Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006
ThC20.5

Practical Issues in Networked Control Systems

I. Lopez**, J. L. Piovesan*, C.T. Abdallah*
Electrical & Computer Engineering Department
MSC01 1100
1 University of New Mexico
Albuquerque, NM 87131-0001
{ilopez,jlpiovesan,chaouki}@ece.unm.edu

D. Lee, O. Martinez, M. Spong
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 W. Main St. Urbana IL 61801
d-lee@control.csl.uiuc.edu,
{pomartin,m-spong}@uiuc.edu

R. Sandoval
Division of Graduate Studies and Research
Chihuahua Institute of Technology
Ave. Tecnologico 2909
Chihuahua, Chih., Mexico
C.P. 31310
rsandova@itchihuahua.edu.mx

***Abstract*— In this paper we expose experimental issues faced in a closed-loop networked control system. We also propose some compensation actions, and evaluate their performance for different experimental setups, focusing specifically on time delays.**

I. Introduction

Networked Control Systems (NCS) applications such as teleoperation and robot formation, require measurement and control signals to travel across communication links. Even when the distance travelled is short a general purpose communication network will introduce new issues into the feedback loop, such as time-varying delays and the potential loss of information [1], [2]. While some communication applications may suffer from the same limitations, a feedback control system is especially vulnerable to the timing of information and control signals. Therefore, the control algorithms should be robust enough to handle such issues. While earlier research, [9] [5], [4], had clearly identified many of the problems discussed in this paper, we develop in this paper some experiments to also show some compensation approaches when a general purpose communication network is used. In order to focus on generic problems, we use standard operating systems and industrial hardware for data acquisition. Two experimental setups were implemented to test the robustness of the control algorithms when constraints induced by the network are present. The first one involves a teleoperation experiment and the second one a cooperative control experiment.

The remainder of this paper is organized as follows: Section II reviews various issues introduced by a network in a feedback loop. Section III describes the experimental setup and provides the results for the teleoperation experiment while Section IV focuses on the coordination experiment. Finally our conclusions and possible future work is mentioned in Section V.

II. Issues introduced by the LAN

In this section we review and illustrate some of the problems encountered in NCS. While many such problems have been noted earlier, we point to specific problems that arise in TCP-data communication and wireless networks.

A. Retention of Packets

In some NCS, the plant's signals are broadcasted to controllers, or to a supervisory system. With the purpose of measuring the difference in latency for various sizes of Ethernet packets, we devised a experiment where the plant is transmitting packets with sizes from 46 to 1500 bytes, and alternating between User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). With the computer's controller inside the Electrical & Computer Engineering (ECE) building (at the University of New Mexico (UNM)) local area network (LAN), and the receiving computer either inside or outside the same LAN, we did not observe a significant difference in latency when transmitting a single packet (independent of its size and regardless whether UDP or TCP are used). However, when the plant broadcasts packets at a given sampling rate, the "slow start" feature in TCP limited the broadcasting rate to 200 ms, irrespective of the packet size. Even when the signals were sampled at a faster rate, TCP retained the packets until the next multiple of 200 ms. Figure 1 shows the arrival time to the controller's computer of time stamps taken at the plant every 20 ms; 9 packets were retained and at the next multiple of 200 ms, the group of 10 packets were transmitted to the controller's computer. From Figure 1 we see that samples with time stamps from 20 to 200 ms arrived to the controller's computer at $t_c = 200$ ms. This problem however, did not manifest itself with

** The research of this author is partially supported by NSF-0233205 and ANI- 0312611 and CoNaCyt.
* The research of these authors is partially supported by NSF-0233205 and ANI- 0312611.

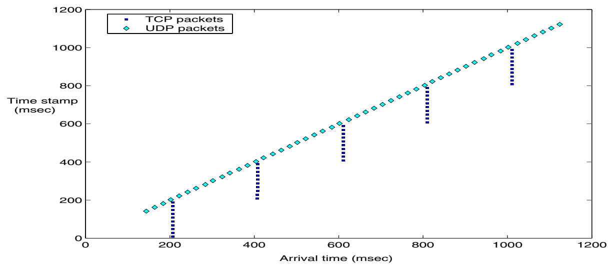

Fig. 1. Arrivals of time stamps using TCP and UDP, sampling at 20msec.

UDP packets which arrived every 20 ms, as sampled. The retention of packets generates later a "bursting" of those packets. If the plant's state samples are not time-stamped, confusion results at the controller's computer as the program simply can not recognize the fresher samples. If bursting occurs, the program in the controller should be able to empty the incoming queue, discard old packets, and only use the last sample of the plant state. We connected the plant's laptop computer to the wired LAN, to verify that this problem occurs with TCP, and not because of the wireless medium. The wired connection also experienced retention of packets when using TCP. *Thus, and because of TCP's slow start, if the broadcast requires sampling times smaller than* 200 *ms, our recommendation is to use UDP, assuming that some lost packets may be tolerated.*

B. Disconnection from the WLAN

In this section, we highlight the disconnection of the plant's computer from the WLAN. This problem is attributed to the re-association procedure that wireless cards execute in order to find the strongest-signal access point. We observed that the disconnection occurs on the average every 60 seconds and lasts 1.5 seconds on average. Figure 2 shows the arrival times of time stamps with a disconnection from the WLAN. The top plot shows a disconnection from the WLAN when using TCP and a sampling time of 200 ms. The sample with time stamp $t_p = 2410$ ms arrives to the controller at $t_c = 2550$ ms, showing a time-delay of $\tau = 140$ ms. This time delay includes the delay due to the asynchronism between the retention feature of TCP and the sampling clock in the plant, plus the propagation time-delay. The next sample with time stamp $t_p = 2610$ ms arrives to the controller at $t_c = 4020$ ms, showing a time-delay of $\tau = 1410$ ms. Subtracting the previous sample time-delay, results in a disconnection time of approximately 1.27 s. The bottom plot in Figure 2 shows the time between two disconnections from the WLAN when using UDP and a sampling time of 200msec. The first disconnection occurred at $t_c = 29133$ ms, while the second disconnection occurred at $t_c = 92296$ ms, resulting in a time between the disconnections of approximately 63.163 s. The time of disconnection, and the period between disconnections seem to be independent of the congestion control protocol and

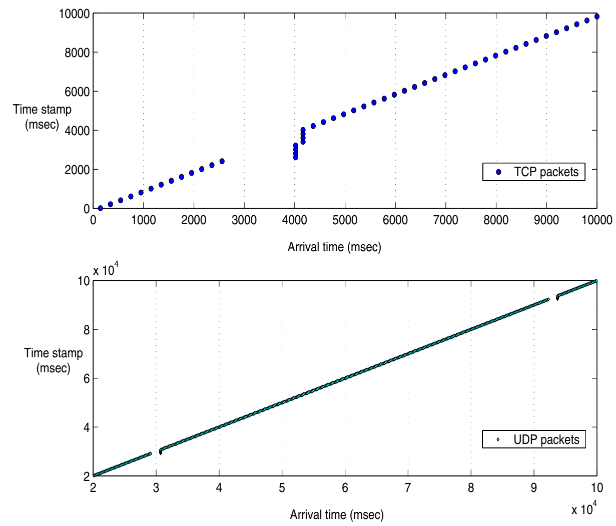

Fig. 2. Disconnection from the WLAN.

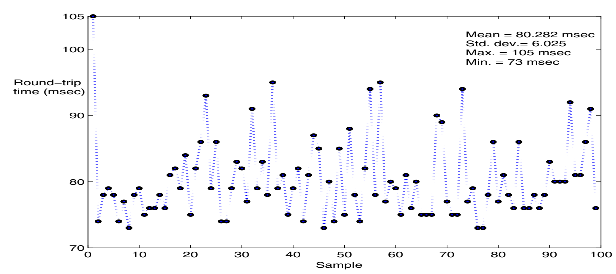

Fig. 3. Round-trip times for 100 samples.

sampling time used.

C. Propagation Time-Delay

For this experiment the controller's computer was connected to a broadband ISP outside the ECE building's LAN, with the purpose of emphasizing the problem of large propagation time delays. We again ran the experiment of reading the plant's clock as a time stamp and sending it to the controller's computer, which sends it back immediately. The plant's computer registers the arrival times and computes the round-trip times. Figure 3 shows the resulting round-trip times of 100 samples. In order to check for symmetry in the channel, we calculated the average arrival time at the controller's computer (41 ms), which was about half the average RTT of Figure 3 calculated a 80.282 ms. We ran these experiments several times at different times of the day. The mean of the round-trip times changed slightly, but the standard deviation was relatively constant. The plant-to-controller and controller-to-plant time-delays were verified to be close, thus establishing that the propagation channel is symmetric. With the purpose of illustrating the effect of propagation time delay and to set a basis for compensation schemes, let us consider the scalar system

$$\dot{x} = ax + bu \tag{1}$$

where $a > 0$, and $b > 0$. Let us also consider state (in this case also output) feedback control with gain K, i.e. $u = -Kx$. The sensing is clock-driven with sampling time t_s, and the control and actuation are event-driven. This means that the controller will compute and send a control signal as soon as it receives a sample, and that the plant will immediately process any received control signal. The time-delay between the plant and the controller is denoted by τ_{pc}, while the time-delay between the controller and the plant is denoted by τ_{cp}, as depicted in Figure 4. At this time, we consider that the combined time-delay is less than the sampling time. We observe that the control signal $u = -Kx[(k-1)t_s]$

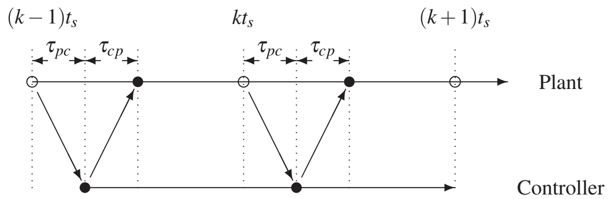

Fig. 4. Time-delay between plant and controller.

arrives to the plant at time $(k-1)t_s + \tau_{pc} + \tau_{cp}$, and is held until time $kt_s + \tau_{pc} + \tau_{cp}$, when it is replaced by the new control signal $u = -Kx[kt_s]$. Thus, two control signals are applied during the interval $kt_s \leq t \leq (k+1)t_s$. Solving for the system's state in equation (1) in the interval $kt_s \leq t \leq kt_s + \tau_{pc} + \tau_{cp}$, yields

$$x[kt_s + \tau_{pc} + \tau_{cp}] = \Phi_1 x[kt_s] + \Gamma_1 x[(k-1)t_s] \tag{2}$$

where $\Phi_1 = e^{a(\tau_{pc}+\tau_{cp})}$ and $\Gamma_1 = -\frac{b}{a}K\left(e^{a(\tau_{pc}+\tau_{cp})} - 1\right)$. Now, solving for the interval $kt_s + \tau_{pc} + \tau_{cp} \leq t \leq (k+1)t_s$, results

$$x[(k+1)t_s] = \Phi_2 x[kt_s + \tau_{pc} + \tau_{cp}] + \Gamma_2 x[kt_s] \tag{3}$$

where $\Phi_2 = e^{a(t_s-\tau_{pc}-\tau_{cp})}$ and $\Gamma_2 = -\frac{b}{a}K\left(e^{a(t_s-\tau_{pc}-\tau_{cp})} - 1\right)$.

Substituting (2) into equation (3), and simplifying $x[(k+1)t_s] = \Psi x[kt_s] + \Upsilon x[(k-1)t_s]$, where $\Psi = e^{at_s} - \frac{b}{a}K\left(e^{a(t_s-\tau_{pc}-\tau_{cp})} - 1\right)$ and $\Upsilon = -\frac{b}{a}K\left(e^{at_s} - e^{a(t_s-\tau_{pc}-\tau_{cp})}\right)$. Consider now the augmented vector $y[kt_s] = \begin{bmatrix} x[kt_s] \\ x[(k-1)t_s] \end{bmatrix}$ leading to the augmented system $y[(k+1)t_s] = \Phi y[kt_s]$, where

$$\Phi = \begin{bmatrix} \Psi & \Upsilon \\ 1 & 0 \end{bmatrix} \tag{4}$$

Thus, given the system parameters a and b, control gain K, and sampling time t_s, there exists an upper bound, τ^*, in the combined time-delay $\tau = \tau_{pc} + \tau_{cp}$, such that if $\tau < \tau^*$ the matrix Φ in equation (4) is Schur. In other words, the system can tolerate the combined time-delay $\tau = \tau_{pc} + \tau_{cp}$, and still converge to the origin. This of course, is more of an analysis result, but forms the basis of a predictive control action that was described by the authors in another paper [9].

III. Teleoperation Experiment

So far, we have described the various issues that arise in a NCS. In this section, we focus on the time-delay issue, in a teleoperation experiment.

A. Experimental Setup

The teleoperation experiment was set up between a PHANToM™ Desktop haptic device as a master device that was locally at the Coordinated Science Laboratory at the University of Illinois, and a slave mobile robot located at the Network Control Systems Laboratory at the UNM. A laptop computer connected to the Internet through an Ethernet Card, was used in the robot. A wired LAN was used instead of the WLAN in order to avoid the disconnection issue highlighted in section II-B. A PCMCIA data acquisition card, DAQ 6024E from National Instruments™, was used to interface the laptop computer to the mobile robot. The software programs used to acquire the state's measurements from the encoders and to apply control signals to the motors, as well as to implement the communication routines, were developed in LabView®, another National Instruments™ product. For the haptic device master station, a PC computer with two Pentium 4 processors at 2.8GHz was connected to the Internet. The control and communication programs in the controller computer were developed using Microsoft Visual C++ v. 6. All computers were running standard Windows XP Professional. UDP was again the transmission protocol chosen to send and receive data from/to haptic device to/from mobile robot through the Internet, to alleviate the bursting phenomenon, described in section II-A.

B. Implementation

It is of course well known that time delay in the communication channel may cause instability either in the teleoperation or in the robot formation algorithm. For the teleoperation experiment, we use the control law proposed in [8], which, by enforcing passivity of the closed-loop teleoperator, ensures stable teleoperation with constant time-delays. Moreover, this control law also addresses kinematic/dynamic discrepancy between the master and slave systems, i.e. master haptic device is holonomic and has confined workspace, but slave mobile robot is nonholonomic and has unlimited workspace [7]. Consider the degrees of freedom defined in figures 5 and 6. We also consider the next model for the the mobile robot: $m\dot{v} = -\eta v + \frac{1}{r}(\tau_r + \tau_l)$ and $J\dot{\omega} = -\psi\omega + \frac{l}{r}(\tau_r + \tau_l)$. Where v, θ are linear velocity and heading angle, m_c, is the cart mass, J is the inertial moment, b is the the half-width of the cart, τ_r, τ_l are the torques for the right and left wheels, η is the viscous friction coefficients and ψ is the rotational friction coefficient (for simplicity, we made two assumptions: wheel inertial equal to zero and the geometrical center of the robot coincides with the center of mass). It was also considered that the robot has the pure rolling non slipping constraint $-\dot{x}sin\theta + \dot{y}cos\theta = 0$, [6].

We determined the parameters of the robot as follows: $m_c = 25$ kg, $J = 1.03$ kgm, $l = 0.203$ m, $r = 0.101$ m, $\psi = 5.51$ kgm/s and $\eta = 133.7$ kg/s. According to [8], we sent $\hat{r}(t) := \dot{r}(t) + \lambda r(t)$ as the reference command for the linear velocity v of the slave mobile robot. Also, the angular position ϕ of the haptic interface was taken as the angular position reference for the slave mobile robot heading angle θ (Figures 5 and 6). The master control law was given by

$$\begin{aligned} T_r(t) &:= -B_r\dot{r}(t) - K_r r(t) - K_{rv}(\hat{r}(t) - v(t-\tau_2)), \quad (5) \\ T_\phi(t) &:= -B_{\phi\theta}(\dot{\phi}(t) - \dot{\theta}(t-\tau_2)) \\ &\quad -B_\phi\dot{\phi}(t) - K_{\phi\theta}(\phi(t) - \theta(t-\tau_2)), \quad (6) \end{aligned}$$

and the slave control law was given by

$$\begin{aligned} T_v(t) &:= -K_{rv}(v(t) - \hat{r}(t-\tau_1)), \quad (7) \\ T_\theta(t) &:= -B_{\phi\theta}(\dot{\theta}(t) - \dot{\phi}(t-\tau_1)) \\ &\quad -K_{\phi\theta}(\theta(t) - \phi(t-\tau_1)). \quad (8) \end{aligned}$$

where $T_\star$ is the control command acting along the $\star$ direction, τ_1, τ_2 are the forward/backward delays, and $K_{rv}, K_r, K_{\phi\theta}, B_{\phi\theta}, B_\phi$ are (positive) control gains. We set the control gains as follows: $B_r = 0.5$, $K_r = 0.001$, $K_{rv} = 100.0$, $B_\phi = 0.1$, $B_{\phi\theta} = 1000.0$, $K_{\phi\theta} = 2500.0$ and $\lambda = 0.04$. Here, λ was determined by trial-and-error without master inertia identification as required in [8], while $B_{\phi\theta}$ was set with the assumption that the maximum round-trip delay can go up to 0.8 sec. For the actual implementation, the round-trip delay between the master and the slave locations was measured repeatedly, and found to have a mean of about 60 ms. In the case of the control law implemented in the robot, the gains were as follows: $K_{rv} = 100.0$, $B_{\phi\theta} = 1\times10^6$ and $K_{\phi\theta} = 2.5\times10^6$.

The force generated at the haptic interface was also scaled to achieve the bilateral power scaling, with which the different size/strength between the master and slave can be matched with each other. In [8], the control law 5-8 was derived for the linear master system with constant delay. However, even with the nonlinear Phantom as the master system and time-varying delays, this control law 5-8 was satisfactory.

In our UDP communication scheme, each packet sent has a unique identification number. Let the packet p_i, $i \in N$ be the one received at time t_i without previously receiving p_{i-1}. If packet p_{i-1} is received a time later than t_i, then p_{i-1} is dropped to avoid time reversing.

C. Experimental Results

To cope with the time-varying delay, two approaches are possible: a) the estimation of the plant state that was explained in section II. C and, b) the addition of a buffer

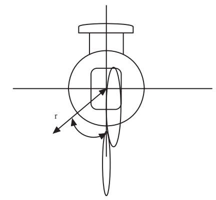

Fig. 5. Master r and ϕ directions

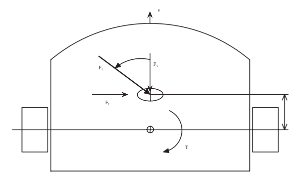

Fig. 6. Slave v and θ directions

[3]. The buffer may be used to save the information that arrives from the opposite side of the teleoperation loop during a time that exceeds the maximum time delay. This information is then feed into the controllers at a constant rate to each controller. By using this method, the time delay may be kept constant at the expense of making it larger.

For the teleoperation experiment, we implemented the buffer idea. However, our control law is capable of producing acceptable performance of the NCS even in the absence of the buffer, and without any other time-varying delay compensation scheme, as shown in Figures 7, and 8, which show the tracking performance of the remote slave robot. The force reflected to the master in this experiment is due to the dynamics of the slave robot. In other words, even though there is no force applied to the robot from an obstacle, gravity, friction and time delays force the robot to have a settling time different from zero. This in turn produces an error between the references sent to the slave and the actual state measurements, which forms the basis for the force reflection control laws designed in equations 5-8. The buffer size was chosen of 20, so this imply a constant time-delay in the loop of almost 20 times the average delay. This size was chosen assuming that neither the network nor the computer

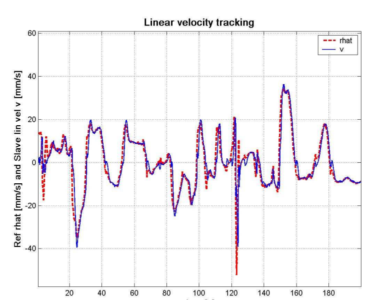

Fig. 7. Linear velocity response.

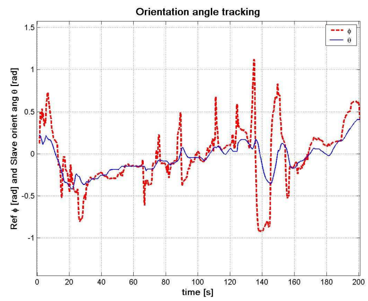

Fig. 8. Heading angle response.

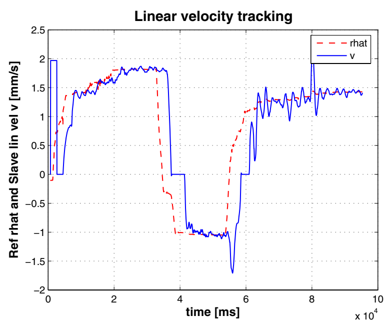

Fig. 9. Linear velocity response using the buffer.

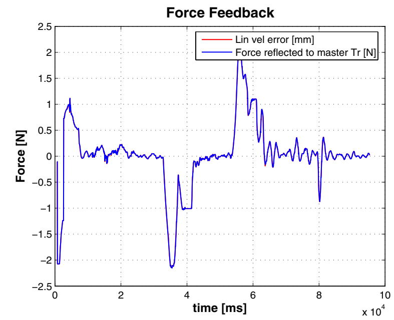

Fig. 11. Force feedback in r direction using the buffer.

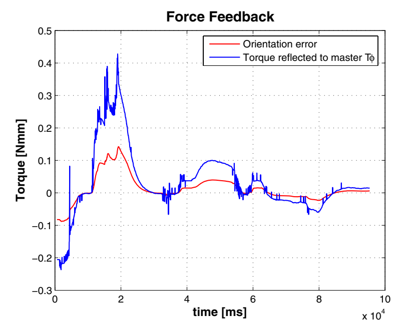

Fig. 12. Force feedback in ϕ direction using the buffer.

processing time will induced any longer delay and it worked reasonable for the experiment. Since the delay time in the loop was incremented by the buffer inclusion, the control gains were tuned again. In the case of the robot control law , the gains were as follows: $K_{rv} = 100.0$, $B_{\phi\theta} = 1 \times 10^5$ and $K_{\phi\theta} = 2.5 \times 10^6$. For the haptic device control law, the gains were: $B_r = 0.5$, $K_r = 0.001$, $K_{rv} = 100.0$, $B_\phi = 0.1$, $B_{\phi\theta} = 100.0$, $K_{\phi\theta} = 2500.0$ and $\lambda = 0.04$.

The experimental obtained by using the buffer are shown in Figures 9 to 12.

From these results we see that the tracking in velocity and angle experience a longer delay, caused by the buffer. However, the tracking in the angle is more accurate than when the buffer was not used.

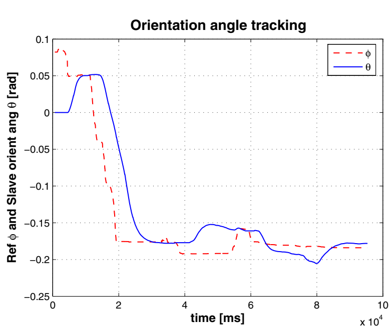

Fig. 10. Heading angle response using the buffer.

IV. Coordination Experiment

The formation task consisted in transmitting a desired trajectory from a central computer (emulating a command center), to the mobile agents via a WLAN, which would build the desired formation, and move along the desired reference.

A. Experimental Setup

The test set-up includes (Figure 13) a computer that acts as central command or virtual leader, a laptop per mobile agent, and a computer that collects and organizes the experimental data. The central command computer essentially converts any operator's command to a reference that can be understood by the agents or generates a task or a set of tasks (if acting autonomously) to be transmitted to the agents. The laptops on top of each mobile agent were used to control, receiving the commands from the central command computer, acquiring the agent's state, calculating the control input and transmitting all the important information to data collection computer. Finally the data collection computer receives all the information from the agents and organizes the data after the experiment is concluded. Communication among the "agents" in the experiment was done via a LAN and Wireless-LAN networks at UNM. Specifically, the virtual leader and the data collector computer were connected to the cabled LAN, while the robots were connected using the wireless LAN (IEEE 802.11b). For the transmission protocol, UDP was again chosen over TCP to avoid the packets retention problem.

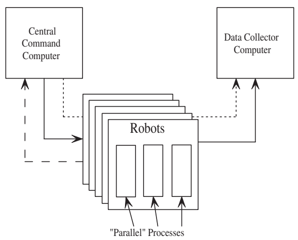

Fig. 13. Experimental implementation structure.

Note in Figure 13 that the reference trajectory was transmitted to the mobile agents who retransmitted this reference together with their state information to the data collector. No information was communicated directly from the central command computer to the data collection computer. An alternative of this approach would have been to acquire the data using an external device such as an overhead camera and a vision estimation system (see for example [10]), but since the robots use dead-reckoning for their current state, the most effective and efficient data acquisition technique is the one used in our experiments.

Two programming styles were tested, and one of them was deemed more effective than the other. The first, and less effective form, used a series of subroutines such that the program would constantly run a single loop that receives the desired data, obtains its current state, calculates and commands the control input and sends all information to the collection computer. With this approach, a failure in any routine (such as in the communication that was the most frequent) would affect the performance of the whole system leading to disastrous results. The second, and successful technique (Figure 13), took advantage of the multithreading capability of LabView®, that allowed us to execute several routines in parallel. The program is composed of several loops, one for each process such as the communication, the state measurement, or the control input calculation, thus avoiding any bottlenecks in the individual processes executions.

B. Relevant Observations

The experimental results for our coordination algorithms are irrelevant for the purposes of this paper (for details on the coordination results see [11]), but several other factors came up during the experimentation that should be commented here. During the various experiments (successful or not) we noted some of issues introduced by the LAN and WLAN addressed in section II. Some of them were so critical that the experiment failed completely due to a complete lack of communication between agents. On the other hand, time delay did not apparently affect the behavior of the system, and packet retention was avoided with the use of UDP as transmission protocol.

V. Conclusions

In this paper, we focused on various issues that arise when experimentally implementing NCS. We first highlighted the issues of bursting which arise with TCP communications, then the disconnection phenomenon associated with wireless communications. Most of the paper and experiments focused on timed-delay issues. We first used a predictive approach on a simple experimental set up to show how time delays may be dealt with. We then focused on a more realistic teleoperation experiment, where a control law that ensures passive bilateral teleoperation of a mobile robot when time delay is present in the communications channel was proposed. This was done based on the dynamics of both master and slave while considering the passivity requirements to ensure stable teleoperation through delayed communications. The inclusion of a buffer give a better response of the teleoperation since the control algorithm was originally designed assuming constant time-delays.

The coordination experiment was not originally designed to test NCS issues, but they came up as a consequence of the nature of the experiment. A possible line of future work would be the study of the network effects on the tested coordination algorithm, and the potential compensatory actions.

References

[1] W. Zhang, M.S. Branicky, and S.M. Philips, Stability of networked control systems, IEEE Control Systems Magazine, 21 (1): 84-99, February 2001.

[2] G.C. Walsh, H. Ye, and L. Bushnell, Stability analysis of networked control systems in Proceedings of the American Control Conference, pp. 2876-2880, San Diego, June 1999.

[3] J. Nilsson, Real-Time Control Systems with Delays, Ph.D. dissertation, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.

[4] P. Hokayem, and C. T. Abdallah, “Inherent Issues in Networked Control Systems: A Survey” in Proceedings of the American Control Conference, pp. 4897-4902, Boston, Massachusetts, July 2004.

[5] F. Lian, J. R. Moyne, and D. M. Tilbury, “Optimal controller design and evaluation for a class of networked control systems with distributed constant delays”, Proceedings of the American Control Conference, Vol. 4, May 2002, pp. 3009-3014.

[6] R. Fierro, F.L. Lewis, “Control of a Nonholonomic Mobil Robot: Backstepping Kinematics into Dynamics”. Journal of Robotics Systems, 14(3):149-163, 1997.

[7] T. Fukao, H. Nakagawa, and N. Adachi. “Adaptive tracking control of a nonholonomic mobile robot”. IEEE Transactions on Robotics and Automation, 16(5):609615, 2000.

[8] D. J. Lee, O. Martinez-Palafox, and M. W. Spong. “Bilateral teleoperation of a wheeled mobile robot over delayed communication network”. In Proceedings of IEEE International Conf. on Robotics & Automation, 2006. Submitted. Preprint also available at http://decision.csl.uiuc.edu/ dlee/ICRA06b.pdf.

[9] R. Sandoval-Rodriguez, C.T. Abdallah, H.N. Jerez, R.H. Byrne, “Experimental Results on the Effects of 802.11b WLAN on Networked Control System”, in Proc. 13th. Mediterranean Conference on Control and Automation, pp. 1113 - 1118, Limassol, Cyprus, June 2005.

[10] T. Chung, L. Cremean, W. Dunbar, Z. Jin, E. Klavins, D. Moore, A. Tiwari, D. van Gogh, S. Waydo, “A Platform for Cooperative and Coordinated Control of Multiple Vehicles”, Technical Report, California Institute of Technology, http://www.cds.caltech.edu/ mvwt/.

[11] J.L. Piovesan, “Formation Control for Mobile Robots: A Tracking Approach”, M.S. Thesis, University of New Mexico, Albuquerque, NM, 2005.