Abstract

This paper concentrates on transparency and synchronization of supermedia in Internet based teleoperation. Supermedia is used to describe the collection of all the feedback streams in teleoperations, such as haptic, video, audio, temperature and others. Transparency and synchronization are introduced and analyzed from the event-based control perspective. The concepts of event-transparency and event-synchronization for event-based control telerobotic systems are developed and their implications are studied. To illustrate those concepts and their benefits, the teleoperation of a mobile manipulator via the Internet, where haptic, video and temperature information is fed back to the operator, is discussed. Experimental results will verify the event-transparency and event-synchronization of this event-based telerobotic system.

1 Introduction

The common forms of feedback in teleoperation are video and force. However, there are many applications that require additional types of feedback for feasibility and safety. An example is classification and identification, where temperature is required to identify certain objects and materials [1]. Also in some medical applications the temperature change of the organ operated on is critical for safety. Therefore, supermedia enhanced teleoperation systems, similar to the one shown in Fig.1, are required. Supermedia is used to describe the collection of all the feedback streams, such as haptic, video, audio, temperature and others. However, for such teleoperation systems to become widely acceptable several safety and performance specifications have to be met. Stability, transparency and synchronization are among the main features that are desired in any teleoperation system. However, ensuring those in teleoperation systems, specifically Internet based ones, is challenging. The difficulties are caused by the properties of delay in such networks. Internet delay is characterized as random, not bounded and not the same for both branches in the control loop. Stability for teleoperation systems has been studied extensively and many promising results have been presented by the authors and others [2]-[7]. On the other hand, there is still much work to be done relating to transparency [8]-[10]. It was even shown that robust stability and transparency can be conflicting design goals in teleoperation systems under time-based control [9]. As for synchronization in such systems, it has not been addressed. Most of the synchronization work found in the literature relate to multimedia synchronization in open loop systems [11] [12]. In previous work presented by the authors the event-
based control was adopted to ensure the stability of bilateral teleoperation systems [7]. This paper studies transparency and synchronization of event-based systems. Event-transparency and event-synchronization concepts, which are features that imply increased efficiency and consistency, are introduced and their advantages are analyzed. Experimental results of the Internet based teleoperation of a mobile manipulator verify the analysis. In those experiments, supermedia is fed back from the environment in the form of force, video and temperature. And those are rendered to the operator in their original forms.

2 Transparency and Synchronization in Event-Based Control

Transparency and synchronization are relative performance measures of teleoperation systems. The current technology does not allow for perfect transparency and synchronization in Internet based teleoperation if time-based control is used. The main limitations, which are communication related, are bandwidth and time delay. Those limitations are augmented in the Internet since it does not offer any guarantees on bandwidth and time delay. Especially, that the Internet, in its current form, was not designed and implemented having in mind applications such as teleoperation. Therefore, there is a need for teleoperation systems to offer consistency in the performance under the dynamic communication conditions encountered.

In the time domain, transparency is simply a measure of how closely does the feel of interaction with the environment in teleoperation resemble the actual feel if the operator was to directly interact with the environment. Formally, perfect time-transparency is satisfied under the condition,

\[ Z_t(t) = Z_e(t) \]  

(1)

where \( Z_t(t) \) is the transmitted or “felt” impedance and \( Z_e(t) \) is the slave environment impedance. Impedance is defined as the ratio of force to velocity. Therefore,

\[ Z_t(t) = \frac{F_h(t)}{V_h(t)} \quad Z_e(t) = \frac{F_e(t)}{V_e(t)} \]  

(2)

where \( F_h \) and \( V_h \) are the force and velocity generated by the joystick respectively; similarly, \( F_e \) and \( V_e \) are the slave force and velocity respectively. It is intuitive that under time-based control perfect time trans-

parency in teleoperation is not possible because of delay [9]. However, in event-based control some consistency in the control can be achieved. This consistency is event-transparency and is defined as,

**Definition 1** A control system is event-transparent if, \( \frac{F_h(s)}{V_h(s)} = C \frac{F_e(s)}{V_e(s)} \) and \( F_e(s)_{rd} = F_e(s)_{nd} \), where \( C \) is a constant, \( s \) is the action reference, \( F_e(s)_{rd} \) and \( F_e(s)_{nd} \) are the supermedia sensed by the system under random delay and no delay conditions respectively.

It is clear that this generalized definition can apply to any control system, where the \( F \)’s are the feedback and the \( V \)’s are the control signals. In the special case of teleoperation systems, \( F \) is usually force or some type of supermedia and \( V \) is the operator’s command. To design event-transparent systems, first design the system as an event-based control system. That is the signals in the system are being sampled uniformly with respect to a reference, \( s \), other than time. In order to ensure stability of the teleoperation system, this action reference has to be a non-decreasing function of time [7]. Being an event-based control system gives,

\[ F_h(s) = \alpha F_e(s) \quad \text{and} \quad V_h(s) = \beta V_e(s) \]  

(3)

\[ \Rightarrow \frac{F_h(s)}{V_h(s)} = C \frac{F_e(s)}{V_e(s)} \]  

(4)

Where \( \alpha, \beta \) and \( C \) are simply scaling factors. In addition, \( s \) has to ensure that \( F_e(s)_{rd} = F_e(s)_{nd} \). The choice of \( s \) will depend on the type of feedback being used.

From the definition, it implies that if a system is event-transparent then regardless of random time delay, the control received under random time delay as a function of the action reference, \( V_e(s)_{rd} \), is equal to the control received by the robot under no time delay as a function of the action reference, \( V_e(s)_{nd} \). This is deduced from the fact that the choice of \( s \) ensured that the haptics detected by the robot under time delay as a function of the event, \( F_e(s)_{rd} \), are equal to the haptics detected by the robot under no time delay as a function of the event, \( F_e(s)_{nd} \). So the delay will have no effect on the control signal with respect to the event-based action reference. Obviously, in the time case this property is not true. The following theorem formally states this,

**Theorem 1** If the event-based action reference, \( s \), of an event-based control system is chosen such that \( F_e(s)_{rd} = F_e(s)_{nd} \) then \( V_e(s)_{rd} = V_e(s)_{nd} \), where \( F_e(s)_{rd} \) and \( F_e(s)_{nd} \) are the supermedia sensed by the system under random delay and no delay conditions.
respectively, and where \( V_e(s)_{rd} \) and \( V_e(s)_{nd} \) are the control received under random time delay and no time delay conditions respectively.

**Proof.** First it will be shown that \( F_h(t) \) and \( V_h(t) \) are being sampled at the same time instances. To prove this, \( F_h \) and \( V_h \) respect to time and event references are analyzed for event-based control and time-based control under time delay conditions. In event-based control, assume that the deviation of the nonuniform samples from a set of uniform samples is represented by \( \delta_{1n} \) and \( \delta_{2n} \) and the event sampling periods are \( S_1 \) and \( S_2 \). So the samples are \( F_h(nKS_1 - \delta_{1n}) \) and \( V_h(nKS_2 - \delta_{2n}) \), where \( K \) is a scaling factor. In order to determine the relationship between the sampling instances of signals \( F_h(t) \) and \( V_h(t) \), the signals \( \theta_1(s) \) and \( \theta_2(s) \) are formed [14] [15].

\[
\begin{align*}
\theta_1(s) &= \sum_{n=-\infty}^{\infty} \delta_{1n} \sin w(s - nKS_1) \\
\theta_2(s) &= \sum_{n=-\infty}^{\infty} \delta_{2n} \sin w(s - nKS_2)
\end{align*}
\]

(5) (6)

Sample values of \( \theta_1(s) \) and \( \theta_2(s) \) are \( \delta_{1n} \) and \( \delta_{2n} \).

\[
\begin{align*}
\theta_1(nKS_1) &= \delta_{1n} \quad \text{and} \quad \theta_2(nKS_2) = \delta_{2n}
\end{align*}
\]

(7)

Let

\[
\begin{align*}
t_1 &= s - \theta_1(s) \\
t_2 &= s - \theta_2(s)
\end{align*}
\]

(8) (9)

It follows that

\[
\begin{align*}
\text{if} \ s &= nKS_1, \text{ then } t_1 &= KnS_1 - \theta_1(nKS_1) \\
&= KnS_1 - \delta_{1n} \quad (10) \\
\text{if} \ s &= nKS_2, \text{ then } t_2 &= KnS_2 - \theta_2(nKS_2) \\
&= KnS_2 - \delta_{2n} \quad (11)
\end{align*}
\]

Therefore, the transformations in eq.(8) and eq.(9) transform the points \( nKS_1 \) and \( nKS_2 \) of the \( s \)-axes into the points \( KnS_1 - \delta_{1n} \) and \( KnS_2 - \delta_{2n} \) of the \( t \)-axes. Considering that the sampling with respect to the event, \( s \), has the same period for all the signals, that is \( S_1 = S_2 \). And assuming that \( V_e \) is generated instantaneously once \( F_e \) is rendered, that is \( \delta_{1n} = \delta_{2n} \). Then, \( KnS_1 - \delta_{1n} = KnS_2 - \delta_{2n} \) and the sampling instances of signals \( F_h(t) \) and \( V_h(t) \) are identical. A similar analysis can be made to show that the sampling instances of signals \( F_e(t) \) and \( V_e(t) \) are identical. For the time-based control case, since the time and event sampling periods are not the same for either \( F_h(t) \) and \( V_h(t) \) or \( F_e(t) \) and \( V_e(t) \), then similar conclusions can not be made.

This conclusion implies that, in the event-based control case if the system is event-transparent then,

\[
F_e(s)_{rd} = F_e(s)_{nd} \Rightarrow F_h(s)_{rd} = F_h(s)_{nd} \quad (12)
\]

Assuming that the operator’s reaction is consistent based on the force felt, then eq.(12) gives,

\[
V_h(s)_{rd} = V_h(s)_{nd} \Rightarrow V_e(s)_{rd} = V_e(s)_{nd} \quad (13)
\]

Thus the system control is the same regardless of random delay as stated previously, implying that its performance with respect to the event-based action reference is transparent to delay and its randomness. Since the event-based action reference can be any system output, which is a non-decreasing function of time [7], this property implies that consistent system behavior with respect to certain system outputs can be achieved. However, special care should be taken in designing the event-based action reference. It has to ensure that \( F_e(s) \) is the same regardless of time delay. For example, \( F_e(s) \) can be a function of the distance the robot moved; therefore, the event-based reference, \( s \), can be chosen to be the distance traveled.

As for synchronization, research found in the literature has mainly focused on time synchronization in open loop systems [11] [12]. However, the control systems presented in this paper are closed loop event-based. Therefore, there is a need to define synchronization in this context. Event-synchronization is defined as:

**Definition 2** An event-synchronized system is one in which all the signals in the system are always referencing events which are within a certain tolerance range of each other.

This definition is similar to the definition of time synchronization but instead of having the time as a reference the event-based action reference is being used. Also an important difference is that this definition includes the control signal too. Which means that the control has also to be synchronized with the feedback. This implies that the event-synchronized system ensures that the feedback sensed is a reflection of the system’s most current state. From the definition it implies that systems have to be designed in a way...
that the video, temperature and force being rendered have action reference stamps that are within a certain range of each other. Also they have to ensure that the control signal has an action reference stamp which is within a certain range of the feedback being rendered.

### 3 Implementation

The experiments were done using a mobile manipulator according to the system architecture shown in Fig.2. A new temperature rendering device has been developed to integrate within the Internet-based teleoperation system. The mobile manipulator is equipped with an infrared non-contact temperature sensor, which was used for the remote sensing.

The operator sends velocity commands, which the mobile manipulator executes. This velocity is divided by the local controller between the arm and the mobile base. This division is done to achieve an acceptable posture for the arm [13]. The haptic feedback in this case is actual force detected by the force/torque sensor mounted on the gripper. Also temperature and real-time video are fed back.

The video is event-synchronized with the force and temperature. Thus the operator can be sure that the video displayed is close in event to the force and temperature felt. The programs required for this system are the control and temperature server, control and temperature client, video server and video client.

The control and temperature server runs on the robot and is responsible for receiving the velocity commands and deciding the appropriate control for the manipulator and the mobile base. It is also responsible for sampling the temperature sensor and the force/torque sensor in order to feedback this information to the control and temperature client. In addition, it forwards the event number to the video server to be used on the client side for synchronization.

The control and temperature client runs on the remote machine. It forwards velocity commands to the server and receives force and temperature information back. The force is rendered with the joystick and the temperature is sent to the temperature rendering device connected to the serial port. Also this client forwards the event reference to the video client so that it can be compared with the reference that the frame is tagged with for synchronization purposes.

The video server tags each video frame with the event reference and forwards it to the video client. The video client before displaying a frame compares its event reference with the event reference of the force and temperature currently being rendered. If the frame’s event reference is much less than the force’s and temperature’s event reference then the frame is discarded. The limit of how old a frame can be and still be displayed is a performance parameter which can be tuned.

As for the temperature rendering device, it is based on thermoelectric technology. Thermoelectric coolers/heaters are solid state heat pumps that operate on the Peltier effect. The Peltier effect states that there is a heating or cooling effect when electric current passes through two conductors. A voltage applied to the free ends of two dissimilar materials creates a temperature difference. Therefore, the temperature on one end can be controlled by controlling the voltage and its polarity across the device. The polarity is controlled based on whether cooling or heating is required and is done by switching between two circuits. A contact sensor is mounted on the device in order to feedback the actual temperature and attain closed loop control.

### 4 Experimental Results

The performance of the system was experimented under no delay and random delay conditions. The mobile manipulator was teleoperated locally from Michigan State University and remotely from The Chinese University of Hong Kong. The operators were asked to move the robot in any direction and once they detect a force they were asked to command the robot in that
direction. The results are shown in Fig.3 for the no delay case and in Fig.4 for the random delay case. The first row in each figure shows the forces detected in the $x$ and $y$ directions with respect to the event reference, the second row gives the actual velocity of the robot in the $x$ and $y$ directions with respect to the event reference. The event-based action reference in this system is taken to be the distance the robot moved.

As seen, the velocity of the robot in both cases is responding to the force similarly. Once a force is applied on the robot as seen in the first rows the velocity changes direction to match that force as seen in the second rows. For example when a force is applied in the $y$ direction the velocity changes to the $y$ direction.

Although different operators did those experiments, the performance is similar in both cases. This illustrates the implications of event-transparency, which states that if a system is event-transparent then the control is consistent regardless of random delay. In addition, it is clear in both figures that the control signal, which in this case is the velocity, is a reflection of the most current state of the system, that is the force detected. This is seen since there is no difference between the reference at which the force is detected and the reference at which the velocity responds to it. This implies that the system satisfies part of the event-synchronization requirements.

As for the event-synchronization requirements that relate to the supermedia being rendered, the results are shown in Fig.5. This figure gives the temperature sensed on the robot, received remotely and rendered to the operator with respect to the event reference in the first column. It also gives the forces detected and received with respect to the event reference in the second column. As seen the received signals are identical to the detected ones with respect to the event reference. Also all the supermedia streams are event-synchronized, which implies that the force and temperature detected at a certain event reference value are rendered at the same event reference value. For example, the force and temperature detected at event reference $s_n$ are rendered to the operator at the same event reference $s_n$. Note that the rendered temperature plotted is slow with respect to the detected temperature. This is due to the fact that the external temperature sensor used to detect the temperature of the rendering device has slow response.

Similarly the video is event-synchronized with the force and the temperature. However, since video feedback is slower than force and temperature feedback, a certain tolerance has to be allowed. So the frame generated at $s_n$ can be rendered at any event reference within $s_n + N$, where $N$ is the accepted tolerance. If the frame does not arrive within this margin then it is discarded. So what the operator will be seeing cannot be older than $N$ events from what is being felt. This results in the video being event-synchronized with the other supermedia streams.

The experimental results presented confirmed the analysis made regarding event-transparent event-based control teleoperation systems. This implies that performance of the system is consistent in the face of random time delay. Also the supermedia streams fed back and the control signal are event-synchronized.
5 Conclusions

Event-transparency and event-synchronization were introduced in this paper. Analysis showed that event-based control under certain conditions can ensure those properties. This analysis was independent of the system or environment model; therefore, it applies to any event-based control system that satisfies the design criteria.

The system developed included supermedia feedback, where supermedia refers to all feedback forms such as video, haptic and temperature. Experimental results confirmed the analysis done and showed that the event-based control system was event-transparent and event-synchronized. Event-transparency allowed the system to have consistent performance under no delay and random delay conditions. Event synchronization ensured that all the rendered streams are within a certain number of events of each other.

References


