Compressive sampling for networked feedback control

by Masaaki Nagahara, Daniel E. Quevedo, Takahiro Matsuda, & Kazunori Hayashi

COMPRESSIVE SAMPLING FOR NETWORKED FEEDBACK CONTROL

Masaaki Nagahara⋆, Daniel E. Quevedo†, Takahiro Matsuda‡, Kazunori Hayashi⋆

⋆ Graduate School of Informatics, Kyoto University
† School of Electrical Engineering & Computer Science, The University of Newcastle
‡ Graduate School of Engineering, Osaka University

ABSTRACT

We investigate the use of compressive sampling for networked feedback control systems. The method proposed serves to compress the control vectors which are transmitted through rate-limited channels without much deterioration of control performance. The control vectors are obtained by an ℓ^1-ℓ^2 optimization, which can be solved very efficiently by FISTA (Fast Iterative Shrinkage-Thresholding Algorithm). Simulation results show that the proposed sparsity-promoting control scheme gives a better control performance than a conventional energy-limiting L^2-optimal control.

Index Terms— compressive sampling, networked control, ℓ^1-ℓ^2 optimization

1. INTRODUCTION

The objective of this article is to design a controller in a *networked control system* [2] that produces sparse control vectors for effective compression before transmissions. Unfortunately, the calculation of optimal sparse vectors will, in general, require significant computational cost and may thereby introduce delays, which are unacceptable for closed-loop operation. To overcome this issue, we subsample the problem to reduce its size and adopt a fast algorithm called *FISTA* (Fast Iterative Shrinkage-Thresholding Algorithm) [1].

Networked control systems are those in which the controlled plants are located away from the controllers, and the communication should be made through rate-limited communication channels such as wireless networks or the Internet [8]. In networked control systems, efficient signal compression or representation is essential to send control data through rate-limited communication channels. For this purpose, we propose an approach of sparse control signal representation using the *compressive sampling* technique [4]. Our contributions in this paper are (1) a new strategy for networked feedback control systems based on compressive sampling, (2) an effective data compression scheme of the control signals with sparse representation, (3) formulation of the design problem by ℓ^1-ℓ^2 optimization which can be efficiently solved.

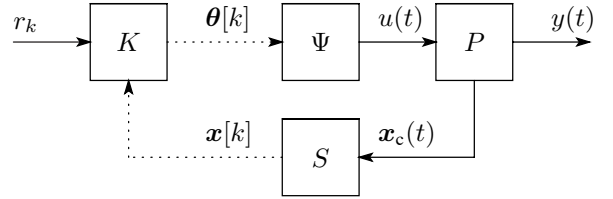The compressive sampling approach will open up a new vista in control theory. To the best of our knowledge, so far only a few studies have applied compressive sampling to control: [3] proposes to use compressive sensing in feedback control systems for perfect state estimation and [6] proposes sparse representation of transmitted control packets for feedback control with packet dropouts. For remote control systems, [7, 5] also propose to use ℓ^1-ℓ^2 optimization (as in this paper). However, [7, 5] consider only feed-forward control systems.

Fig. 1. Networked control system. The dotted line indicates a rate-limited communication channel.

2. CONTROL PROBLEM

Fig. 1 shows the networked control system which we consider in this article. The system consists of a controlled plant P, a sensor (or sampler) S, a decoder or a digital-to-analog (DA) converter Ψ, and a digital controller K. The definitions of these systems are given as follows:

Plant P: The controlled plant P is modeled by the following state-space representation:

$$P : \begin{cases} \dot{\boldsymbol{x}}_{\mathrm{c}}(t) = A\boldsymbol{x}_{\mathrm{c}}(t) + \boldsymbol{b}u(t), \\ y(t) = \boldsymbol{c}^{\top}\boldsymbol{x}_{\mathrm{c}}(t), \ t \in [0, \infty), \end{cases} \quad (1)$$

where $\boldsymbol{x}_{\mathrm{c}}(0) = \boldsymbol{0}$, $A \in \mathbb{R}^{\nu \times \nu}$, and $\boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^{\nu \times 1}$.

Sensor S: The sensor (or sampler) S converts the continuous-time state $\boldsymbol{x}_{\mathrm{c}}$ into a discrete-time signal $\boldsymbol{x}[k] := \boldsymbol{x}_{\mathrm{c}}(kT)$, $k = 0, 1, 2, \ldots$, where $T > 0$ is the sampling period.

Decoder Ψ: The decoder (or DA converter) Ψ converts a vector valued signal $\boldsymbol{\theta}[k]$ into a continuous-time signal

$\{u_k(t)\}_{t \in [0,T)}, k = 0, 1, 2, \ldots$ via

$$\Psi : \boldsymbol{\theta}[k] \mapsto u_k := \sum_{m=-M}^{M} \theta_m[k] \psi_m \in L^2[0,T], \quad (2)$$

where M is a positive integer, $\theta_m[k]$ is the m-th element of the vector $\boldsymbol{\theta}[k]$, and

$$\psi_m(t) := \frac{1}{\sqrt{T}} \exp(j\omega_m t), \ \omega_m := \frac{2\pi m}{T}, \ t \in [0,T). \quad (3)$$

We call the vector $\boldsymbol{\theta}[k]$ a *control vector*. Note that the continuous-time signal u_k is band-limited to $\omega_M = 2\pi M/T$ [rad/sec]. That is, u_k belongs to the following subspace of $L^2[0,T]$:

$$V_M := \mathrm{span}\{\psi_{-M}, \ldots, \psi_M\} \subset L^2[0,T]. \quad (4)$$

The input u to the plant P is defined by $u(t + kT) = u_k(t), t \in [0,T), k = 0, 1, 2, \ldots$.

Controller K: The controller K uses a continuous-time reference signal $r_k \in V_M$, $k = 0, 1, 2, \ldots$, and the sampled state $\boldsymbol{x}[k]$ to produce the control vector $\boldsymbol{\theta}[k]$. The latter defines the input signal u_k as per (2).

We assume that S, Ψ, and K are synchronized at $t = kT$, $k = 0, 1, 2, \ldots$. We also assume that we can transmit the control vector $\boldsymbol{\theta}[k]$ and the sampled state $\boldsymbol{x}[k]$ through communication channels without any delays nor packet dropouts. In this article, we consider a situation where the size $N = 2M + 1$ of $\boldsymbol{\theta}[k]$ is much larger than the size ν of the state $\boldsymbol{x}[k]$, and should be compressed because $\boldsymbol{\theta}[k]$ needs to be transmitted through a rate-limited communication channel.

Under these assumptions, we then formulate our control problem. Let y_k be the continuous-time signal y on the interval $[kT, (k+1)T)$, that is, $y_k(t) := y(t + kT), t \in [0,T)$, $k = 0, 1, \ldots$. We design the controller K to achieve the following objectives:

1. The first objective is to attenuate the tracking error between the reference r_k and the output y_k on the interval $[kT, (k+1)T), k = 0, 1, 2, \ldots$. The error is measured by the L^2 norm:

$$\|y_k - r_k\|_2^2 := \int_0^T |y_k(t) - r_k(t)|^2 \, \mathrm{d}t.$$

2. The second objective is to reduce the data size of the control vector $\boldsymbol{\theta}[k]$ which defines the control u_k via (2). For this objective, we adopt the so-called 0-norm of u_k defined by $\|u_k\|_0 := \|\boldsymbol{\theta}[k]\|_0$, the number of the nonzero elements in $\boldsymbol{\theta}[k]$.

In general, there is a trade-off between tracking-error attenuation and data-size reduction. For example, the sparsest solution $u_k \equiv 0$ leads to very large error, and the control u_k which

minimizes only the first objective function may not be sparse. To solve this problem, we adopt *regularization*. The problem is formulated as follows.

Problem 1 Given reference signal $r_k \in V_M, k = 0, 1, 2, \ldots$, find the control $u_k \in V_M$ (or the control vector $\boldsymbol{\theta}[k]$) which minimizes

$$J(u_k) := \|y_k - r_k\|_2^2 + \mu\|u_k\|_0, \quad (5)$$

where $\mu > 0$ is the regularization parameter to reconcile the trade-off between the tracking error and the sparsity.

3. COMPRESSIVE SAMPLING FOR SPARSE CONTROL VECTORS

The objective function $J(u_k)$ in (5) is defined on an infinite-dimensional signal subspace V_M defined in (4). We here relax the objective function into a finite-dimensional convex ℓ^1-ℓ^2 optimization problem by using the technique of compressive sampling.

Since the signals r_k and u_k are assumed to be band-limited up to the frequency $\omega_M = 2\pi M/T$ [rad/sec], we can safely discretize the signals by sampling them at a sampling frequency higher than $2\omega_M$ based on Shannon's sampling theorem [9]. However, if M is very large, it may take very long time to compute the optimal vector. It follows that there may exist a large delay in the feedback loop, which may lead to instability and control performance deterioration. Hence it is preferable to use a more efficient method than Shannon's sampling. For this purpose, we adopt the technique of compressive sampling [4] with *random sampling*, which can reduce the computational load for the optimization.

Random sampling is modeled as follows: we first split the interval $[0,T)$ with sampling points $t_n := (n-1)/f_M, n = 1, 2, \ldots, N = 2M + 1$, where $f_M := 2M/T$ is the Nyquist rate. Then we randomly choose K sampling points ($K < N$) from $\{t_1, \ldots, t_N\}$. To model this, we define a random matrix $U := [\boldsymbol{e}_{i(1)}, \boldsymbol{e}_{i(2)}, \ldots, \boldsymbol{e}_{i(K)}]^\top \in \{0,1\}^{K \times N}$, where $i(1), \ldots, i(N)$ are discrete random variables chosen from the uniform distribution on $\{1, 2, \ldots, N\}$ such that $i(l) < i(l+1), l = 0, 1, \ldots, N-1$, and $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N\}$ is the standard basis in $\mathbb{R}^N$, that is, $\boldsymbol{e}_n$ ($n = 1, 2 \ldots, N$) denotes a unit vector whose n-th element is equal to one and the other elements are equal to zero. The random matrix U is re-chosen at every sampling step k.

By using the random variables $i(1), i(2), \ldots, i(K)$, we define the random sampling points by $t_{i(l)} := i(l)h, h := T/(N-1), l = 1, 2, \ldots, K < N$. Then we consider random sampling of the output y_k. The sampled output $y_k(t_n)$ with the control signal $u_k \in V_M$ defined in (2) is computed by

$$y_k(t_n) = \boldsymbol{c}^\top \exp(t_n A)\boldsymbol{x}_0 + \sum_{m=-M}^{M} \theta_m[k]\langle \phi_n, \psi_m \rangle, \quad (6)$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $L^2[0,T]$, and

$$\phi_n(t) := \begin{cases} c^\top \exp\left[(t_n - t)A\right] b, & t \in [0, t_n), \\ 0, & t \in (t_n, T]. \end{cases}$$

Define the randomly sampled output vector

$$y[k] := [y_k(t_{i(1)}), \ldots, y_k(t_{i(K)})]^\top \in \mathbb{R}^K.$$

Then by (6), we have $y[k] = UG\theta[k] + UHx[k]$, where G is an $N \times N$ matrix defined by $(G)_{ij} = \langle \phi_i, \psi_j \rangle$, $i = 1, \ldots, N, j = -M, \ldots, M$, and H is an $N \times \nu$ matrix defined by $H := \left[\exp(t_1 A^\top)c, \ldots, \exp(t_N A^\top)c\right]^\top$. Let $r[k] := [r_k(t_1), r_k(t_2), \ldots, r_k(t_N)]^\top \in \mathbb{R}^N$ and $\Phi := UG$, $\alpha[k] = U(r[k] - Hx[k])$. Then the tracking error at the random sampling points $\{t_{i(1)}, t_{i(2)}, \ldots, t_{i(K)}\}$ is given by $y[k] - r[k] = \Phi\theta[k] - \alpha[k]$. It follows that the cost function (5) in Problem 1 is approximately described in a finite-dimensional one:

$$J_0(\theta[k]) := \|\Phi\theta[k] - \alpha[k]\|_2^2 + \mu\|\theta[k]\|_0. \quad (7)$$

The minimization of the cost function (7) is still difficult to solve when M is large since the optimization is a combinatorial one. To reduce this, we adopt a convex relaxation by replacing the ℓ^0 norm with the ℓ^1 norm:

$$J_1(\theta[k]) := \|\Phi\theta[k] - \alpha[k]\|_2^2 + \mu\|\theta[k]\|_1. \quad (8)$$

The cost function $J_1(\theta[k])$ in (8) is convex in $\theta[k]$ and hence the optimal value uniquely exists. To obtain the ℓ^1-ℓ^2 optimal vector, we use an iterative algorithm called FISTA [1]. This algorithm is very simple and fast; it can be effectively implemented in digital devices, which leads to a real-time computation in the feedback loop. For this algorithm, see [1].

4. SIMULATION RESULTS

In this section, we illustrate simulation results to show the effectiveness of the compressive sampling technique in networked feedback control systems.

The matrices in the state-space representation (1) of the controlled plant P are taken as

$$A = \begin{bmatrix} 0 & 1 \\ -\alpha\beta & -\alpha-\beta \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad c = \begin{bmatrix} -\alpha \\ 1 \end{bmatrix},$$

where $\alpha = 5$ and $\beta = 10$. We assume the initial state $x(0) = 0$. The control period T is set to be 2π. The number of the basis functions $\{\psi_m\}$, or the size of the control vector $\theta[k]$ is $N = 2M + 1 = 101$ ($M = 50$). We use the reference

$$r_k(t) = \sin(10t) + \cos(5t), \ k = 0, 1, 2, \ldots.$$

The sparsity of the reference r_k is given by $\|r_k\|_0 = 8 \ll N = 101$. Therefore, r_k is a sparse vector when it is represented by the basis functions $\{\psi_m\}$ defined in (3). That
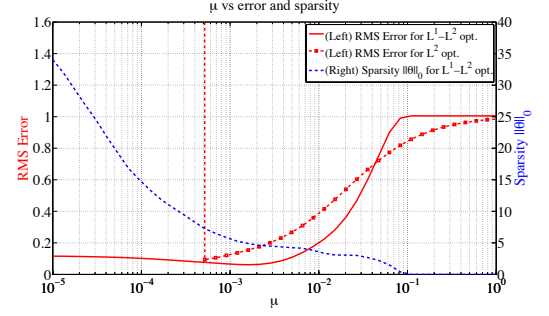

Fig. 2. Regularization parameter versus RMS tracking error and average sparsity

is, the reference r_k is sparse with respect to the basis $\{\psi_m\}$. The shortest sampling interval in random sampling is $h = T/(N-1) = 2\pi/100$. We set the number of random sampling $K = 33$. The iteration steps in FISTA for minimizing the ℓ^1-ℓ^2 optimization in (8) is 10. We run the simulation of the feedback control for $k = 0, 1, \ldots, 100$, that is, the length of simulation time is $T_{\mathrm{f}} := T \times 101 = 202\pi$.

First, we compute the relation between the regularization parameter μ in (8) and metrics for control performance to be achieved by the optimal control vector $\theta[k]$. We use two metrics: RMS (Root Mean Square) of the tracking error $e := y - r$ and the average sparsity of control vector. $\{\theta[k]\}_{k=0}^{100}$. The RMS is defined as:

$$\mathrm{RMS}(e) := \sqrt{\frac{1}{T_{\mathrm{f}}} \int_0^{T_{\mathrm{f}}} |e(t)|^2 \mathrm{d}t} = \sqrt{\frac{1}{T_{\mathrm{f}}} \sum_{k=0}^{100} \|y_k - r_k\|_2^2}.$$

The average sparsity is defined as: $\|\theta\|_0 := \sum_{k=0}^{100} \frac{\|\theta[k]\|_0}{101}$. Fig. 2 shows the performance as a function of the parameter μ. To compare the proposed method with a conventional one, we consider the L^2-optimal control which minimizes $J_2(u_k) := \|y_k - r_k\|_2^2 + \mu_2\|u_k\|_2^2$. This cost function limits the energy (i.e., the L^2 norm) of the control u_k, which has been widely used in control. The optimal control vector, say $\theta_2[k]$, is given by

$$\theta_2[k] = (\mu_2 I + G^\top G)^{-1} G^\top (r[k] - Hx[k]), \quad (9)$$

where we assume the control u_k is in the subspace V_M defined in (4). The RMS error performance for the L^2-optimal control is also shown in Fig. 2. In this case, the feedback system becomes unstable for $\mu_2 < 0.0005$. The sparsity of the optimal control vector is $\|\theta_2[k]\|_0 = 101$ for all $k = 0, 1, \ldots, 101$. That is, the L^2-optimal control does not produce any sparse vectors at all.

Fig. 2 suggests that the optimal parameter is $\mu \approx 0.002$ for ℓ^1-ℓ^2 optimization, and $\mu_2 \approx 0.0005$ for L^2 optimization. With these parameters, we simulate the feedback control. Fig. 3 (a) shows the absolute value of the nonzero elements in the ℓ^1-ℓ^2 optimal control vector $\theta[k]$ at $k = 50$. We can see that the number of the nonzero elements is 8 out
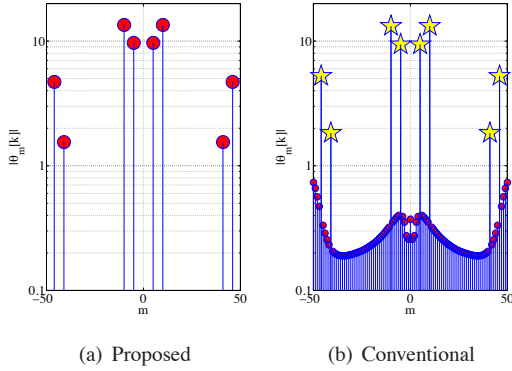
(a) Proposed   (b) Conventional

Fig. 3. The absolute values of the nonzero coefficients of $\boldsymbol{\theta}[k]$ based on compressive sampling with ℓ^1-ℓ^2 optimization (left)and $\boldsymbol{\theta}_2[k]$ based on L^2 optimal control (right) at $k = 50$. The stars in the right figure are the 8 elements of the truncated vector.
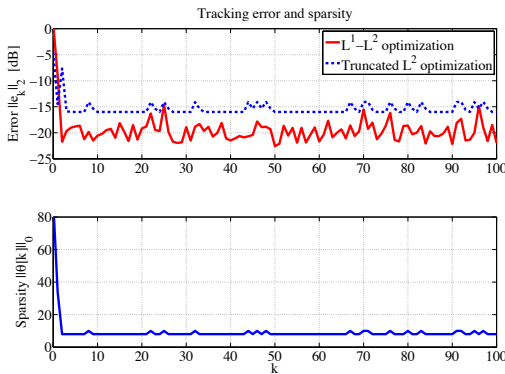

Fig. 4. Top figure: Tracking error $\|e_k\|_2$ in dB (solid: proposed, dash: conventional), bottom figure: sparsity $\|\boldsymbol{\theta}[k]\|_0$

of 101 (the size of the vector), and hence the vector is very sparse. Then, the L^2 norm of the tracking error $e_k := r_k - y_k$ on the k-th period $\|e_k\|_2$ $(k = 0, 1, \ldots, 100)$ is shown in the top figure in Fig. 4. The sparsity history $\{\|\boldsymbol{\theta}[k]\|_0\}_{k=0}^{100}$ is shown in the bottom figure in Fig. 4.

To compare the proposed method with conventional L^2-optimal control, we compute the control vector $\boldsymbol{\theta}_2[k]$ by the formula (9) with $\mu_2 = 0.0005$. Fig. 3 (b) shows the absolute value of the nonzero elements in the control vector $\boldsymbol{\theta}_2[k]$ at $k = 50$. We can see that all the elements in this vector are nonzero (cf Fig. 3 (a)). One may think that the vector $\boldsymbol{\theta}_2[k]$ is *compressible* since almost all the elements are nearly zero. To see the difference, we truncate the full vector $\boldsymbol{\theta}_2[k]$ by using the sparsity history in Fig. 4.

The stars in Fig. 3 (b) are the 8 elements of the truncated vector. The tracking error by the truncated vectors is shown in the top figure in Fig.4. The proposed method shows the better performance than the truncated L^2-optimal control with the

same data size. This shows the effectiveness of our method.

In an additional simulation study, we considered a step function for the reference, that is, $r_k(t) = r \in \mathbb{R}$, $k = 0, 1, 2, \ldots$. This signal is also sparse in the space V_M and produces a sparse control vector (we omit details due to space limitations).

5. CONCLUSION

We have studied the use of compressive sampling for feedback control systems with rate-limited communication channels. Simulation studies indicate that the method proposed can effectively compress the signals transmitted. Control vectors are obtained via an ℓ^1-ℓ^2 optimization, which is solved by the FISTA algorithm. Future work could include further investigation of bit-rate issues and the study of closed loop stability.

6. REFERENCES

[1] 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.

[2] A. Bemporad, M. Heemels, and M. Johansson, *Networked Control Systems*, Springer, Oct. 2010.

[3] S. Bhattacharya and T. Başar, "Sparsity based feedback design: a new paradigm in opportunistic sensing," *Proc. of ACC*, pp. 3704–3709, Jul. 2011.

[4] E. J. Candes, "Compressive sampling," *Proc. International Congress of Mathematicians*, vol. 3, pp. 1433–1452, Aug. 2006.

[5] M. Nagahara, T. Matsuda, and K. Hayashi, "Compressive Sampling for Remote Control Systems," to appear in *IEICE Trans. on Fundamentals*, Vol. E95-A, No. 4, Apr. 2012.

[6] M. Nagahara and D. E. Quevedo, "Sparse representations for packetized predictive networked control," *Proc. IFAC 18th World Congress*, pp. 84–89, Aug. 2011.

[7] M. Nagahara, D. E. Quevedo, J. Østergaard, T. Matsuda, and K. Hayashi, "Sparse command generator for remote control," *Proc. 9th IEEE ICCA*, pp. 1055–1059, Dec. 2011.

[8] A. S. Tanenbaum, *Computer Networks*, 4th Ed., Prentice Hall, 2002.

[9] M. Unser, "Sampling — 50 years after Shannon," *Proc. of the IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.