

圧縮センシングの遠隔制御系への応用

Compressed Sensing Application to Remote Control Systems

永原正章[†]

松田崇弘[‡]

林和則[†]

[†] 京都大学大学院情報学研究科

[‡] 大阪大学大学院工学研究科

Masaaki Nagahara[†]

Takahiro Matsuda[‡]

Kazunori Hayashi[†]

[†] Graduate School of Informatics, Kyoto University

[‡] Graduate School of Engineering, Osaka University

アブストラクト In remote controlled systems, efficient representation of control signals is one of the crucial issues because of the bandwidth-limited communication channel, such as a wireless communication link, between the controller and the controlled object. In this article, we propose a compressed sensing-based sparse representation of control signal, which enables us to reduce the differential entropy of the control signal to be transmitted from the controller to the plant, while keeping the controlled performance. The sparse representation can be effectively obtained by solving l^1 - l^2 optimization problem with the iterative shrinkage method. We show design examples to illustrate the effectiveness of the proposed method.

1 はじめに

圧縮センシングまたは圧縮サンプリングは、元のデータがスパースであるという仮定を置くことによって、非常に少ない観測データから元のデータをかなり正確に復元できる手法として、近年、とくに信号処理や情報理論の分野で盛んに研究されている [2], [5]。この手法の特徴は、スパースネスの指標であるベクトルの非ゼロ要素の数（しばしば“ l^0 ノルム”と呼ばれる）の最小化を、その凸緩和である l^1 ノルムの最小化に置き換えて逆問題を解く点にある。これは、最小二乗法の正則化項に l^1 ノルムを用いる LASSO (Least Absolute Shrinkage and Selection Operator) と呼ばれるものとも関係がある [5]。この最適解は、内点法を用いて計算することもできるが、近年、繰り返し縮小法 (iterative shrinkage) と呼ばれる手法が提案され、より高速に最適解を見つけることが可能となった [1], [6]。

一方遠隔制御では、制御入力信号は無線伝送路などの伝送帯域の制約が強い通信路を介してシステムに加えら

れる場合が多い。また、通信路で雑音等の原因によりビット誤りが生じる場合には、誤り訂正符号等で冗長化した信号を送る必要があり、制御入力信号に対する伝送帯域の制約はより強いものとなる。従って、与えられた伝送帯域で効果的に制御信号を送るための手法が重要である。本稿では、圧縮センシングの手法を用いてスパースな（すなわち多くの要素が 0 であるような）制御入力信号を得ることにより、これを表現するのに必要な情報量を低減する手法を提案する。本手法は制御入力信号の情報圧縮法の一つととらえることができるが、通常の情報源符号化における情報圧縮とは本質的に異なり、制御性能が劣化しないように情報源の確率分布そのものを変更することで、その微分エントロピーを減少させるものである。

具体的には、制御対象の出力軌道に対する追従誤差を l^2 ノルムで測り、かつ送信信号（設計パラメータ）のスパースネスに対して l^1 ノルムのペナルティ項を加えることにより、制御性能の向上とスパースネスとを同時に達成する制御信号を設計する。この設計は、 l^1 ではなく l^2 のペナルティを考えた場合、S. Sun ら [4] によって提案された制御理論的スプラインの問題となる。一方、我々の問題は、 l^1/l^2 最適化として定式化されるが、上記で述べた繰り返し縮小法により効率的に最適解を得ることができる。この設計によって、制御性能を大幅に劣化させることなく、制御信号の微分エントロピーを減少させることが可能であることを設計例により示す。

記法

ベクトル $v = [v_1, \dots, v_n]^T \in \mathbb{R}^n$ に対して、その l^1 ノルムおよび l^2 ノルムを

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

で定義する．また行列 Φ に対して，そのノルムを

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

で表す．ここで $\sigma_{\max}(\Phi)$ は行列 Φ の最大特異値である．実数 $x \in \mathbb{R}$ に対して

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

$$(x)_+ := \max\{x, 0\}$$

と定義する．区間 $[0, T]$ ($T > 0$) 上の実数値 Lebesgue 2 乗可積分関数の全体を $L^2[0, T]$ と書く． $f, g \in L^2[0, T]$ に対して，内積を

$$\langle f, g \rangle := \int_0^T f(t)g(t)dt$$

で表す．

2 問題設定

次の線形システム P を考える：

$$P: \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases} \quad t \in [0, \infty). \quad (1)$$

ここで $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times 1}$ and $C \in \mathbb{R}^{1 \times n}$ とし， $x(0) = 0$ とする．また， (A, B) は可制御， (C, A) は可観測とする．このシステムの出力信号 y に対して，次のデータ点を与える：

$$\mathcal{D} := \{(t_1, \alpha_1), (t_2, \alpha_1), \dots, (t_N, \alpha_N)\}.$$

ここで，各 t_i は時刻を表すデータで

$$0 < t_1 < t_2 < \dots < t_N =: T$$

とし，その各時刻において，システムの出力 $y(t_i)$ が α_i に近くなるように，すなわち $y(t_i) \approx \alpha_i$ となるように制御入力 u を設計したい．

この設計問題に対して，次の 2 ノルム評価関数を考える：

$$J_2(u) := \lambda_2 \int_0^T u(t)^2 dt + \sum_{i=1}^N w_i (y(t_i) - \alpha_i)^2. \quad (2)$$

ここで， $w_i > 0$ は各データ点へのフィッティングに対する重みである．この評価関数 $J_2(u)$ を最小化する最適制御入力は，次式で表されることが知られており，制御理論的スプラインと呼ばれる [3], [4]:

$$u(t) = \sum_{i=1}^N \theta_i g_i(t). \quad (3)$$

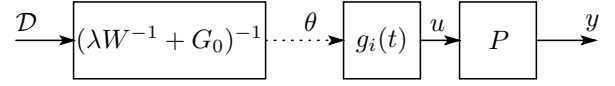


図 1: Remote control system optimized with $J_2(u)$ in (2). The vector θ is transmitted through a communication channel.

関数 $g_i(t)$ は

$$g_i(t) := \begin{cases} Ce^{A(t_i-t)B}, & \text{if } t_i > t, \\ 0, & \text{if } t_i \leq t \end{cases} \quad (4)$$

で定義され，係数 $\theta := [\theta_1, \dots, \theta_N]^T$ は

$$\theta = (\lambda_2 W^{-1} + G_0)^{-1} \alpha \quad (5)$$

で与えられる．ここで， $\alpha := [\alpha_1, \dots, \alpha_N]^T$ であり，また

$$G_0 := \begin{bmatrix} \langle g_{t_1}, g_{t_1} \rangle & \dots & \langle g_{t_N}, g_{t_1} \rangle \\ \vdots & \ddots & \vdots \\ \langle g_{t_1}, g_{t_N} \rangle & \dots & \langle g_{t_N}, g_{t_N} \rangle \end{bmatrix},$$

$$W := \text{diag}\{w_1, \dots, w_N\}$$

である．遠隔制御系の枠組みで考えると，この制御では，まずデータ \mathcal{D} が与えられたとき，(5) によって係数ベクトル θ が計算され，この θ が通信路に送信される．受信側では，(3) により制御入力信号 u が計算される．この状況では，通信路の容量に制約があるのが通常で， θ のデータ量は少ない方が望ましい．したがって，圧縮センシングの考え方 [2], [5] に従い，送信信号 θ のスパース表現を考える．まず，制御入力 u は， $L^2[0, T]$ の部分空間

$$\mathcal{V}_M := \left\{ u \in L^2[0, T] : u = \sum_{i=1}^M \theta_i \psi_i, \theta_i \in \mathbb{R} \right\}. \quad (6)$$

に属する信号と仮定する ($M \geq N$ とする)．ここで， $\{\psi_i\}$ は $L^2[0, T]$ の線形独立なベクトルとする．このとき，評価関数 $J_2(u)$ の第二項は次のように表現できる：

$$\begin{aligned} & \sum_{i=1}^N w_i (y(t_i) - \alpha_i)^2 \\ &= \theta^T G^T W G \theta - 2\theta^T G^T W \alpha + \alpha^T W \alpha \\ &= \|\Phi \theta - v\|_2^2. \end{aligned}$$

ただし，

$$G := \begin{bmatrix} \langle g_1, \psi_1 \rangle & \dots & \langle g_N, \psi_1 \rangle \\ \vdots & \ddots & \vdots \\ \langle g_1, \psi_M \rangle & \dots & \langle g_N, \psi_M \rangle \end{bmatrix}$$

とし、 $\Phi := W^{1/2}G$, $v := W^{1/2}\alpha$ とする．これより、係数ベクトル θ に ℓ^1 のペナルティを与えた次の評価関数を導入する：

$$J_1(\theta) := \lambda \|\theta\|_1 + \|\Phi\theta - v\|_2^2. \quad (7)$$

上で述べたように、第一項の ℓ^1 ノルムはベクトル θ のスパースネスを表す指標である．この評価関数を最小化する最適解を θ^* とすると、最適制御入力は

$$u^*(t) = \sum_{i=1}^N \theta_i^* \psi_i(t), \quad t \in [0, T]$$

で与えられる．

3 ℓ^1 - ℓ^2 最適化によるスパース表現

(7) の評価関数は θ に関して凸であり、最適解は必ず存在する．しかし、その最適解は (5) のように解析的には表すことができず、最適解を求めるためには繰り返し計算が必要である．この繰り返し計算には、内点法などが従来用いられてきたが、近年、繰り返し縮小法 (iterative shrinkage) [1], [6] と呼ばれる手法が提案され、これにより高速な数値計算が可能となった．

繰り返し縮小法のアルゴリズムは以下で与えられる [1], [6]: 初期値 $\theta[0] \in \mathbb{R}^N$ を与え、 $\beta[1] = 1$, $\eta[1] = \theta[0]$ とする．定数 c を

$$c > \|\Phi\|^2 = \sigma_{\max}(\Phi)^2$$

となるように選ぶ．次を繰り返す¹:

$$\begin{aligned} \theta[k] &= S_{\lambda/c} \left(\frac{1}{c} \Phi^\top (v - \Phi\eta[k]) + \eta[k] \right) \\ \beta[k+1] &= \frac{1 + \sqrt{1 + 4\beta[k]^2}}{2} \\ \eta[k+1] &= \theta[k] + \frac{\beta[k] - 1}{\beta[k+1]} (\theta[k] - \theta[k-1]) \\ k &= 1, 2, \dots \end{aligned}$$

関数 $S_{\lambda/c}$ は $\eta = [\eta_1, \dots, \eta_N]^\top$ に対して

$$S_{\lambda/c}(\eta) := \begin{bmatrix} \text{sgn}(\eta_1)(|\eta_1| - \lambda/c)_+ \\ \vdots \\ \text{sgn}(\eta_N)(|\eta_N| - \lambda/c)_+ \end{bmatrix}$$

で定義される非線形関数である．関数 $S_{\lambda/c}$ の各要素に対する非線形関数 $\text{sgn}(\eta)(|\eta| - \lambda/c)$ の入出力関係を Fig. 2 に示す．条件 $c > \|\Phi\|^2$ が成り立つような $c > 0$ を選べば、上のアルゴリズムは任意の Φ と v に対して収束し、

¹繰り返し縮小法にも様々な方法が提案されているが [6]、ここで用いた方法は FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) と呼ばれる方法である [1]．

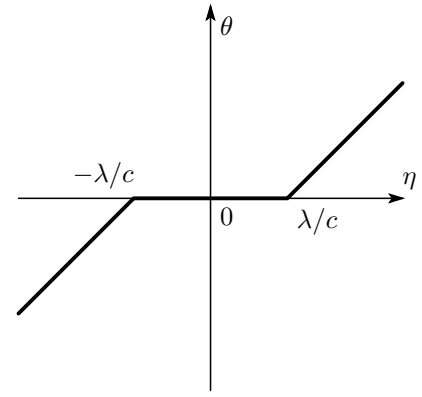


図 2: Nonlinear function $\theta = \text{sgn}(\eta)(|\eta| - \lambda/c)_+$

$\lim_{k \rightarrow \infty} \theta[k]$ は (7) を最小化する最適解に一致する [1]．さらに、最適解への収束の早さは最悪評価で $O(1/k^2)$ であることも知られている [1]．上記のアルゴリズムは、計算機に容易に実装が可能であり、収束も早いことから、リアルタイムでスパースなベクトル θ を得ることが可能である．

4 設計例

線形システム P の実現を

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

とする．すなわち、 $P(s) = 1/s^3$ である．データ D のサンプル時刻を $t_i = 0.1 + (i-1) \times 0.5$, $i = 1, 2, \dots, 11$ とし、データ α_i を正弦波 $\alpha_i = \sin t_i$ とする (図 3 を参照)．また、 $L^2[0, T]$ の線形独立なベクトル $\psi_i(t)$ は、上記の A, B, C および $\{t_1, \dots, t_{11}\}$ を使って (4) によって定義される $g_i(t)$ と同じものとする²．また、(2) に対する λ を 0.1、(7) に対する λ を 0.02 とする．これらの設定のもとで、(3) で計算される、(2) の ℓ^2 評価関数を最適化する θ を θ_2^* とおき、また (7) の ℓ^1 - ℓ^2 評価関数を最適

²(A, B) が可制御かつ (C, A) が可観測ならば、(4) で定義される $\{g_i\}$ は線形独立となることが知られている [3]．

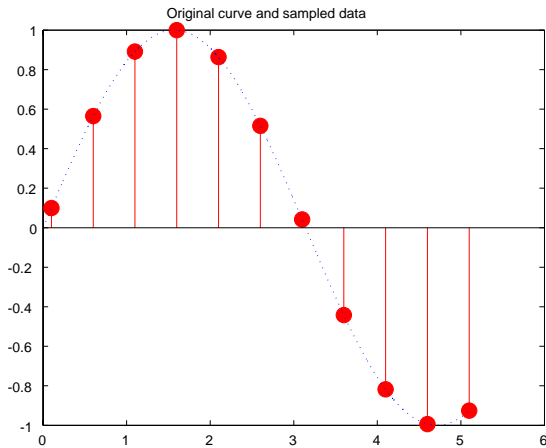


図 3: Original curve $\sin t$ and sampled data \mathcal{D} .

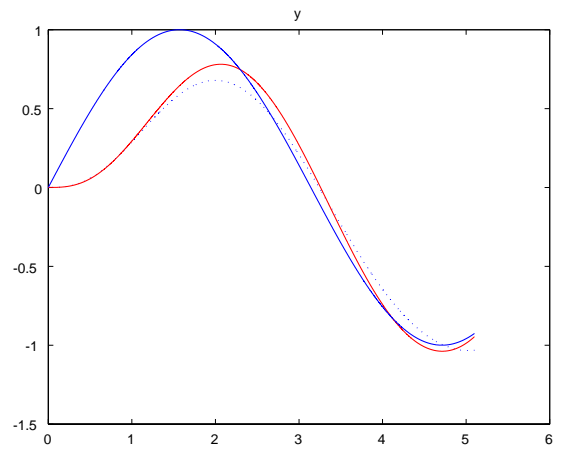


図 4: Original curve $\sin t$: (blue line), Output $y(t)$ of ℓ^1 - ℓ^2 optimal input with $\lambda = 0.02$ (red line), ℓ^2 optimal input with $\lambda = 0.1$ (blue dots).

化する θ を θ_1^* とおくと, それぞれ以下で与えられる.

$$\theta_2^* = \begin{bmatrix} 0.9919 \\ 4.7303 \\ 5.4459 \\ 3.9878 \\ 1.8978 \\ 0.1898 \\ -0.8228 \\ -1.2059 \\ -1.0451 \\ -0.3015 \\ 1.0642 \end{bmatrix}, \quad \theta_1^* = \begin{bmatrix} 0 \\ 0 \\ 2.0665 \\ 3.8879 \\ 3.6590 \\ 1.1288 \\ 0 \\ -1.7358 \\ -2.6657 \\ -0.3231 \\ 1.7432 \end{bmatrix}$$

ℓ^1 - ℓ^2 最適化によりスパースな解 θ_1 が得られることがわかる. これらのパラメータに対するシステムの出力を Fig. 4 に, 入力 $u(t)$ を Fig. 5 に示す. また復元誤差 $e(t) = |\sin t - y(t)|$ を Fig. 6 に示す. ℓ^1 - ℓ^2 による最適化によるスパース表現にもかかわらず, ℓ^2 最適化とほぼ同じ性能を示していることがわかる.

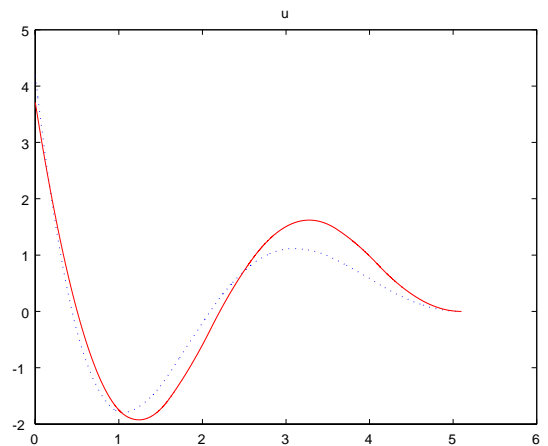


図 5: ℓ^1 - ℓ^2 optimal input $u(t)$ with $\lambda = 0.02$ (red line), ℓ^2 optimal input $u(t)$ with $\lambda = 0.1$ (blue dots).

さらに, (7) の ℓ^1 - ℓ^2 評価関数の λ を 0.05 にした場合

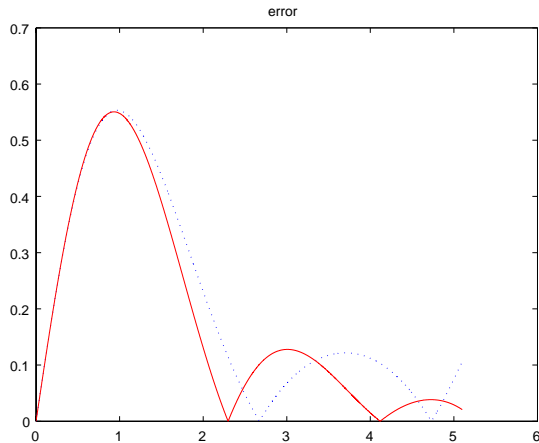


図 6: Absolute error with θ_1^* (solid) and θ_2^* (dots).

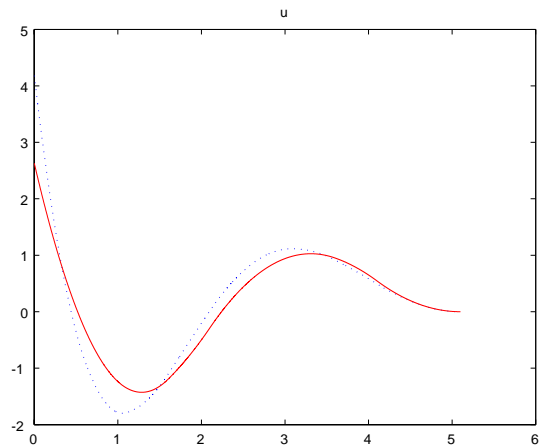


図 7: ℓ^1 - ℓ^2 optimal input $u(t)$ with $\lambda = 0.05$ (red line), ℓ^2 optimal input $u(t)$ with $\lambda = 0.1$ (blue dots).

の最適パラメータ θ_1^* は以下のようになる .

$$\theta_1^* = \begin{bmatrix} 0 \\ 0 \\ 0.3672 \\ 3.2113 \\ 3.2282 \\ 0 \\ 0 \\ -0.6546 \\ -2.2377 \\ 0 \\ 1.0940 \end{bmatrix} .$$

定数 λ の値を大きくしたことで、よりスパースな表現が得られていることがわかる . この θ_1^* を用いた入力 $u(t)$, および復元誤差をそれぞれ Fig. 7 , Fig. 6 に示す . 入力 $u(t)$ は $\lambda = 0.02$ の場合に比べて小さくなっていることがわかる . これは , ℓ^1 ノルムのペナルティをより大きくとったことに対応している . さらに , このペナルティを大きくすることにより , 復元誤差が上昇していることがわかる . この例題から , 入力信号のパラメータ θ のスパースネスと復元誤差 (性能) との間にはトレードオフの関係があることがわかる .

次に信号 θ のスパース性と情報圧縮との関係を調べるために , (1) のシステムをランダムに 1000 サンプル発生させる . ここで , システムの次数は $n = 2$ に固定する . なお , システムに対するランダムサンプルの発生には SCILAB の `ssrand` 関数を用いた . ℓ^2 最適なベクトル θ_2^* と ℓ^1 - ℓ^2 最適なベクトル θ_1^* の要素の分布を調べる . 得られた θ の確率分布を図 9 に示す . なお , ℓ^1 - ℓ^2 最適化によって得られた分布は原点 $\theta = 0$ で急峻に立ち上が

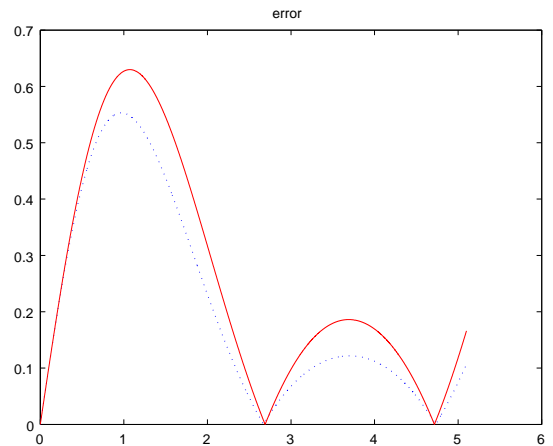


図 8: Absolute error with θ_1^* (solid) and θ_2^* (dots).

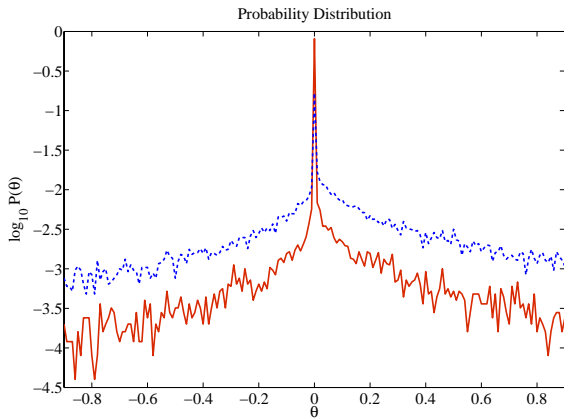


図 9: Probability distribution $\log_{10} P(\theta)$: ℓ^2 optimal (dots) and ℓ^1 - ℓ^2 optimal (solid). A logarithmic scale is used.

る特性（デルタ関数のような特性）を持つため，比較を容易にするために $\log_{10} P(\theta)$ を図示している．これより， ℓ^2 最適化によって得られた分布よりも原点付近に分布が集中しており，微分エントロピーが小さくなること予想される．実際，この分布にもとづき微分エントロピーを計算すると，

$$H_1(\theta) = 2.4885,$$

$$H_2(\theta) = 7.3817$$

となる．ただし， H_1 は ℓ^1 - ℓ^2 最適化によって得られた θ の微分エントロピー， H_2 は ℓ^2 最適化によって得られた θ の微分エントロピーである．これより， ℓ^1 - ℓ^2 最適化による微分エントロピーが ℓ^2 最適化の微分エントロピーの約 $1/3$ になっていることがわかる．したがって， ℓ^1 - ℓ^2 最適化によるスパース表現により，送信信号のより大きな情報圧縮が可能となる．

5 おわりに

本稿では，遠隔制御系の枠組みで，送信信号（最適制御入力を定義するパラメータ θ ）のデータ量を抑えるために， ℓ^1 - ℓ^2 最適化の枠組みで問題を定式化し，繰り返し縮小法により効率的に最適解を求めることができることを示した．また設計例を示し，性能をそれほど下げずにある程度のスパース表現が可能なこと，またパラメータのスパースネスと制御性能の間にトレードオフの関係があることを示した．また，スパース表現により，送信信号の微分エントロピーが小さくなることを数値例によって示し，送信信号の大きな情報圧縮が可能であることを示した．

参考文献

- [1] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sciences*, vol. 2, No. 1, 183/202 (2009)
- [2] E. J. Cnades and M. B. Wakin, An introduction to compressive sampling, *IEEE Signal Processing Magazine*, vo. 25, 21/30 (2008)
- [3] M. Egerstedt and C.F. Martin, *Control Theoretic Splines*, Princeton University Press (2010)
- [4] S. Sun, M. B. Egerstedt, and C. F. Martin, Control theoretic smoothing splines, *IEEE Trans. on Automatics Control*, Vol. 45, No. 12, 2271/2279 (2000)
- [5] 田中，圧縮センシングの数理，*IEICE Fundamental Review*, Vol. 4, No. 1, 39/47 (2010)
- [6] M. Zibulevsky and M. Elad, L1-L2 optimization in signal and image processing, *IEEE Signal Processing Magazine*, Vol. 27, 76/88 (2010)