi\|$, the sequence $\{\boldsymbol{\theta}[0], \boldsymbol{\theta}[1], \dots\}$ converges the optimal solution of (6) for any initial vector $\boldsymbol{\theta}[0]$. The convergence rate is known to be $O(1/k^2)$.

IV. Simulation Results

We here perform a comparative study of existing methods for the optimization (3) based on simulation. The state-space matrices of the controlled plant P are given by

$$A = \begin{bmatrix} 0 & 1 \\ -0.5 & -1.5 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} -0.5 \\ 1 \end{bmatrix}.$$

The transfer function of this system is given by

$$P(s) = \frac{s - 0.5}{(s+0.5)(s+1)}.$$

The period T is 2π. The number of basis $\{\psi_m\}$ is $N = 2M + 1 = 51$ ($M = 25$). The reference signal $r(t)$ is given by

$$r(t) = \sin(5t) + \cos(12.5t).$$

For compressive sampling, we take $K = 51/3 = 17$ random samples among $N = 51$ sampled data, that is the compression ratio is $1/3$.

With these parameters, we run 100 simulations with randomly generated initial state $\boldsymbol{x}_0$ and matrix U. We adopt four existing methods: OMP, CoSaMP, NESTA, and FISTA introduced in the previous section. Table I shows the average values of the control performance (or tracking error) $\|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2$, sparsity $\|\boldsymbol{\theta}\|_0$, computational (CPU) time (sec), and the number of iterations. The result indicates that OMP shows the best control performance and needs the fewest iterations, while FISTA gives the sparsest control vector and needs the shortest computational time. Note that NESTA gives dense vectors since the algorithm uses Huber's function f_δ, which leads to vectors many elements of which are almost zero (not exactly zero). To obtain a sparse vector with NESTA, one needs truncation. FISTA is quite fast and leads to very sparse vectors, but one should choose an appropriate parameter $\mu > 0$ in the algorithm. This requires some trial-and-error process. As a result, OMP is the best choice for the remote control system in the simulation.

TABLE I
COMPARISON

method	$\|\Phi\boldsymbol{\theta} - \boldsymbol{\alpha}\|_2$	$\|\boldsymbol{\theta}\|_0$	CPU time (sec)	# of iteration
OMP	0.33068	6.66	0.026208	9.81
CoSaMP	1.454	5	0.056219	345.58
NESTA	0.62325	51	1.183	1000
FISTA	1.1084	3.79	0.0016118	17.59

V. Conclusion

In this article, we have shown a comparative study of sparsity-promoting methods in remote control systems by simulation. The simulation result shows OMP (orthogonal matching pursuit) is the best for the remote control system considered in the simulation. Since many other algorithms have been also proposed for sparsity, it is important to seek for more suitable algorithm for remote control systems.

References

[1] C. Sayers, *Remote Control Robotics*. Springer, 1999.
[2] A. F. T. Winfield and O. E. Holland, "The application of wireless local area network technology to the control of mobile robots," *Microprocessors and Microsystems*, vol. 23, pp. 597–607, Mar. 2000.
[3] R. C. Luo and T. M. Chen, "Development of a multibehavior-based mobile robot for remote supervisory control through the internet," *IEEE/ASME Trans. Mechatron.*, vol. 5, no. 4, Dec. 2000.
[4] M. Nagahara, T. Matsuda, and K. Hayashi, "Compressive sampling for remote control systems," *IEICE Trans. on Fundamentals*, vol. E95-A, no. 4, pp. 713–722, Apr. 2012.
[5] M. Nagahara, D. E. Quevedo, T. Matsuda, and K. Hayashi, "Compressive sampling for networked feedback control," *Proc. IEEE ICASSP*, pp. 2733–2736, Mar. 2012.
[6] E. J. Candes, "Compressive sampling," *Proc. International Congress of Mathematicians*, vol. 3, pp. 1433–1452, Aug. 2006.
[7] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
[8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[9] M. Elad, *Sparse and Redundant Representations*. Springer, 2010.
[10] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse Image and Signal Processing*. Cambridge University Press, 2010.
[11] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.
[12] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, Apr. 1991.
[13] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.
[14] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Nov. 1993.
[15] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
[16] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Jan. 2009.
[17] M. A. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Processing*, vol. 12, no. 8, pp. 906–916, Aug. 2003.
[18] M. Zibulevsky and M. Elad, "L1-L2 optimization in signal and image processing," *IEEE Signal Processing Magazine*, vol. 27, pp. 76–88, May 2010.
[19] S. Becker, J. Bobin, and E. J. Candes, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imaging Sci.*, vol. 4, no. 1, pp. 1–39, 2011.
[20] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.