2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Bi-Directional Communication among "Smart" Components in a Networked Control System

Melih Cakmakci and A. Galip Ulsoy,
Department of Mechanical Engineering,
University of Michigan, Ann Arbor, MI, 48109-2125.

***Abstract*— Increased availability of electronics at much lower costs have helped to create a new breed of control system components; so called "smart" components, which can perform control responsibilities in the actuator and sensor components as well as in the controller. "Smart" components can communicate bi-directionally in networked control systems. Our results show that an "ideal" system with bi-directional communications is equivalent to a single loop control system in terms of performance. However, performance improvements are possible in "non-ideal" cases. Using bi-directional communications can also improve the control system design process since it facilitates the configuration and re-configuration. Building cost effective modular control systems using bi-directional controls appears increasingly attractive.**

I. INTRODUCTION

Components of a feedback control system, usually one per feedback loop, are controllers (i.e. "brains"), actuators (i.e., "brawn"), sensors (i.e., "senses") and the controlled system, or plant. Traditionally actuator and sensors are considered "brainless" devices which perform actuation and sensing tasks. However with the increased availability of electronics at much lower costs many commercially available sensors and actuators currently have on-board computers (i.e., CPU, memory, I/O interface) which enable them to perform diagnostics and component specific control functions such as mapping, filtering noise and saturation. Figure 1 compares the block diagram representations for traditional feedback control systems which have no control algorithm in the actuator and sensor devices, to emerging systems with so called "smart" components which can perform control responsibilities in the actuator and sensor components as well as in the controller.

Traditionally, the three components are sequentially connected via dedicated wiring. However, the current trend is to close feedback loops via communication networks. Such control systems, whose feedback loop is closed via a network, are called Networked Control Systems (NCSs) [1]. Although using networks provides many benefits such as less wiring, better interfacing, lower costs [1,2] and a open architecture, there are some disadvantages such as communication delays, bandwidth and non-delivery of the message carrying the loop information from one component to the other [1-4].

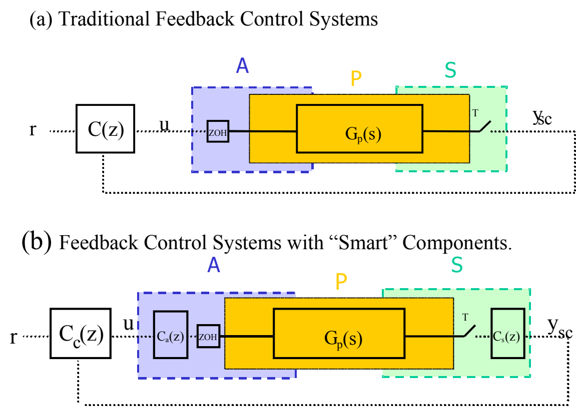


Figure 1 Evolution of Control System Components: (a) Traditional Feedback Control Systems. (b) Feedback Control Systems with "Smart" Components.

Current research on NCS has primarily focused on understanding the effects of network delays. Yook and co-workers present a mathematical framework to analyze the effect of time delays on the mechanical performance of the distributed control systems [4]. This framework is then used to evaluate different control schemes and choose the best one. Lian, and co-workers present performance evaluation of three control networks: Ethernet, ControlNet and DeviceNet based on characteristics and requirements of control systems [5]. They conclude that different types of network protocols are suitable for different control applications. For example, Ethernet can be used for aperiodic/non-time critical and large data size communication, DeviceNet for short and/or prioritized messages and ControlNet for time-critical, non-time critical and large data size communication. Zhang and co-workers state that conventional control theories such as synchronized control and non-delayed sensing and actuation must be reevaluated before they can be applied to NCSs [1]. A NCS model with packet drop out and multiple packet transmission as an asynchronous system is given and the system stability at different data loss rates is investigated.

Walsh and Ye demonstrate that improved performance is possible by dispensing with queues and dynamically scheduling the network traffic in networked control systems [2]. Lian and co-workers provide design considerations related to control quality of performance as well as network quality of service [6]. Performance comparison of continuous control, digital control and networked control cases are given. It is also stated that although networked control systems are a class of digital control systems, by definition their performance degrades when the sampling rate is increased due to increased message traffic among components. Improvement opportunities for NCS performance is stated in two areas: (1) decrease delay due to hardware components and (2) design of the controller so that performance loss due to uncertainty is minimized. Walsh, Ye and Bushnell propose a new protocol, Try-Once-Discard, to improve NCS performance and provide a proof of stability for this protocol and the more common statically scheduled protocols [7].

As discussed in the articles mentioned above, uncertainty of information exchange (packet loss, delay, etc.) is the main focus of researchers interested in NCSs. It seems inevitable that with the improvements to network hardware and network protocols there will be increasing utilization of networks for real-time applications. One good example of currently available hardware (at very high cost) is the SCRAMNet+ Network provided by Curtiss Wright Controls Inc. [8], which enables remotely located components to share a global memory. ControlNET [9] network protocol designed specifically for real-time control applications is mentioned in Lian et. al [5] as suitable for time-critical, non-time critical and large data size communication.

Control engineers have not conducted significant research on the communication flexibility emerging from having control system components use networks instead of dedicated wiring which runs from one specific component to another. There has been research on sensor and actuator networks [10-11] and networks of controllers [12]. However, there is very little research on the effects of information exchange between "smart" components in a networked feedback control system. The purpose of this article is to investigate the potential benefits of bi-directional communication in a feedback control loop, which is the heart of so many control applications. First we define what we mean by bi-directional communications in networked control systems. Next we discuss the potential benefits of bi-directional communications in terms of traditional control-loop performance, as well as the control system design and life-cycle economics. We present our initial conclusions on bi-directional communications using NCSs in the final section of this article. In terms of the performance of an "ideal" traditional feedback control system, we show that when all the components are linear and time-invariant (LTI) there is no performance benefit due to bi-directional communications. However, some benefits of bi-directional communications may be realized in the non-ideal case that includes actuator nonlinearities, sensor noise, etc. Furthermore, we discuss the considerable potential benefits of bi-directional communications in the higher-level functions of NCS design and operations.

II. Defining Bi-directional Communications among Control System Components

Figure 1 includes a block diagram representation of systems which have control algorithm execution capability in actuators and sensors as well as the controller. Traditionally a feedback control system has two types of communication: measured plant output from sensor(s) to the controller (y_{sc}) and a command signal issued by the controller to the actuator(s) (u_{ca}). With the bi-directional communication capability of using networks, it is now possible to define four additional communication paths among the controller, the "smart" actuator and the "smart" sensor: (1) communication from controller to sensor(s) (u_{as}), (2) communication from actuator(s) to sensor(s) (y_{as}), (3) communication from actuator(s) to controller (y_{ac}), and finally (4) communication from sensor(s) to actuator(s) (y_{sa}).

In Figure 2 we incorporate these new communication paths, arising from bi-directional communications in an NCS to the original feedback controller block diagram with smart components.

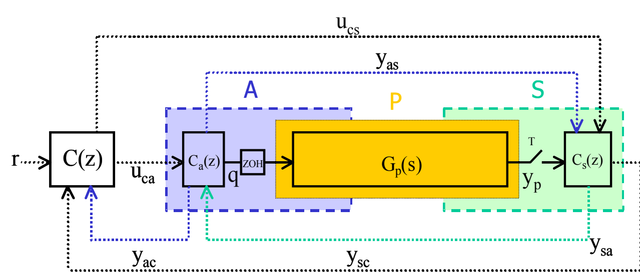


Figure 2 Networked Control System with Bi-Directional Communications among Smart Components.

As a first step to analyze these new communication paths, and their effect on the feedback control system performance, we model the whole system as a linear time-invariant (LTI) system with no communication loss or delays (see Figure 3). Note that it is assumed that the "smart" components do not *hide* internal states from each other. In other words, we do not constrain what can be communicated among components.

In such an "ideal" case, the controller using bi-directional communications in a LTI system can in fact be reduced to a single loop controller as shown in Figure 10. The analytical derivation of this interesting (i.e., non-intuitive) fundamental result is lengthy, but straightforward, and is summarized in the Appendix.

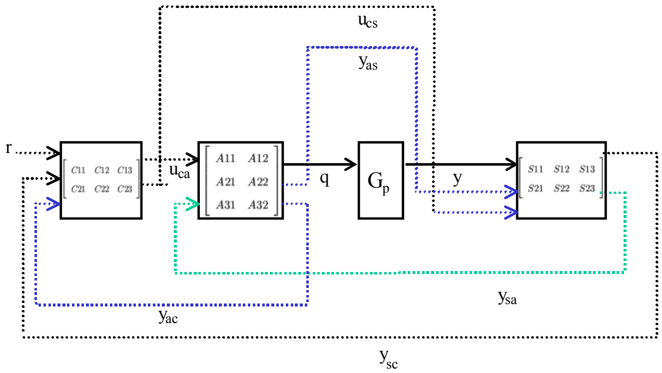


Figure 3 Bi-Directional Communications in a "Ideal" Continous Control System

The conclusion is that, assuming all components are LTI, the performance of an NCS with "smart" components and bi-directional communications is equivalent to, but not improved over, the performance of a feedback system with no bi-directional communication.

In the following sections, we consider the potential benefits of bi-directional communications among "smart" components over a network in more general contexts. In the next section, we consider a "non-ideal" version of the system in Figure 3, which may include, for example, actuator non-linearity, sensor noise, communication delays etc. However, we will assume the effect of communication delays is insignificant, because of expected hardware and network software improvements over time and because the single feedback loop system is localized (i.e. components are connected to the same hub/bus). For cases where communication delay is significant, the addition of new paths may introduce performance degradation due to increased network traffic. In subsequent sections, we also discuss potential benefits of bi-directional communications for the control system design, as well as configuration, reconfiguration, etc. over its life-cycle.

III. IMPROVING TRADITIONAL CONTROL SYSTEM PERFORMANCE USING BI-DIRECTIONAL COMMUNICATIONS:

We have shown in the previous section that for an "ideal" LTI system, there is no benefit in terms of traditional performance of using bi-directional communications since this new controller structure can be simplified to a traditional single loop feedback controller. There are, however, some benefits when "non-ideal" (e.g., non-LTI) systems are considered.

A full-state feedback controller with state estimation is a very common controller structure and implemented successfully for many applications. For cases with actuator saturation communication of the actual actuator output will improve the performance of the state estimation, and therefore, the overall system performance. Such a system controller is shown in Figure 4. For other controller structures this output can also be used to reset integrators (i.e., anti-reset windup) when saturation is detected by comparison in the controller (e.g., see Figure 9).

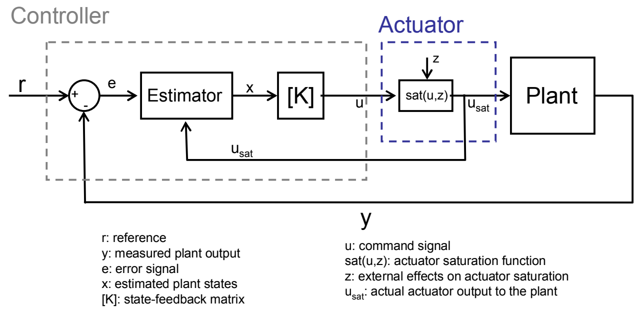


Figure 4 State Feedback Controller with Saturation Feedback from the Actuator.

It is also possible to use the communication path from the sensor to the actuator in order to create a "shortcut" in the feedback loop to bypass the controller. The controller will still dictate the final value that the actuator should apply, but when actuator receives the same message going to the controller, it uses the information as a "heads-up" to improve its overall response time. Figure 5 (a) shows such a controller architecture and Figure 5 (b) compares the actuator response times and presents the improvement. Since the actuator predicts the direction of the next command signal by using the previous command value and the sensor measurement, there will be an improvement in the actuator response time (denoted as T_i).

Another example of the benefits of using bi-directional communications can be given for cases when messages from the controller and the actuator are sent to the sensor in order to improve the noise reduction at the sensor. Controller and actuator outputs can be sent to the sensor and, by using a plant model, an *expected* plant output with certain amount of variance can be calculated without the need for complex actuator and controller algorithm models at the sensor.

Although it is possible to find many applications of bi-directional communications among "smart" components of a feedback controller to improve traditional performance, these improvements highly depend on the algorithms being used and the overall operating conditions.

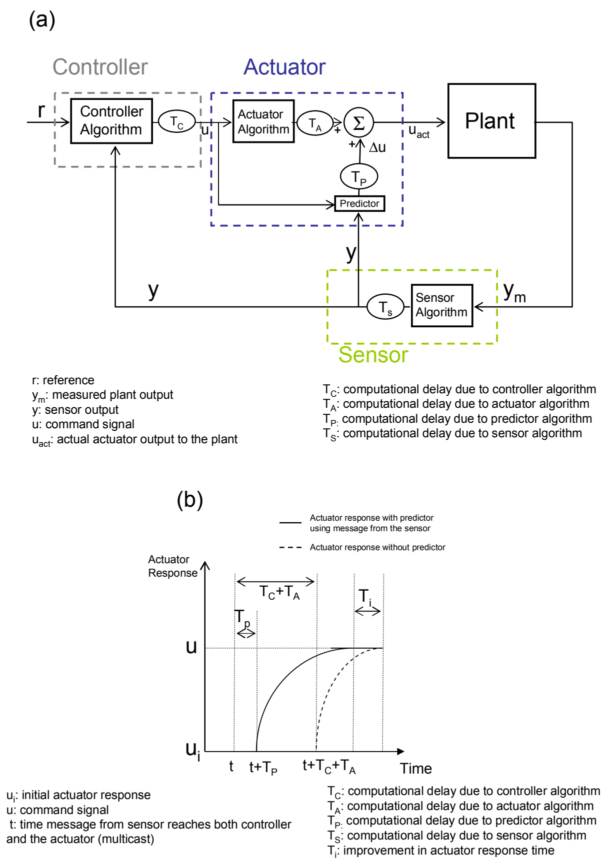


Figure 5 Improving Actuator Response Time by Bi-Directional Communications (oval blocks represent computational delays).

IV. Improving Control System Design Process Using Bi-Directional Communications:

Besides the benefits of improving traditional performance for non-ideal systems, the control system design process can be improved by using the flexibility introduced by using networks using dedicated wiring. This may happen in two ways: (1) automating some lengthy but critical steps of design and implementation, and (2) initially design and partition the control algorithm among components such that life-cycle cost of the system is minimized due to increased modularity.

The control system design and implementation phase has many tedious steps which are prone to simple but detrimental mistakes such as using incorrect sensor calibration data or actuator saturation limits. Many of these steps must be repeated when a change in the system configuration occurs. Figure 6 presents two layers of a NCS, a physical layer where the plant including actuator and sensor physical hardware is located, and the network layer where all the digital communication among the components take place. A new component, the NCS configurator gathers information from the network about individual components, monitor the system for hardware changes and distribute this information as needed. This new device, or computing thread, also can be used to build systems which handle component and algorithm faults better (i.e. graceful degradation). The RoSES (Robust Self-configuring Embedded Systems) project employs different reconfiguration approaches to build self-customizing, distributed, embedded control systems [13-15].

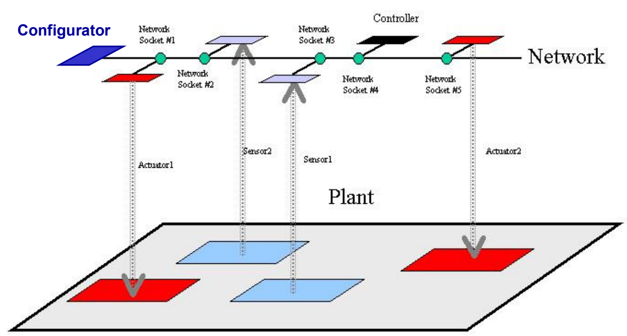


Figure 6 Network and Physical Layers in a NCS.

Building control systems with improved modularity is also important to decrease the life-cycle cost of the system. Figure 7 summarize the life cycle of a control system from a given task to retirement and disposal of the overall system. It can be seen from this chart that maintenance and upgrades always mean additional re-work, and minimizing this re-work will minimize the overall cost of the system. Modular components of the system can be defined such that maintenance and upgrades are well confined so that re-design, re-implementation and re-validation work is minimized. Because component resources shared on the network centralized controller performance is still achievable. Butts and co-workers report that their systems engineering process involved up to as much as 60 % of re-engineering of requirements and defects during powertrain controller development process [16].

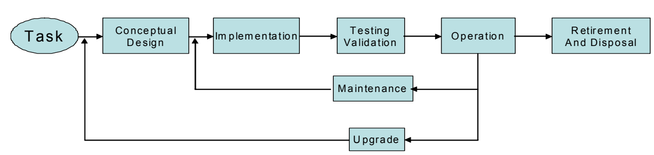
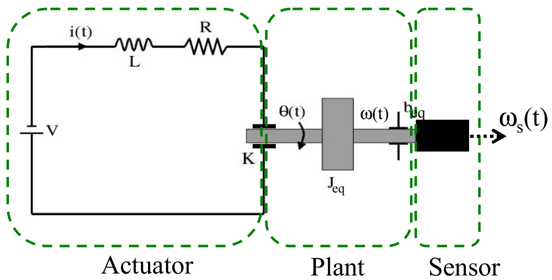


Figure 7 Life Cycle of a Control System.

In order to explain the improved modularity by using bi-directional communications among components, an example related to speed control of a driveshaft driven by a dc-motor is considered (Figure 8). Linearization assumptions such as $b_{eq}(\omega)=b_{eq}(\omega_{operation})$; $K(\omega)=K(\omega_{operation})$ are used in order to model the overall system and design a feedback controller with integral control and state estimation. DC-motors have nonlinear characteristics including velocity drift and saturation.



Parameter	Representation	Units
Motor Constant	K	rad/sV or Nm/A
Rotational Damping	b_{eq}	Nms/rad
Inertia	J_{eq}	kgm^2
Inductance	L	H
Electrical Resistance	R	Ω

State Equations:

$$\frac{d}{dt}\begin{bmatrix} w \\ i \end{bmatrix} = \begin{bmatrix} -b_{eq}/J_{eq} & K/J_{eq} \\ -K/L & -R/L \end{bmatrix}\begin{bmatrix} w \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 1/L \end{bmatrix} V$$

Transfer Function:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K}{LJ_{eq}s^2 + (Lb_{eq} + RJ_{eq})s + Rb_{eq} + K^2}$$

Figure 8 Model of a driveshaft driven by a dc-motor.

Figure 9 shows the controller algorithm which is distributed over the sensor, the controller and the actuator. This distribution has superior modularity because when a component (i.e., controller, sensor or actuator) is replaced with another one as a part of an upgrade or maintenance operation, only portion of the algorithm related to the particular component gets replaced. For example, actuator saturation and integrator anti-wind up algorithm is replaced with the new saturation and anti-wind up algorithm and/or measurement conversion factor and state estimator is replaced with new versions when the sensor is replaced.

When a component and the algorithm inside the component gets replaced the rest of the system stays valid in terms of conceptual design, implementation, testing and validation sign off already obtained. This will significantly reduce the maintenance and upgrade costs since the distribution of the algorithm minimizes the scope of the work involved with the re-design, re-implementation, re-validation phases. In fact, with well defined boundaries, such as in the example of dc-motor speed control, this work can be done at the supplier level which would completely eliminate the re-work on the system. When component interactions in the previous driveshaft driven by a dc-motor example is considered, the importance of bi-directional communications for effective distribution of the algorithm for increased modularity becomes evident in that the suggested algorithm and distribution uses all of the newly proposed communication paths in the networked control system infrastructure.

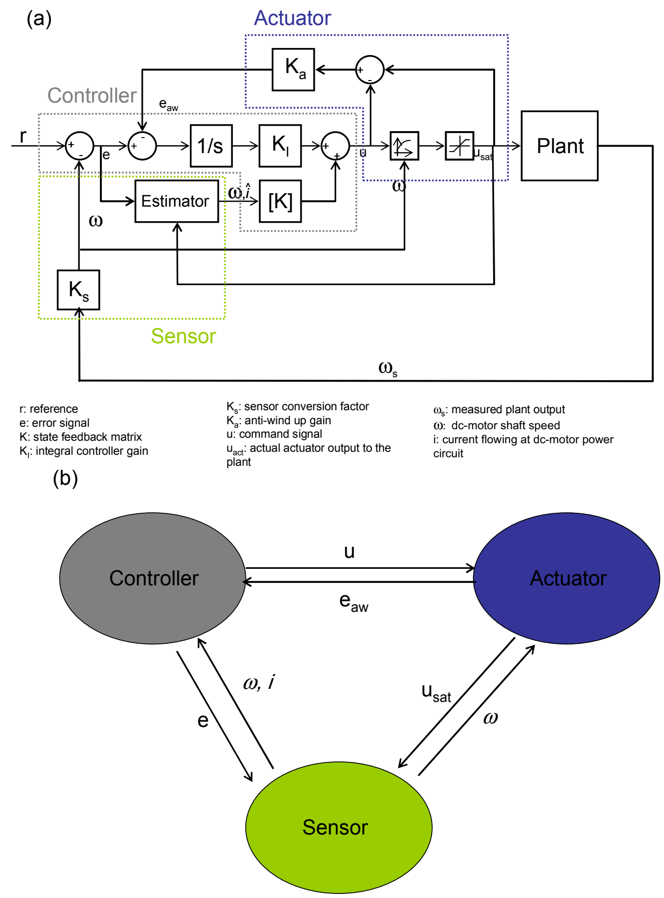


Figure 9 Distributed Modular Control System Example: (a) Block diagram representation, (b) Communication among components/

V. Conclusion

In this article we discuss two important changes that are affecting overall control system design and operation: The increasing availability of (1) embedded electronics (i.e. "smart" components) and (2) network capabilities in components. Current research on networked control systems primarily focuses on communication loss and delay of information transfer. In this paper we have discussed potential benefits of using bi-directional communications among "smart" components in a networked control system, both to improve traditional performance and the control system design process and operations. We have shown that for an "ideal" LTI system, the controller using bi-directional communications can be reduced to a single loop controller. Therefore, there is no benefit in this case in terms of improving traditional control performance. In the "non-ideal" case, however, performance improvement is possible but dependent on the algorithm and the operating conditions used in the system. Bi-directional communications can also be used to improve how a system is designed and re-designed. Modularity of control systems can be improved by freely distributing the overall algorithm among components such that the overall cost building maintaining and operating a control system is reduced.

APPENDIX

In order to show that in an "ideal" case, the controller using bi-directional communications in a continuous LTI system can be reduced to a single loop controller, derivation by block diagram manipulation will be used. The complete analytical derivation of this result is lengthy, but straightforward so we provide an outline for brevity. When the system in Figure 3 examined, it can be seen that

$$\begin{aligned} u_{ca} &= C11r + C12y_{sc} + C13y_{ac} \\ y_{sa} &= S21y + S22u_{cs} + S23y_{as} \\ u_{cs} &= C21r + C22y_{sc} + C23y_{ac} \end{aligned} \quad \textbf{(1)}$$

Using block diagram manipulation and equations in (1) we can get a *simpler* controller transfer function matrix as

$$\begin{aligned}
C_{simple}(1,1) &= A11C11 + A12S22C21 + (A11C12 + A12S21C22)S12C21/(1-S12C22) \\
C_{simple}(1,2) &= A12S21 + (A11C12 + A12S21C22)S13/(1-S12C22) \\
C_{simple}(1,3) &= A12S23 + (A11C12 + A12S21C22)S13/(1-S12C22) \\
C_{simple}(1,4) &= A11C13 + A12S22C23 + (A11C12 + A12S21C22)S12C23/(1-S12C22) \\
C_{simple}(2,1) &= A21C11 + A22S22C21 + (A21C12 + A22S22C22)S12C21/(1-S12C22) \\
C_{simple}(2,2) &= A22S21 + (A21C12 + A22S22C22)S11/(1-S12C22) \\
C_{simple}(2,3) &= A22S23 + (A21C12 + A22S22C22)S13/(1-S12C22) \\
C_{simple}(2,4) &= A21C13 + A22S22C23 + (A21C12 + A22S22C22)S12C23/(1-S12C22) \\
C_{simple}(3,1) &= A31C11 + A32S22C21 + (A31C12 + A32S22C22)S12C21/(1-S12C22) \\
C_{simple}(3,2) &= A32S21 + (A31C12 + A32S22C22)S11/(1-S12C22) \\
C_{simple}(3,3) &= A32S23 + (A31C12 + A32S22C22)S13/(1-S12C22) \\
C_{simple}(3,4) &= A31C13 + A32S22C23 + (A31C12 + A32S22C22)S12C23/(1-S12C22)
\end{aligned} \quad \textbf{(2)}$$

Where r,y, y_{as} and y_{ac} inputs to the C_{simple} and q, y_{as} and y_{ac} are the outputs from this controller respectively. Since both y_{as} and y_{ac} are internal states of the controller, overall system will reduce to a single loop feedback control system as shown in Figure 10.

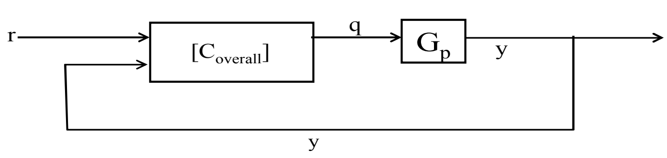


Figure 10 Final form of the distributed algorithm as a single loop feedback controller

One other observation is that although obtaining the single control algorithm from the distributed version is unique, reverse may not be true. Simplest way to see this fact is when distribution of a gain controller between controller and the actuator is considered. There many possibilities (infinitely many in a continuous formulation case) of obtaining the overall control gain, K_o, by multiplying the controller gain, K_c, and actuator gain, K_a.

REFERENCES

[1] Zhang, W., M.S. Branicky and S. M. Phillips, "Stability of Networked Control Systems", *IEEE Control Systems Magazine*, February 2001, pp.84-99.

[2] Walsh, G. C. and Hong Ye,"Scheduling of Networked Control Systems", *IEEE Control Systems Magazine*, February 2001, pp.57-65.

[3] Walsh G.C., H. Ye and L. Bushnell, "Stability Analysis of Networked Control Systems", *IEEE Transactions on Control Systems Technology*, Vol. 10, No 3, May 2002, pp. 438-446.

[4] Yook, J. K., D.M. Tilbury and N.R. Soparkar, "A Design Methodology for Distributed Control Systems to Optimize Performance in the Presence of Time Delays", *Proceedings of the American Control Conference*, Chicago, June 2000, pp. 1959-1964.

[5] Lian, Feng-Li, J. Moyne and D. Tilbury, "Performance Evaluation of Control Networks: Ethernet, ControlNet and DeviceNet", *IEEE Control Systems Magazine*, February 2001,pp. 66-83.

[6] Lian, Feng-Li, J. Moyne and D. Tilbury," Network Design Consideration for Distributed Control Systems", *IEEE Transactions on Control Systems Technology*, Vol. 10, No 2, March 2002, pp. 297-307.

[7] Walsh, G. C., O. Beldiman and L. Bushnell, "Asymptotic Behavior of Networked Control Systems", *Proceedings of the IEEE International Conference on Control Applications*, Hawaii, 1999, pp. 1448-1454.

[8] Curtiss Wright Controls Inc. Scramnet+ Website: www.systran.com/scmain.html

[9] ControlNET Protocol Website: www.controlnet.org

[10] Sinopoli B, Sharp C, Schenato L, Schaffert S, Sastry S.. "Distributed control applications within sensor networks", *Proceedings of the IEEE*, vol.91, no.8, Aug. 2003, pp.1235-46

[11] Liu, J., Chu, M., Liu, J. J., Reich, J. E., Zhao, F., "State-centric programming for sensor and actuator network systems", *IEEE Pervasive Computing*, 2003 October/December; 2 (4): 50-62.

[12] Wong, K.C and J.H.van Schuppen, "Decentralized Supervisory Control of Discrete-Event Systems with Communication", *Proceedings International Workshop on Discrete Event Systems (WODES96)* , IEE, London, 1996, 284-289.

[13] Nace, W. and P. Koopman, "A Product Family Based Approach to Graceful Degradation", *DIPES 2000*, 8-19 October 200, Paderborn, Germany.

[14] Nace, W. and P. Koopman, "A Graceful Degradation Framework for Distributed Embedded Systems", *Workshop on reliability in Embedded Systems (in conjunction with SDRS-2001),* October 2001, New Orleans, L.A..

[15] The **R**obust **Se**lf-configuring **E**mbedded **S**ystems (RoSES) Group Website: www-2.cs.cmu.edu/afs/cs.cmu.edu/project/ices-eris2/roses/public_html/

[16] Butts, K., J. Cook, C. Davey, J. Friedman, P. Menter, S. Raman, N. Sivashankar, P.Smith, S.Toeppe, "Automotive Powertrain Controller Development Using CACSD", *Perspectives in Control: New Concepts and Applications,* Tariq Samad (ed.), IEEE Press.