

The Study of Packet Loss Effect on Network Control System Error Function Model

M. Jahanshahi^a, S. M. Houshyar^b, A. R. Zare Bidaki^c

^a*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*

^b*Department of Electrical Engineering Science & Research Branch, Islamic Azad University, Tehran, Iran.*

^c*Young Researchers and Elite Club, Buinzahra Branch, Islamic Azad University Buinzahra, Iran.*

Abstract. Modern control systems widely use network to decrease the implementation cost and also increase the performance. Although they have several advantages, they suffer from some limitations and deficiencies. Packet loss is one of the main limitations which affect the control system in different conditions and finally can lead to system instability. To prevent such problems it is important to model the system properly. In this paper, a new function has been proposed for error modeling. Moreover, different influences of system parameters and packet loss on error function have been investigated and it has been determined that changes of which parameter has more contribution on packet effect in error function. Considering these parameters is very critical in designing systems.

Received: 13 August 2013; Revised: 20 October 2013; Accepted: 17 November 2013

Keywords: Network control system (NCS), Packet Loss, Transport layer, Bernoulli binary, Seed, system model.

1. **Introduction**
2. **NCS Phenomenon**
3. **Network System**
4. **Study Packet Loss Parameters Effects in NCS Output Error**
5. **Conclusion**

1. Introduction

With the extension of the control system scale and fast development of network technologies, control systems based on control networks are widely applied in the modern industry control, which have advantages of less wires, high reliability, good flexibility and share of information resources, etc. However, since the communication network has been

*Corresponding author. Email: mohsenjahanshahi@gmail.com

inserted in the networked control system, some inherent characteristics of network such as limited bandwidth, carrying capacity and service ability make data transmission possibly suffer time delay, multi-packet transmission and data dropouts [13].

Today's technological advances in wireless communications and in the fabrication of inexpensive embedded electronic devices, are creating a new paradigm where a large number of systems are interconnected, thus providing an unprecedented opportunity for totally new distributed control applications, commonly referred as networked control system [13].

Network-induced time delay and data dropouts are the main problems and often exist at the same time in the networked control system. However, in the mostly current research results, only one problem is considered and study considering the two problems at the same time is scarce.

Examples of NCSs are available in manufacturing systems, intelligent vehicle highway systems, teleoperation of robots, aircraft systems, etc. Compared with conventional point-to-point control systems NCSs have the advantages of low installation cost, reduced wiring, easy maintenance and diagnosis, and so on. In NCSs, however, network-induced delays and data packet dropout may be inevitable during transmission of digital data between control devices. Recently, the effect of data packet dropout on the stability and performance of NCSs has received great attention.

2.NCSPhenomenon

2-1. Packet Loss

One of the most common problems in networked control systems, especially in wireless sensor networks, is packet loss, i.e. packets can be lost due to communication noise, interference, or congestion. If the controller is not co-located with the sensors and the actuators and it is placed in a remote location, then both sensor measurement packets and control packets can be lost [13].

In the past decade, much attention has been paid to the impact of the delayed data packets of NCSs. In fact, the transmission packet dropouts are also the potential source of instability and poor performance in NCSs because of the critical real-time requirement in control systems. Therefore the impact of packet dropouts is an important aspect in the analysis and synthesis of NCSs and this issue has received wide attention recently [12].

2-2. Packet loss models

In general, in most of the literature two different strategies are considered for dealing with packet drops. In the first one, which we refer as zero-input, the actuator input to the plant is set to zero when the control packet from the controller to the actuator is lost, while in the second, which we refer as hold-input, the latest control input stored in the actuator buffer is used when a packet is lost. These are not the only strategies that can be adopted. In fact, if smart actuators are available, i.e. if actuators are provided with computational resources, then the whole controller or a compensation filter can be placed on the actuator [13]. Another strategy is to use a model predictive controller which sends not only the current input but also a finite window of future control inputs into a single packet so that if a packet

is lost the actuator can pop up from its buffer the corresponding predicted input from the latest received packet. Although the zero-input strategy in many situations it performs better than the hold-input strategy, thus encouraging further investigation in experimental settings and justifying its use in networked control systems.

NCS with short time delay can be modeled as a linear switching model with certain switching rules and the periodic dynamic output feedback controller [13]. The packet transmission sequence of sensor nodes is non-deterministic. Therefore, the packet transmission process is stochastic and satisfies the Markov characteristic. Figure 1 present two type of system that in Figure 2 compare their responsibility to random lossy input.

Packet loss can exist in two way, sensor to controller and controller to actuators. The best model is a system that model loss in two way independently.

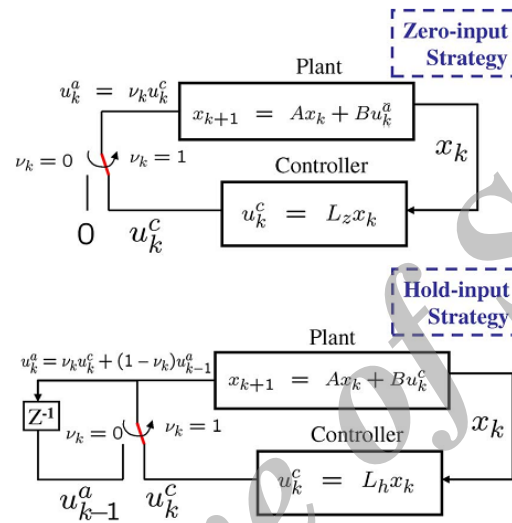


Figure 1: Compensation approaches for actuators with no computational resources when a control packet is lost: zero-input approach (top) and hold-input approach (bottom) [13].

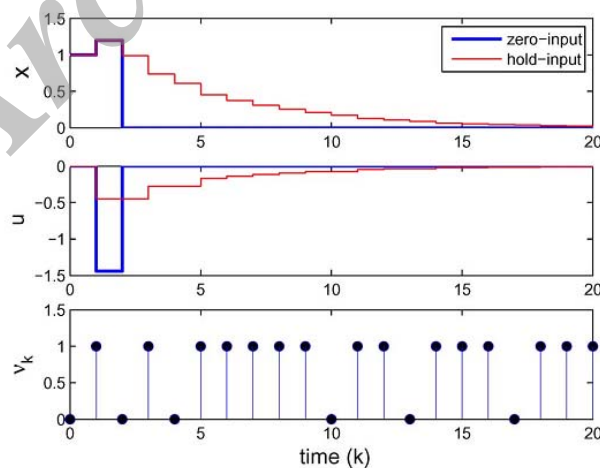


Figure 2: Specific realization under optimal zero-input control and optimal hold-input control [13].

2-3.Mathematics lost models

In the previous literature, the models of packet dropouts in NCSs can be roughly categorized into two types. The first type of model assumes that the packet dropouts follow certain probability distributions and describes NCSs with packet dropouts via stochastic models, such as Markov chains or binary switching sequences. Using the Markov chains to represent random packet dropout models for the NCSs, a few results have been developed recently [4-6]. Another stochastic model approach is to view the packet dropouts as the Bernoulli binary distributed white sequence taking the values of 0 and 1 with certain probability. Recently, there have been some control methods presented on such a model [11-13].

A Markov chain (discrete-time Markov chain or DTMC) named after Andrei Markov, is a mathematical system that undergoes transitions from one state to another, between a finite or countable number of possible states. It is a random process usually characterized as memory less: the next state depends only on the current state and not on the sequence of events that preceded it. This specific kind of "memorylessness" is called the Markov property. Markov chains have many applications as statistical models of real-world processes [10].

In probability and statistics, a Bernoulli process is a finite or infinite sequence of binary random variables, so it is a discrete-time stochastic process that takes only two values, canonically 0 and 1. The component Bernoulli variables X_i are identical and independent. Prosaically, a Bernoulli process is a repeated coin flipping, possibly with an unfair coin (but with consistent unfairness). Every variable X_i in the sequence is associated with a Bernoulli trial or experiment. They all have the same Bernoulli distribution. Much of what can be said about the Bernoulli process can also be generalized to more than two outcomes (such as the process for a six-sided die); this generalization is known as the Bernoulli scheme.

So if loss occurs according to bandwidth limitation or small packet size, Bernoulli binary is a good method for loss modeling.

3.NetworkSystem

3-1.System elements behavior

All sensor and actuators can behave in two state. Clock driven and event driven is two main state of elements behavior. Time driven elements are work in constant time that work with sample time. But Event driven element wait for signal to act. In most NCS, sensors and actuators act clock driven because packet loss in these system can cause big fault for event driven elements.

3-2.Network layer

OSI (Open Systems Interconnection) is a standard description or "reference model" for how messages should be transmitted between any two points in a telecommunication network. Its purpose is to guide product implementers so that their products will

consistently work with other products. The reference model defines seven layers of functions that take place at each end of a communication. Although OSI is not always strictly adhered to in terms of keeping related functions together in a well-defined layer, many if not most products involved in telecommunication make an attempt to describe themselves in relation to the OSI model. It is also valuable as a single reference view of communication that furnishes everyone a common ground for education and discussion. OSI divides telecommunication into seven layers. The layers are in two groups. The upper four layers are used whenever a message passes from or to a user. The lower three layers (up to the network layer) are used when any message passes through the host. Messages intended for this host pass to the upper layers. Messages destined for some other host are not passed up to the upper layers but are forwarded to another host. The seven layers are:

Layer 7: The application layer...This is the layer at which communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. (This layer is not the application itself, although some applications may perform application layer functions.)

Layer 6: The presentation layer...This is a layer, usually part of an operating system, that converts incoming and outgoing data from one presentation format to another (for example, from a text stream into a popup window with the newly arrived text). Sometimes called the syntax layer.

Layer 5: The session layer...This layer sets up, coordinates, and terminates conversations, exchanges, and dialogs between the applications at each end. It deals with session and connection coordination.

Layer 4: The transport layer...This layer manages the end-to-end control (for example, determining whether all packets have arrived) and error-checking. It ensures complete data transfer.

In many control systems this four layers merge together and assume as one layer.

Layer 3: The network layer...This layer handles the routing of the data (sending it in the right direction to the right destination on outgoing transmissions and receiving incoming transmissions at the packet level). The network layer does routing and forwarding.

Layer 2: The data-link layer...This layer provides synchronization for the physical level and does bit-stuffing for strings of 1's in excess of 5. It furnishes transmission protocol knowledge and management.

Layer 1: The physical layer...This layer conveys the bit stream through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier.

3-3. 4th layer

The transport layer provides transparent transfer of data between end users, providing reliable data transfer services to the upper layers. The transport layer controls the reliability of a given link through flow control, segmentation/desegmentation, and error control. Some protocols are state- and connection-oriented. This means that the transport layer can keep track of the segments and retransmit those that fail. The transport layer also provides the acknowledgement of the successful data transmission and sends the next data if no

errors occurred. This mean this layer can prevent loss occurring. The main protocols in this layer are TCP and UDP. In brief, TCP protocol assure that packet receive to end user successfully. This protocol is not suitable for NCS because Network Control Systems speed is so high and packet amount is large in time. So in these system, the other protocol can be used and it is UDP. UDP uses a simple transmission model with a minimum of protocol mechanism. It has no handshaking dialogues, and thus exposes any unreliability of the underlying network protocol to the user's program. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system.

4. Study Packet Loss Parameters Effects in NCS Output Error

4-1. self-robustness

All systems have self-robustness in addition to occurring uncertainty in their dynamics. Packet loss in NCS is an uncertainty. Therefor the system must have self-robustness against it. In this paper a simple system is provided which has self-robustness against packet loss up to 70%. Figure 3 illustrates this system. As the result shows system is stable despite the performance of system responses is very bad and unclear [11-13].

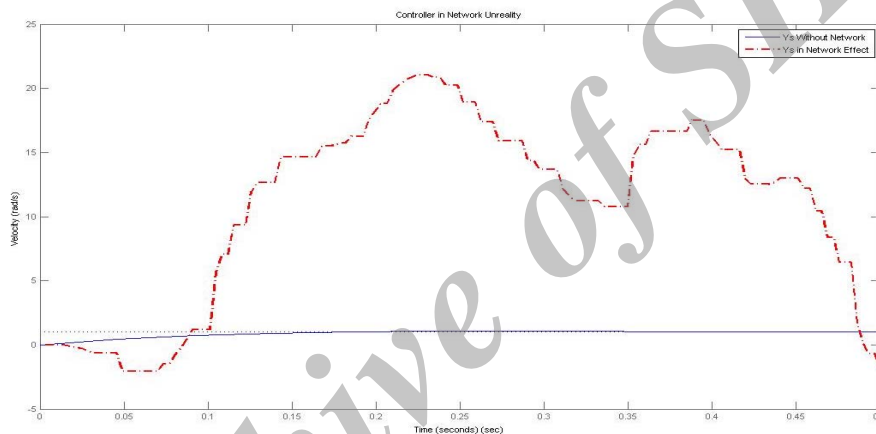


Figure 3: Self-robustness against packet loss

4-2. Packet loss parameters

In this paper the effect of random packet loss for a class of NCSs has been investigated which their sensors, controllers, and actuators are clock driven. Both the S/C and C/A packet loss are modeled by two mutually independent stochastic variables that satisfied Bernoulli binary distribution. And 4th layer protocol is UDP.

Main assumption:

- a. System elements like sensors and actuators are clock driven.
- b. M& N Memory block save last packet. So system is memorable.
- c. System input is A, and output is B. the difference between B and real system output is error.
- d. System is linear time invariant.

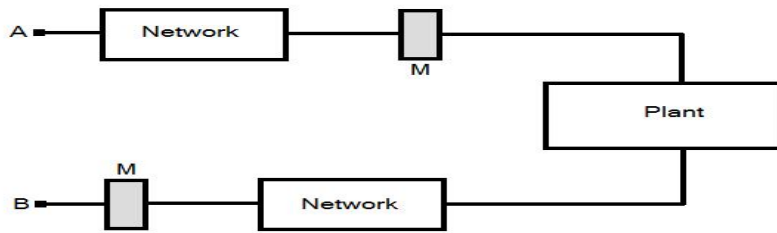


Figure 4: NCS model

Error function in this system can be defined as below:

$$J(\alpha) = \sum_k (|Y(k)_{Ideal_Network} - Y(k)_{Lossy_Network}|^2) \quad 1$$

Now, the role of parameters can be investigated. The two main parameters which have the main role in occurring packet loss are Bernoulli binary seed generator and system frequencies. The other parameter according to error function is modeling length.

4-2-1. Seeds

Seed parameter is Bernoulli vector order generator. As a matter of fact, in Bernoulli binary vector there are 1 and 0 that manipulate to link data and allow packet to pass or discard. If the result of this manipulation equals to zero it means that loss occurred. The order of 1 and 0 is characterized by seed number [2]. Different seeds with constant loss percent is shown in the table below:

A1=[1	1	1	1	1	1	1	1	1	1]
A2=[1	1	1	1	1	1	1	1	1	1]
A3=[1	1	1	0	0	0	1	1	1	0]
A4=[1	1	1	1	1	1	0	0	0	1]
A5=[0	0	0	1	1	1	0	0	0	1]

Figure 5: show seed effect in error function

Error function increases according to loss percent increasing, although seed effect in some special value has a large difference with similar seeds in constant loss presents.

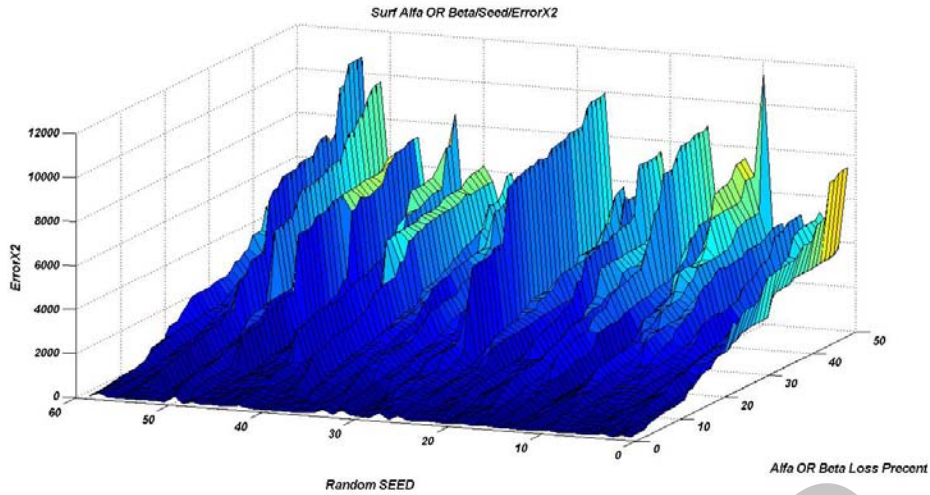


Figure 5: Seed effect in error function

4-2-2. Frequency

The other main parameter which is able to have a big role in error function is frequency. Different simulations show that error has direct relation with frequency that means if frequency increases, error will be increased too. It is reasonable because signals information in high frequency will be narrow. It means that a signal information in high frequency system exists in a few packet in comparison with low frequency system. Figure 6 present frequency effect on error function:

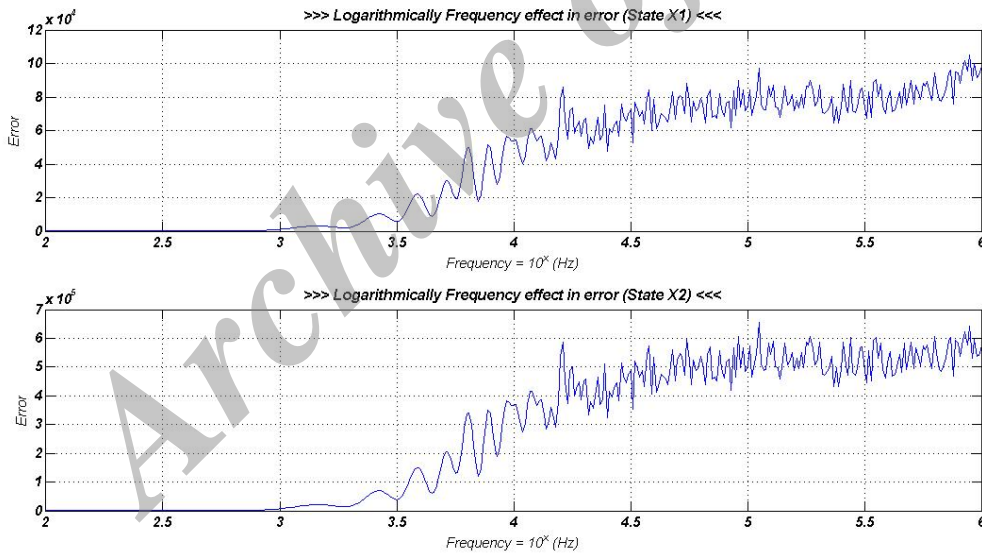


Figure 6: Frequency effect on error function

4-2-3. Packet loss percent

Other main parameter which packet loss will be defined with is loss percent. According to increasing loss percent, error increasing is tangible. But NCS simulation shows unexpected result. There is dents in some area of error with loss percent surface. This research cannot give a suitable reason for this phenomenon and need to research

separately. Figure 7 shows percent surface with attention to dent in high packet loss percent.

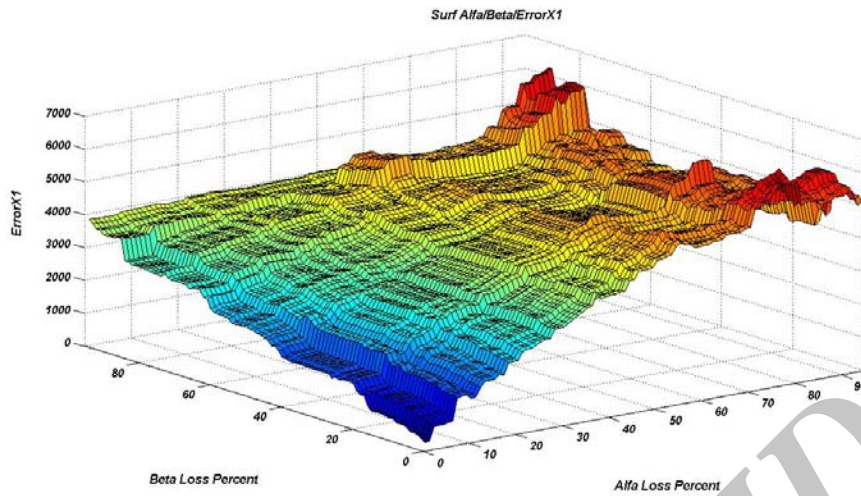


Figure 7: Loss percent effect on error with attention to dent in surface

4-2-4. Simulation length

The last point is error function that deeply depends on simulation length. In this article the relation between seed and simulation length has been investigated. Figure 8 shows that different seeds with increasing length make a strictly increasing surface that changes in specific bound.

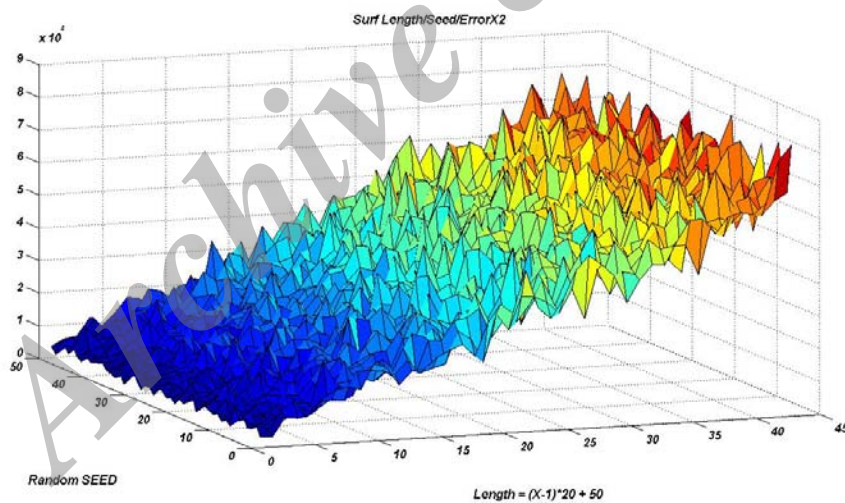


Figure 8: Seed with increasing length effects on error

5. Conclusion

Network control Systems have been used in wide ranges of industrial applications. As a result, studying and analyzing these systems are very vital. To perform a better system analyzes, it is necessary to propose an appropriate model of system. Generally, with network system layering it is possible to partially analyze the performance of system.

Moreover, packet loss is another important factor in network systems. In this research, the maximum effects of packet loss were occurred in fourth layer. By analyzing various parameters of system, it has been concluded that input frequency of system and possibility percentage of occurring packet loss are the most important factors that can lead to instability in system.

Acknowledgment

The authors gratefully acknowledge the financial and other support of this research, provided by Central Tehran Branch, Islamic Azad University, Tehran, Iran.

References

- [1] Huiying C., Wanliang W., Zuxin L., Synthetic Modeling and Control of Networked Control Systems with Multi-Packet Transmission, *Advanced Comp. Control*, (2012) 661-664.
- [2] Heemels W.P.M.H., Teel A.R., van de Wouw N, Networked Control Systems with Communication Constraints Tradeoffs between Transmission Intervals Delays and Performance, *IEEE Trans, Autom. Control*, **55** (8)(2010)1781-1796.
- [3] Lun S., Wang S, Robust H_∞ cost control for Networked control systems based on T-S model, *IEEE Trans, Net. Sens. Control*, (2010)447-451.
- [4] Ling Q., Lemmon M.D., Soft real-time scheduling of networked control systems with dropouts governed by a Markov chain. *Proc. American Control Conf.*,**6** (2003) 4845–4850.
- [5] Seiler P., Sengupta R., An H approach to networked control, *IEEE Trans. Autom. Control*, **50** (3)(2005) 356–364.
- [6] Schenato L., To Zero or to Hold Control Inputs With Lossy Links, *IEEE Trans Autom. Control*, **54**(5) (2009) 1093-1099.
- [7] Sanaiee A. M, Habibi J., Lucas C, Intrusion detection using a fuzzy genetics-based learning algorithm, *Net. Comp Appli*,**30**(1) (2007) 414-428.
- [8] Wang Z., Yang F., Ho D.W.C., Liu X. ,Robust H_1 control for networked systems with random packet losses, *IEEE Trans. Syst. Man Cybern. B, Cybern.*,**37** (2007)916–924.
- [9] Wu J., Chen T, Design of networked control systems with packet dropouts”, *IEEE Trans. Autom. Control*, **52**(2007) 1314–1319.
- [10] Wang Y, Yang G , H_∞ Controller Design for Networked Control Systems via Active-varying Sampling Period Method, *Auto sinica*,**34** (7) (2008) 414-418.
- [11] Wang Z., Ho D.W.C., Liu X ,Variance-constrained control for uncertain stochastic systems with missing measurements, *IEEE Trans. Syst. ManCybern. A, Syst. Humans*, **35** (5) (2005) 746–753.
- [12] Xiaosheng F., Jingcheng W., Stochastic Observer-based Guaranteed Cost Control for Networked Control Systems with Packet Dropouts, *Automatic Control* ,**9** (11) (2008) 980-989.
- [13] Yang F., Wang Z.D., Ho W.C., GaniM ,Robust H control with missing measurements and time-delays, *IEEE Trans. Autom. Control*, **52** (2007)1666–1672.
- [14] ZhivoglyadovP. V, Middleton R. H, Networked control design for linear system, *Automatica*, **39** (4) (2003) 743-750.