

## A gain scheduling strategy for the control and estimation of a remote robot via Internet

Wenjuan Jiang, Alexandre Kruszewski, Jean-Pierre Richard, Armand  
Toguyeni

► **To cite this version:**

Wenjuan Jiang, Alexandre Kruszewski, Jean-Pierre Richard, Armand Toguyeni. A gain scheduling strategy for the control and estimation of a remote robot via Internet. CCC'08 - 27th Chinese Control Conference, Jul 2008, Kunming, Yunnan, China. IEEE, 2008, <10.1109/CHICC.2008.4605866 >. <inria-00266451>

**HAL Id: inria-00266451**

**<https://hal.inria.fr/inria-00266451>**

Submitted on 22 Mar 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A gain scheduling strategy for the control and estimation of a remote robot via Internet

W. Jiang<sup>1</sup>, A. Kruszewski<sup>1</sup>, J.-P. Richard<sup>1,2</sup>, A. Toguyeni<sup>1</sup>

1. LAGIS CNRS UMR 8146, Ecole Centrale de Lille, BP 48, 59651 Villeneuve d'Ascq Cedex, France.

2. Equipe-Projet ALIEN, INRIA.

E-mail: wenjuan.jiang@ec-lille.fr, Alexandre.Kruszewski@ec-lille.fr, jean-pierre.richard@ec-lille.fr and armand.toguyeni@ec-lille.fr

**Abstract:** In this article, a gain scheduling strategy for the controller of a remote robot based on Internet and Bluetooth networks is designed and implemented. The Internet communication is based on the Master-Slave structure, UDP protocol. The Slave comprises a PC and a mobile robot, interconnected through the protocol Bluetooth. The Master is a second PC which realizes the remote control, the design of which is based on a remote observer achieving a state prediction of the robot (Slave), despite the variable communication delays, sampling and packets losses. The detected variable time-delays serve as the switching signals. The gain scheduling state feedback controller is based on Lyapunov-Krasovskii functional and the approach of LMI, which guarantee the *uniform* stabilization performance.

**Key Words:** Remote control, Switching signal, Exponential stability, Linear time-delay system, LMIs, Internet, UDP, Robot.

## 1 INTRODUCTION

As Internet is well developed, remote control system has been widely used in industrial, communicational, medical and even biological systems. However, alongside the advantage of low costs, the Internet inevitably brings problems to the closed-loop controlled system, such as delay variation, data-packets loss (M. Yu *et al.*, 2004) and disorder, which cause poor performance (J.P. Richard and T. Divoux, 2007), instability or danger. How to diminish the effect of time delay in the remote system is critical in the system design. The main solution can split into two (combinable) strategies (J.P. Richard and T. Divoux, 2007; J. Chiasson and J.J. Loiseau, 2007): 1) Increase the network performances(QoS) or 2) design an adapted control that can compensate the network influence. In this article, we consider this last approach for an Internet and Bluetooth network. Our aim is to ensure suitable stabilization and speed performances, i.e. exponential stabilization, despite the dynamic variations of the network. Note that, in the Internet case, the network delays cannot be modeled nor predicted. Moreover, the (variable) transmission delays are asymmetric, which means that the delay  $h_1(t)$  from Master to Slave (shortly, M-to-S), and the return one (S-to-M)  $h_2(t)$  normally satisfy  $h_1(t) \neq h_2(t)$ . Because of this lack of knowledge, predictor-based control laws (E. Witrant *et al.*, 2007) cannot be applied. A delay maximizing strategy (J.P. Richard, 2003; A. Lelevé *et al.*, 2001) (“virtual delay”, “buffer”, or “waiting” strategy) can be carried out so to make the delay constant and known. This requires the knowledge of the maximum delay values  $h_m \geq h_1(t)$  and  $h_m \geq h_2(t)$ . However, it is obvious that maximizing the de-

lay up to its largest value decreases the speed performance of the remote system.

Our solution relies on the theoretical results of (A. Seuret *et al.*, 2004) (exponential stabilization of systems with unknown, varying delays), as well as (A. Seuret *et al.*, 2006) (networked control), the main lines of which will be shortly recalled in the next section. It allows for applying a waiting strategy only to the M-to-S communication, whereas the S-to-M communication takes the information into account as soon as received. In order to enhance the performance of the system, a gain scheduling strategy is adapted according to the variable time-delay of the Internet. The additional contribution of this paper is the design of an adapted computer implementation (M/S structure) based on the UDP protocol and involving lists as buffers. The choice of UDP is preferred to TCP because in our NCS (Networked Control System) situation, packets re-emitting is not needed and setting up the TCP connection between two PCs is time-consuming.

## 2 FEATURES OF THE REMOTE SYSTEM

The remote system is based on Master-Slave structure. In order to simplify the work of the Slave PC, the control and observation complexity is concentrated on the Master. The main features of the system refer to Fig.1. In the system,

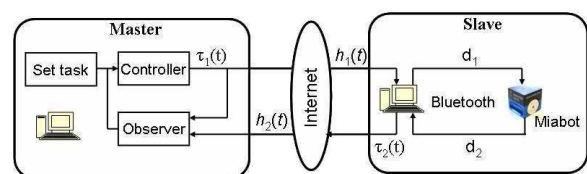


Figure 1: Structure of the global system

the robot Miabot of the company Merlin Systems Corp. Ltd together with a PC serve as the Slave. The Miabot works as an accessory device which communicates with the PC by the port of Bluetooth.

## 2.1 The four delay sources

In such a situation, the variable delays come from: 1) the communication through the Internet; 2) the data-sampling ;3) time delay of the Bluetooth and 4) the possible packet losses. In the sequel,  $h_1(t)$  and  $h_2(t)$  denote the communication delays and  $\tau_1(t)$  and  $\tau_2(t)$  include the sampling delays and possible packet losses. The total Master-to-Slave delay  $\delta_1(t)$  results from the addition of  $h_1(t_k)$  and  $\tau_1(t)$ .

1) Both computers dates are automatically synchronized in the system. The strategy of NTP (Network Time Protocol)(D.L. Mills, 1995) is used in the program to calculate the time clock difference between the two sides. By this way, whenever the Master or the Slave receives the data including the time stamp, it knows the instant  $t_k$  of data sending out and the resulting transmission delay  $h_i(t_k), i = 1, 2$ .

2) The real remote system, including Master, Slave and network, must involve some data sampling. However, following (A. Seuret *et al.*, 2005; E. Fridman *et al.*, 2004), this phenomenon is equivalent to a time-varying, discontinuous delay  $\tau_i(t)$  (to be defined in (1)), which allows for keeping a continuous-time model. If the packets exchange between the Master and the Slave is of high speed, then  $\tau_i(t)$  constitutes a disturbance that should be considered in the stabilization design (M. Yu *et al.*, 2004).  $\tau_i(t)$  is variable but it is supposed there is a known  $T$  (maximum sampling period) so that  $\tau_i(t) \leq T$ .

3) Miabot works as a device of the PC, so we can not use buffer strategy on the robot. Because there is no time information included in the command of Miabot, we do not know when the command has been applied. That means, we can't use time stamp directly on Miabot. To simplify our problem, we treat the time-delay of Bluetooth between the PC and Miabot as a constant one. We add the delays  $d_1, d_2$  into the respectively variable delays  $h_1(t), h_2(t)$ .

4) If some packet  $p_{t_k}$  containing the sample at  $t_k$  is lost, or arrives later than the packet  $p_{t_{k+1}}$ , then the Master only considers the most recent data (i.e., those from  $p_{t_{k+1}}$ ). If it is assumed that the maximum number of successive packets that can be lost is  $N$ , then the maximum resulting delay is  $NT$ . The same lines also holds for the control packets. From 2) and 3), the delay  $\delta_i(t)$  has a known maximum  $\delta_i^m(t) = (N + 1)T + h_m + d_i$  and the delay variation satisfies  $\dot{\delta}_i(t) \leq 1$ . In order to keep simple expressions, the notation  $T$  will be kept preferably to  $T' = (N + 1)T$ .

Summarizing, given a signal  $g(t)$  and the global delay  $\delta(t)$  which represents the combination of the bluetooth delay  $d$ , the sampling and packet loss delay  $h(t_k)$  that the transmission line subjects to the packet containing the  $k^{th}$  sample at time  $t_k$ ,  $g(t)$  can be written as:

$$\begin{aligned} g(t_k - h(t_k)) &= g(t - h(t_k) - (t - t_k)), \\ &= g(t - \delta(t)), \\ t_k \leq t < t_{k+1}, \quad \delta(t) &\triangleq h(t_k) + t - t_k + d. \end{aligned} \quad (1)$$

5) The time-delay of the Internet varies a lot especially between the rush hour and idle time period, so we have designed two controllers corresponding with two bounds of time-delay. The controller switches on the function of time-delay. The switching signal is given by  $\sigma(t) = \gamma(t - \xi)$ , where  $\xi$  is the time-delay of the signal due to the Internet and calculation. In order to guarantee the *uniform* exponential stability, our solution is to find a common Lyapunov function for both closed loops (D. Liberzon, 2003). Of course, for greater delay values, the performance cannot be guaranteed anymore and an alternative solution has to be considered. In our system, we give a command for the robot to stop until the communication comes back to a sufficient quality.

## 2.2 Calculation of the time difference between the M/S

To synchronize the different time in the two PCs, we can add GPS into the system (A. Seuret *et al.*, 2006), but this increases the cost and it is not flexible. Another way is to use a certain protocol such as NTP(Network Time Protocol) (D.L. Mills, 1995), due to different time clock of PCs, we have to make synchronization from time to time. So, our solution is to directly adapt the strategy of NTP in the program to calculate the time differences.

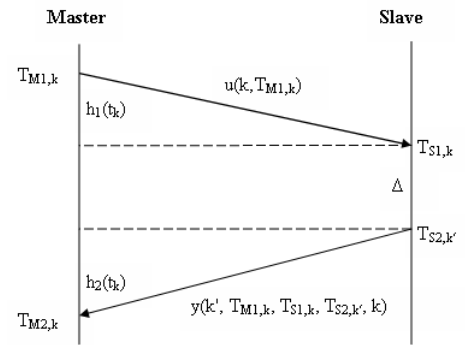


Figure 2: Packets communication between the M/S

As showed in Fig. 2, Here  $k$  is the sequential number of the packets sending from the Master and  $k'$  is the number sent back.  $h_1(t_k)$  and  $h_2(t_k)$  refer to the respective delays of the communication on Internet. To simplify the problem, we assume that  $h_1(t_k) = h_2(t_k)$ . If we define the time clock difference between the M/S as  $\theta$ ,  $\Delta$  is the time for Slave to apply the commands, then

$$\begin{aligned} \theta &= (T_{S2,k'} - T_{M2,k} - T_{M1,k} + T_{S1,k})/2; \\ h_1(t_k) = h_2(t_k) &= (T_{M2,k} - T_{M1,k} - T_{S2,k'} + T_{S1,k})/2; \end{aligned}$$

That is to say, every time the Master receives a packet, the time clock difference between the M/s and the time delay of the Internet can be measured. The values of the time-delay serve as the signals for switching enter the different observers and controllers.

## 2.3 Transmission and receipt of the control data

The  $k^{th}$  data sent by the Master to Slave includes the control  $u(t_{1,k})$  together with the time stamp when the packet is sent out. At time  $t'_{1,k}$ , when the Slave receives the data it can calculate the time delay because of the time stamp. If

the delay is greater than  $h_{1m}$ , the Slave should apply immediately the command.

The control  $u$ , sent out by the Master at time  $t_{1,k}$ , is received by the Slave at time  $t'_{1,k} > t_{1,k}$ . It will be injected in the Slave input only at the pre-defined ‘‘target time’’  $t_{1,k}^{target} = t_{1,k} + h_{1m} + d_1$ . The corresponding waiting time  $h_{1m}$  is depicted on Fig. 3. This is realistic because the transmission delay is bounded by a known value  $h_{1m}$ . By this way, the Master knows the time  $t_{1,k} + h_{1m} + d_1$  when this control  $u(t_{1,k})$  will be injected at the Slave input.

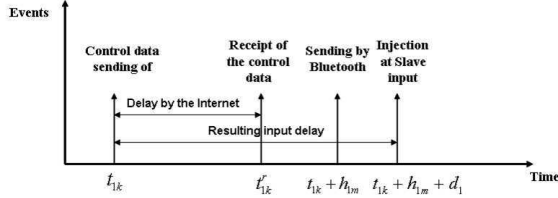


Figure 3: Control data processing

### 3 Problem formulation and preliminaries

Consider the Slave as a linear system. It is described as following form, in which  $(A,B,C)$  is controllable and observable.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t - \delta_1(t)), \\ y(t) = Cx(t), \end{cases} \quad (2)$$

where  $\delta_1(t) = \delta_1 + \eta_1(t)$ ,  $\|\eta_1(t)\| \leq \mu_1$ .

In order to guarantee the closed-loop stability whatever the delay variation, the exponential stability with the rate  $\alpha$  must be achieved. In other words, there must be a real  $\kappa \geq 1$  so that the solution  $x(t; t_0, \phi)$  starting at any time  $t_0$  from any initial function  $\phi$  satisfies:  $\|x(t; t_0, \phi)\| \leq \kappa \|\phi\| e^{-\alpha(t-t_0)}$ . In this paper, it is achieved using a state observer and a state feedback.

To ensure this exponential global stabilization, one can use the results of (A. Seuret *et al.*, 2006) which considers a Lyapunov-Krasovskii functional with descriptor representation:

$$\begin{aligned} V(t) = & \bar{x}_\alpha^T(t) E P \bar{x}_\alpha(t) + \int_{-\delta_1}^0 \int_{t+\theta}^t \bar{x}_\alpha^T(s) R \dot{x}_\alpha(s) ds d\theta + \\ & \int_{t-\delta}^t \bar{x}_\alpha^T(s) S \dot{x}_\alpha(s) ds + \\ & \int_{-\mu_1}^{\mu_1} \int_{t+\theta-\delta_1}^t \bar{x}_\alpha^T(s) R_a \dot{x}_\alpha(s) ds d\theta, \end{aligned} \quad (3)$$

where  $\bar{x}_\alpha(t) = \text{col}\{x_\alpha(t), \dot{x}_\alpha(t)\}$ ,  $x_\alpha(t) = x(t)e^{\alpha t}$  and  $E = \text{diag}\{I, 0\}_{(2 \times 2)}$ .

Because of the separation principle, one can divide the analysis of the global stabilization into two smaller problems: the observer design and the controller design. The result are recalled hereinafter using a non-switching observer/controller.

#### 3.1 Observer design

For a given  $k$  and for any  $t \in [t_{1,k} + h_{1m}, t_{1,k+1} + h_{1m}]$ , there exists a  $k'$  such that the proposed observer is of the form:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t_{1,k}) - L(y(t_{2,k'}) - \hat{y}(t_{2,k'})), \\ \hat{y}(t) = C\hat{x}(t). \end{cases} \quad (4)$$

The index  $k'$  corresponds to the most recent output information that the Master has received. Note that the Master knows the time  $t_{1,k}$  and the control  $u(t_{1,k})$  (see Section 2.3), which makes this observer realizable.

Using the delay (1) re-writing, one obtains:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t - \delta_1(t)) \\ \quad - L(y(t - \delta_2(t)) - \hat{y}(t - \delta_2(t))), \\ \hat{y}(t) = C\hat{x}(t). \end{cases} \quad (5)$$

with  $\delta_1(t) \triangleq h_{1,k} + t - t_{1,k} + d_1$  and  $\delta_2(t) \triangleq h_{2,k'} + t - t_{2,k'} + d_2$ . In other words, the observer is realizable because the times  $t_{1,k}$  and  $t_{2,k'}$  defining the observer delays are known thanks to the time stamps. The system features lead to  $\delta_1(t) \leq h_{1m} + T + d_1$  and  $\delta_2(t) \leq h_{2m} + T + d_2$ .

We define the error vector between the estimated state  $\hat{x}(t)$  and the present system state  $x(t)$  as  $e(t) = x(t) - \hat{x}(t)$ . From (2) and (5), this error is ruled by:

$$\dot{e}(t) = Ae(t) - LCe(t - \delta_2(t)). \quad (6)$$

**Theorem 1** (A. Seuret *et al.*, 2006) Suppose that, for some positive scalars  $\alpha$  and  $\varepsilon$ , there exists  $n \times n$  matrices  $0 < P_1, P, S, Y_1, Y_2, Z_1, Z_2, Z_3, R, R_a$  and a matrix  $W$  with appropriate dimensions such that the following LMI conditions are satisfied for  $j = 1, 2$ :

$$\begin{bmatrix} \Psi_2 & \begin{bmatrix} \beta_{2j} WC - Y_1 \\ \varepsilon \beta_{2j} WC - Y_2 \end{bmatrix} & \mu_2 \beta_{2j} \begin{bmatrix} WC \\ \varepsilon WC \\ 0 \end{bmatrix} \\ * & -S & \\ * & * & -\mu_2 R_a \end{bmatrix} < 0,$$

$$\begin{bmatrix} R & Y \\ * & Z \end{bmatrix} \geq 0,$$

where  $\beta_{2j}$  are defined by:

$$\begin{aligned} \beta_{11} &= e^{\alpha(\delta_1 - \mu_1)}, & \beta_{12} &= e^{\alpha(\delta_1 + \mu_1)}, \\ \beta_{21} &= e^{\alpha(\delta_2 - \mu_2)}, & \beta_{22} &= e^{\alpha(\delta_2 + \mu_2)}, \end{aligned} \quad (7)$$

and the matrices  $Y, Z$  and  $\Psi_2$  are given by:

$$Y = [Y_1 \ Y_2], \quad Z = \begin{bmatrix} Z_1 & Z_2 \\ * & Z_3 \end{bmatrix}, \quad (8)$$

$$\begin{aligned} \Psi_2^{11} &= P^T (A_0 + \alpha I) + (A_0 + \alpha I)^T P + S \\ &\quad + \delta_2 Z_1 + Y_1 + Y_1^T, \\ \Psi_2^{12} &= P_1 - P + \varepsilon P^T (A_0 + \alpha I)^T + \delta_2 Z_2 + Y_2, \\ \Psi_2^{22} &= -\varepsilon (P + P^T) + \delta_2 Z_3 + 2\mu_2 R_a + \delta_2 R. \end{aligned}$$

Then, the gain:

$$L = (P^T)^{-1} W, \quad (9)$$

makes the error (6) of observer (5) exponentially converge to the solution  $e(t) = 0$ , with a decay rate  $\alpha$ .

In the following, the solution of the LMI problem corresponding this theorem is written:

$$L = LMI_{obs}(\mu_2, \delta_2, \alpha) \quad (10)$$

### 3.2 Control design

We first consider a controller  $u = Kx, i = 1, 2$ , i.e. the ideal situation  $e(t) = 0, x(t) = \hat{x}(t)$  and:

$$\dot{x}(t) = Ax(t) + BKx(t - \delta_1(t)). \quad (11)$$

**Theorem 2** (A. Seuret et al., 2006) *Suppose that, for some positive numbers  $\alpha$  and  $\varepsilon$ , there exists a positive definite matrix  $\bar{P}_1$ , matrices of size  $n \times n$ :  $\bar{P}, \bar{U}, \bar{Z}_1, \bar{Z}_2, \bar{Z}_3, \bar{Y}_1, \bar{Y}_2$  similarly to (8) and a  $n \times m$  matrix  $W$ , such that the following LMI conditions hold:*

$$\Gamma_{3i} = \begin{bmatrix} \Psi_3 & \begin{bmatrix} \beta_{1i}BW - \bar{Y}_1^T \\ \varepsilon\beta_{1i}BW - \bar{Y}_2^T \\ -\bar{S} \end{bmatrix} & \mu_1 \begin{bmatrix} \beta_{1i}BW \\ \varepsilon\beta_{1i}BW \\ 0 \\ -\mu_1\bar{R}_a \end{bmatrix} \\ * & * & * \\ * & * & * \end{bmatrix} < 0, \quad \forall i = 1, 2,$$

$$\begin{bmatrix} \bar{R} & \bar{Y}_1 & \bar{Y}_2 \\ * & \bar{Z}_1 & \bar{Z}_2 \\ * & * & \bar{Z}_3 \end{bmatrix} \geq 0,$$

where  $\beta_{1i}$ , for  $i = 1, 2$ , are defined by (7) and

$$\begin{aligned} \bar{\Psi}_3^{11} &= (A_0 + \alpha I)\bar{P} + \bar{P}^T(A_0 + \alpha I)^T + \bar{S} \\ &\quad + \delta_1\bar{Z}_1 + \bar{Y}_1 + \bar{Y}_1^T, \\ \bar{\Psi}_3^{12} &= \bar{P}_1 - \bar{P} + \varepsilon\bar{P}^T(A_0 + \alpha I)^T + \delta_1\bar{Z}_2 + \bar{Y}_2, \\ \bar{\Psi}_3^{22} &= -\varepsilon(\bar{P} + \bar{P}^T) + \delta_1\bar{Z}_3 + 2\mu_1\bar{R}_a + \delta_1\bar{R}. \end{aligned}$$

Then, the gain:

$$K = W\bar{P}^{-1},$$

exponentially stabilizes the system (11) with the decay rate  $\alpha$  for all delay  $\delta_1(t)$ .

In the following, the solution of the LMI problem corresponding this theorem is written:

$$K = LMI_{con}(\mu_1, \delta_1, \alpha) \quad (12)$$

### 3.3 Performance enhancement

In order to reach higher value for the exponential convergence, one proposes switching controller and observer gains. The switching signals  $\sigma_1(t)$  and  $\sigma_2(t)$  chosen are function of the time delays  $\delta_1(t)$  and  $\delta_2(t)$ . For sake of simplicity, they can only take two values:

$$\sigma_i(t) = j, \text{ if } \delta_i(t) \in [h_{Min}^{ij}, h_{Max}^{ij}], i, j = 1, 2 \quad (13)$$

Consider every zone of time-delay, we have to compute the gains  $K_1, K_2$  and  $L_1, L_2$  in such a way that they exponentially stabilize the global Master-Slave-Observer system despite the variable delays  $\delta_1(t)$  and  $\delta_2(t)$ . This global system is:

$$\begin{cases} \dot{x}(t) = Ax(t) + BK_{\sigma_1(t)}x(t - \delta_1(t)) + BK_{\sigma_1(t)}e(t - \delta_1(t)), \\ \dot{e}(t) = Ae(t) + L_{\sigma_2(t)}Ce(t - \delta_2(t)), \\ \sigma_1(t) = \gamma_1(t - \delta_1(t) - \delta_2(t)), \\ \sigma_2(t) = \gamma_2(t - \delta_2(t)). \end{cases} \quad (14)$$

$\gamma_i(t)$  are the detection functions of  $\delta_i(t)$ . These functions are delayed because the Master needs to receive the last packet to calculate the delays. So each gain is activated for a certain period ( $\delta_1(t) + \delta_2(t)$  or  $\delta_2(t)$ ).

Each gain is computed using

$$K_j = LMI_{con}((h_{Max}^{1j} - h_{Min}^{1j})/2, (h_{Max}^{1j} + h_{Min}^{1j})/2, \alpha_{con})$$

$$L_j = LMI_{obs}((h_{Max}^{2j} - h_{Min}^{2j})/2, (h_{Max}^{2j} + h_{Min}^{2j})/2, \alpha_{obs}).$$

A sufficient condition to prove the uniform stability of the switching closed loop is to find a common Lyapunov-Krasovskii functional for all gains. This functional has to take into account all admissible delays i.e.  $\forall \delta_i(t) \in [h_{Min}^{i1}, h_{Max}^{i2}]$ . For each gain Fig. 4 shows the regions where the exponential stability is proven.

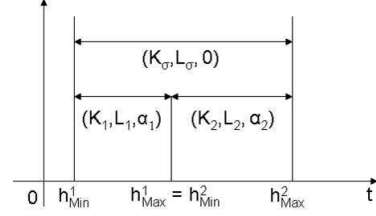


Figure 4: Uniform stability of the system

The following theorems give sufficient conditions to prove the uniform stability of the switching closed loop.

**Theorem 3** *Suppose that, for a given switching observer gains  $L_{\sigma_2(t)}$ , for some positive scalars  $\alpha$  and  $\varepsilon$ , there exists  $n \times n$  matrices  $0 < P_1, P, S, Y_1, Y_2, Z_1, Z_2, Z_3, R, R_a$  such that the LMI conditions 1 and the following ones hold for  $i = 1, 2, j = 1, 2$*

$$\begin{bmatrix} \Psi_2 & \begin{bmatrix} \beta_{2j}P^T L_i C - Y_1 \\ \varepsilon\beta_{2j}P^T L_i C - Y_2 \\ -S \end{bmatrix} & \mu_2\beta_{2j} \begin{bmatrix} P^T L_i C \\ \varepsilon P^T L_i C \\ 0 \\ -\mu_2 R_a \end{bmatrix} \\ * & * & * \\ * & * & * \end{bmatrix} < 0,$$

where the matrices  $\Psi_2$  and  $\beta_{2j}$  are the same as in theorem 1.

Then the error of observer exponentially converge to the solution  $e(t) = 0$ , with a decay rate  $\alpha$ .

Proof: Consider the Lyapunov-Krasovskii functional (3). Following the same proof as in (A. Seuret et al., 2006), one gets the following sufficient conditions for  $j = 1, 2$ :

$$\begin{bmatrix} \Psi_2 & \begin{bmatrix} \beta_{2j}P^T L_{\sigma_2(t)} C - Y_1 \\ \varepsilon\beta_{2j}P^T L_{\sigma_2(t)} C - Y_2 \\ -S \end{bmatrix} & \mu_2\beta_{2j} \begin{bmatrix} P^T L_{\sigma_2(t)} C \\ \varepsilon P^T L_{\sigma_2(t)} C \\ 0 \\ -\mu_2 R_a \end{bmatrix} \\ * & * & * \\ * & * & * \end{bmatrix} < 0,$$

Then by convexity, one obtains the conditions of the previous theorem.

**Theorem 4** *Suppose that, for a given switching state feedback  $K_{\sigma_1(t)}$ , for some positive numbers  $\alpha$  and  $\varepsilon$ , there exists a positive definite matrix  $\bar{P}_1$ , matrices of size  $n \times n$ :  $\bar{P}, \bar{U}, \bar{Z}_1, \bar{Z}_2, \bar{Z}_3, \bar{Y}_1, \bar{Y}_2$  similarly to (8), such that LMI conditions 1 and the following ones hold for  $i = 1, 2, j = 1, 2$*

$$\begin{bmatrix} \Psi_3 & \begin{bmatrix} \beta_{1j}BK_j\bar{P} - \bar{Y}_1^T \\ \varepsilon\beta_{1j}BK_j\bar{P} - \bar{Y}_2^T \\ -\bar{S} \end{bmatrix} & \mu_1 \begin{bmatrix} \beta_{1j}BK_j\bar{P} \\ \varepsilon\beta_{1j}BK_j\bar{P} \\ 0 \\ -\mu_1\bar{R}_a \end{bmatrix} \\ * & * & * \\ * & * & * \end{bmatrix} < 0,$$

where the matrices  $\Psi_3$  and  $\beta_{1j}$  are the same as in theorem 1. Then the closed loop is exponentially stable with the decay rate  $\alpha$  for all delay  $\delta_1(t)$ .

Proof: Same as in the observer case.

## 4 IMPLEMENTATION OF THE REMOTE CONTROL SYSTEM

The transmission protocol UDP is applied to communicate the data between Master and Slave. In order to know the instant of data-sent, time stamps are added to the data packets. The data structure of list served as buffers is introduced for the program to search for the data of the right instance. In all the lists, the control data are restored in the decreasing order of its sending time. That is to say, the most recent sent data is always at the head of the list.

### 4.1 The structure of the Master

In order to implement the model for the remote control system, four-thread program is designed to fulfill the functions of Controller and Observer of Fig.1.

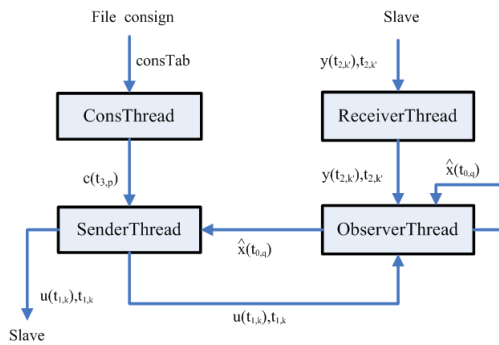


Figure 5: Structure of the Master

These four threads are parallel working as showed in Fig.5. There are two buffers,  $list\_U$  and  $list\_X$  which respectively keep the data sent out from the Master and the data of the estimated state of the process. The most recent calculated data is inserted at the beginning of the lists so that it is easier to find the right data we need and delete the useless ones.

(a) ConsThread gets the tasks (it is the position where the user wants the motor to arrive) from a file given by the user. In this way, the user can freely change the task. The time period  $T_3$  for this thread to work continuously is also set by the user, e.g. 10 seconds.

(b) SenderThread gets the different consign ( $c(t_{3,p})$ ) every time period of ConsThread. Then it calculates the control data to send out according to the following equation:

$$u(t_{1,k}) = K_{\sigma_1(t)}(\hat{x}(t_{0,q}) - c(t_{3,p})). \quad (15)$$

The most recent  $\hat{x}(t_{0,q})$  can be found at the beginning of the  $list\_X$ ; then, the data of command together with the system time is sent out to the slave through the socket. While, at the same time it is inserted into the  $list\_U$  for the ObserverThread to use.

In order to adjust the  $u(t)$  with the value of  $x(t)$  which is the estimated state of the motor, the time period of this thread should be chosen much smaller than that of ConsThread, here 0.1 second is applied.

(c) ReceiverThread is a event-driven thread. As there is data arrived from the slave, it first check whether there is

packet loss. As the time period for the Slave to send out the data is so small that several packets lost will not affect much. Then according to the time stamp, the time clock difference and the time-delay are calculated, meanwhile, the most recent data is sent to the thread of ObserverThread.

(d) ObserverThread is the most important part of the program. It mainly serves as the Observer in the system model. The main task is to estimate the *present* position and speed of the motor. To work it out, it is needed to find out the command  $u$  which has been applied to the slave system and the estimated motor position at the time when the information is sent out from the slave.

As it is illustrated in Fig.6, in order to determine  $\hat{y}(t_{2,k'})$ , it is necessary to find in the  $list\_X$  the closer state estimation  $\hat{x}$  with regard to the date  $t_{2,k'}$ . And we can get the control data  $u$  in the  $list\_U$  with the time stamp of time  $h_{1m}^i (i = 1, 2)$  before. So, according to the equation (4), the estimated state can be obtained. As we can see from Fig.6, in order to find

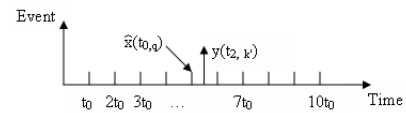


Figure 6: Packet Sequences

the state  $\hat{x}(t_{0,q})$  at the time nearly to  $t_{2,k'}$ , the time period of this thread should be small enough. We choose here 0.02 second. As the results showed after, it is sufficient.

### 4.2 The structure of the Slave

The Slave does not need power computation abilities, which is designed to communicate with the Master and the Miabot. As we can see from Fig.7, this program is divided into two threads: ReceiveThread and SendThread. As we need to apply the control data with the time delay of  $h_{1m}^i (i = 1, 2)$  after the time stamp, a  $list\_Y$  is used to contain the control data temporarily.

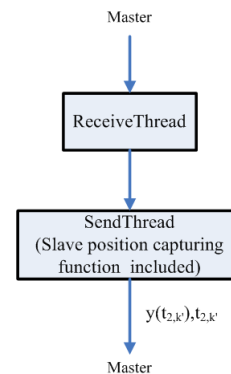


Figure 7: Structure of the Slave

(a) ReceiveThread is an event-driven thread which is activated by the control data arrived from the Master. The control data is inserted into the proper position of the list  $list\_Y$  according to its time stamp. If the time stamp is before the

oldest data of the list, that means there is disorder of the packets through Internet, then the data is discarded. If there are several packets lost, as we have a high frequency of the Master's SenderThread, it does not affect the *uniform* stability of the system.

(b) SendThread includes a function to get the real position of the Miabot. Considering the character of the motor, we choose a time period  $T_2$  for this thread as 0.1 second. In every circulation, it look for the control data to be applied in *list\_Y* and then send it to the Miabot by the port of Bluetooth. The state data of the Miabot is then sent to the Master.

## 5 RESULTS AND ANALYSIS

### 5.1 Parameters of the system

After identification of the Miabot, we get the model of the following form

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -10 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.014 \end{bmatrix} u(t - \delta_1(t)) \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t). \end{cases}$$

The application is implemented in the two PCs which are separated more than 40 kilometers away.

Considering the initial approach, i.e. without switching gains, the maximum exponential convergence obtained is  $\alpha_{control} = \alpha_{observer} = 0.96$ .

Consider two zones of delay with  $\delta_1^1 = \delta_2^1 = 0.04sec.$ ,  $\mu_1^1 = \mu_2^1 = 0.04sec.$ , and  $\delta_1^2 = \delta_2^2 = 0.29sec.$ ,  $\mu_1^2 = \mu_2^2 = 0.21sec.$  It means that the gains switch when the delay crosses the value of  $0.08sec.$  According to *Theorem (3)* and *(4)*, the maximum exponential convergence ensuring the global stability are:  $\alpha_{control1} = 2.1$ ,  $\alpha_{observer1} = 2.2$ , and  $\alpha_{control2} = \alpha_{observer2} = 1$ .

Note that because the global stability is checked after the computation of the gains, these values are not optimal. To get optimal value, it is needed to be able to find the control/observer gains ensuring the exponential convergence and the global stability at the same time with a LMI problem.

The gains  $K_i$  and  $L_i$  ( $i = 1, 2$ ) are:

$$L_1 = \begin{bmatrix} -3.01 \\ -0.77 \end{bmatrix}, K_1 = \begin{bmatrix} -1659 & -260 \end{bmatrix}.$$

$$L_2 = \begin{bmatrix} -1.4 \\ -0.16 \end{bmatrix}, K_2 = \begin{bmatrix} -1015 & -100 \end{bmatrix}.$$

### 5.2 Result of remote experiment

The experiment is done on two computers separated about 40 kilometers away. The Master program runs on the remote computer with an advanced computing capability, the slave one on the local one which also communicates with the Miabot by the port of Bluetooth. The result is shown in Fig. 8, in which the blue curve represents the set values; the green and red represent respectively the robot's estimated position and speed; the black correspond to the real position of the Miabot. Fig.9 is the corresponding variable time-delays, which comprises the time-delay of sampling and communication of Bluetooth (we consider it as

constant time-delay, here we take the value of 40ms). In Fig. 10, the red curve represents the sampled command sent to Slave, and the green and black represent the command for the zone one and two respectively.

Note that all the data in the figure is obtained from the Master, so the data of the real position of the Miabot (curve black) lags behind the estimated one. This illustrates the fact that, despite the time delays of Internet and Bluetooth, the Master estimates a prediction of the Slave's state.

Because the maximal speed of the Miabot is  $3.5m/sec.$ , the command value corresponding is 2000 (in open loop). But to guarantee the linear character of the Miabot, we make the command not surpass 1000 and the speed  $1.7m/sec.$  The controller gains are those of the last section. Despite of their high value, one can notice that the control signal (Fig.10) to not exceed the limit value as well as the speed (Fig. 8 red line) which validates the linearity assumption.

On 8, one can notice three kinds of step response. The first one corresponds to the case where the control switches a lot during the response. In that case, only the global stability is guaranteed. During the second step, only the second zone is active, i.e. only the gains  $K_2$  and  $L_2$  are active ( $\alpha = 1$ ). In this case, some performances are guaranteed. In the last kind of response, only the first zone is active because the delays are small. In that case, the performances are better ( $\alpha = 2.1$ ): the response time is smaller and the damping is greater.

As it is clearly shown in Fig. 8, the global stability of the closed loop is maintained despite that some assumptions made are not satisfied. On the bluetooth, it was considered constant whereas in reality it varies (the minimum delay recorded os less than 40ms). And the other one is on the synchronization, symmetric delays were needed and in the experiment it was clearly not the case.

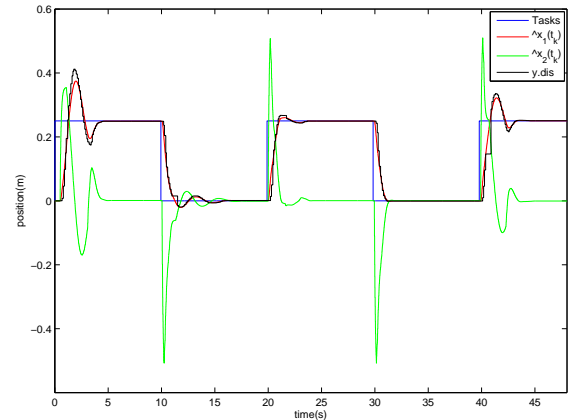


Figure 8: Results of remote experiment

## 6 CONCLUSION

An experimental platform has been developed to illustrate the results of the network-based control theory. This platform has needed a lot of development in order to be able to control a slave through a network and has needed skills in automatic control, computer science and networks.

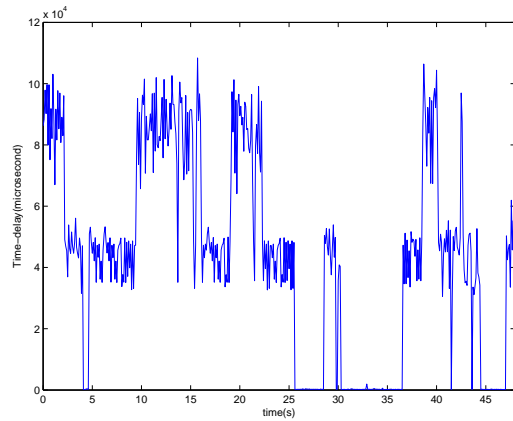


Figure 9: The corresponding variable time-delays

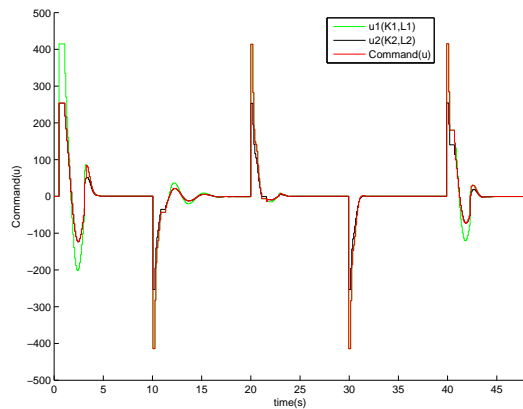


Figure 10: The corresponding Slave control

The experimental results confirm the theory. The exponential stability is obtained in both the time-delay zones and the *uniform* stability is guaranteed. The performances are shown, through our experiment, to be better when considering two zones of time delay.

Considering the variety of time-delay, more than two zones switching signals for the state feedback controller can be selected in order to enhance the performance of the global system. The LMI conditions, in that case, have a bigger size and are directly inspired from *Theorem (3) and (4)* and does not need to be investigated.

A way to improve the presented results is to propose a “one shot algorithm” which allows finding the optimal gains in term of exponential convergence. The next work will investigate on finding a solution without the input buffer systems. Without buffer, the input delay will be smaller ensuring more performances and the slave will need less memory to run.

At last, one consider the improvement of the network communication by for example developing dedicated protocols which minimizes the time delays and enhancing the clock synchronization.

## REFERENCES

- A. Lelevé, P. Fraisse and P. Dauchez (2001). Telerobotics over IP networks: Towards a low-level real-time architecture. *IROS'01 International conference on intelligent robots and systems, Maui, Hawaii*.
- A. Seuret, E. Fridman and J.P. Richard (2005). Sampled-data exponential stabilization of neutral systems with input and state delays. *Proc. of IEEE MED 2005, 13th Mediterranean Conference on Control and Automation, Cyprus*.
- A. Seuret, F. Michaut, J.P. Richard and T. Divoux (2006). Networked control using gps synchronization. *Proc. of ACC06, American Control Conf., Mineapolis, USA*.
- A. Seuret, M. Dambrine and J.P. Richard (2004). Robust exponential stabilization for systems with time-varying delays. *Proc. of TDS04, 5th IFAC Workshop on Time Delay Systems, Leuven, Belgium*.
- D. Liberzon (2003). *Switching in Systems and Control*. Birkhäuser.
- D.L. Mills (1995). Improved algorithms for synchronizing computer network clocks. *IEEE/ACM Transactions On Networking* **3**(3), 245–254.
- E. Fridman, A. Seuret and J.P. Richard (2004). Robust sampled-data stabilization of linear systems: An input delay approach. *Automatica* **40**(8), 1441–1446.
- E. Witrant, C. Canudas-De-Wit and D. Georges (2007). Remote stabilization via communication networks with a distributed control law. *IEEE Transactions on Automatic control*.
- J. Chiasson and J.J. Loiseau (2007). *Applications of time delay systems*. Vol. 352. Springer.
- J.P. Richard (2003). Time delay systems: an overview of some recent advances and open problems. *Automatica* **39**, 1667–1694.
- J.P. Richard and T. Divoux (2007). *Systèmes commandés en réseau*. Hermes-Lavoisier, IC2, Systèmes Automatisés.
- M. Yu, L. Wang and T. Chu (2004). An LMI approach to network control systems with data packet dropout and transmission delays. *MTNS '04 Proc. of Mathematical Theory Networks and Systems, Leuven, Belgium*.