Integrated Computation, Communication and Control: Towards Next Revolution in Information Technology

Feng Xia, Zhi Wang, and Youxian Sun

National Laboratory of Industrial Control Technology,
Institute of Modern Control Engineering
Zhejiang University, Hangzhou 310027, China
{fxia,wangzhi,yxsun}@iipc.zju.edu.cn

Abstract. There is a strong trend in modern industrial systems to integrate computation, communication, and control theories into networked control systems (NCSs). This is anticipated to be the next wave in the information technology revolution. From a control perspective, the interdisciplinary relationship between computation, communication and control is illustrated through control loop timing analysis of NCSs. Critical issues in the emerging field of integrated computation, communication and control (ICCC) are discussed. Since it is difficult to analytically quantify the impacts of computation and communication constraints on the quality of control (QoC), a simulation-based approach is proposed. A numerical example of networked DC motor control is utilized in simulations, with different scheduling schemes and communication protocols employed. Results and analysis give valuable suggests for improving the control performance of NCSs which feature the integration of control with computation and communication.

1 Introduction

With advances in the information technologies of communication, computation and control, a successful implementation of real-time control systems requires a good understanding of not only control theory but also computation and communication theories. It is witnessed [1,2] that the integration of these fields will provide new capabilities. For instance, integration of computation and communication has resulted in the Internet, which provides us the ability to exchange information in the form of email or to share useful resources.

From a control perspective, we observe the trends of integrating computation, communication and control in the field of information technology. Already today networked control systems (NCSs) play an important role in real-time control, and they have attracted much attention from both academia and industry. A networked control system [3,4] is an integration of sensors, controllers, actuators and communication network of certain local field and is used to provide data transmission between devices in order that users of different sites in this location can realize resource sharing and coordinating manipulation. The primary advantages of a networked control system include flexible design, simple and fast implementation, ease of diagnosis and maintenance, and increased agility. The change of communication architecture from point-to-point to common-bus, however, introduces different forms of communication time delay between sensors, actuators, and controllers. The characteristics of this time delay could be constant, bounded, or random, mainly depending on the different

G. Das and V.P. Gulati (Eds.): CIT 2004, LNCS 3356, pp. 117–125, 2004.

features of the employed communication protocol. Particularly, most NCSs are embedded systems that built upon embedded microprocessors. These systems are often subject to hard economic constraints, which in turn give rise to resource constraints on the computing platform level, e.g., limited CPU time and communication bandwidth. In reality this is true in spite of the fast development of computing hardware [5,6]. Furthermore, several tasks may exist in the same microprocessor. And hence, the CPU time constitutes a shared resource that the tasks compete for, just as the network bandwidth for interconnected nodes.

In the context of resource constraints, integrating communication, computation and control into networked control systems is anticipated to be the next phase of the information technology revolution. It is well-known in control systems that time delays can degrade a system's performance and even cause system unstable. In order to guarantee good Quality of Control (QoC), it is necessary to effectively deal with resource constraints using a more holistic approach. The success of this approach involves the confluence of control with computation and communication. The computation theory intended for attacking CPU time constraint covers different real-time scheduling schemes, such as Rate Monotonic (RM) and Earliest Deadline First (EDF) [7]. Different communication protocols including CAN (Controller Area Network) and Ethernet may render different impact on control performance [8].

In this paper, we examine the integration of control with computation and communication, in the context of NCSs and from a perspective of real-time control. Control loop timing analysis of NCS shows the underlying principle of interaction between real-time scheduling, communication protocols and QoC. Trends and issues in the emerging field of integrated computation, communication and control (ICCC) are discussed. Since it is difficult to analytically quantify the impacts of real-time scheduling schemes and network protocols on control performance, we follow a simulation-based approach [6] to analyze the QoC of NCSs with resource constraints. A numerical example of NCS with DC motor problem is utilized in our simulations, where different scheduling schemes and communication protocols are employed. Considering several scenarios with: 1) only computation constraint, 2) only communication constraint, and 3) both computation and communication constraints, we show the impacts of computation and communication constraints with different scheduling schemes and communication protocols on the QoC of NCS. Simulation results and analysis give valuable suggests for the design of NCS which features the integration of computation, communication and control.

The rest of this paper is structured as follows. Section 2 illustrates the interaction between scheduling, communication and real-time control in NCS through control loop timing analysis. Section 3 gives emerging issues in the field of integrated computation, communication and control. The simulation-based approach to analyze the impacts of computation and communication constraints on control performance is illustrated in Section 4. The QoCs based on different scheduling schemes and communication protocols are presented. This paper concludes with Section 5.

2 Integrating Computation, Communication and Control in NCS

In this section, we illustrate, from a control perspective, the interdisciplinary relationship between computation, communication and control through control loop timing analysis of a networked control system. Although typical characteristics of NCS timing problems involve both control delay and jitter, we only focus on control delay in this paper. The reasons for this include: 1) control delay is the primary aspect of NCS timing problems that affect control performance, and 2) the components of control delay can sufficiently explain the relationship between computation, communication and control.

In the control context, a networked control system is typically implemented by a set of computational devices (sensors, actuators, controllers, etc) that run one or several tasks, which communicate data across a field level communication network (fieldbus). Due to data transmissions and controller processing, the control delay of certain control loop includes not only the execution time of controller algorithm and the AD/DA conversion delays, but also the delays associated to the communication network. That is, the control delay [3,6,9] contains both computation delay and communication delay, and will then be:

$$\tau_k = \tau_{sc}{}^k + \tau_c{}^k + \tau_{ca}{}^k \tag{1}$$

where $\tau_c{}^k$ is the computation delay. $\tau_{sc}{}^k$ and $\tau_{ca}{}^k$ are sensor-to-controller delay and controller-to-actuator delay. In (1), the processing delay of sensor has been included into $\tau_{sc}{}^k$, while the processing delay of actuator into $\tau_{ca}{}^k$, since they are relatively constant and small.

In general, the variation of $\tau_c{}^k$ is not substantial compared to $\tau_{sc}{}^k$ and $\tau_{ca}{}^k$ if the controller is designed properly. Therefore, this delay is insignificant in many control techniques. However, in case of implementation upon embedded microprocessor with limited CPU time, the computation delay can be dramatically large. It is cannot still be neglected, and should be treated separately. When designing a networked control system, engineers must reduce this computation delay as much as possible, e.g., by employing certain real-time scheduling schemes in computation community, in order that the resulting QoC is improved. Thus the employed scheduling scheme greatly affects the magnitude of computation delay, which in turn impacts the QoC. In this sense, the integration of computation and control will come to effectively schedule the limited CPU time, in a manner of minimizing the computation delay.

The communication delay, i.e. $\tau_{sc}{}^k$ and $\tau_{ca}{}^k$, includes both the medium access and the message transmission delays. While the message transmission delay is approximately constant, the medium access delay is highly variable as it depends on network characteristics such as communication protocols and available bandwidth [3]. Several fieldbus protocols have been proposed for constructing NCSs, including CAN, Ethernet, and token bus like Profibus. These communication protocols hold different features such as MAC sublayer protocol, packet size, and typical data rate, etc. With a token bus, the variation of $\tau_{sc}{}^k$ and $\tau_{ca}{}^k$ can be periodic and deterministic. While in random access networks such as CAN and Ethernet, $\tau_{sc}{}^k$ and $\tau_{ca}{}^k$ become stochastic processes. That is, CAN and Ethernet may significantly affect the communication delay and the QoC can be impacted. When developing control systems over these protocols, this impact must be well understood.

3 Emerging Issues in ICCC

As mentioned above, this paper is intended to examine the integration of control with computation and communication from a control perspective. Since the primary objective of NCS design is to efficiently use the limited communication and computation capacity while maintaining good control performance, the timing analysis in Section 2 naturally leads to two research directions: 1) integrated control and real-time scheduling, and 2) integrated communication and control.

From a historical perspective, the control community and real-time scheduling community have normally been separated [5,10]. This separation has allowed the control community to focus on its own problem domain without worrying about how scheduling is being done, and it has released the scheduling community from the need to understand what impact scheduling has on the stability and performance of the plant under control. However the separated development of control and scheduling theories has led to a lack of mutual understanding between the fields. In order to effectively cope with computation constraints in NCSs, control and real-time issues should be discussed from the point of view of integration. In recent years, a number of research efforts have been devoted to control and scheduling codesign of real-time control systems. A state-of-the-art on this field can be found in [10].

In the field of integrated communication and control [11], the interaction of between control performance and real-time communication theory must be examined in order to effectively use the communication resources and minimize delays in the design process. A proper communication protocol is necessary to guarantee the network Quality of Service (QoS), whereas advanced controller design is desirable to guarantee the QoC. Because of the integral link between network and control in a NCS, it is important to consider network and control parameters simultaneously to assure both network QoS and QoC. Many works have been conducted on ways to minimize the influence of communication on the QoC of the NCS. For instance, the impact of network architecture on control performance in NCS was discussed in [8] and design considerations related to QoC and network QoS are provided.

Besides the above-mentioned two issues, a more challenging one is to improve control performance by the way of fully integrating computation and communication theories. In NCSs where the microprocessor holds high computing speed and large memory size while the communication bandwidth is limited, is it possible to trade computation for communication in order to minimize the effect of communication delay on system performance and enhance the utilization of computing resources? A feasible approach is to reduce communication by employing state estimators [12]. By using the estimated values instead of true value at every node, a significant savings in the required bandwidth is achieved. At the same time, the problem of how to trade increased communication demands for decreased computation when there is only computation constraint still lacks investigation. In order to develop effective methods for this issue, the interaction between computation and communication and their synthetic effect on control performance should be thoroughly identified.

Another important issue still missing is the accompanying theoretical confluence of computation, communication and control communities. With the integration of control with computation and communication involved in the technological development of NCS, there is a further convergence of theories [2] at a more integrated view of system theory. As far as we know, little work has been reported on this issue to date.

4 Simulation Based Analysis

4.1 Modeling Overview and Simulation Setup

In the NCS model (see Fig. 1) [13] employed for simulations, the time-driven sensor samples the process periodically and sends the samples to the controller over the network. Upon receiving a sample, the controller computes a control signal that is sent to the actuator, where it is subsequently actuated. The threads executing in the controller and actuator nodes are both event-driven. In order to obtain computation constraint, there is an interfering task that may execute in the controller node. A disturbance node with high priority is also introduced to generate communication constraint. The main reasons behind the choice of this model are that: 1) it is in existence in many fields as a typical networked control system, 2) the system components are enough to examine the impacts of computation and communication constraints on control performance, and 3) it can be easily complexified e.g. by extending it with more nodes or adding dependencies across control loops.

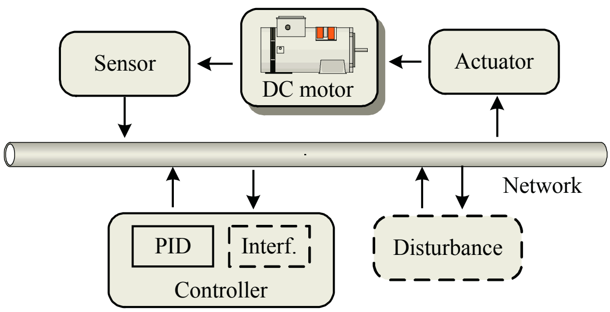

Fig. 1. The NCS setup employed for simulations

We consider networked PID control of a DC servo motor. The goal of the control is to make the servo position *y(t)* follow the reference position *r(t)* as closely as possible. Let the servo model be discribed by the transfer functions:

$$G(s) = \frac{980}{s^2 + 20s} \tag{2}$$

The employed PID controller is implemented [6,13] as follows.

$$\begin{aligned}
P(t) &= K_P(r(t) - y(t)),\\
I(t) &= I(t-h) + \frac{K_P h}{T_I}(r(t) - y(t)),\\
D(t) &= \frac{T_D}{Nh + T_D} D(t-h) + \frac{NK_P T_D}{Nh + T_D}(y(t-h) - y(t)),\\
u(t) &= P(t) + I(t) + D(t).
\end{aligned} \tag{3}$$

The corresponding parameters are K_P=0.9, T_I=0.1, T_D=0.007 and N=10. A sampling period of h=6ms is chosen. Default network is assumed to be of CAN-type with a data rate of 500Kbps. All simulations are made using Matlab/Simulink based on the TrueTime toolbox [13].

4.2 QoC with Computation Constraint

In this subsection, we observe the impact of computation constraint on the system performance. The RM and EDF scheduling schemes are, respectively, utilized to schedule the CPU in the controller node, where an interfering task is introduced. Due to the competition from interfering task, the CPU time dedicated to execute the PID algorithm becomes limited. Let h_i be the sampling period of the interfering task. It is evident that the computation constraint will be more serious as h_i decreases. Therefore, we adjust the value of h_i to achieve different limitation on the CPU time for control.

With the RM scheduling scheme employed, the system responses are given in Fig.2, with different values for h_i. In scenarios holding that h_i>4ms, the control system performs pretty well, and the QoC is almost unaffected by the interfering task. Because the CPU time constraint is nonsignificant in these situations, the PID control task can be guaranteed schedulable. This is especially true when h_i is larger than the control period, since the control task will hold higher priority according to the principle of RM. With h_i increasing, the impact of computation constraint on QoC gradually becomes clear. The network control system becomes unstable when h_i is set to be 3.1ms, although the QoC is satisfactory when h_i=3.2ms. Similarly, Fig.3 shows the system performance when the EDF scheduling scheme is employed. Satisfactory control performance can be achieved given that h_i>3.2ms in the EDF context. The system still remains stable when h_i=3.1ms, and the output response is not bounded as h_i decreases to 3.0ms. Other than comparing the effectiveness of RM and EDF, we attempt to examine the impact of computation constraint on control system performance. As we can see from Fig.2 and 3, the QoC of networked control system can be dramatically affected by computation constraint in a way that this constraint may result in long computation delay, and different real-time scheduling schemes may result in different system performance even under the same constraint.

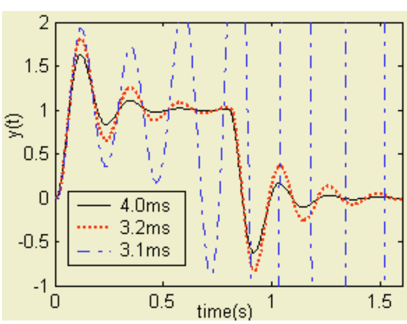

Fig. 2. System responses when CPU is scheduled by RM, with different h_i

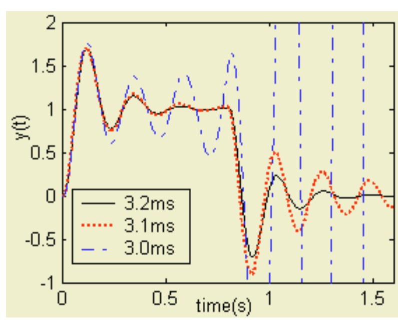

Fig. 3. System responses when CPU is scheduled by EDF, with different h_i

4.3 QoC with Communication Constraint

In order to observe the impact of communication constraint on QoC, the interfering task inside the controller node is removed, getting rid of the effect of CPU constraint. Two popular communication protocols, namely CAN and Ethernet, are involved in this subsection. The bandwidths of CAN and Ethernet are set to be their typical data rates, i.e. 500K and 1M bps, respectively. Interfering traffic with highest-priority will be generated by the disturbance node, which periodically sends certain data (with 10 bytes for CAN and 46 bytes for Ethernet as default sizes) to itself via the network. The expected fraction of the network bandwidth occupied by this node can be specified to reflect the constraint of communication bandwidth. Let r_o denote the ratio (in percentage) of this fraction to the total network bandwidth. Hence the communication constraint will be more serious with bigger r_o value. Note that the value of r_o differs from the runtime bandwidth occupational ratio of the disturbance node, depending on the features of the underlying communication protocol. In the context of CAN bus, the r_o value may be used to represent the runtime bandwidth occupation of the interfering traffic. However, this is not the case for Ethernet. To comply with the MAC sublayer protocol of Ethernet, the disturbance node also needs to compete for communication bandwidth although it holds the highest priority. Therefore, the actual bandwidth occupational ratio of the interfering traffic must be lower than the value of r_o, when Ethernet is used.

Fig.4 presents the system performance with CAN as the communication protocol. Because the amount of data flow in the control loop is relatively small, the system shows satisfactory QoC even when the interfering traffic occupys up to 80% of the communication bandwidth. When the r_o value climbs to 90%, the system will become unstable, due to long communication delay caused by waiting for network access. In the context of Ethernet, we find that the QoC remains acceptable even when the r_o value is set to be 99.99%. This is mainly because the actual bandwidth occupational ratio of the interfering traffic is remarkably lower than the value of r_o, and the control loop always holds chances to access the Ethernet. In order to examine how different communication constraints impact QoC, we adjust the data size in the interfering traffic, while setting r_o=99%. As shown in Fig.5, the system performance almost remains unaffected with a data size smaller than 65 bytes. When the data size reaches 66 bytes, the system goes unstable. The reason behind this involves that the actual

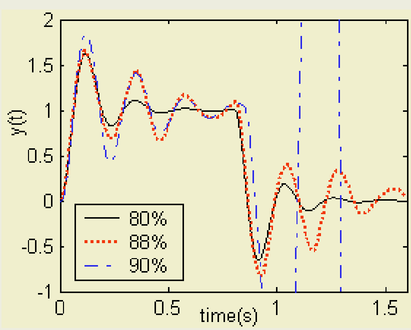

Fig. 4. System responses over CAN, with different r_o values

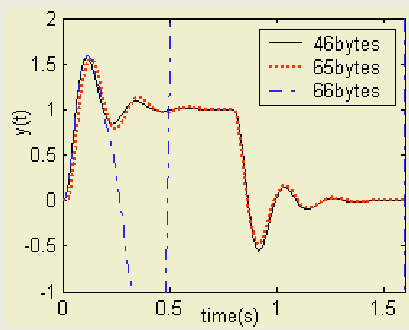

Fig. 5. System responses over Ethernet, with different interfering data sizes

bandwidth occupation of the interfering traffic is strongly enhanced with larger data size, thus increasing the communication delay in the control loop.

4.4 QoC with Computation and Communication Constraints

In this subsection, we focus on the integrated impact of CPU computation and network communication constraints on QoC. Both the interfering task and the disturbance node are introduced. The limited CPU is scheduled by EDF, and the network is of CAN-type. The values of h_i and r_o are selected as the basic factors that affect the resulting QoC, and will be adjusted to achieve different computation and communication constraints. The generally used performance criteria IAE [14] is employed to measure the QoC, which is given by

$$IAE = \sum_{k=0}^{K} | y(kh) - r(kh) | \times h \tag{4}$$

where K is the final time of the evaluation period in discrete time.

In Fig.6, QoCs have been evaluated with IAE values for different h_i and r_o. As we can see, the control performance can be greatly affected by both computation and communication constraints. Serious constraints may cause the system unstable, which is represented with an infinite IAE value. When the impact of a certain constraint significantly outweighs another, there may exist approaches to improving QoC in the sense of reducing IAE. These approaches involve achieving a tradeoff between computation constraint and communication constraint through proper combination of advanced technologies from computation and communication communities.

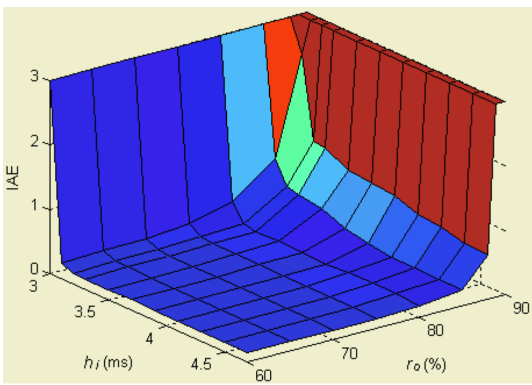

Fig. 4. IAE values as QoC, with different h_i and r_o

5 Conclusions

It is envisioned that the fields of communication, computation and control will converge. As can be seen from NCS, this convergence tends to provide new capabilities. In this paper, we observe the trends of integration of control with computation and communication, from a control perspective. Issues in this emerging field are discussed. The interaction between computation, communication and control is examined and analyzed, using a simulation-based approach. Some useful suggestions for improving the overall system performance are also described.

Due to the high complexity of the interdisciplinary relationship between computation, communication and control theories, many works require to be done. Our future efforts in this field will focus on design methodology for NCS, tradeoff between computation and communication constraints, and control-oriented scheduling schemes.

Acknowledgement

This work is supported by National Natural Science Foundation of China (NSFC 60084001, 60203030, 60304018), National Basic Research Program of China (2002CB312200) and Advance Research Program of France-China (PRA SI03-02).

References

1. Scott Graham and P. R. Kumar, "The Convergence of Control, Communication, and Computation", pp.458–475, Lecture Notes in Computer Science, Volume 2775, Springer-Verlag, Heidelberg, 2003
2. Scott Graham, Girish Baliga and P. R. Kumar, "Issues in the convergence of control with communication and computing: Proliferation, architecture, design, services, and middleware", to appear in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Bahamas, Dec. 2004
3. Mo-Yuen Chow and Yodyium Tipsuwan. "Network-Based Control Systems: A Tutorial", in *Proc. of The 27th Annual Conf. of the IEEE Industrial Electronics Society*, 2001, 1594-1602
4. Linda G. Bushnell. "Networks and control" (Guest editorial), *IEEE Control System Magazine* (Special section), vol. 21, no.1, pp. 22-23, February 2001.
5. Karl-Erik Årzén, Anton Cervin, Dan Henriksson. "Resource-Constrained Embedded Control Systems: Possibilities and Research Issues", in *Proc. of CERTS'03 – Co-design of Embedded Real-Time Systems Workshop*, Porto, Portugal, July 2003.
6. Feng Xia, Zhi Wang, and Youxian Sun. "Simulation Based Performance Analysis of Networked Control Systems with Resource Constraints", to appear in *Proc. of the 30th Annual Conference of the IEEE Industrial Electronics Society*, Busan, Korea, Nov. 2004
7. C. Liu and J. Layland. "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment", *J.ACM*, 20, 46-61. 1973
8. Feng-Li Lian, "Analysis, Design, Modeling, and Control of Networked Control Systems," Ph.D. dissertation, University of Michigan, 2001.
9. B. Wittenmark, J. Nilsson, and M. Torngren. "Timing problems in real-time control systems". In *Proceedings of American Control Conference,* pp: 2000-2004, 1995.
10. K-E. Årzen, et al. "Integrated Control and Scheduling". Research report ISSN 0820-5316. Dept. Automatic Control, Lund Institute of Technology, 1999
11. Y. Halevi and A. Ray, "Integrated communication and control systems: Part I – Analysis," *J. Dynamic Syst., Measure. Contr.*, vol. 110, pp. 367-373, Dec. 1988.
12. J. K. Yook, D. M. Tilbury, and N. R. Soparkar. "Trading Computation for Bandwidth: Reducing Communication in Distributed Control Systems Using State Estimators", *IEEE Trans. on control systems technology*, Vol.10, No. 4, July 2002, 503-518
13. Anton Cervin, et al. "How Does Control Timing Affect Performance?", *IEEE Control Systems Magazine*, 23:3, pp. 16–30, June 2003
14. Feng Xia, Zhi Wang and Youxian Sun. "Design and Evaluation of Event-driven Networked Real-time Control Systems with IEC Function Blocks", to appear in *Proc. of IEEE International Conf. on Systems, Man and Cybernetics*, Hague, Netherlands, Oct. 2004