Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Hybrid event based control architecture for tele-robotic systems controlled through Internet*

LI Xiao-ming (李晓明)[†1], YANG Can-jun (杨灿军)[1], CHEN Ying (陈 鹰)[1], HU Xu-dong (胡旭东)[2]

([1]State Key Laboratory of Fluid Power Transmission and Control, Zhejiang University, Hangzhou 310027, China)

([2]Zhejiang Institute of Science and Technology, Hangzhou 310033, China)

[†]E-mail: lxmzju2000@hotmail.com

Received Apr. 16, 2003; revision accepted Sept. 11, 2003

Abstract: A new hybrid event based control architecture for tele-robotic systems controlled through the Internet is proposed in this paper. Different from the traditional event based control method, the new framework does not require every part of the system to be strictly event synchronized. Instead, it allows time referenced control components to be integrated into this framework, which makes it more convenient to develop Internet based control systems. Since there are two reference variables, time and event, in this architecture, how to coordinate these components with different references to keep the stability of the whole system is discussed in detail in this paper. To verify this new idea, an experiment was conducted to control the end effector of a PUMA robot tracking a continuous state trajectory given on-line by the remote operator. Experimental results confirmed the stability of such systems being controlled through the Internet in real-time.

Key words: Tele-robot, Event based control, System integration
Document code: A **CLC number:** TP202

INTRODUCTION

With the rapid development of Internet technology, more and more Internet based applications in industries have been brought forth, ranging from web based data acquisition (Hu *et al*., 2002) to robots sharing through the Internet (Safaric *et al*., 1999). In recent years, many scientists and engineers have devoted enthusiastically much time to study of Internet based tele-robotic systems (Brady and Tarn, 2001). Internet, as a channel of communication, provides a new media for control signals and sensory information delivery controllable through the Internet (Mirfakhrai and Payandeh, 2001). All of these have enabled worldwide integration of mechatronical devices. On the other hand, there are still many mechatronical devices and systems (e.g. Robots) which exist as isolated islands. It is necessary to integrate these devices with the Internet and make them open structured and inter-operative.

To control devices through the Internet, two factors must be taken into consideration carefully: the loss of data packages and the time-variable time delay (TVTD) in the control loop. The event based plan and control theory (Fig.1) has been proved to be effective in dealing with the TVTD in the control cycle (Xi and Tarn, 1999; Kang *et al*., 1999). Because it is event, not time, which is used to reference the system's behaviors, the stability of the system will not be affected by the delay of the Internet. The essential feature of the event based planning and control theory is that it integrates low

* Project (No. 50305035) supported by the National Natural Science Foundation of China

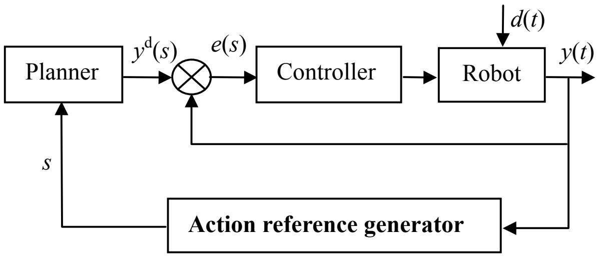

Fig.1 Event based control block

level sensing and control with high level task scheduling and action planning, which enables the robotics systems to cope with unexpected events and uncertainties. The system is converted to be time independent so that it is immune to time delay. The robot controller should be strictly event based in order to apply the event based method, which requires designing a new event based robot controller or converting existing controller to be event based using the method provided in the literature (Xi and Tarn, 1999). The event based plan and control theory also provides action reference for synchronization of local robots with remote controllers.

The requirement of the whole system to be strictly event referenced increases the difficulty of applying the event based control method. Event if the design of event based controller is not a problem, it is still troublesome to define a proper event reference *s* because that the desired state trajectory is usually generated randomly on-line instead of being a predefined one in most of the teleoperaton applications. However, with the existence of TVTD, the event reference variable cannot refresh very frequently, which will lead to the degrading of the system response. In order to use the existing time based robot controller and its control algorithms, it is preferable to have two action references: the time reference variable *t* for local robot controllers and the event reference *s* for synchronization between the remote and local sites. The two reference variables should be correlated with each other through the state of the robot being controlled.

The purpose of this paper is to introduce the hybrid event based control architecture together with the implementation of the experimental system. With this method, the local robot controller is kept unchanged, runs with the time reference *t* and cooperates with the remote operator with event reference *s*.

INTERNET IN CONTROL LOOP

The Internet plays an important role in the Internet based control system, especially in some real-time based systems. As mentioned above, sending the control command and sensory information through the Internet is liable to data packets loss and time variable time delay. The data packets loss can be eliminated by choosing appropriate communication protocol but the time delay naturally exists because of the switching principle of data communication of the Internet and will lead to the two main characteristics of the Internet: asynchronism and buffer effect (Elhajj *et al*., 2001).

The communication on the Internet is asynchronous essentially because the local robot site and remote operator site are actually both independent systems running on different processors with their own system clocks. Just being given the delayed sensory data received from the robot site, the operator cannot infer what the up to date state of the robot site is; on the other hand, the robot controller cannot infer what is the actual state of the robot generating the control command. This confusion caused by the asynchronism always leads to the system instability. So there should be a reference which can be used to coordinate the two sites when the tele-control task is performed. This reference should be relevant to the system state and can mark the system state uniquely. The event variable *s* can meet this requirement perfectly (Xi and Tarn, 1999).

Buffer effect is another trouble caused by the time delay. Suppose the value of time delay of the Internet is $\tau=\tau_1+\tau_2$ (τ_1, τ_2 are the values of delay in the two directions between the robot site and the controller site), the sensory data at time instant t_0 is described by $X(t_0)$ and the control command is symbolized as $u(t)$. At this time instant, the sensory information is $X(t_0-\tau_1)$ on the remote operator site and the command generated by the operator is $u(t_0-\tau_1-\tau_2)$ on the local robot site. This means that the control command generated according to $X(t_0)$ now applies to the system state $X(t_0+\tau_1+\tau_2)$. This is the buffer effect of the Internet, which causes the system to behave unexpectedly with the control command from the remote operator. Although the event s can be used to synchronize the coordination of the two parts of the tele-control system, the buffer effect still exists if the control system is not totally event based. To avoid the buffer effect, some alternations to the traditional event based control architecture (Fig.1) must be made on the robot site.

First, The Planner block in event based architecture is changed into Step Planner block. Step Planner is used to control the local robot from one stable state to another stable state. Step Planner can guarantee the stability of local robot system with interference from the remote operator. Then the command from operator should be filtered through the Command Filter block so that only the command encoded with the latest event variable sequence will be accepted. With this alternation, the buffer effect can be eliminated entirely. Fig.2 presents the loop sequence in the Internet based control system. The hybrid event based telerobot system will be detailedly introduced in the next section.

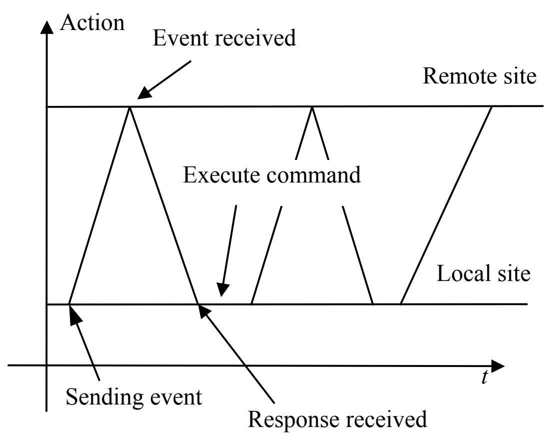

Fig.2 Action sequence in the Internet based control system

HYBRID EVENT BASED ARCHITECTURE

In order to apply the event based control method in the Internet based teleoperation of robot, a new hybrid event based tele-robot control architecture is put forward. According to the above analysis of the characteristics of the Internet, which is based on the traditional event based control architecture (Xi and Tarn, 1999). Fig.3 shows the control block diagram. The upper part above the dashed line presents the local robot site while the other part presents the remote operator site. These two sites are connected through the Internet. In this section, the stability of such system will be discussed first and then the details of each block will be presented, including their functions, the operation mechanism and the roles they play in the whole system.

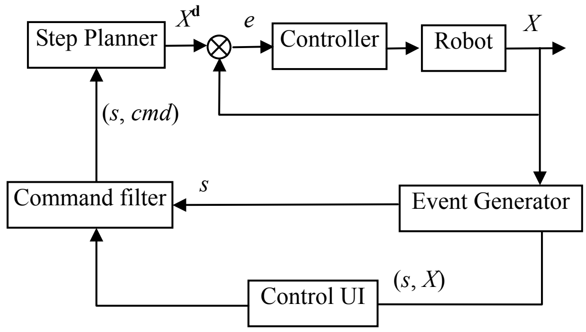

Fig.3 Hybrid event based control block

Stability analysis

The stability of the hybrid system will be proved first. The local robot controller is kept untouched, which is one of the reasons why this system is brought out, so it still runs with the time variable t as its action reference. The system can then be described generally as

$$\dot{X} = f(X(t), u(t), t) \tag{1}$$

and it is stable asymptotically without the remote operator. According to the definition of stability, there must exist a neighborhood $W(\delta)$ (see next subsection for more details) of the current system state. Given the desired new state within this neighborhood, the output error satisfies $\lim_{t\to\infty} e(t) = \lim_{t\to\infty}(X(t) - X^{\mathrm{d}}) = 0$, where $\|X^{\mathrm{d}} - X(0)\| < W(\delta)$. Generally, the system will reach a stable state (most of the time it is the desired state X^{d}) in limited time intervals, which can be denoted as t_{s}.

The desired state trajectory S can be described as a sequential controlling command flow $S=(X_1^{\mathrm{d}}, X_2^{\mathrm{d}}, X_3^{\mathrm{d}}, X_4^{\mathrm{d}}, \ldots)$. Denoting the time interval between two adjacent control commands as t_{v}, the telerobot system is stable if only the following conditions are satisfied:

$$t_{\mathrm{v}} \geq t_{\mathrm{s}} \tag{2}$$

$$\left\| X^{\mathrm{d}} - X(0) \right\| < W(\delta) \tag{3}$$

It is easy to understand that with this limitation the remote operator's control command (e.g. $X_i^{\mathrm{d}} \in S$) can only make the robot's operating state transfer from one stable point to another stable point in the robot's state trajectory, so that the stability of the whole telerobotic system is ensured.

To fulfill this stability condition given by Eq.(2) and Eq.(3), and considering the characteristics of the Internet in the control loop discussed in Section 2, the architecture of the control system is adjusted and some modules are inserted in order to filter or regenerate the proper control command sequence. This design of architecture is useful especially when the control command sequence S is generated online by the operator according to the feedbacks he/she receives from the robot site through the Internet, which have been delayed certainly.

Architecture introduction

In this part each block in the block diagram shown in Fig.3 will be introduced, along with the discussion of their functions, purposes and contributions to the system state trajectory control process.

1. Step Planner

Step Planner is a key component to the system stability and performance. It is used mainly to ensure the controlled object "stepping" from one stable state to another stable state, which guarantees the stability of the whole system. Some restrictions should be applied on it to avoid violent state oscillation and improve the system performance.

In the Step Planner model, there are three important parameters: the execution time t_{s}, the interval between two sequential commands t_{v} and the command distance $d_x=d(X^{\mathrm{d}})$.

The control command sequence is:

$$X^{\mathrm{d}}(s_m) = X_m^{\mathrm{d}}, \quad (0 \leq m < \infty) \tag{4}$$

where the $X \in \Omega$, Ω is the task space of robot, $DIM(\Omega)=n$, and s_m is the event sequence. The command distance is defined as:

$$d(X_i^{\mathrm{d}}) = \left\| \overline{X_{i+1} X_i} \right\| = \left\| \boldsymbol{X}_{i+1} - \boldsymbol{X}_i \right\| \quad (i \in [0, \infty)) \tag{5}$$

According to the discussions on system stability in Section 1, d_X should satisfy the condition of stability of the local robot system. Denote this restriction condition as

$$d_X(i) < r \tag{6}$$

where r is a scalar value. The collection of d_X which satisfies Eq.(6) forms an open set, i.e.

$$U(X_i, r) = \{(X_i \in IR^n \times IR \ \left| d_X(i) < r \right.\} \tag{7}$$

To satisfy the stability condition described by Eq.(2), the set $U(X_i, r)$ should be within the neighborhood of X_i, i.e.

$$U(X_i, r) \subseteq W(X_i, \delta) \tag{8}$$

So the control command sequence must fulfill the conditions expressed with Eqs.(5), (6) and (8).

Another function of the Step Planner is the interpretation of the command from the remote operator. The command sent by the remote operator varies when different control UI (User Interface) is used by the operator, e.g. the mouse and pen can be used to generate the position command while the joy stick can only be used to generate the velocity command. The Step Planner will convert these different control commands to corresponding command sequence which would be sent to the local robot controller with appropriate time intervals to ensure the fulfillment of stability condition described by Eq.(3).

2. Event Generator

Event Generator is used to generate event sequence and send the sensory data marked by event sequence index to the remote operator site. The event sequence is indexed by increasing integers. i.e. $s(i)=i, i=1, 2, 3,\ldots$. In order to preserve the credits of event based control method, the event should be generated directly from the sensory data. In our system, the event is generated when any of following conditions is satisfied:

(a) The robot finishes the command from step planner and reaches a new stable state.

(b) The status of the robot remained unchanged within specific time intervals.

(c) Predefined event occurs triggered by specific sensors.

The first condition is used to keep the system continuously controlled; the second condition is designed for unexpected events and the third is required for interfacing with other discrete event sources. The operator cannot generate control command until the event from the robot server is received. Meanwhile, there must be a control command generated accordingly when any event is received. Unexpected control commands are filtered and removed by the Command Filter block.

3. Command Filter

In this architecture, the Internet together with the operator is considered as being passive: one signal in and one signal out. The remote site can only temporarily save the information but cannot generate it or swallow it. This restriction is applied by using the Command Filter block. When the event is sent to the remote site, the Command Filter block is locked with the latest event sequence index at the same time. The received control command from remote site will be filtered by this block and only the one marked with the corresponding event sequence index can pass through this block and reach the Step Planner block.

EXPERIMENTAL SYSTEM IMPLEMENTATION

We have applied the hybrid event based architecture to internetize the existing robotic system. The object being controlled was a PUMA560 6DOF industry robot, which is a typical robotic system and is widely equipped in industry and laboratories. Because the traditional PUMA robot is not open-structured so it has to be reconstructed and has the network interface added. Fig.4 shows the experimental system structure and Fig.5 shows the experimental rigs on the robot site. The PUMA robot is controlled now by use of the PMAC multi-axis controller, which is plugged into the robot server PC and communicates with it through the BUS of the motherboard. The server is connected to the Internet and allow client from anywhere to log in to control the robot.

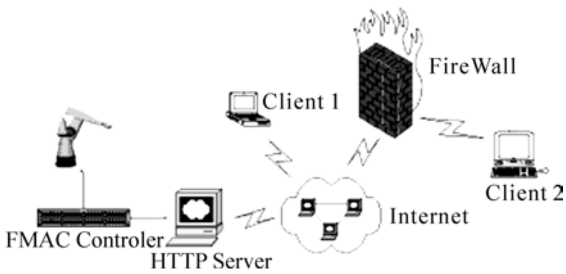

Fig.4 Experimental telerobot system

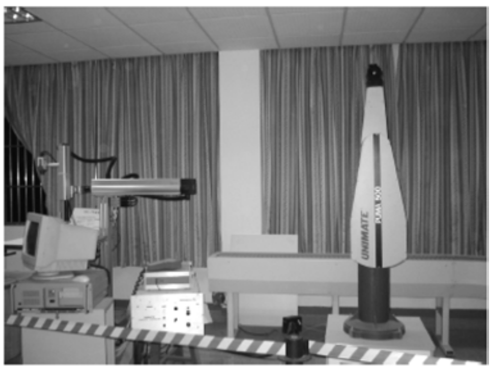

Fig.5 Experimental telerobot devices

The communication protocol we have selected was HTTP (Hyper Text Transportation Protocol) because we believe that the HTTP is the only “true” Internet communication protocol as it is supported widely by almost all of the proxy servers/fire walls. Another advantage of HTTP is that it is connection oriented and meet our assumptions of the Internet.

The HTTP based server push technique is adopted to decrease the effect of time delay. It should be pointed out that the HTTP transfer protocol increases the time delay, so other protocols such as UDP or TCP could be a good alternative in some cases where the systems have more restrictions to the time delay and system responses.

The experiment was carried out in the local intranet of the laboratory. Considering the protocol and the event based architecture, the experimental result and conclusion can be generalized to the Internet environment.

The software was developed using Java programming language, both the server site and the client site. The integration of PMAC controller with the robot server was achieved by utilizing the Ch programming language (Cheng, 1995). Ch is a C-extended script language, which is quite useful in mechatronical system control, system integration, scientific calculation and distributed computing. One can refer to literature (Cheng, 2002) for more information about Ch. The Ch program and Java program could communicate with each other through Internet socket connection and an inner data communication protocol was designed for data exchanging.

In this experiment, an autonomous controller was used instead of the operator to control the PUMA Robot so the problems caused by the limitation of the control UI could be avoid. Graphic control UI was also provided for observing the running states of the PUMA robot together with other auxiliary operations.

EXPERIMENTAL RESULTS

The curves in Fig.6 show the experimental results for the task of tracking a trajectory described by a predefined function. The accurate desired trajectory was:

$$\begin{cases} x = x \\ y = 750 \\ z = \sin(\dfrac{2\pi}{400.0}x)\cdot 200.0 \end{cases} \quad (x \in [0, 400.0]) \qquad (19)$$

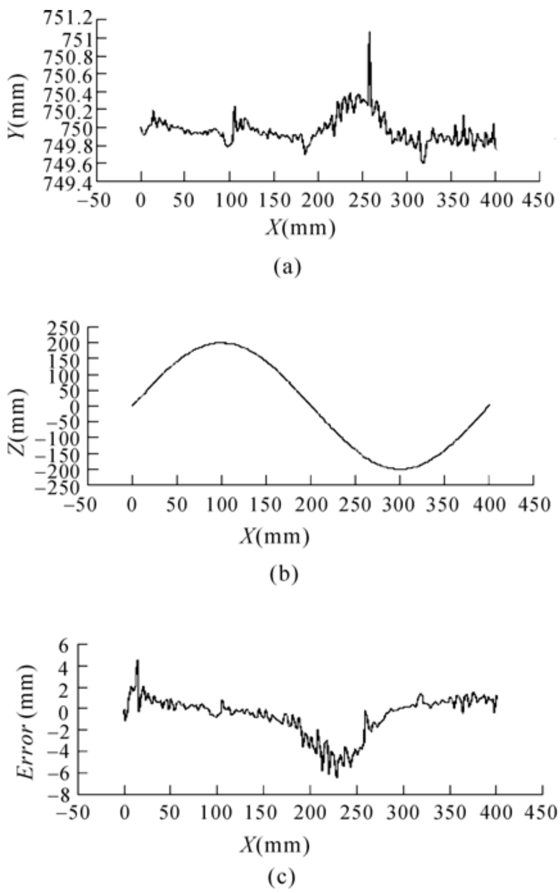

Fig. 6 Trajectory tracking experimental results
(a) the actual trail of the end effector of PUMA560 projected on the *X*-*Y* plane; (b) the actual trail of the end effector of PUMA560 projected on the *X*-*Z* plane; (c) the tracking error in the control process

The first two curves in Fig.6 present the actual trail of the end effector of PUMA560 projected on the *X*-*Y* plane and *X*-*Z* plane. The third one shows the tracking error in the control process. From these curves we can see that the system was stable; and that the control process was smooth. There was a big time delay at time instant near 25, but the trajectory was not effected by this delay. The tracking error came mainly from the local controller, which was originally a PID controller. Fig.7 presents the movement of the terminal in the time field. This experiment proved the effectiveness of the hybrid system architecture; and that devices integrated by use of this architecture could work together with harmony. Oscillations in the direction of axis y were introduced by Step Controller when sending

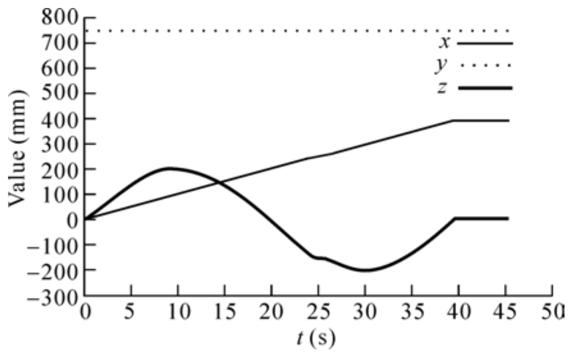

Fig.7 Movement of the end effector of PUMA robot

commands to local robot controller and compelling the system state to follow the desired state trajectory. The oscillations could be eliminated by coordinating the control policy of the step planner and local robot controller.

CONCLUSION

We presented in this paper a hybrid event based control architecture which can be used to integrate mechatronical devices through the Internet. Because of the asynchronism and buffer effect of the Internet, the normal control method cannot be applied. One possible solution is the event based plan and control theory, which turns the action reference from the traditional time variable t to the event variable s. Based on this method, we proposed a new hybrid event based architecture, which can preserve the advantages of the event based control theory and does not require redesigning of the local robot controller at the same time. The system is a hybrid containing both time variable t and event s in it. The event is used mainly on the synchronization between the local site and the remote operator site. Thus the robot system can be easily converted into Internet Based Tele-robot System. The stability of this architecture has been proved in theory and an experimental system had been implemented. The results of continuous trajectory tracking experiments presented in this paper showed that this control architecture can effectively be used for building Internet based control systems. Because the control effect is mainly determined by the local robot controller, more efforts should be made and concentrated on trying to improve the performance of the time based local controller.

References

Brady, K., Tarn, T.J., 2001. Internet-Based Teleoperation. Proceedings of the 2001. IEEE International Conference on Robotics & Automation, Seoul, Korea, p.644-650.

Cheng, H.H., 1995. Ch Shell for Integration of Mechatronic Systems. Proceedings of the 1995 NSF Design and Manufacturing Grantees Conference, La Jolla, California, p.95-96.

Cheng, H.H., 2002. The Ch Language Environment. Soft Integration, Inc. http://www.softintegration.com.

Elhajj, I., Xi, N., Wai, K.F., Liu, Y.H., Li, W.J., Kaga, T., Fukuda, T., 2001. Super Media in Internet-based Telerobotic. *In*: Operations, E.S. Al-Shaer and G. Pacific (Eds.), MMNS 2001, LNCS 2216, p.359-372.

Hu, X.D., Yu, H., Chen, Y., 2002. Web-based data acquisition. *Journal of Zhejiang University SCIENCE*, **3**(2): 135-139.

Kang, W., Xi, N., Tan, J.D., 1999. Analysis and Design of Non-time Based Motion Controller for Mobile Robots. Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan, p.2964-2969.

Mirfakhrai, T., Payandeh, S., 2001. A Model for Time-Delays for Teleoperaion Over the Internet. Proceedings of 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Alberta, Canada, p.1-6.

Safaric, R., Jezernik, K., Calkin, D.W., Parkin, R.M., 1999. Telerobot Control via Internet. Proceedings of the IEEE International Symposium on Industrial Electronics, 1999, Bled , Slovenia, p.298-303.

Tarn, T.J., Song, M., Xi, N., 1998. Intelligent Planning and Control for Hybrid System. Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, B.C., Canada, p.972-977.

Xi, N., Tarn, T.J., 1999. Action Synchronization and Control of Internet Based Telerobotic Systems. Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan, p.219-224.