Some Remarks on Delay Effects in Motion Synchronization in Shared Virtual Environments

Joono Cheong, Silviu-Iulian Niculescu, Yonghwan Oh, Irinel-Constantin Morarescu

SOME REMARKS ON DELAY EFFECTS IN MOTION SYNCHRONISATION IN SHARED VIRTUAL ENVIRONMENTS

Joono Cheong * **Silviu-Iulian Niculescu** ** **Yonghwan Oh** ***
Irinel Constantin Morărescu ****

* *Department of Control and Instrumentation Engineering, Korea University, Jochiwon, South Korea*
E-mail: `jncheong@korea.ac.kr`

** *L2S (UMR CNRS 8506), CNRS-Supélec, 3, rue Joliot Curie, 91192, Gif-sur-Yvette, France.*
E-mail: `Silviu.Niculescu@lss.supelec.fr`

*** *Bio-Mimetic Robotics Laboratory, Korea Institute of Science and Technology (KIST)*
E-mail: `oyh@kist.re.kr`

**** *INRIA Rhône Alpes, BIPOP project*
Inovalée, 655 avenue de l'Europe, 38330, Montbonnot, France.
E-mail: `constantin.morarescu@inrialpes.fr`

Abstract: This paper addresses the motion synchronisation problem in shared virtual environments in the presence of communication delays. More precisely, we consider the case of multiple users interacting with the same dynamics. Unlike the conventional synchronization, the technological attempt we are interested in pursues a more *robust* and *better synchronization* that gives an almost concurrent evolution of motions between the distributed systems in absolute time-frame (earth's time). Physically, the existence of time delay prevents immediate information exchange, which disables concurrent motions between the distributed systems. Using the delay information available, the proposed controller preserves natural local dynamics and compensate for de-synchronization error caused by mismatched initial conditions. Simulation tests are conducted in order to validate the considered methodology.

Keywords: delay, virtual environments, synchronisation, Smith predictors, stability.

1. INTRODUCTION

It is well-known that the interconnection of two or more dynamical systems leads to an increasing complexity of the overall system's behavior due to the effects induced by the emerging dynamics (in the presence or not of feedback loops) in strong interactions (sensing, communication) with the environment changes. Decision making in such systems is challenging and is subject to multiple competitive objectives. The development of technology in the last years is accompanied with increasing computing, sensing, communications in decision making systems and processes. Among these systems, there exists a lot of examples where the *control* (or the decision) is based on the *information* changed and transferred between systems (units) or sub-systems (sub-units). As examples, we can cite: teleoperation, networked control systems (NCS), and shared virtual environment. Without any lack of generality, such systems are simply called *"information-based systems"*. Further details and various references on such topics can be found in [Murray (2002)].

One of the major problems appearing in such information-based systems is related to the *propagation*, *transport*, and *communication delays* acting "through" and "inside" the interconnections. The *origin* of such *delays* can be: the physical separation between the systems defining the interconnections, or due to the presence of the human factor in the decision process, or finally due to some hierarchy, and synchronization at the lowest levels in the decision process in real-time.

This paper addresses the analysis of *delay effects* in some class of *information-based interconnected systems*, namely the shared virtual environment simulation [Lawrence (1993); Cheong *et al.* (2005)]. An extremely brief presentation of the synchronization methodology for these inter-connected systems is presented in Section 2. The construction of the controller and related closed-loop stability analysis are proposed in Section 3. A particular attention will be paid to the sensitivity of the scheme with respect to the overall delay parameter (round-trip time). Some simulation results are illustrated in section 4, and finally, some concluding remarks end the paper.

2. SHARED VIRTUAL ENVIRONMENT SIMULATION AND DELAY

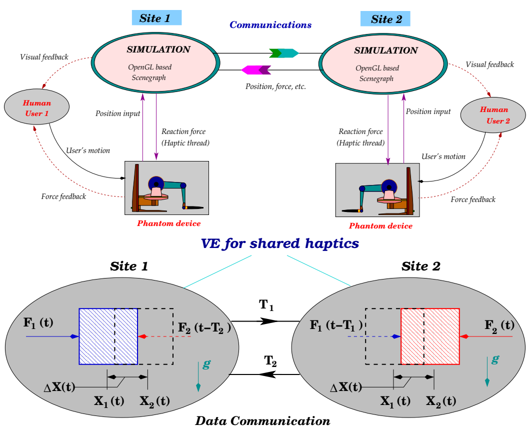


Fig. 1. Shared virtual environment with force input (Up) Implementation of peer-to-peer shared virtual environment; (Down) Time delay and graphic de-synchronization

Shared virtual environment requires synchronized visualization of virtual environment and real-time stable haptic interaction between separate users to carry out collaborative tasks in virtual assembly, CAD modeling, or medical training [Singhal and Zyda (1999)]. The use of communication networks complicates the task since we need to consider the *communication constraints*, and, in particular, the communication time-delay. In the context of shared virtual environment applications, time-delay in the data communication becomes the most difficult part so as to meet synchronized visualization and immediate response from user interaction. Due to the time delay, a change of a virtual environment in one site cannot be immediately displayed in the remote site, and de-synchronized graphic display between users may lead to unstable interaction between them [Katz and Graham (1994)]. Furthermore, in case when users are interacting through mechanical haptic interface, the instability can cause damage to the device and the users also.

For illustration, let's consider a shared virtual environment with solid cube as shown in Fig.1. Two remote users are interacting with the cube at the same time. The *challenging problem* here is the *difficulty of synchronizing the virtual environment at both sites*. The motion of the cube at site 1, $X_1(t)$, is computed by local force $F_1(t)$ and remote force $F_2(t - T_2)$ by the Newton's law, while the motion of the cube at site 2, $X_2(t)$, is computed by remote force $F_1(t - T_1)$ and local force $F_2(t)$, where T_1 and T_2 are communication delays from site 1 to site 2, and from site 2 to site 1, respectively. As time goes on, because the input histories are different at both sites, the deviation of the position, $\Delta X(t) = X_2(t) - X_1(t)$, develops and would accumulate without any synchronization treatment. Not only the graphical de-synchronization, the time-delay destabilizes the force interaction between users. Because of delay, the force data is being lagged and interaction forces can easily be out-of-phases. Thus, a special care must be paid while dealing with shared interactive system in the presence of time delay.

For synchronization of shared virtual environment in the *"TransAtlantic Haptic Project"* [Kim *et al.* (2004)], a long distance haptic experiment was done, and a motion synchronization scheme was further combined to better achieve consistency between users, based on a feedback control using Smith principle [Cheong *et al.* (2005)]. The scheme took into consideration the possibility of delay variation, and the robustness bound of the variation was computed. The analysis result showed that, for large controller gain, the synchronization ability was nice but the tolerance level of delay variation was low and vice versa. However, direct user-to-user interaction became easily unstable under much smaller amount of delay like 150ms. And, though the analysis considered the variability of delay, they did not preform real experimentation in environment of variable delay.

Another method for collaborative visualization was addressed by [Li *et al.* (2006)] to present the synchro-

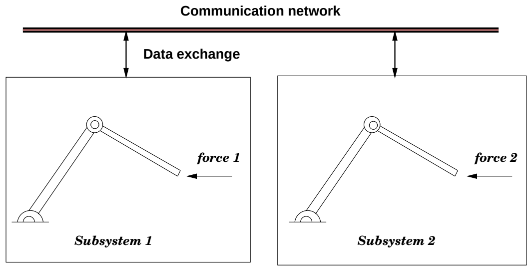


Fig. 2. Configuration of distributed systems via network communication

nized view of the virtual environment. They first considered how interaction of dynamic object is perceived by the remote users and accordingly a trajectory was extrapolated using the information of motion velocity and polynomial based motion model. This work did not consider the possibility of delay variation contrary to the reality of Internet connections.

3. MOTION SYNCHRONIZATION IN DELAYED MEDIA

3.1 Synchronization controller

The synchronization scheme addressed in this section is directly related to the shared virtual environment, where multiple users are interacting with the same dynamics. However, unlike the conventional synchronization, the technological attempt we are interested in pursues a more robust and better synchronization that gives an almost concurrent evolution of motions between the distributed systems in absolute time-frame (earth's time). Physically, the existence of time delay prevents immediate information exchange, which disables concurrent motions between the distributed systems. For example, two mechanical systems shown in Fig.2 cannot yield concurrent evolution of motion because of time delay between them, while a strict motion synchronization (i.e., the concurrency) is very much necessary for a stable direct user-to-user interaction. Physically, the concurrent evolution of motion does not seem possible, but a sophisticated utilization of Smith principle, disturbance estimation and time-delay analysis in the communication channel, and optimized prediction of input sequences may overcome physical delay and allows a near concurrent evolution of motion between the systems.

First, we define two coupled, but distributed systems to be synchronized (shown in Fig.2), modeled as a simple rigid body with viscous damping as follows:

$$\begin{aligned} m\ddot{x}_1(t) + b\dot{x}_1(t) &= f_1(t) + f_2(t-T_2) \\ m\ddot{x}_2(t) + b\dot{x}_2(t) &= f_1(t-T_1) + f_2(t), \end{aligned} \tag{1}$$

where m and b are mass and damping coefficient of the systems, $x_1(t)$ and $x_2(t)$ denote positions of the systems in sites 1 and 2, respectively, and $f_1(t)$ and

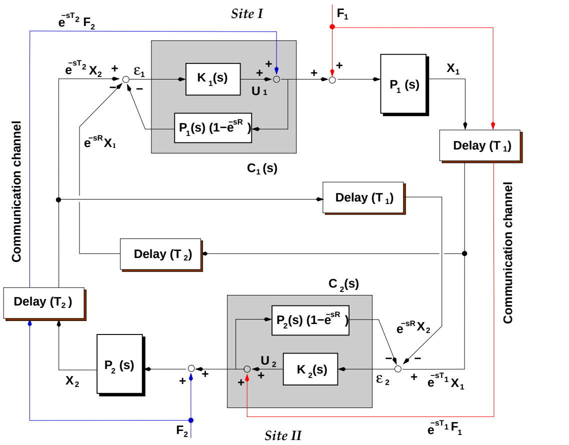


Fig. 3. Synchronization control scheme with two collaborators

$f_2(t)$ are input forces acting on sites 1 and 2, respectively. Constant time delays, T_1 and T_2, represent unidirectional delays for data communication from site 1 to site 2 and from site 2 to site 1, respectively. At this stage we assume the delay is constant, but the later we deals with the effect of variable but a rather smooth delay.

To overcome possible de-synchronization between the sites, we develop a motion synchronization controller which is of the structure shown in Fig.3. This is robust under data loss or any corruption during the communication since we feed back signals and continuously compensate for de-synchronization. This structure also shows the property that natural dynamics of the given system is not affected by the addition of motion control [Cheong *et al.* (2006)]. The controller, $C_i(s)$, in site i consists of primary compensator, $K_i(s)$, that generates ultimate control command and the internal model of dynamics that produces estimated states. Two kinds of state estimations, that is, the current state and the state delayed by R time unit, are generated through the internal dynamics with exact knowledge of dynamic parameters and time delay, similar to Smith predictor [Smith (1957)]. However, the structure is not just a copy of the conventional Smith predictor, but rather we utilize its principle so as to enforce exquisite timing between signals of feedforward and feedback information. For example, our Smith principle is for the canceling feedback and reference input by utilizing the internal model with the knowledge of plant dynamics and amount of delay, with careful consideration of signal timing.

To be more specific, two equations are obtained from the structure as:

$$\begin{aligned} u_i(s) = K_i(s)\left(X_j(s)e^{-sT_j} - X_i(s)e^{-sR}\right) - \\ K_i(s)P_i(s)\left(1-e^{-sR}\right)\left(u_i(s) + F_j(s)e^{-sT_j}\right) \end{aligned} \tag{2}$$

$$X_i(s) = P(s)\left(u_i(s) + F_j(s)e^{-sT_j} + F_i(s)\right) \tag{3}$$

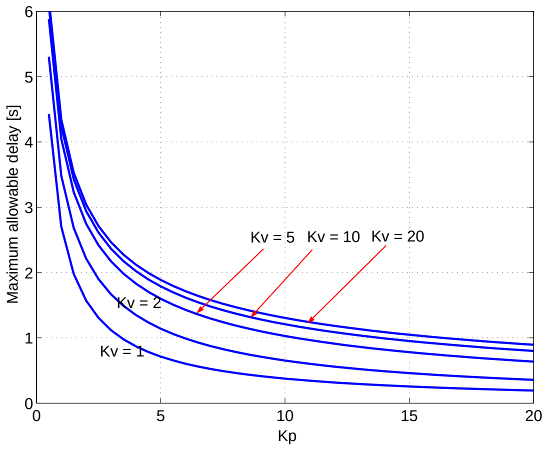


Fig. 4. Numerical values of R_m for $P(s) = 1/(s^2 + 0.01s)$ and $K(s) = k_v s + k_p$.

for $i, j = 1, 2,$ and $i \neq j$, and combining these yields

$$K(s)X_i(s) + P^{-1}(s)X_i(s) - K(s)X_j(s)e^{-sT_j} = F_i(s) + K(s)P(s)(1 - e^{-sR})F_i(s) + F_j(s)e^{-sT_2}, \quad (4)$$

assuming $K_1(s) = K_2(s) = K(s)$ and $P_1(s) = P_2(s) = P(s)$. If writing this in a matrix form after a simple matrix manipulation, we have

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} P(s) & P(s)e^{-sT_2} \\ P(s)e^{-sT_1} & P(s) \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}. \quad (5)$$

As shown, the closed loop system is exactly the same as the natural motion given in (1). This implies that the closed-loop coupled system follows the behavior as natural dynamics of the coupled system shows, while any disturbance effect during communication can be overcome.

This nice and strange property is due to the pole/zero cancelation of the following form of quasi-polynomial factor:

$$\begin{aligned} \Phi(s, R) &\triangleq \alpha(s) + \beta(s)e^{-sR} \\ &= \left\{P^{-1}(s) + K(s)\right\}^2 - K^2(s)e^{-sR}. \end{aligned} \quad (6)$$

The stability condition of the whole system is, thus, equivalent to finding the *delay margin* R_m, for given plant and controller parameters (see, for instance, the analysis suggested by [Niculescu (2001)]). However, the variation of delay creates uncertainty in the above quasi-polynomial and the stability bound in the worst case must be tremendously reduced.

For an illustration, refer to Fig.4, where a numerical values of maximum allowable delay (i.e. delay margin) is computed under $P(s) = 1/(s^2 + 0.01s)$ with different sets of control gains k_v and k_p. Result shows that R_m tends to increase as the proportional gain becomes smaller and velocity gain becomes larger.

It is well known that a linear time invariant delay system is stable if and only if all the roots of its characteristic quasipolynomial have negative real parts. According to the continuity properties of zeros with respect to the delay parameters [Datko (1978)] (see also [Niculescu (2001)], the number of roots in the right-half plane (RHP) can change only when some zeros appear and cross the imaginary axis. Thus, it is natural to consider the *frequency crossing set* (see also, [Morărescu (2006)] and [Gu *et al.* (2005)]) Ω consisting of all real positive ω (obviously, $\omega \in \Omega \Leftrightarrow -\omega \in \Omega$) such that there exist at least a pair (k_v, k_p) for which

$$\begin{aligned} H(j\omega, k_v, k_p, R) := \left(P^{-1}(j\omega) + K(j\omega)\right)^2 \\ -K^2(j\omega)\mathrm{e}^{-j\omega R} = 0. \end{aligned} \quad (7)$$

Using the modulus we arrive to:

$$\left|P^{-1}(j\omega) + K(j\omega)\right|^2 = \left|K^2(j\omega)\right|. \quad (8)$$

In conclusion, Ω consists of the values ω such that $\dfrac{P^{-1}(j\omega)}{K(j\omega)}$ belongs to the circle with radius 1 and centered in $(-1, 0)$. Next,

$$k_p = -\frac{|P^{-1}(j\omega)|^2 + 2k_v Im(P^{-1}(j\omega))}{2Re(P^{-1}(j\omega))} \quad (9)$$

Remark 1. Since $\left|\dfrac{P^{-1}(j\omega)}{K(j\omega)} + 1\right| \to \infty$ when ω approaches ∞ one obtains that there exists $M > 0$ such that $\Omega \subset (0, M]$.

On the other hand, from equation (7) we can derive the following expression

$$\angle(P^{-1}(j\omega)\bar{K}(j\omega) + |K(j\omega)|^2) = -\frac{\omega R}{2} \quad (10)$$

where $\angle(z)$ denotes the argument of the complex number z, Using (9), one can replace k_p in (10) and thus, we obtain a simple polynomial equation of the form:

$$A(\omega)k_v^2 + B(\omega)k_v + C(\omega) = 0$$

where A, B, C are polynomial functions of ω and $\tan(\omega R)$. Imposing that $B^2(\omega) - 4A(\omega)C(\omega) \geq 0$ we get the explicit expression of Ω as an union of intervals of finite lengths. When ω sweeps Ω the corresponding pair (k_v, k_p) defined by (9) and (10) moves on some *stability crossing curves*. Every time the pair of controller parameters (k_v, k_p) crosses such a curve the number of characteristic roots in RHP changes.

We note that for a fixed value of k_v when R rise the stability boundary in terms of k_p become smaller. The same result is depicted in Fig.4.

Now let us find out the effect of unreliability in the data communication. If there is a data loss in the network or disturbance in any form, subsidiary responses from these uncertainties are created and superimposed to the ideal response in (5), which makes

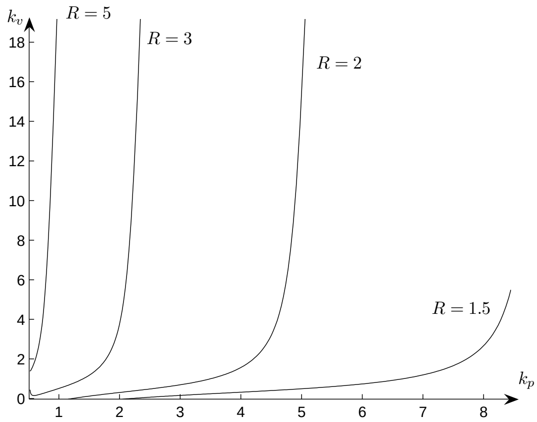


Fig. 5. The geometry of the crossing curves in the parameter space (k_v, k_p)

unexpected de-synchronization. The proposed controller then compensates for the de-synchronization error, where we can observe the transient behavior that the characteristic modes of (6) are involved in. Assume that $f_j(t)$ is the input force at site j and $f_j(t-T_j)$ is the available information of $f_j(t)$ at site i with communication delay, T_j. If $f_j(t-T_j)$, for some reason, is disturbed to $\widetilde{f}_j(t-T_j) = f_j(t-T_j) + \Delta f_j(t-T_j)$, where $\Delta f_j(t-T_j)$ is the amount of disturbance, the closed-loop input-output equation becomes

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} P(s) & P(s)e^{-sT_2} \\ P(s)e^{-sT_1} & P(s) \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} + \begin{bmatrix} \dfrac{K(s)e^{-sT_2}}{\Phi(s,R)} & \dfrac{(P^{-1}(s)+K(s))}{\Phi(s,R)} \\ \dfrac{(P^{-1}(s)+K(s))}{\Phi(s,R)} & \dfrac{K(s)e^{-sT_1}}{\Phi(s,R)} \end{bmatrix} \cdot \begin{bmatrix} e^{-sT_1}\Delta F_1 \\ e^{-sT_2}\Delta F_2, \end{bmatrix} \tag{11}$$

where ΔF_j is the Laplace transform of $\Delta f_j(t)$. In the above, the output response for the transient and finite disturbance dies out whenever $\Delta f_j(t)$ is transient and $\Phi(s,R)$ is asymptotically stable. So, we can say $\Phi(s,R)$ is the *de facto* characteristic function.

Remark 2. To implement the synchronization controller using the network medium, we need a data packet having the following information fields:

```
PACKET = {
        Subsystem 1: time, state, force;
        Subsystem 2: time, state, force;
        }
```

where data field values of time, state, and force at a certain subsystem refer to the corresponding local values of the site. Packets of this simple form are continuously being sent and received via network.

4. SIMULATION STUDY

A set of basic simulation is done to examine two elementary abilities of the proposed synchronization scheme: (i) to preserve natural local dynamics and (ii) to compensate for de-synchronization error caused by mismatched initial conditions. The considered system comprises two connected identical subsystems modeled as $P(s) = 1/(s^2+0.01s)$ with $T_1 = T_2 = 0.15$s, and the synchronization controller is designed followed by structure shown in Fig.3 with $K(s) = 2s+2$. (Here we assume the delay is constant.)

First, we assess the sameness between the uncontrolled natural response and the response with the proposed synchronization scheme to the sinusoidal forces given by

$$f_1(t) = \sin(t),$$
$$f_2(t) = \begin{cases} -\sin(0.4t+1) & 0 < t \leq 32\text{s} \\ 0 & t > 32\text{s} \end{cases} .$$

We set up the system so that the initial condition is the same and no information loss occurs during data communication. According the analysis the controlled response must be the same as that of the uncontrolled natural response, which is verified in Fig.6.

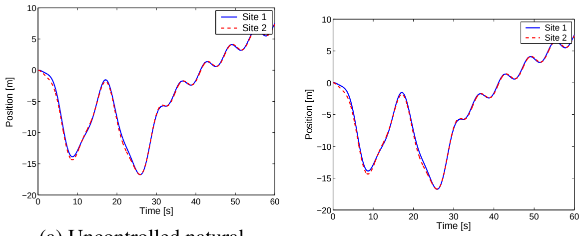
Second we simulate the case where both subsystems have different initial conditions such that

$$x_1(0) = 1,\ x_2(0) = 0,\ \text{and}\ \dot{x}_1(0) = 0,\ \dot{x}_2(0) = 0$$

and no external force is applied here. With no doubt, the natural response without control remains to the initial state, but in the controlled response, however, the differences of initial states are overcome and they become synchronized. Fig.7 shows the transient behavior of synchronization and the applied control forces for the case of controlled motion. The speed and shape of the transient behavior are governed by the roots of quasi-polynomial:

$$\Phi(s,R) = (s^2+2.01s+2)^2 - (2s+2)^2 e^{-0.3s} .$$

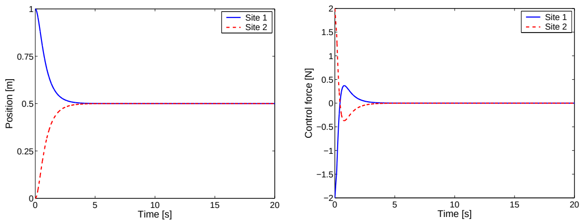
By applying a 5-th order Padé approximation [Franklin *et al.* (1994)] for delay $e^{-0.3s}$, we get nine closed loop poles. Among them the slowest mode, responsible for the sluggish behavior in Fig.7, is from the pole located at $s = -0.2417$. If k_p gain is increased to 4, the speed of response will be faster because the slowest mode becomes located at $s = -0.4822$. However much further increase brings some (other) pairs of complex poles near the imaginary axis, resulting in oscillatory response. A remedy for the oscillation is that k_v gain must be increased simultaneously together with k_p gain. In doing so, we make sure that under the chosen gains the maximum allow delay is sufficiently larger than the current amount of delay.



(a) Uncontrolled natural response

(b) Controlled response

Fig. 6. Comparison of natural and controlled responses



(a) Position

(b) Control input

Fig. 7. Controlled motion for different initial positions

5. CONCLUDING REMARKS

This paper focused on a motion control scheme for synchronization of distributed subsystems connected via communication network. Remarkably the scheme enables us to achieve the property of invariant local dynamics of each subsystem under the operation of feedback control. Due to this property, a near concurrent evolution of motion between subsystems was possible, even in the presence of communication time-delays in the network, by some particular appealing way of a combined utilization of the proposed scheme and an input prediction algorithm. Some illustrative example concluded our presentation.

REFERENCES

Cheong, J., Niculescu, S.-I., Annaswamy, A.M. and Srinivasan, M.A. (2005). Motion synchronization in virtual environments with shared haptics and large time delays. in *Proc. of Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pp. 277-282.

Cheong, J., Lee, S. and Kim, J. (2005). Motion duplication control for coupled dynamic systems by natural damping. in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 387-392.

Datko, R. (1978). A procedure for determination of the exponential stability of certain differential-difference equations. in *Quart. Appl. Math.*, vol. **36**, pp. 279-292.

Franklin, G.F., Powell, J.D. and Emami-Naeini, A. (1994). *Feedback Control of Dynamic Systems*, Addison-Wesley.

Gu, K., Niculescu, S.-I. and Chen, J. (2005). “On stability crossing curves for general systems with two delays,” in *J. Math. Anal. Appl.*, vol. 311, pp. 231-253.

Katz, A. and Graham, K. (1994). Dead Reckoning for Airplanes in Coordinated Flight in *Proc. of the Tenth Workshop on Standards for the Interoperability of Defense Simulations*, pp. II.5-13.

Kim, J., Kim, H., Manivannan, M., Srinivasan, M.A., Jordan, J., Mortensen, J., Oliveira, M. and Slater, M. (2004). Transatlantic Touch: A study of haptic collaboration over long distance. in *Presence: Teleoperators and Virtual Environments*, vol. **13**, pp. 328-337.

Lawrence, D.A. (1993). Stability and Transparency in Bilateral Teleoperation in *IEEE Trans. on Robotics and Automation*, vol. **9**, pp. 624-637.

Li, L.W.F, Li, F.W.B and Lau, R.W.H. (2006). A Trajectory-Preserving Synchronization Method for Collaborative Visualization. in *IEEE Transactions on Visualization and Computer Graphics*, vol.**12**, pp. 989-996.

Morărescu, C.I. (2006). *Qualitative analysis of distributed delay systems: Methodology and algorithms* (Ph.D. thesis, University of Bucharest/Université de Technologie de Compiègne, September 2006).

Murray, R.M. (2002). *Control in information rich world*, Report of the panel on future directions in control, dynamics and systems.

Niculescu, S.-I. (2001). *Delay effects on stability of time delay systems. A robust control approach* (Springer-verlag: Heidelberg, LNCIS, vol. **269**).

Rodriguez-Angeles, A. and Nijmeijer, H. (2004). Mutual synchronization of robots via estimated state feedback: a cooperative approach. in *IEEE Trans. on Control Systems Technology*, vol. **12**, pp. 542–554.

Singhal, S. and Zyda, M. (1999). *Networked Virtual Environments: Design and Implementation*, Addison-Wesley.

Smith, O.J.M. (1957). Closer Control of Loops with Dead Time. in *Chem. Eng. Prog.*, vol. **53**, pp. 217-219.

Taoutaou, D., Niculescu, S.-I. and Gu, K. (2004). Closed-loop stability of a teleoperation control scheme subject to time-varying communication delays. in *Advances in time-delay systems* (S. I. NICULESCU, K. GU, EDS.), Springer-Verlag: Berlin, LNCSE, vol. **38**, pp. 327-338.