Front. Mech. Eng. China 2008, 3(3): 299–306
DOI 10.1007/s11465-008-0049-7

RESEARCH ARTICLE

**Shihua WANG, Bugong XU, YunHui LIU, Yeming ZHOU**

# Real-time mobile robot teleoperation via Internet based on predictive control



**Abstract** A remote control system that can control a mobile robot in real time via the internet is proposed. To compensate for the network delay and counteract its impact on the teleoperation system, a predictive control scheme based on the modified Smith predictor proposed is selected. To ensure the stability and transparency of the system, a dynamic model manager is designed based on the information exchange between the sensors at the master and slave sides. To precisely predict the time delay, a new timer synchronization algorithm is proposed. To decrease delay- jitter, a new data buffer scheme is performed. Force feedback and a virtual predictive display are introduced to enhance the real-time efficiency of teleoperation. The usefulness and effectiveness of the proposed method and system are proven by teleoperation experiments via the internet over a long distance.

**Keywords** teleoperation, internet robots, predictive control, delay algorithm

## 1 Introduction

In recent years, with the rapid development of information technology, the use of the internet for remote control of robots has been one of the hottest topics in robotics and automation. Goldberg et al. [1] set up the earliest teleoperation system in the Mercury project in 1994, which enables users to control a manipulator with a camera through an internet browser. Many telerobotics systems have accomplished some tasks through web browsers or remote client software [2–9].

For bilateral teleoperation systems, their stability and transparency are significantly destroyed due to time delay in signal transfer through communication channels [10,11]. Moreover, for the systems via the internet, time delay is variable because the network structure is dynamic and data flows are unbalanced. In recent years, main research on conquering time delay has been classified into three subjects: a control scheme, IP packets transmission control protocol, and the freedom of a slave robot. Control schemes, except predictive control schemes, are difficult to compensate for time delays to realize a real-time operating system. An event-based control scheme and its system stability, transparency, and event synchronization caused by time delay were first proposed and discussed in Refs. [12,13]. The control scheme can keep the system stable by selecting events as its control paramrters; however, the system real-time operating capability is significantly damaged. All the improvements based on the IP packets transmission control protocol include modifying the TCP data transmitted algorithm [14], adding an IP packet buffer scheme [15,16], and increasing the priority of a transmission program to a protocol handler task [17], which can only slightly reduce time delay or delay jitter. As to the freedom of slave robots, increasing freedom can ensure the security of robots in avoiding obstacles, seeking goals, and tracking paths at a task programming level [18–21]. However, the transparency and real-time operating capacity of the system are strongly deteriorated.

From the analysis above, and on the premise of ensuring stability, the predictive control scheme is better for improving the system transparency in a real-time control state. In 1959, Smith proposed the well-known Smith predictor (SP) structure based on models to deal with a linear system with constant time delay [22]. Many teleoperation systems have been successfully applied with the SP control scheme [23]. In recent years, teleoperation systems with

Translated from *Robot*, 2007, 29(4): 305–312 [译自: 机器人]

Shihua WANG (✉), Bugong XU
Laboratory of Real-Time Control Through Internet and Fieldbuses, College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China
E-mail: wsh188881@163.com

Shihua WANG, Yeming ZHOU
Naval Arms Command Academy of Guangzhou, Guangzhou 510430, China

YunHui LIU
Department of Automation and Computer Aided Engineering, The Chinese University of Hong Kong, Shatin, N.T., HK, China

the SP control scheme have been classified into two types: 1) model prediction, i.e., the output of the slave model directly feeds back to controller at the master side and operators feel the output through a force feedback device such as a joystick; 2) vision prediction, i.e., the state of the slave side feeds back to the virtual display at the master side and determines the next command based on the predictive display operators. For a force-reflecting teleoperation system, Andreu et al [16] designed an SP predictive control scheme over an IP network based on force prediction. In the same way, a time-delay adaptive control was added to force prediction based on extended temporal correlations characteristics of packet traffic [24]. Casavola et al [25] designed a command governor (CG) in model predictive strategy to keep predictive system state errors within an allowable bound. It is obvious that the inactive model of ''slave robot + environment'' can be calibrated in advance. If this model is provided with the virtual mobile robot at the master side, operators can telecontrol the robot well without real vision feedback [26,27]. Similarly, Wang et al [20] proposed an inactive model together with an active virtual robot to accomplish the interactive control through an IP network. Itoh et al [28] developed a superimposed display for motion prediction and force information from the predictive and real images of the slave site without any virtual model of environment. Kheddar [29] realized a ''hidden robot'' based on virtual reality technology and used it to control the real robot with a predictive display.

The use of model output to control the remote robot can make a system robust on tracking errors. However, the operator perceives the system mainly based on touch and a video, and the transparency and tele-presence cannot thus be enhanced. The control scheme with vision prediction can not only help operators in task programming, real-time supervising, control evaluation, and other functions, but also accordingly enhance system transparency and real-time control capability. By using the advantages of both, this paper presents the real-time mobile tele-robot system based on a predictive control scheme derived from our modified SP. Correspondingly, several delay algorithms are proposed.

## 2 System architecture and models

### 2.1 System architecture

As for a slave environment, immovable buildings or inactive obstacles can be calibrated on a map ahead of the schedule, and then form a static model. The movable subjects can be detected by sensors of the mobile robot in its workaround, and then form an active model. The slave model at the master side is composed of static and active models and can be displayed by the dynamic model display (DMD) in virtual reality. The virtual display is updated by a dynamic model manager (DMM). Operators generate commands based on the slave model with the SP predictive architecture. To overcome the delay jitter and transfer the variable delay to a constant, the buffer scheme is implemented on both sides. The implementing and programming structure of the real-time mobile system is shown in Fig. 1. The real image feedback only enhances the tele-presence of operators. When it is turned off, operators can also successfully telecontrol the mobile robot.

### 2.2 System models

1) Joystick: The joystick is simplified as a mass-spring system whose dynamics is as follows:

$$M_{\mathrm{m}}\ddot{X}(t)+K_{\mathrm{m}}X(t)=F_{\mathrm{d}}(t)-F_{\mathrm{m}}(t), \tag{1}$$

where $M_m$ is the mass of the joystick; $K_m$ is the spring constant; $X(t)=[X_{\mathrm{X}},X_{\mathrm{Y}}]^{\mathrm{T}}$ denotes the displacement of the joystick from the centre; $F_{\mathrm{d}}$ is the force applied by a operator; and $F_{\mathrm{m}}$ is the reflective force.

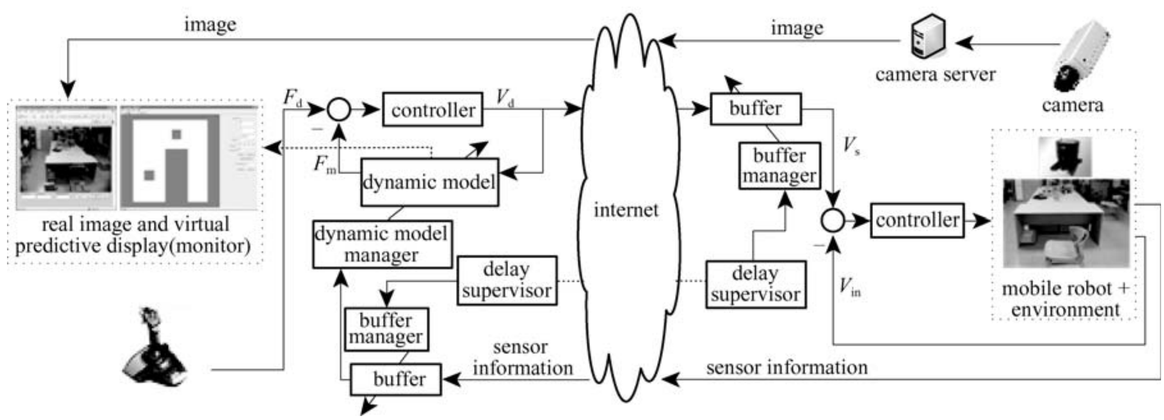


**Fig. 1** Real-time mobile robot teleoperation architecture

2) Model of a mobile robot: The mobile robot is an omnidirectional robot that can freely rotate and translate without rotation-translation coupling. The mobile robot is driven by the velocity commands in a closed-loop.

The desired velocity generated by the joystick can be expressed as follows:

$$V_{\mathrm{d}}=K_{\mathrm{p}}X(t), \tag{2}$$

where $V_{\mathrm{d}}$ denotes the desired velocity; and $K_{\mathrm{p}}$ denotes the transfer ratio of position-velocity.

3) Slave environment: Slave environment models can be classified into two parts: static and active models, whose parameters such as size, position, and velocity can be transferred to digital information. With the Jacobian matrix, the Cartesian velocity of the mobile robot is derived as follows:

$$\dot{\boldsymbol{p}}=J(q)\dot{q}, \tag{3}$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}\cos\theta & 0\\ \sin\theta & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}v\\ \omega\end{bmatrix}, \tag{4}$$

where $\boldsymbol{p}=[x,y,\theta]^{\mathrm{T}}$ represents the robot state; and $v$ and $\omega$ denote the translation and rotation velocities, respectively. Integrating Eq. (3), the position of the robot is as follows:

$$\begin{bmatrix}x\\ y\\ \theta\end{bmatrix}=\begin{bmatrix}x_0\\ y_0\\ \theta_0\end{bmatrix}+\begin{bmatrix}\int v(t)\cos(\theta(t))\mathrm{d}t\\ \int v(t)\sin(\theta(t))\mathrm{d}t\\ \int \omega(t)\mathrm{d}t\end{bmatrix}. \tag{5}$$

When the mobile robot meets obstacles, its seniors will transfer the information of the obstacles to the master side, which can be converted to a repulsive force as follows:

$$\begin{aligned}&F_{\mathrm{m}}(t)=-K_{\mathrm{f}}\cdot r-K_{\mathrm{v}}f(\theta)\cdot(|V_{\mathrm{d}}(t)|-|V_{\mathrm{r}}(t)|),\\ &f(\theta)=\{\cos\theta,\ \sin\theta\}\ \text{as}\ X,Y\ \text{axis}.\end{aligned} \tag{6}$$

where $K_{\mathrm{f}}$ and $K_{\mathrm{v}}$ are the distance and velocity gains, respectively; $V_{\mathrm{r}}$ is the velocity of the movable subjects; and $r$ is a variable depending on the distance between the mobile robot and those subjects or obstacles:

$$r=\begin{cases}0, & D\geqslant 100,\\ 0.5, & 50\leqslant D<100,\\ 0.7, & 10\leqslant D<50,\\ 1, & D<10,\end{cases} \tag{7}$$

where $D$ is the distance mentioned above. The above equations are used to update the joystick force feedback along the directions of $X$ and $Y$.

# 3 Predictive control scheme

## 3.1 Smith predictor principle

The Smith predictor shown in Fig. 2 is proposed to compensate a dead-time system. Its transfer function is as follows:

$$Y_{\mathrm{u}}(s)=\frac{G_{\mathrm{c}}G_0\mathrm{e}^{-\tau_0 s}}{1+G_{\mathrm{c}}G_{\mathrm{m}}+G_{\mathrm{c}}(G_0\mathrm{e}^{-\tau_0 s}-G_{\mathrm{m}}\mathrm{e}^{-\tau_{\mathrm{m}}s})}, \tag{8}$$

where $G_0$ is the real process. When $G_{\mathrm{m}}=G_0$, the predictive delay $\tau_{\mathrm{m}}$ is thus equal to the real delay $\tau_0$, which obviously denotes that the system is delay-free.

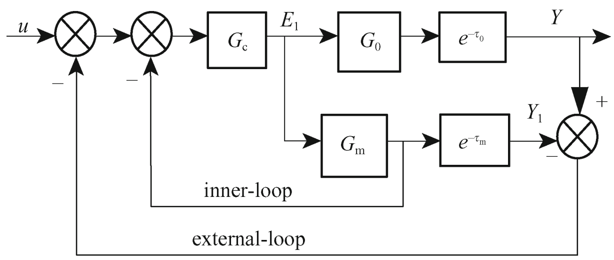


**Fig. 2** Classical SP structure

The above systems with the traditional SP structure can be divided into two closed loops: an inner-loop and external-loop. Derived from the dual-diagram method with the Nyquist contour, the system is sensitive to system uncertainty and variation of time delay. When predictive and real delay increase, the system gain ($K$) significantly decreases and the system becomes extremely unstable [30]. Therefore, as shown in Fig. 3, a modified SP proposed in Ref. [30] is selected to overcome the impact of uncertainty with variable time delay on the teleoperation system.

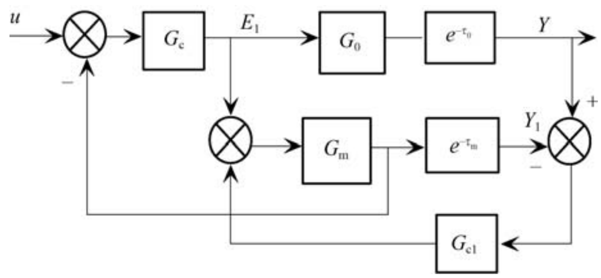


**Fig. 3** Modified SP structure

As shown in Fig. 3, the system real output feeds back not to the controller input but indirectly to the model input, which makes the system model not only coincide with the real process, but also makes the uncertainty robust with variable time delay. This method is extended to the proposed real-time mobile robot teleoperation. Based on the modified SP, the real system output, i.e., the output from the mobile robot and environment, indirectly feeds back to the master model to ensure that the model equals the slave model during the information exchange process between the two sides by DMM.

### 3.2 Force feedback prediction

Force feedback can enhance the sense of telepresence. Due to the time delay occurring in the communication channel, the image or force feedback coming from the slave side cannot represent the robot and its environment in real time. Thus, the force feedback comes from the system model at the master side. To strengthen the system robustness on the variable delay of force feedback, one method is to predict force feedback of the real mobile robot by using the following form:

$$\hat{F}_s(t) = F(t|F_\mathrm{m}(t-\tau_\mathrm{b}), V_\mathrm{s}(t-\tau_\mathrm{b}), V_\mathrm{d}(t-\tau_\mathrm{f})), \quad (9)$$

where $\hat{F}_\mathrm{s}(t)$ is the predictive force; $\tau_\mathrm{f}$ and $\tau_\mathrm{b}$ describe the forward and back network delay, respectively; and $F(\cdot)$ are the functions of the velocities at both sides and the force at the master side. The force ($F$) feeds back to the master side to compare with the force generated by the slave model next time. If the difference is large, the force generated by the slave model will be modified under a fuzzy controller until the errors are within an allowable bound.

### 3.3 Model display

The slave model consists of the mobile robot and its environment. The environment model is calibrated before the schedule so that it is constant. Movable subjects or active obstacles cannot be calibrated, so that their positions, velocity, and other properties/characteristics are thus changeable. For those moving objects, when their movement is limited in the work space of the mobile robot, their information can be perceived by the robot through sensors. The perceived information will be transmitted to DMM over an IP network. Once DMM receives the information, it will calculate the position, velocity, and acceleration with predictive network delay and then judge whether these objects have existed in the master model. If they have not been in the master model, DMM will add them into the model, otherwise, DMM will compare them with their previous information stored in the model and determine whether the model in the master side should be modified based on the fuzzy rules. After DMM has completed the above tasks, the model display changes and the force feedback to operators is calculated according to Eq. (6). The position of the mobile robot can be calculated exactly based on its initial position and velocity according to Eq. (5). The position of the mobile robot in the model is periodically updated at a fixed sampling time if no DMM commands are generated. Due to the dynamic network delay, there are position errors between the master model and the real position at the slave side. When the errors reach a threshold value, DMM will decide whether or not to modify the model display and regulate the force feedback with predictive delay under the fuzzy controller until the errors are within an allowable bound.

## 4 Delay process

The network delay regulator (NDR) shown in Fig. 4 is designed to ensure that the delay is constant. It is clear that both the master and slave sides can have their own sampling times at a fixed time with a proper stack size $S$ under the NDR principle. $T_\mathrm{m}$ and $T_\mathrm{s}$ are the sampling times at the master and slave sides, respectively. The fixed sampling time must achieve network delay $\tau$ to regulate the buffer length. Therefore, three algorithms are designed as follows.

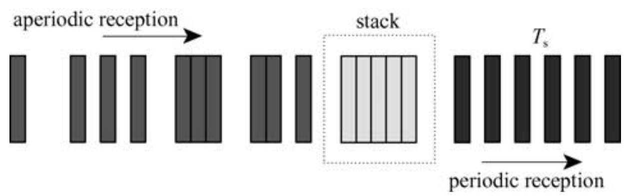


**Fig. 4** NDR principle

### 4.1 Timer synchronization algorithm

Recently, network delay has been obtained with two methods. 1) An ICMP time request in an IP layer is sent with a "ping" function. However, the network delay obtained by ICMP cannot reflect the real delay of TCP flows. 2) Datagrams with a time tag are sent and the network delay can then be obtained by calculating the time difference between the sending and receiving times. This method requires time synchronization in advance. Although a network timer protocol (NTP) can synchronize the timer, it is difficult to connect the special time servers in the internet. Thus, a new method for timer synchronization is proposed and the steps are as follows.

1) Synchronization. A datagram is sent from the master side to the slave side with the time tag, $\hat{t}_\mathrm{s} = t_\mathrm{m} + \Delta t$, where $t_\mathrm{m}$ is the current system time of the master side. $\Delta t = \hat{\tau}_\mathrm{f}$ denotes the predictive network forward delay and its initial value $\Delta t = 0$. The slave side sends the datagram back as soon as it is received, and $\hat{t}_\mathrm{s}$ and its corresponding real system time $t'_\mathrm{s}$ are saved.

2) Correction. Once the master side receives the datagram back from the slave side, then $\Delta t_1 = t_1 - \hat{t}_s$, where $t_1$ denotes the current system time. If $\Delta t_1 < 0$, then $\Delta t_1 = 0$. If $|\Delta t_1 - \Delta t| \geqslant \xi$, where $\xi$ is the time errors threshold, return to the first step, otherwise go to the next step.

3) Confirmation. An accomplished timer synchronization symbol sends from the master side to the slave side

and the slave side then confirms the master side with its synchronized time $\hat{t}_s$ and $t'_s$.

This algorithm is absolutely convergent. The total time errors obtained at both sides is only 1 ms. By adding the errors threshold $\xi$, the total errors in timer synchronization can be controlled less than $(\xi+1)$ ms.

### 4.2 Delay prediction algorithm

After the timer synchronization, network delay for an IP datagram at the slave side can be calculated as follows:

$$\tau_n = \hat{t}_s(n) - t_m(n), \tag{10}$$

where $t_m(n)$ is the $n$th IP datagram transmitting time at the master side; and $\hat{t}_s(n)$ is the slave side time corresponding to the master side when the $n$th IP datagram arrives at the slave side. If $t_s(n)$ denotes the system time of the slave side, $\hat{t}_s(n)$ can be calculated as follows:

$$\hat{t}_s(n) = \hat{t}_s + \left(t_s(n) - t'_s\right). \tag{11}$$

The delay prediction algorithm is proposed as follows:

$$\begin{aligned} &\hat{\tau}_{n+1} = \tau_n + \beta_n(\tau_n - \tau_{n-1}),\\ &\hat{\beta}_{n+1} = (\tau_{n+1} - \tau_n)/(\tau_n + T_m),\\ &\text{If } \hat{\tau}_{n+1} > \tau_{\max}, \text{ then } \hat{\tau}_{n+1} = \tau_{\max},\\ &\text{If } \hat{\tau}_{n+1} < \tau_{\min}, \text{ then } \hat{\tau}_{n+1} = \tau_{\min}, \end{aligned} \tag{12}$$

where $\tau_{\max}$ and $\tau_{\min}$ are the maximum and minimum delay, respectively; $\hat{\tau}_{n+1}$ is the predictive delay; and $\beta_n$ are the dependent coefficients that are initialized as:

$$\hat{\tau}_2 = \tau_1 + \beta_0(\tau_1 - \tau_0), \text{ where } \beta_0 = 0 \text{ and } \tau_0 = 0. \tag{13}$$

### 4.3 Buffer length regulator algorithm

Using a long buffer length, $S$ may artificially increase delay in the communication channel while a short $S$ may lead the slave side to not receive commands in a sampling time. In Ref. [15], the buffer length is simply evaluated as $S = 6\sigma/T$ with $\sigma = 2T$. However, the added delay reaches $6\sigma$ with this method, and the system will receive no commands when the delay variation reaches $3\sigma$. Reference [16] lets $S = RTT_{\max}/RTT_{\text{mean}} + 1$ to regulate the buffer length, where $RTT$ is the abbreviation for "round trip time". The algorithm is obviously impossible. The sampling time is mainly dependent on the generating rate of commands. The system will receive no commands once the network delay reaches 2 $RTT_{\text{mean}}$. As shown in Fig. 4, based on our previous algorithms, a buffer length regulator algorithm is proposed as follows:

$$\begin{aligned} &\text{If } S > b_2 \text{ then } T_s(n) = T_m - |\tau_{\text{mean}} - \hat{\tau}_{n+1}|,\\ &\text{where if } T_s(n) < T_m - \Delta T \text{ then } T_s(n) = T_m - \Delta T.\\ &\text{If } b_1 \leqslant S \leqslant b_2 \text{ then } T_s(n) = T_m.\\ &\text{If } S < b_1 \text{ then } T_s(n) = T_m + |\tau_{\text{mean}} - \hat{\tau}_{n+1}|,\\ &\text{where if } T_s(n) > T_m + \Delta T \text{ then } T_s(n) = T_m + \Delta T. \end{aligned} \tag{14}$$

$\tau_{\text{mean}}$ denotes the mean delay; $T_s(n)$ is the $n$th IP datagram sampling time; $b_1$ and $b_2$ represent the thresholds of the buffer length, and $b_1 < b_2$; and $\Delta T$ is the maximum variation of the sampling time.

## 5 Experimental Results

To verify the feasibility and evaluate the performance of the developed system, experiments were performed in Guangzhou and Chengdu at the same time via the internet, which took the TCP/IP protocol. The real time delay and the predictive delay of the network for one trip are shown in Fig. 5. It can be seen that the predictive delay tracks the real delay well and even perfectly when the network real delay jitters are minimal. When the real delay suddenly changes, larger errors occur due to the progres-

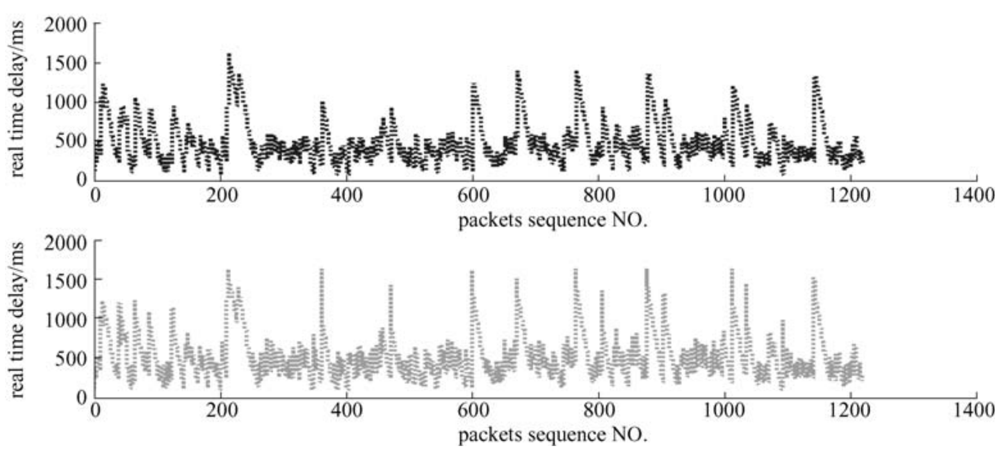


**Fig. 5** Real network time delay and its predictive time delay

siveness of the proposed algorithm. It should be noted that the errors rapidly drop back and close to zero when the real delay becomes slightly smooth. Figure 6 shows the mean network time delay and the buffer sampling time. Clearly, for real TCP data the mean delay flows vary slightly for a long time. According to Fig. 6 drawn for the slave side, the buffer length regulator algorithm works well. The sampling time error between the slave and master sides is limited in $\pm$ 10 ms when the master buffer sampling time is 40 ms, and the sampling time at the slave side can be kept constant for most time. Moreover, we have successfully finished timer synchronization at different places over the internet 1000 times. The longest time to complete this task is less than 3 s when $\xi$ is set as 2 ms.

The first and second columns in Fig. 7 show the designed and actual positions in the $x$ and $y$ directions, respectively. The actual path tracks the designed path well with some delayed sampling times, which represent the actual network delay. Figure 8 shows that the reflective force rapidly changes when the mobile robot encounters obstacles. The first change denotes the static obstacle and the second one denotes the movable obstacle. The first and second columns in Fig. 9 show the designed and actual positions without virtual predictive display, respectively. The force reflected to operators without virtual predictive display is shown in Fig. 10.

The comparison of the experimental results with (Figs. 7 and 8) and without (Figs. 9 and 10) the virtual display shows that the former are better. Without the virtual predictive display, operators control the mobile robot mainly according to a "look-move" scheme. At the same time, both in the $x$ and $y$ directions (Fig. 9), many points are unchangeable and the reflective force (Fig. 10) is often equal to zero. To accomplish the same task, the time consumed without the virtual predictive display is about 4 times that with the virtual predictive display. Moreover, operators should more carefully manipulate the mobile robot to avoid obstacles, especially movable obstacles. In Fig. 10, the second sudden change is larger than the first one, which shows that the mobile

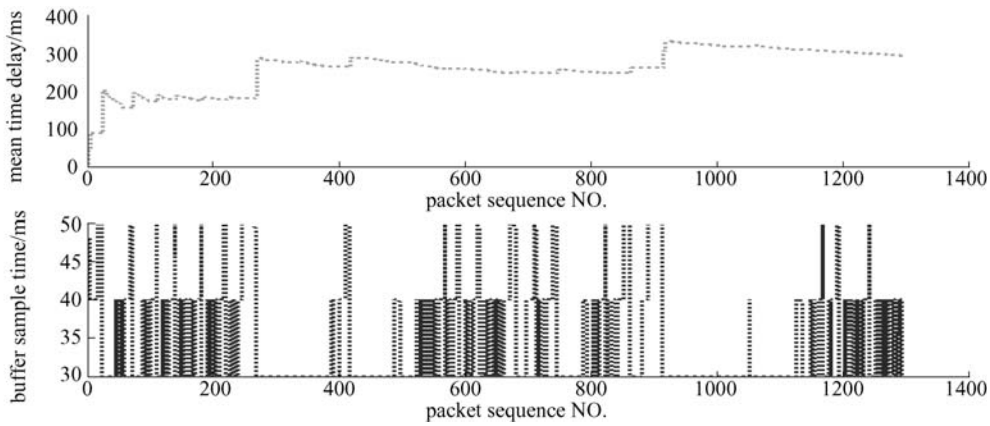


**Fig. 6** Mean network time delay and buffer sample time

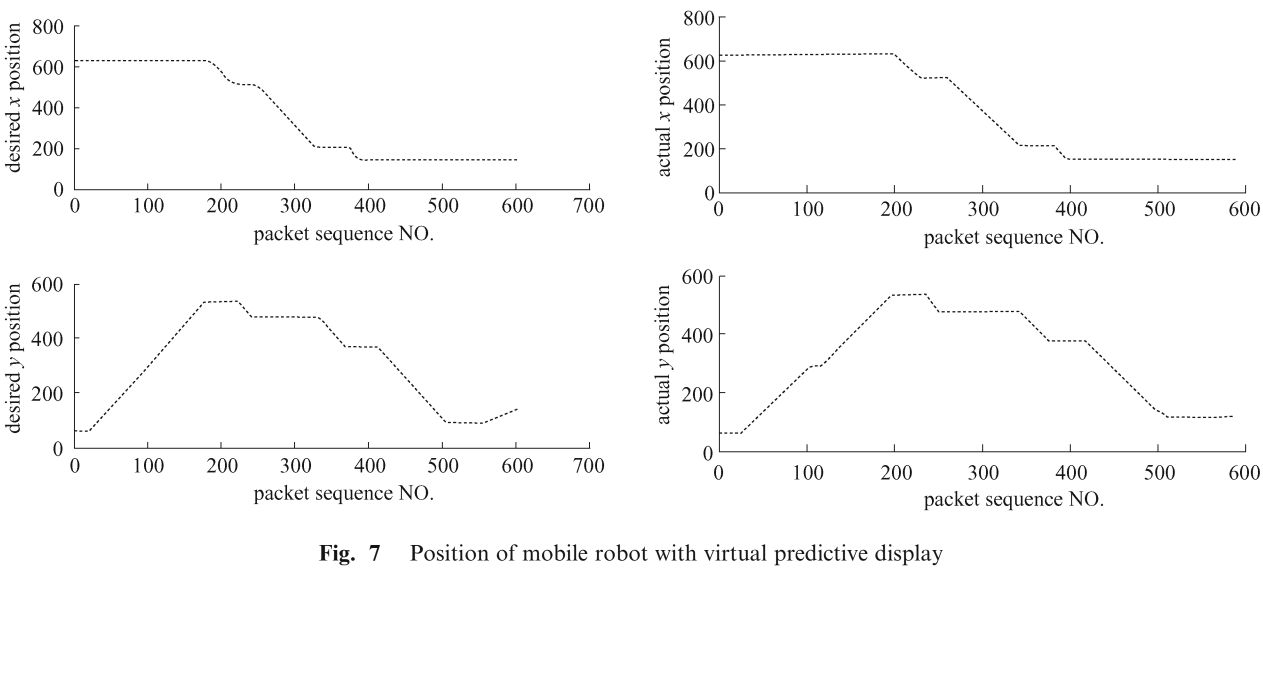


**Fig. 7** Position of mobile robot with virtual predictive display

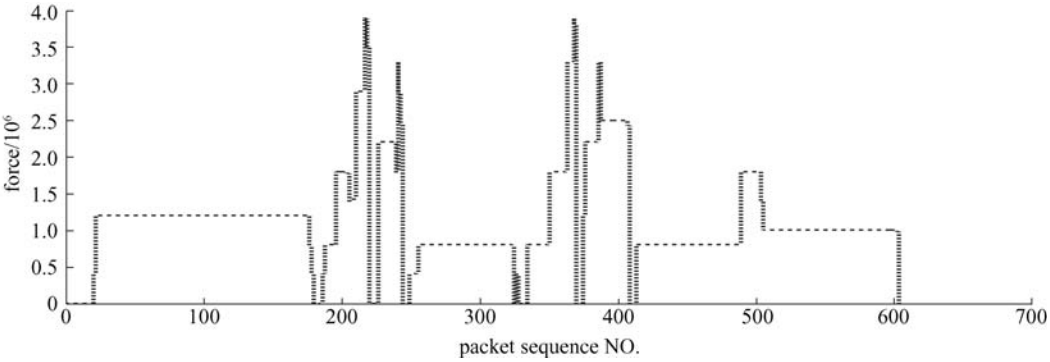


**Fig. 8** Reflective force with virtual predictive display

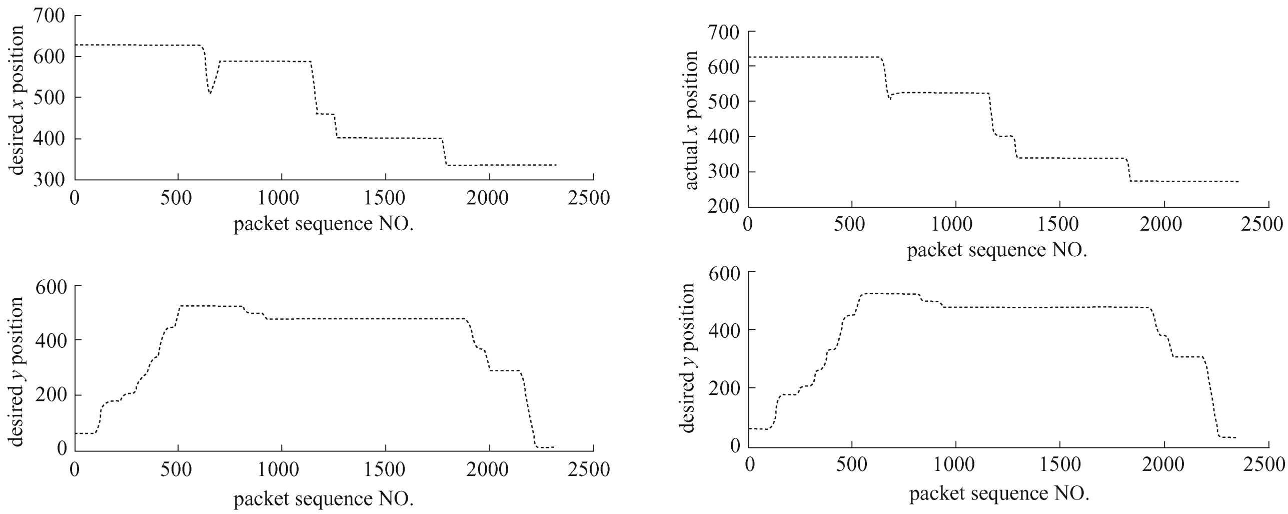


**Fig. 9** Position of mobile robot without virtual predictive display

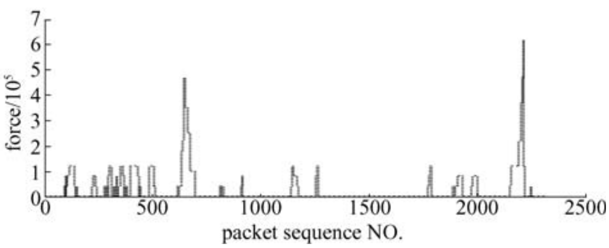


**Fig. 10** Reflective force without virtual predictive display

robot has collided with the movable obstacle. Therefore, by using the proposed methods, the real-time efficiency of the mobile robot system over a network is possible.

## 6 Conclusions

The stability and real-time efficiency of teleoperation systems can be improved at the predictive control scheme with DMM based on the proposed modified SP principle. The vital prediction and modification of network delay can be achieved with the proposed three delay algorithms. Force feedback and the virtual predictive display are used with real image feedback to enhance the sense of telepresence. The theoretical and experimental results show that time delay is not encountered in the real-time teleoperation system. The major contribution of this paper is the development of the general predictive scheme and the three algorithms.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China-Guangdong Joint Foundation Key Project (Grant No. U0735003) and Guangdong Province Natural Science Foundation of China Project (06105413).

## References

1. Goldberg K, Mascha M, Gentner S, et al. Desktop teleoperation via the World Wide Web. In: Proceedings of the IEEE International Conference on Robotics & Automation. Piscataway, NJ, USA: IEEE, 1995: 654–659
2. Paulos E, Canny J. Delivering real reality to the World Wide Web via telerobotics. In: Proceedings of the IEEE International Conference on Robotics and Automation. Piscataway, NJ, USA: IEEE, 1996: 1694–1699
3. Baruch J E F, Cox M J. Remote control and robots: an Internet solution. Computing & Control Engineering Journal, 1996, 7(1): 39–44

4. Brady K, Tarn T J. Internet-based remote teleoperation. In: Proceedings of the IEEE International Conference on Robotics and Automation.Piscataway, NJ, USA: IEEE, 1998: 65–70
5. Simmons R, Fernandez J L, et al. Lessons learned from Xavier. IEEE Robotics & Automation Magazine, 2000, 7(2): 33–39
6. SteinM R. Interactive Internet artistry. IEEE Robotics & Automation Magazine, 2000, 7(2): 28–32
7. Churms C L, Prozesky V M, Springhorn K A. The remote control of nuclear microprobes over the Internet. Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, 1999, 158(1–4): 124–128
8. Saucy P, Mondada F. KhepOn TheWeb: open access to a mobile robot on the Internet. IEEE Robotics & Automation Magazine, 2000, 7(1): 41–47
9. Taylor K, Dalton B. Internet robots: A new robotics niche. IEEE Robotics & Automation Magazine, 2000, 7(1): 27–34
10. Anderson R J, Spong M W. Bilateral control of teleoperators with time delay. IEEE Transactions on Automatic Control, 1989, 34(5): 494–501
11. Lawrence D A. Stability and transparency in bilateral teleoperation. IEEE Transactions on Robotics and Automation, 1993, 9(5): 624–637
12. Elhajj I, Xi N, Fung W K, et al. Haptic information in Internet-based teleoperation. IEEE/ASME Transactions on Mechatronics, 2001, 6(3): 295–304
13. Elhajj I, Xi N, Song B H, et al. Transparency and synchronization in supermedia enhanced Internet2based teleoperation. In: Proceedings of the IEEE International Conference on Robotics &Automation. Piscataway, NJ, USA: IEEE, 2002. 2713–2718
14. Liu P X, Meng M Q H, Liu P R, et al. An end-to-end transmission structure for the remote control of robots over IP networks. IEEE/ASME Transactions on Mechatronics, 2005, 10(5): 560–570
15. Oboe R. Force-reflecting teleoperation over the Internet: The JBIT project. In: Proceedings of the IEEE, 2003, 91(3): 449–462
16. Andreu D, Fraisse P, Segovia De Los Rios J A, et al. Teleoperation over an IP Network: From control to architectural considerations. In: Proceedings of the International Conference on Control, Automation, Robotics and Vision. Piscataway, NJ, USA: IEEE, 2004: 765–770
17. Uchimura Y, Yakoh T. Bilateral robot system on the real-time network structure. IEEE Transactions on Industrial Electronics, 2004, 51(5): 940–946
18. Tsuji T, Kato A, Ohnishi K, et al. Safety control of teleoperation system under time varying communication delay. In: Proceedings of the 8th IEEE International Workshop on Advanced Motion Control. Piscataway, NJ, USA: IEEE, 2004. 463–468
19. Fong T, Thorpe C, Baur C. Robot, asker of questions. Robotics and Autonomous Systems, 2003, 42(3–4): 235–243
20. Wang M, Liu J N K. Interactive control for Internet-based mobile robot teleoperation. Robotics and Autonomous Systems, 2005, 52(2–3): 160–179
21. Sim K B, Byun K S, Harashima F. Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller. IEEE Transactions on Industrial Electronics, 2006, 53(4): 1362–1372
22. Smith O J. A controller to overcome dead time. ISA Journal, 1959, 6(2): 28–33
23. Sheridan T B. Space teleoperation through time delay: Review and prognosis. IEEE Transactions on Robotics and Automation, 1993, 9(5): 592–606
24. Wang S H, Xu B G, Wang Q Y. Delays analysis for teleoperation over Internet and Smith p redictor with adaptive time-delay control. In: Proceedings of the IEEE International Conference on Robotics and Biomimetics. Piscataway, NJ, USA: IEEE, 2005. 664–669
25. Casavola A, Mosca E, Papini M. Predictive teleoperation of constrained dynamic systems via Internet-like channels. IEEE Transactions on Control Systems Technology, 2006, 14(4): 681–694
26. Ramos J J G, Maeta S M, Mirisola L G B, et al. Internet-based solutions in the development and operation of an unmanned robotic airship. Proceedings of the IEEE, 2003, 91(3): 463–474
27. Bruemmer D J, Few D A, Walton M C, et al. "Turn off the television!": Real-world robotic exploration experiments with a virtual 3-D display. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences. Piscataway, USA: IEEE, 2005: 1–8
28. Itoh T, Yudate K, Ito S, et al. New predictive display method of motion and force information for network teleoperation without using virtual environment model. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems. Piscataway, NJ, USA: IEEE, 2003: 2815–2822
29. Kheddar A. Teleoperation based on the hidden robot concept. IEEE Transactions on Systems, Man, and Cybernetics, 2001, 31(1): 1–13
30. Wang S H, Xu B G, Wang Q Y, et al. Modified Smith predictor and controller for time-delay process with uncertainty. In: Proceedings of the 6th World Conference on Intelligent Control and Automation. Piscataway, NJ, USA: IEEE, 2006: 623–627