A compensated method for Networked Control System with packet drops based on Compressed Sensing

Ruifeng Fan[1,a], Xunhe Yin[2,b,*] and Huaqing Liang[3,c]
[1,2,3] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[a]16120197@bjtu.edu.cn, [b]xhyin@bjtu.edu.cn, [c]17111049@bjtu.edu.cn
*Corresponding author

Keywords: Networked Control System, Compressed Sensing, Packet drops.

Abstract. Due to unreliable and bandwidth-limited characteristics of communication link in networked control systems, a real-time compensated method for multi-output system is presented in this paper based on Compressed Sensing and sliding window, by which estimates of dropping data packets in feedback channel is obtained and the performance degradation induced by packet drops is reduced. The merits of the compensated method are that it can compensate the dropping entire data packet and make the measurement matrix always keep instant in the whole process of Compressed Sensing which reduces the memory requirements of the buffer. Finally, simulation results illustrate that the proposed method could perform well when packet-drop occurs.

1. Introduction

Networked control systems (NCS) are those in which the controller is not co-located with the sensors and the actuators, especially the measurement information and command information should be made through rate-limited communication channels such as wireless networks or the Internet. In NCS, packet drops will inevitably affect the performance of the system, so efficient strategies for handling packet drops to maintain system performance are essential in unreliable communication channels. In this paper, the main aim is to find a compensated method based on Compressed Sensing (CS) to process the packet drops. CS is an emerging field which is better than the traditional signal processing methods. It has been considered in many applications including wireless sensor network, speech processing, WAMS, communication, imaging, etc.

However, only a few studies used CS in the packet-drop processing literatures. [1] uses the CS technique as a new predictive method to compensate the missing data, but its proposed algorithm only can estimate the missing data of single-output system. In the actual engineering environment, most systems have multiple sensors to get more outputs, that is, multi-output system. Under such circumstances, the method proposed by [1] cannot be used again. [2] also introduces the CS method to process the packet drops. In [2], the authors propose that the original complete data can be compressed into a small amount of data. Then the small amount of compressed data need to be only transmitted through the unreliable network, and [2] points out that the signals can be reconstructed using the residual data even if some data is dropped in the small amount of data. One obvious fact is that [2] uses one single packet transfer in NCS. However, the actual situation is that it is the entire packet-drop for the one single packet transfer when the packet-drop happens at most of the time. The phenomenon in [2] that only part of the single packet-drop is possible but very unlikely. Therefore, there is an urgent need to develop advanced techniques based on CS to support the entire packet-drop. In this paper, in order to preferably compensate the dropping entire data packet in NCS which has multiple outputs, a sequential CS framework based on sliding window motivated by [2,3,4] is introduced to effectively compensate the dropping packet.

2. Proposed scheme

2.1 Sliding data window

Let $\boldsymbol{X}_{(k)} \in R^{N\times W}$ denotes a data window at time instant k with window size $W \geq 1$. It consists of W consecutive readings of all N sensors at time instants $\{k-W+1,\cdots,k\}$ as [5]

$$\boldsymbol{X}_{(k)} = \begin{bmatrix} x_1(k-W+1) & \cdots & x_1(k) \\ \vdots & \ddots & \vdots \\ x_N(k-W+1) & \cdots & x_N(k) \end{bmatrix} \tag{1}$$

where $x_i(k)$ is the reading of sensor $i=1,2,\cdots,N$ at time instant k which is the signal that needs to be transmitted in the unreliable network at each sampling instant. Let $\tilde{\boldsymbol{x}}_i^T(k) \triangleq \left[x_i(k-W+1) \ \cdots \ x_i(k)\right]$ denotes the i^{th} row of $\boldsymbol{X}_{(k)}$, which contains the readings of the i^{th} sensor at time instants $\{k-W+1,\cdots,k\}$. Let $\hat{\boldsymbol{x}}(k) \triangleq \left[x_1(k) \ \cdots \ x_N(k)\right]^T$ denotes the k^{th} column of $\boldsymbol{X}_{(k)}$, which contains the readings of all sensors at time instant k.

Assume that there exists a basis $\boldsymbol{\Psi_S} \in R^{N\times N}$ for the spatial domain in which each column of $\boldsymbol{X}_{(k)}$ has a compressible representation, i.e., $\hat{\boldsymbol{x}}(k)=\boldsymbol{\Psi_S\theta_S}(k)$, where $\boldsymbol{\theta_S}(k)\in R^{N\times 1}$ contains the spatial transform coefficients at time instant k. Then $\boldsymbol{X}_{(k)}$ can be written as

$$\begin{aligned} \boldsymbol{X}_{(k)} &= \left[\hat{\boldsymbol{x}}(k-W+1) \ \cdots \ \hat{\boldsymbol{x}}(k)\right] \\ &= \boldsymbol{\Psi_S}\left[\boldsymbol{\theta_S}(k-W+1) \ \cdots \ \boldsymbol{\theta_S}(k)\right] = \boldsymbol{\Psi_S\Theta_S}(k) \end{aligned}. \tag{2}$$

Similarly, there exists a temporal domain basis $\boldsymbol{\Psi_T} \in R^{W\times W}$ in which each row of $\boldsymbol{X}_{(k)}$ has a compressible representation, i.e., $\tilde{\boldsymbol{x}}_i(k)=\boldsymbol{\Psi_T\theta_{T,\mathit{i}}}(k)$, where $\boldsymbol{\theta_{T,\mathit{i}}}(k)$ contains the temporal transform coefficients of the i^{th} sensor. Hence, $\boldsymbol{X}_{(k)}$ can be expressed as

$$\begin{aligned} \boldsymbol{X}_{(k)} &= \left[\tilde{\boldsymbol{x}}_1(k) \ \cdots \ \tilde{\boldsymbol{x}}_N(k)\right]^T \\ &= \left[\boldsymbol{\theta_{T,\mathrm{1}}}(k) \ \cdots \ \boldsymbol{\theta_{T,\mathit{N}}}(k)\right]^{\mathrm{T}} \boldsymbol{\Psi_T}^T = \boldsymbol{\Theta_T}^T(k)\boldsymbol{\Psi_T}^T \end{aligned}. \tag{3}$$

Kronecker sparsifying basis can succinctly combine the individual sparsifying bases of each signal dimension into a single transformation matrix. Thus, the transformations in (2) and (3) can be merged. In order to put the readings of a single sensor in the sliding window together, $\boldsymbol{X}_{(k)}$ can be converted to $\bar{\boldsymbol{X}}_{(k)}=\mathrm{vec}\left(\boldsymbol{X}^T{}_{(k)}\right)$, where vec denotes the vectorization of a matrix which converts the matrix into a column vector. So $\bar{\boldsymbol{X}}_{(k)}$ is a column vector composed of readings of N consecutive sensors and each sensor includes readings of W consecutive time instants. On the basis of Kronecker sparsifying basis, $\bar{\boldsymbol{X}}_{(k)}$ can be represented as

$$\begin{aligned} \bar{\boldsymbol{X}}_{(k)} &= \mathrm{vec}\left(\boldsymbol{X}^T{}_{(k)}\right) = \mathrm{vec}\left(\boldsymbol{\Psi_T\Theta_T}(k)\right) \\ &= \mathrm{vec}\left(\boldsymbol{\Psi_T Z}_{(k)}\boldsymbol{\Psi_S}^T\right) = \left(\boldsymbol{\Psi_S}\otimes\boldsymbol{\Psi_T}\right)\mathrm{vec}\left(\boldsymbol{Z}_{(k)}\right) = \boldsymbol{\Psi}\bar{\boldsymbol{Z}}_{(k)} \end{aligned} \tag{4}$$

where $\boldsymbol{\Psi} = \boldsymbol{\Psi}_S \otimes \boldsymbol{\Psi}_T \in R^{NW \times NW}$ is the Kronecker sparsifying basis, and $\bar{\boldsymbol{Z}}_{(k)} = \mathrm{vec}\left(\boldsymbol{Z}_{(k)}\right) = \mathrm{vec}\left(\boldsymbol{\Theta}_T(k)\boldsymbol{\Psi}_S{}^{-T}\right)$ contains the joint transformation domain coefficients. For the choice of $\boldsymbol{\Psi}_S$ and $\boldsymbol{\Psi}_T$, we can choose them as Fourier transformation basis, discrete cosine transformation basis or wavelet basis, and so on.

2.2 Compensated method

When the data packet is lost, the encoding process of CS is considered. For each sensor, the continuous W data in the sliding window is used as a complete signal; the consecutive $W-1$ data in the sliding window before this time instant of packet-drop is regarded as compression of the signal, which is called the measurement vector $\boldsymbol{y}_i(k)$. Obviously, the dimension of the measurement matrix $\boldsymbol{\Phi}_i(k)$ for each compression is $(W-1)\times W$. Based on this, the losing data of the N sensors are processed. The measurement signal of the window $\bar{\boldsymbol{X}}_{(k)}$ at the time instant k is expressed as

$$\begin{bmatrix} \boldsymbol{y}_1(k) \\ \vdots \\ \boldsymbol{y}_N(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1(k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\Phi}_N(k) \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}}_1(k) \\ \vdots \\ \tilde{\boldsymbol{x}}_N(k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_1(k) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\Phi}_N(k) \end{bmatrix} \bar{\boldsymbol{X}}_{(k)} \tag{5}$$

where the measurement signal of the window $\bar{\boldsymbol{X}}_{(k)}$ is expressed as $\boldsymbol{y}(k) = \begin{bmatrix} \boldsymbol{y}_1{}^T(k) & \cdots & \boldsymbol{y}_N{}^T(k) \end{bmatrix}^T \in R^{N(W-1)\times 1}$ and the block-diagonal measurement matrix is expressed as $\boldsymbol{\Phi}(k) = \mathrm{diag}\left(\boldsymbol{\Phi}_1(k),\cdots,\boldsymbol{\Phi}_N(k)\right) \in R^{N(W-1)\times NW}$. Accordingly, then the encoding process of the window $\bar{\boldsymbol{X}}_{(k)}$ at time instant k in (5) can be compactly written as $\boldsymbol{y}(k) = \boldsymbol{\Phi}(k)\bar{\boldsymbol{X}}_{(k)}$.

In order to obtain effective measurement information, a particular sparse structure of $\boldsymbol{\Phi}_i(k)$ is considered [5]: all its entries are zeros, except for a single "1" in each row, and at most a single "1" in each column. It is necessary to pay attention to the measurement matrix $\boldsymbol{\Phi}_i(k)$, which needs to be kept completely consistent in the encoder and decoder of CS. Since the measurement matrix at each sampling instant all may be different in the process of compression in [2], the current measurement matrix needs to be buffered at each sampling instant, which inadvertently increases the pressure of the buffer's storage space. How to make the decoder know that what is the specific measurement matrix of the encoder is also a big problem at each sampling instant. These problems are all not considered in [2]. Here the measurement matrix $\boldsymbol{\Phi}(k)$ under each window is consistent, which would not cause too much pressure on the other aspects of the system.

By exploiting the joint spatio-temporal compressibility, the decoding process of CS is considered. The actual situation that the signals extracted from the real-world sensing device would have more or less noise is considered. Then the Basis Pursuit De Noising (BPDN) algorithm is employed due to its robustness to noise property. Assume the signals have bounded errors of the form $\|e\|_2 \le \varepsilon$, where ε is a small non-zero constant. Each data window $\bar{\boldsymbol{X}}_{(k)}$ can be recovered from measurements (5) by solving the BPDN algorithm

$$\bar{\boldsymbol{Z}}'_{(\mathrm{k})} = \arg\min \left\|\bar{\boldsymbol{Z}}_{(k)}\right\|_1 \qquad \text{subject to } \left\|\boldsymbol{\Phi}(k)\boldsymbol{\Psi}\bar{\boldsymbol{Z}}_{(k)} - \boldsymbol{y}(k)\right\|_2 \le \varepsilon \tag{6}$$

and $\bar{\boldsymbol{X}}_{(k)} = \boldsymbol{\Psi}\bar{\boldsymbol{Z}}'_{(k)}$. Then the losing data packet at the time instant k can be gotten as $\hat{\boldsymbol{x}}(k) = \begin{bmatrix} \bar{\boldsymbol{X}}_{(k)}(W) & \bar{\boldsymbol{X}}_{(k)}(2W) & \cdots & \bar{\boldsymbol{X}}_{(k)}(NW) \end{bmatrix}^T$.

For the case of continuous packet-drop, according to the principle of the sliding window, when the packet-drop continues to occur at the time instant $(k+1)$, the $\hat{\boldsymbol{x}}(k-W+1)$ in the sliding window $\boldsymbol{X}_{(k)}$ is removed and the compensation data calculated at the time instant k is added to the buffer. On the basis of this, the losing data is compensated again using the method described above.

3. Simulation study

In order to validate the effectiveness of the proposed real-time CS based compensated method, the following continuous-time plant and its state feedback controller are considered

$$\dot{x}=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\-1&-3&-2&-4\end{bmatrix}x+\begin{bmatrix}0\\1\\1\\1\end{bmatrix}u,\quad y=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}x,$$

$$K=-[2.8618 \quad 3.9677 \quad 1.0461 \quad -0.0138].$$

Set the sampling period as 0.1s, $N=4$, $W=4$. The spatial and temporal sparsifying bases are both given by the discrete cosine transform (DCT) to generate a DCT matrix. It is simulated by the approximate network environment TrueTime toolbox of MATLAB.

In the case of the packet dropping probability P equals to 0, the state response curve of the system is shown in Fig. 1. When the packet dropping probability equals to 10%, the state response curve of the system is shown in Fig. 2, which can be seen the states begin to oscillate but soon converge to zero with the increase of time. When the packet dropping probability is 50%, the state response curve of the system is shown in Fig. 3 and can be seen that the oscillations of the states are relatively large, but in the end they also approach zero to reach steady states. When the packet dropping probability is 70%, the state response curve of the system is shown in Fig. 4, which can be seen that the states oscillate violently and the adjustment time is obviously increased, but this system is still stable. When the packet dropping probability is 80%, the system is no longer stable. If this compensated method is not applied to deal with the missing packets, the number of data packets that the system can tolerate at most is 30%, that is, if the packet continues to be lost, the system would not be stable. We can see that the performance degradation induced by packet-drop is reduced preferably by the proposed compensated method.

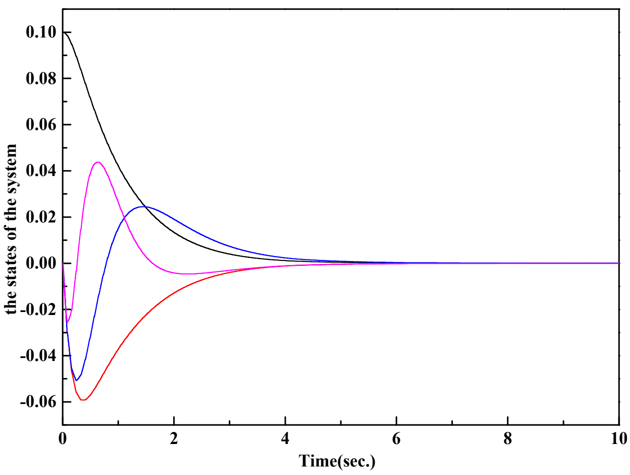

Fig. 1. The state response curve of the system at $P=0$

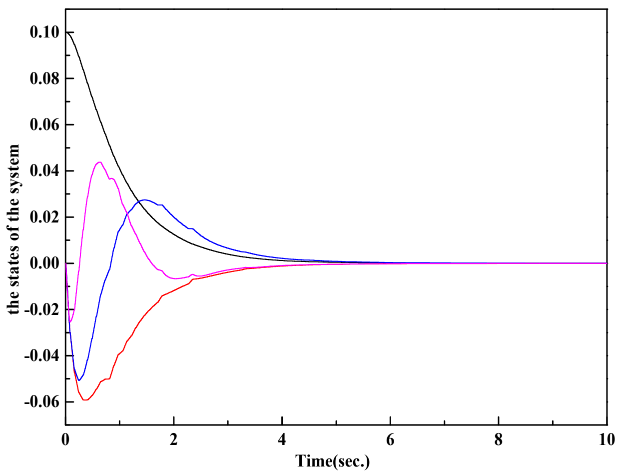

Fig. 2. The state response curve of the system at $P=10\%$

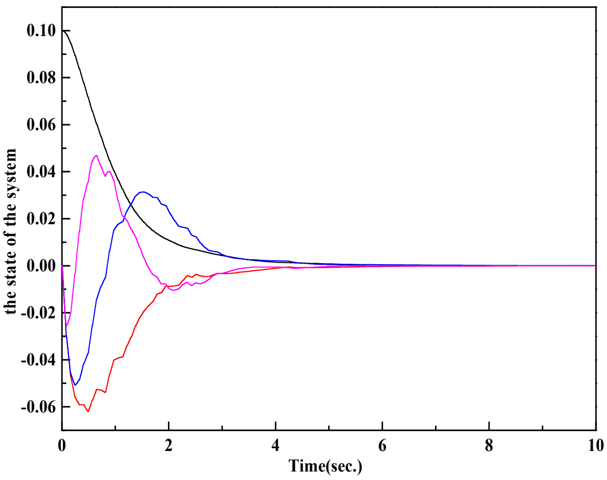

Fig. 3. The state response curve of the system at P =50%

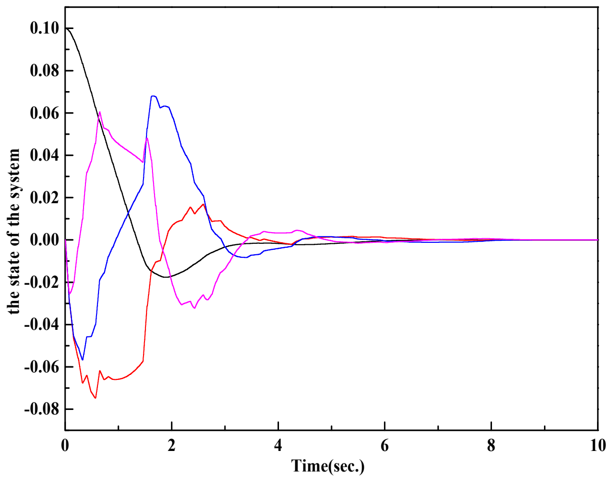

Fig. 4. The state response curve of the system at P =70%

4. Summary

In this paper, considering the feedback channel of NCS which has multiple outputs, a real-time method based on CS and sliding window is proposed to compensate the dropping packet, which reduces the performance degradation induced by packet-drop. It compensates the dropping entire data packet in real time, and does not increase the amount of data transmitted, which satisfies the limited bandwidth requirements of NCS. Especially, the new compensated method makes the measurement matrix always keep instant in the encoder and decoder, which satisfies the requirement of CS. Compared to the other compensated method, it effectively reduces the memory requirements of the buffer. Simulation results demonstrate that the proposed CS-based scheme shows a good action when the packet-drop occurs. In the future work, we will investigate on more complex packets dropping situation such as packets drop when multiple packets are transmitted.

References

[1] S. Perepu and A. Tangirala, Classical PID control in presence of missing data using compressed sensing techniques, *11th Global Congr. Process Safety*, pp. 74-78, 2010.

[2] Z. Li, Y. Xu, and H. Huang, Sparse control and compressed sensing in networked switched systems, *IET Control Theory Appl.,* vol. 10, pp. 1078-1087, 2016.

[3] H. Tang, Y. Xu, and Z. Li, Compressed sensing based real-time control in a smart grid, *IEEE Conf. Power Energy Eng.*, pp. 1782-1786, 2016.

[4] Y. Xu, Z. Yang, and J. Zhang, Real-time compressive sensing based control strategy for a multi-area power system, *IEEE Trans. Smart Grid*, vol. 8, pp. 1-1, 2017.

[5] M. Leinonen, M. Codreanu, and M. Juntti, Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks, *IEEE Trans. Wireless Communications*, vol. 14, pp. 1622-1635, 2015.