

ネットワーク化制御のためのパケット化予測制御とスパース表現

永原 正章 (京都大学) Daniel E. Quevedo (The University of Newcastle)
Jan Østergaard (Aalborg University)

Packetized Predictive Control and Sparse Representation for Networked Control

*M. Nagahara (Kyoto University) D. E. Quevedo (The University of Newcastle)
J. Østergaard (Aalborg University)

Abstract— In this article, we introduce packetized predictive control (PPC) that is known to be robust against packet dropouts in networked control. We first characterize the control vectors (packets) that asymptotically stabilize the feedback control system. Then, we formulate the control problem as an ℓ^0 optimization to obtain the sparsest vector among the stabilizing control vectors. This problem is combinatorial optimization that is difficult to solve, and hence we adopt the OMP (Orthogonal Matching Pursuit) method. Examples show the effectiveness of the proposed methods compared with the conventional ℓ^2 optimization method. We also compare with another sparsity-promoting optimization method based on ℓ^1/ℓ^2 optimization, and discuss advantages and disadvantages of the two sparsity-promoting methods.

Key Words: Networked control, predictive control, sparse representation, compressed sensing

1 はじめに

電力ネットワークなどの制御において、ネットワーク化制御は重要な課題の一つである。ここで、ネットワーク化制御とは、無線通信などのネットワークを介した制御を指す^{10, 4, 1}。通常、ネットワークにはパケットロスや通信帯域の制限など、データ通信上のさまざまな制約が存在する。したがってネットワーク化制御では、それらの制約による安定性や制御性能への影響を考慮する必要がある。本稿では、パケットロスに対してロバストな制御手法であるパケット化予測制御⁹を紹介し、その枠組みにもとづき、通信帯域の制約に対処するための新しい安定化手法を提案する。

パケット化予測制御では、モデル予測制御⁵において計算される未来の制御入力信号系列をパケット化して送信する。送信が成功すれば、それを受信側のバッファで保存する。保存された制御入力系列は、以降、パケットロスが生じたときに使用される。この方式により、パケットロスに対してロバストなネットワーク制御系が得られる。

送信する制御入力系列は長ければ長いほど、より多くのパケットロスに対応できるが、ネットワークは帯域制限されており送信できるビットレートは限られるので、あまり長くはできない。そこで本稿では、圧縮センシング等の研究で用いられているスパース表現のテクニック^{2, 3}を導入し、送信パケットのスパース表現を試みる。この枠組みでのスパースな送信パケットは、OMP (Orthogonal Matching Pursuit) 法⁶により効率的に計算可能であることを示す。また数値例により、提案手法の有効性を示す。

記法

本稿では、以下の記法を用いる。 \mathbb{N}_0 で非負整数の集合を表す。すなわち、 $\mathbb{N}_0 := \{0, 1, 2, \dots\}$ である。 \mathbb{R}^n は n 次元実ベクトル空間であり、 $n = 1$ のときは \mathbb{R} と略記する。ベクトル $x \in \mathbb{R}^n$ と正定値対称行列 $P > 0$ に対して、 $\|x\|_P := \sqrt{x^\top P x}$ と定義する。ただし、 \top は転置を表す。特に $P = I$ (単位行列) の

ときは $\|x\|_I := \|x\|_2$ と表す。対称行列 P に対して、最小および最大固有値をそれぞれ、 $\lambda_{\min}(P)$, $\lambda_{\max}(P)$ と表す。ベクトル $x = [x_1, \dots, x_n]^\top \in \mathbb{R}^n$ に対して、 $\text{supp}\{x\}$ を x の非ゼロ要素の添え字の集合とする。すなわち、 $\text{supp}\{x\} := \{i : x_i \neq 0\}$ と定義する。

2 パケット化予測制御

本稿では、制御対象として以下の状態方程式でモデル化された線形時不変系 Σ を考える。

$$\Sigma: \quad x(k+1) = Ax(k) + Bu(k), \quad k \in \mathbb{N}_0, \quad x(0) = x_0. \quad (1)$$

ここで $x(k) \in \mathbb{R}^n$, $u(k) \in \mathbb{R}$ とし、 (A, B) は可到達であると仮定する。この制御対象 Σ に対して、パケットロスが発生する信頼性の低いネットワークを介して制御を行うことを考え、パケット化予測制御⁹を導入する。

パケット化予測制御では、まず、時刻 $k \in \mathbb{N}_0$ での制御対象 Σ の状態 $x(k)$ の観測から、(1) にしたがって、 N ステップ先までの制御入力を計算する。通常モデル予測制御⁵では、計算された N ステップ先までの制御入力のうち、一番はじめのもののみを送信するが、パケット化予測制御では、すべての制御入力を次式のようにパケット化し送信する。

$$u(x(k)) \triangleq [u_0(x(k)) \quad \dots \quad u_{N-1}(x(k))]^\top \in \mathbb{R}^N. \quad (2)$$

送信されたパケット $u(x(k))$ は受信側のバッファに記憶される (Fig. 1 を見よ)。もし次の時刻 $k+1$ で送信パケット $u(x(k+1))$ がロスしたとすると、制御対象 Σ に対する制御入力 $u(k+1)$ として、バッファに記憶された $u_1(x(k))$ を用いる。さらに時刻 $k+2$ でもパケットロスが起これば、 $u(k+2) = u_2(x(k))$ とする。パケット化予測制御のより詳細な定式化は文献⁹を参照せよ。上記の操作は N 回連続でパケットロスが起こるまで可能である。

仮定 1 本稿では、 $N+1$ 回以上連続でパケットロスは起こらないと仮定する。

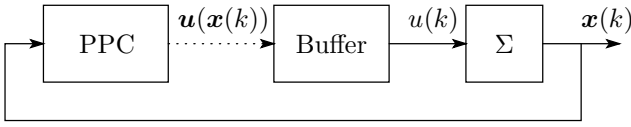


Fig. 1: パケット化予測制御のブロック線図: PPC はパケット化予測制御 (Packetized Predictive Control) を表し, 点線は通信ネットワークを表す.

3 パケットロスのもとでの漸近安定化

本節では, 仮定 1 を満たす任意のパケットロスに対して, フィードバック系が漸近安定となるような制御ベクトルを特徴付ける.

まず, 状態ベクトル $x \in \mathbb{R}^n$ が与えられたとき, 制御入力を $u = (u_0, u_1, \dots, u_{N-1})$ としたときのモデル (1) にもとづく状態の予測 $\{x_i\}_{i=0}^{N-1}$ は以下のように与えられる.

$$x_0 = x, \quad x_{i+1} = Ax_i + Bu_i, \quad i = 0, 1, \dots, N-1. \quad (3)$$

ここで, 与えられた正定値対称行列 P, Q, W に対して, 次のような集合を定義する.

$$\mathcal{F}(x) \triangleq \left\{ u \in \mathbb{R}^N : \|x_N\|_P^2 + \sum_{i=1}^{N-1} \|x_i\|_Q^2 \leq \|x\|_W^2 \right\}.$$

この集合は, N ステップ先までの状態の予測値 x_1, x_2, \dots, x_N が次の不等式

$$\|x_N\|_P^2 + \sum_{i=1}^{N-1} \|x_i\|_Q^2 \leq \|x\|_W^2$$

を満たすような制御入力 $u = (u_0, u_1, \dots, u_{N-1}) \in \mathbb{R}^N$ の集合である.

次の各行列を定義する.

$$\Phi \triangleq \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix} = \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \vdots \\ \Phi_{N-1} \end{bmatrix},$$

$$\Phi_i \triangleq [A^i B, A^{i-1} B, \dots, B, 0, \dots, 0], \quad i = 0, 1, \dots, N-1,$$

$$\Upsilon \triangleq \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \bar{Q} \triangleq \text{blockdiag}\{Q, \dots, Q, P\},$$

$$G \triangleq \bar{Q}^{1/2} \Phi, \quad H \triangleq -\bar{Q}^{1/2} \Upsilon.$$

このとき, 集合 $\mathcal{F}(x)$ は次のように表すことができる.

$$\mathcal{F}(x) = \{u \in \mathbb{R}^N : \|Gu - Hx\|_2^2 \leq \|x\|_W^2\}.$$

以上の準備のもとで, 次の定理が成り立つ.

定理 1 行列 $P > 0, Q > 0, W > 0$ を次の手順で与えるとする.

1. $Q > 0$ を任意に与える.

2. $P > 0$ を次の Riccati 方程式の解とする:

$$P = A^\top P A - \frac{A^\top P B B^\top P A}{B^\top P B} + Q. \quad (4)$$

3. 次の各定数を求める.

$$c_1 \triangleq \max_{i=0, \dots, N-1} \lambda_{\max}\{\Phi_i^\top P \Phi_i (G^\top G)^{-1}\},$$

$$\rho \triangleq 1 - \lambda_{\min}(QP^{-1}), \quad c \triangleq \frac{(1 - \rho^N)c_1}{1 - \rho},$$

4. 次の不等式を満たす行列 \mathcal{E} を与える.

$$0 < \mathcal{E} < \frac{1 - \rho}{c} P.$$

5. 行列 W を $W = P - Q + \mathcal{E}$ とする.

このとき, 任意の $x \in \mathbb{R}^n$ に対して, 集合 $\mathcal{F}(x)$ は少なくとも一つの要素を持ち, かつ各時刻 $k \in \mathbb{N}_0$ において, 送信パケットを

$$u(x(k)) \in \mathcal{F}(x(k))$$

を満たすように選べば, 仮定 1 を満たす任意のパケットロスに対して, Fig. 1 のネットワーク化制御系は漸近安定, すなわち,

$$\lim_{k \rightarrow \infty} x(k) = 0$$

を満たす.

証明: 省略. □

4 制御パケットのスパース表現

Fig. 1 のパケット化予測制御において, ネットワークを介して送信される制御パケットがスパースであれば, パケットを符号化するとき, 密なパケットよりもより効率的なデータ圧縮が可能である. そこで, 時刻 $k \in \mathbb{N}_0$ での状態の観測値 $x(k)$ にもとづいて決定される制御パケット (2) をスパースにする, すなわち, パケット $u(x(k))$ の要素 $u_i(x(k))$ がなるべく 0 となるような制御を考える. 定理 1 の条件を満たす正定値対称行列 P, Q, W をあらかじめ決めておき, 時刻 $k \in \mathbb{N}_0$ における送信パケット $u(x(k))$ を $\mathcal{F}(x(k))$ の中から選べば, フィードバック系の漸近安定性が保証される. これにもとづき, 送信パケット $u(x(k))$ として, 集合 $\mathcal{F}(x(k))$ のなかで最もスパースなものを選ぶ. すなわち, 次の最適化問題を考える.

$$u(x(k)) = \arg \min_{u \in \mathcal{F}(x(k))} \|u\|_0, \quad k \in \mathbb{N}_0. \quad (5)$$

この最適化問題は, 組み合わせ最適化であり, ベクトル u のサイズ N が大きくなると, 指数関数的に計算量は増大する⁸⁾. しかし, (5) の最適化は各時刻 $k \in \mathbb{N}_0$ においてオンラインで計算する必要があり, その計算時間はなるべく短いほうが望ましい. そこで, (5) の近似解を高速に求めることが可能な OMP (Orthogonal Matching Pursuit) 法⁶⁾を導入する. 状態ベクトル x が得られたときに, スパースな送信パケット $u(x)$ を得るための OMP アルゴリズムを Algorithm 1 に示す.

Algorithm 1 OMP for sparse control vector $\mathbf{u}(\mathbf{x})$

Require: $\mathbf{x} \in \mathbb{R}^n$ {observed state vector}
Ensure: $\mathbf{u}(\mathbf{x})$ {sparse control packet}

```

 $k := 0$ .
 $\mathbf{u}[0] := \mathbf{0}$ .
 $\mathbf{r}[0] := H\mathbf{x} - G\mathbf{u}[0] = H\mathbf{x}$ .
 $S[0] := \text{supp}\{\mathbf{x}[0]\} = \emptyset$ .
while  $\|\mathbf{r}[k]\|_2^2 > \mathbf{x}^\top W \mathbf{x}$  do
  for  $j = 1$  to  $N$  do
     $z_j := \frac{\mathbf{g}_j^\top \mathbf{r}[k]}{\|\mathbf{g}_j\|_2} = \arg \min_{z \in \mathbb{R}} \|\mathbf{g}_j z - \mathbf{r}[k]\|_2^2$ .
     $e_j := \|\mathbf{g}_j z_j - \mathbf{r}[k]\|_2^2$ .
  end for
  Find a minimizer  $j_0 \notin S[k]$  such that  $e_{j_0} \leq e_j$ , for all  $j \notin S[k]$ .
   $S[k+1] := S[k] \cup \{j_0\}$ 
   $\mathbf{u}[k+1] := \arg \min_{\text{supp}\{\mathbf{u}\}=S[k+1]} \|G\mathbf{u} - H\mathbf{x}\|_2^2$ .
   $\mathbf{r}[k+1] := H\mathbf{x} - G\mathbf{u}[k+1]$ .
   $k := k+1$ .
end while
return  $\mathbf{u}(\mathbf{x}) = \mathbf{u}[k]$ .

```

5 設計例

本節では提案法の有効性を確認するために、数値例題を示す。

5.1 制御対象が不安定な場合

制御対象として、次の状態空間表現でモデル化された連続時間線形時不変系を考える¹：

$$\dot{\mathbf{x}} = A_c \mathbf{x} + B_c \mathbf{u},$$

$$A_c = \begin{bmatrix} -1.2822 & 0 & 0.98 & 0 \\ 0 & 0 & 1 & 0 \\ -5.4293 & 0 & -1.8366 & 0 \\ -128.2 & 128.2 & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} -0.3 \\ 0 \\ -17 \\ 0 \end{bmatrix}.$$

この制御対象に対して、サンプリング周期 $T_s = 0.5$ (sec) でゼロ次ホールドにより離散化して得られる離散時間系に対して、パケット化予測制御を行う。状態予測の長さ、すなわち制御パケット (2) のサイズを $N = 20$ とする。

以上の準備のもと、Algorithm 1 で与えられる OMP 法を用いてパケット化予測制御のシミュレーションを時刻 $k = 0$ から 199 まで行う。比較のため、 ℓ^1/ℓ^2 最適化にもとづくスパース表現の方法⁷⁾ および標準的な ℓ^2 最適化による制御パケットの設計も考える。ここで、 ℓ^1/ℓ^2 最適化にもとづく方法は、次の評価関数を最小化するパケットを求める：

$$\min_{\mathbf{u} \in \mathbb{R}^N} \|G\mathbf{u} - H\mathbf{x}\|_2^2 + \nu_1 \|\mathbf{u}\|_1. \quad (6)$$

いっぽう、標準的な ℓ^2 最適化では以下の評価関数を用いる：

$$\min_{\mathbf{u} \in \mathbb{R}^N} \|G\mathbf{u} - H\mathbf{x}\|_2^2 + \nu_2 \|\mathbf{u}\|_2^2.$$

この ℓ^2 最適化の解を $\mathbf{u}_2(\mathbf{x})$ とおくと、これは、以下のように解析的に求めることができる：

$$\mathbf{u}_2(\mathbf{x}) = (\nu_2 I + G^\top G)^{-1} G^\top H \mathbf{x}. \quad (7)$$

定数 ν_1 および ν_2 は、 $\nu_1 = 500$, $\nu_2 = 1$ と定める。

シミュレーションは 100 回行った。それぞれのシミュレーションにおいて、パケットロス率は仮定 1 の条件を

¹このモデルは、文献⁵⁾の 2.7 節の例題 (Cessna Citation 500 aircraft) で与えられたものである。

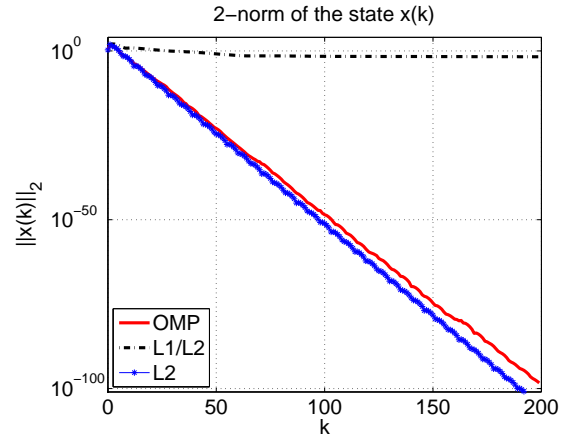


Fig. 2: 状態 $\mathbf{x}(k)$ のノルムの平均値

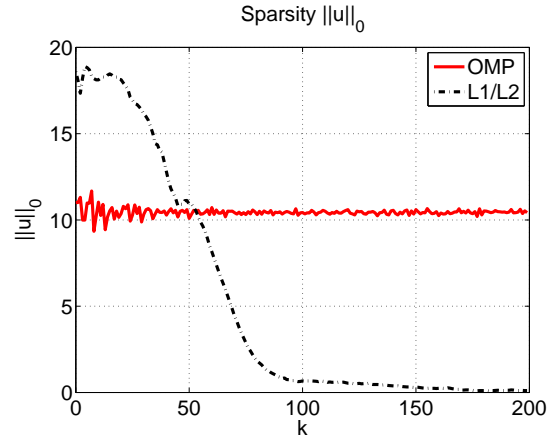


Fig. 3: パケットのスパースさ $\|\mathbf{u}(\mathbf{x}(k))\|_0$ の平均値

満たすようにランダムに生成し、また制御対象の初期状態 $\mathbf{x}(0) = \mathbf{x}_0$ は、その要素を平均 0、分散 1 の正規分布から独立にランダムサンプルして選ぶ。

Fig. 2 に時刻 k における状態 $\mathbf{x}(k)$ のノルム $\|\mathbf{x}(k)\|_2$ の平均値を示す。この図より、OMP 法および ℓ^2 最適化による制御パケットは漸近安定性を示しているが、 ℓ^1/ℓ^2 最適化による制御パケットによる制御では状態のノルムがある値以下にはならない。これらの結果は、本稿の定理 1 および文献⁷⁾で示された ℓ^1/ℓ^2 最適化による安定性定理の結果と一致している。特に、提案法である OMP 法による制御と従来法である ℓ^2 最適化による制御で、制御性能はほぼ互角であることが Fig. 2 よりわかる。

Fig. 3 に OMP 法および ℓ^1/ℓ^2 最適化による制御パケットのスパースさ、すなわち ℓ^0 ノルムの平均値を示す。なお、 ℓ^2 最適化による制御パケットは常に密なベクトル、すなわち ℓ^0 ノルムが $N = 20$ となる。この図より、提案手法では、ほぼすべての時間にわたって約 50% のスパースさ (すなわち、約 50% の要素が 0) を得ていることがわかる。いっぽう、 ℓ^1/ℓ^2 最適化では、はじめの段階ではあまりスパースとは言えないが、時間がたつにつれスパース性が増していき、最終的にかなりのスパース性が得られていることがわかる。すなわち、 ℓ^1/ℓ^2 最適化では、漸近安定性は成り立たないものの、提案法の OMP 法による制御ベクトルよりもはるかにスパースなベクトルが得られていることがわかる。

次にそれぞれの方法による最適制御ベクトルの計算

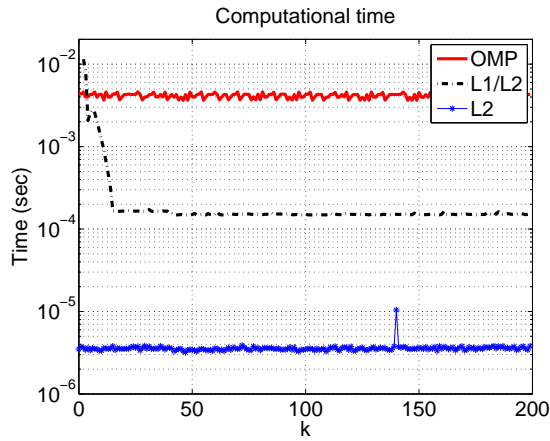


Fig. 4: 計算時間の平均値

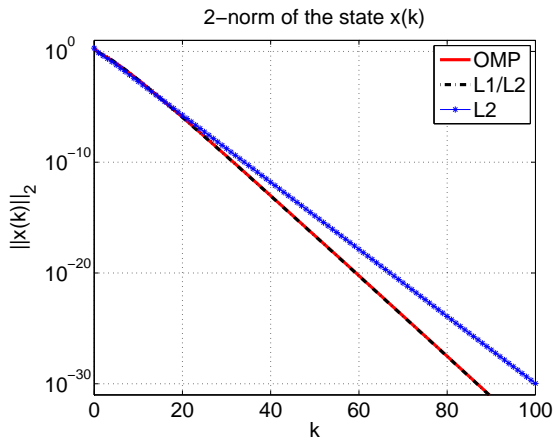


Fig. 5: 状態 $x(k)$ のノルムの平均値

時間の平均値を Fig. 4 に示す． ℓ^1/ℓ^2 最適化のほうが OMP 法よりもかなり早く計算できることがわかる．なお， ℓ^2 最適化の計算は，(7) で与えられる公式により，行列 $(\nu_2 I + G^T G)^{-1} G^T H$ をオフラインであらかじめ計算しておいて，それを状態の観測ベクトル $x(k)$ に掛けるだけなので，非常に高速に計算できる．また，今の例題ではサンプリング周期は 0.5 (sec) であり，最適化計算にかかる時間は，サンプリング周期よりも十分短いことがわかる．

5.2 制御対象が安定な場合

次に制御対象が安定である場合を考える．次の離散時間の状態空間行列 A , B を考える²:

$$A = \begin{bmatrix} 0.0211 & 0 & 0 & 0 \\ 0.1062 & 0.4266 & 0 & 0 \\ 0 & 0 & 0.2837 & 0 \\ 0.1012 & -0.6688 & 0.2893 & 0.4266 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.6462 \\ 0.2800 \\ 1.5237 \\ 0.9929 \end{bmatrix}.$$

制御パケットのサイズを $N = 30$ とする．比較のために (6) の ℓ^1/ℓ^2 最適化 ($\nu_1 = 1$)，および (7) の ℓ^2 最適化 ($\nu_2 = 0.6$) を用いる．Fig. 5 に状態 $x(k)$ のノルムの平均値を示す．安定な制御対象に対しては， ℓ^1/ℓ^2 最適化による制御法でも漸近安定を達成できていることがわかる．Fig. 6 に提案法および ℓ^1/ℓ^2 最適化による制御パケットのスパースさ $\|u(x(k))\|_0$ の平均値を示す．

²このモデルは，文献⁵⁾の Example 2.4 (Paper machine head-box) で与えられたものである．

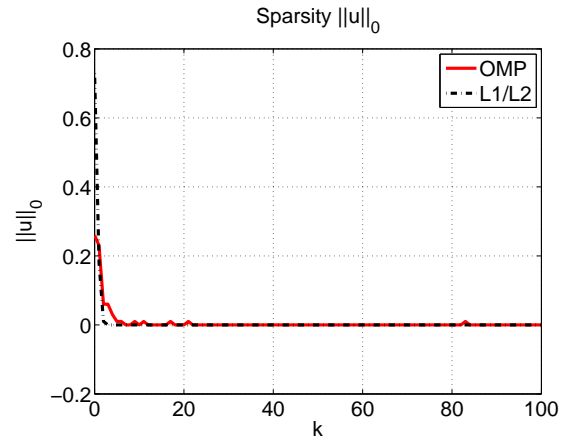


Fig. 6: パケットのスパースさ $\|u(x(k))\|_0$ の平均値

OMP 法， ℓ^1/ℓ^2 最適化法ともに，ある時刻以降，ほとんどすべての制御パケットが 0 となっていることがわかる．これは，制御対象が安定であれば，制御入力がかゼロでも状態は 0 に漸近するからである．

6 おわりに

本稿では，パケットロスに対してロバストな制御手法であるパケット化予測制御を紹介し，その枠組みにもとづき，通信帯域の制約に対処するために，制御ベクトルのスパース表現を提案した．フィードバック系を漸近安定にする制御ベクトルを特徴付け，そのなかで最もスパースなベクトルを求める ℓ^0 最適化として設計問題を定式化した．この問題はそのままでは組み合わせ最適化であり，その近似解を求めるための高速解法として OMP 法を導入した．数値例により従来法である ℓ^2 最適化と比較した提案手法の有効性を示し，さらに OMP 法および ℓ^1/ℓ^2 最適化の利点と欠点を明らかにした．

参考文献

- 1) A. Bemporad, M. Heemels, and M. Johansson, *Networked Control Systems*, Springer, 2010.
- 2) M. Elad, *Sparse and Redundant Representations*, Springer, 2010.
- 3) Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.
- 4) J. Hespanha, P. Naghshtabrizi, and Y. Xu, A survey of recent results in networked control systems, *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
- 5) J. M. Maciejowski, *Predictive Control with Constraints*, Pearson Education, 2002.
- 6) S. G. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries, *IEEE Trans. Signal Process.*, Vol. 41, No. 12, pp. 3397–3415, 1993.
- 7) M. Nagahara and D. E. Quevedo, Sparse representations for packetized predictive networked control, *IFAC world congress*, pp. 84–89, 2011.
- 8) B. K. Natarajan, Sparse approximate solutions to linear systems, *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.
- 9) D. E. Quevedo and D. Nešić, Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts, *IEEE Trans. Automat. Contr.*, Vol. 56, No. 2, pp. 370–375, 2011.
- 10) W. Zhang, M. Branicky, and S. Phillips, Stability of networked control systems, *IEEE Control Syst. Mag.*, vol. 21, no. 1, pp. 84–99, 2001.