Communication in automation, including networking and wireless

Nicholas Kottenstette and Panos J. Antsaklis[1]

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556 USA
{nkottens,antsaklis.1}@nd.edu

Abstract — An introduction to the fundamental issues and limitations of communication and networking in automation is given. Digital communication fundamentals are reviewed and networked control systems together with *teleoperation* are discussed. Issues in both wired and wireless networks are presented.

1 Introduction

1.1 Why communication is necessary in automated systems

Automated systems use local control systems that utilize sensor information in feedback loops, process this information and send it as control commands to actuators to be implemented. Such closed loop feedback control is necessary, because of the uncertainties in the knowledge of the process and in the environmental conditions. Feedback control systems rely heavily on the ability to receive sensor information and send commands using wired or wireless communications.

In automated systems there is control supervision, and also health and safety monitoring via SCADA (Supervisory Control and Data Acquisition) systems. Values of important quantities (which may be temperatures, pressures, voltages etc) are sensed and transmitted to monitoring stations in control rooms. After processing the information, decisions are made and supervisory commands are sent to change conditions such as set points or to engage emergency procedures. The data from sensors and set commands to actuators are sent via wired or wireless communication channels.

So communication mechanisms are an integral part of any complex automated system.

1.2 Communication Modalities

In any system there are internal communication mechanisms that allow components to interact and exhibit a collective behavior, the system behavior. For example, in an electronic circuit, transistors, capacitors, resistances are connected so current can flow among them and the circuit can exhibit the behavior was designed for. Such internal communication is an integral part of any system. At a higher level, subsystems that each can be quite complex interact via external communication links that may be wired or wireless. This is the case for example in antilock brake systems, vehicle stability systems, and engine and exhaust control systems in a car or among unmanned aerial vehicles that communicate among themselves to coordinate their flight paths. Such external to subsystems communication is of prime interest in automated systems.

There are of course other types of communication for example machine to machine via mechanical links and human to machine, but here we will focus on electronic transmission of information and communication networks in automated systems.

Such systems are present in refineries, process pants, manufacturing, automobiles to mention but a few. Advances in computer and communication technologies coupled with lower costs are the main driving forces of communication methods in automated systems today. Digital communications, shared wired communication links, and wireless communications make up the communication networks in automated systems today.

In the following, after an introduction to digital communication fundamentals, the focus is on networked control systems that use shared communication links which is common practice in automated systems.

2 Digital Communication Fundamentals

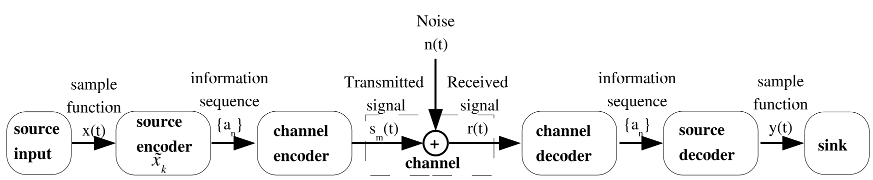

Fig. 1. Digital communication network with separate source and channel coding.

A digital communication system can generally be thought of as a system which allows either a continuous $x(t)$ or discrete random source of information to be transmitted through a channel to a given (set of) sink(s) (Figure 1). The information that arrives at a given destination can be subject to delays, signal distortion and noise. The digital communication channel typically is treated as a physical medium through which the information travels as an appropriately modulated analog signal, $s_m(t)$, is subjected to a linear distortion and additive (typically Gaussian) noise $n(t)$. As is done in [1] we choose to use the simplified single channel network shown in Figure 1 in which the source encoder/decoder and channel encoder/decoder are separate entities. The design of the source encoder/decoder can usually be performed independently of the design of the channel encoder/decoder. This is possible due to the *source-channel separation theorem* (*SCST*) stated by Claude Shannon [2], which states that as long as the *average* information rate of bits per second from the source encoder R_s is strictly below the channel capacity C then information can be reliably transmitted with an appropriately designed channel encoder. Conversely, if R_s is greater than or equal to C then it is impossible to send any information reliably. The interested reader should also see [3] for a more recent discussion as how the *SCST* relates for the single channel case; [4] discusses a *SCST* as it applies to a single source broadcasting to many users and [5] discusses how the *SCST* relates to many sources transmitting to one sink.

In Section 2.1 we will restate some of Shannon's key theorems as they relate to digital communication systems. With a clear understanding of the limitations and principles associated with digital communication systems we will address source encoder and decoder design in Section 2.2 and channel encoder and decoder design in Appendix.

2.1 Entropy, Data Rates and Channel Capacity

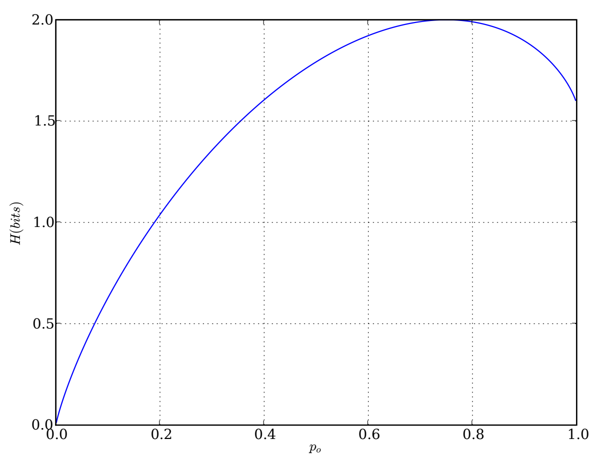

Fig. 2. Entropy of four symbol source $p_i = \{\frac{p_o}{3}, \frac{p_o}{3}, \frac{p_o}{3}, 1 - p_o\}$.

Entropy is a measure of uncertainty of a data source and is typically denoted by the symbol H. It can be seen as a measure of how many *bits* are required to describe a specific output *symbol* of the data source. Therefore, the natural unit of measure for entropy is bits/symbol and can also be used in terms of bits/second depending on its context. Assuming the source could have n outcomes in which each outcome has a probability p_i of occurrence the entropy has the form [2, Theorem 2]:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

The entropy is greatest from a source where all symbols are equally likely. For example, given a two bit source in which each output symbol is $\{00, 01, 10, 11\}$ with respective output probabilities $p_i = \{\frac{p_o}{3}, \frac{p_o}{3}, \frac{p_o}{3}, 1 - p_o\}$. Will have the following entropy which is maximized when all outcomes are equally likely:

$$\begin{aligned} H &= -\frac{p_o}{3}\sum_{i=1}^{3} \log_2 \frac{p_o}{3} - (1-p_o)\log_2(1-p_o) \\ &= -\frac{1}{4}\log_2(\frac{1}{4}) = \log_2(4) = 2\ ;\ p_o = \frac{3}{4} \end{aligned} \tag{2}$$

($p_o = \frac{3}{4}$). Figure 2 shows a plot relating entropy as a function of p_o, note that $H = 0$ bits when $p_o = 0$ since the source would only generate the symbol 11 there is no need to actually transmit it to the receiver. Note that our two bit representations of our symbols is an inefficient choice, for example if $p_o = 0.2$ we could represent this source with only one bit. This can be accomplished by encoding groups of symbols as opposed to considering individual symbols. By determining the redundancy of the source, efficient compression algorithms can be derived as discussed further in Section 2.2.

In digital communication theory we are typically concerned with describing the entropy of joint events $H(x, y)$ in which events x and y have respectively m and n possible outcomes with a joint probability of occurrence $p(x, y)$. The joint probability can be computed using

$$H(x, y) = -\sum_{i,j} p(i, j) log_2 p(i, j)$$

in which it has been shown that the following inequalities hold [2]:

$$H(x, y) \le H(x) + H(y) \tag{3}$$

$$= H(x) + H_x(y) \tag{4}$$

$$H(y) \ge H_x(y) \tag{5}$$

Equality for (3) holds if and only if both events are independent. The uncertainty of y ($H(y)$) is never increased by knowledge of x ($H_x(y)$) as indicated by the conditional entropy inequality in (5). These measures provide a natural way of describing channel capacity when digital information is transmitted as an analog waveform through a channel which is subject to random noise. The effective rate of transmission, R is the difference of the source entropy $H(x)$ from the average rate of conditional entropy $H_y(x)$. Therefore, the channel capacity C is the maximum rate R achievable.

$$R = H(x) - H_y(x) \tag{6}$$

$$C = \max(H(x) - H_y(x)) \tag{7}$$

This naturally leads to the discrete channel capacity theorem given by Shannon [2, Theorem 11]. The theorem states that if a discrete source with entropy H is less than the channel capacity C their exists an encoding scheme such that data can be transmitted with an arbitrarily small frequency of errors (small equivocation), otherwise, the equivocation will approach $H - C + \epsilon$ where $\epsilon > 0$ is arbitrarily small.

2.2 Source Encoder/Decoder Design

Source Data Compression

Shannon's fundamental theorem for a noiseless channel is the basis for understanding data compression algorithms. In [2, Theorem 9] states that for a given source with entropy H (bits per symbol) and channel capacity C (bits per second). Then a compression scheme exists such that you can transmit data at an average rate $R = \frac{C}{H} - \epsilon$ (symbols per second) in which $\epsilon > 0$ is arbitrarily small. For example, if you had a 10 bit temperature measurement of a chamber which 99% of the time is at 25 C and all other measurements are uniformly distributed for the remaining 1% of the time then you would only send a single bit to represent 25 C instead of all 10 bits. Assuming that the capacity of the channel is 100 (bits per second), then instead of sending data at an *average* rate of $10 = \frac{100}{10}$ measurements per second you will actually send data at an *average* rate of $99.1 = (.99\frac{100}{1} + .01\frac{100}{10})$ measurements per second.

Note as this applies to source coding theory, we can also treat the channel capacity C as the ideal H for the source, and so H is the actual bit rate achieved, R, for a given source. Then $R = \sum_{i=1}^{n} p_i n_i$ where p_i is the probability of occurrence for each code word of length n_i bits. When evaluating a source coding algorithm we can look at the *efficiency* of the algorithm which equals $100H/R\%$.

As seen in Figure 2 if $p_o = .19$ then $H = 1.0$ bits/symbol. If we used our initial encoding for the symbols, we would transmit on average two bits per symbol with an *efficiency* of 50%. We will discover that by using a variable length code and by making the following source encoder map $x_k = \{11, 00, 01, 10\} \rightarrow a_k = \{0, 01, 011, 111\}$ we can lower our average data rate $R = 1.32$ (bits/symbol) which improves the *efficiency* to 76%. Note that both mappings satisfy the *prefix condition* which requires that a given code word $\mathbf{C}_k$ of length k having bit elements $(b_1, b_2, \ldots, b_k)$, there is no other code word of length $l < k$ with elements $(b_1, b_2, \ldots, b_l)$ for $1 \leq l < k$ [6]. Therefore, both codes satisfy the Kraft inequality [6, p. 93].

In order to get closer to the ideal $H = 1.0$ (bit/symbol) we will use the Huffman coding algorithm [6, p.95-99] and encode pairs of letters before transmission (which will naturally increase $H = 2.0$ (bits/symbol-pair)).

Figure 3 shows the resulting code words for transmitting pairs of symbols. We see that the encoding results in an *efficiency* of 95% in which $H = 2.0$ and the average achievable transmission rate is $R = 2.1$. The table is generated by sorting in descending order each code word pair and its corresponding probability of occurrence. Next a tree is made in which pairs are generated by matching the two least probable events and are encoded with a corresponding 0 or 1. The probability of either event occurring is the sum of the two least probable events as indicated. The tree continues to grow until all events have been accounted for. The code is simply determined by reading the corresponding 0 and 1 sequence from left to right.

Symbol 1	Symbol 0	P_i1	P_i0	P_i	Self Information	Si*P_i	Code	R
11,	11,	0.81	0.81	0.656	0.608	0.399	0,	0.66
11,	10,	0.81	0.06	0.051	4.285	0.220	011,	0.15
11,	00,	0.81	0.06	0.051	4.285	0.220	0001,	0.21
10,	11,	0.06	0.81	0.051	4.285	0.220	1001,	0.21
01,	11,	0.06	0.81	0.051	4.285	0.220	0101,	0.21
11,	01,	0.81	0.06	0.051	4.285	0.220	1101,	0.21
00,	11,	0.06	0.81	0.051	4.285	0.220	0111,	0.21
10,	01,	0.06	0.06	0.004	7.962	0.032	0001111,	0.03
10,	10,	0.06	0.06	0.004	7.962	0.032	01001111,	0.03
00,	00,	0.06	0.06	0.004	7.962	0.032	11001111,	0.03
01,	00,	0.06	0.06	0.004	7.962	0.032	0101111,	0.03
00,	10,	0.06	0.06	0.004	7.962	0.032	1101111,	0.03
00,	01,	0.06	0.06	0.004	7.962	0.032	0011111,	0.03
10,	00,	0.06	0.06	0.004	7.962	0.032	1011111,	0.03
01,	10,	0.06	0.06	0.004	7.962	0.032	0111111,	0.03
01,	01,	0.06	0.06	0.004	7.962	0.032	1111111,	0.03
					H	2.005	R	2.1
							Efficiency:	95.6%

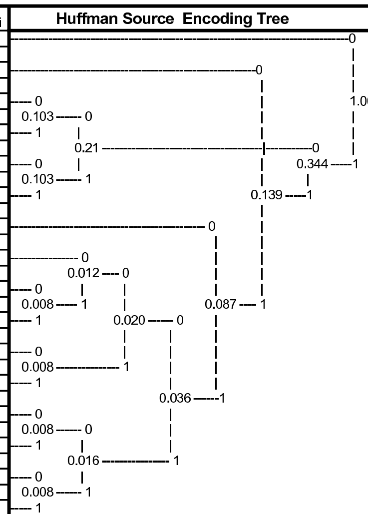

Fig. 3. Illustration of Huffman Encoding Algorithm.

Source Quantization

Due to the finite capacity (due to noise and limited bandwidth) of a digital communication channel, it is impossible to transmit an exact representation of a continuous signal from a source $x(t)$ since it requires an infinite number of bits. The question to be addressed is how can the source be encoded in order to guarantee some minimal *distortion* of the signal when constrained by a given channel capacity C. For simplicity we will investigate the case when $x(t)$ is measured periodically at time T and the continuous sampled value is denoted as $x(k)$ and the quantized values is denoted as $\hat{x}(k)$. The *squared-error distortion* is a commonly used measure of distortion and is computed as follows:

$$d(x_k, \hat{x}_k) = (x_k - \hat{x}_k)^2 \tag{8}$$

Using $\mathbf{X}_n$ to denote n consecutive samples in a vector and $\hat{\mathbf{X}}_n$ to denote the corresponding quantized samples the corresponding distortion for the n samples is

$$d(\mathbf{X}_n, \hat{\mathbf{X}}_n) = \frac{1}{n}\sum_{k=1}^{n} d(x_k, \hat{x}_k) \tag{9}$$

Assuming the source is stationary, the expected value of the distortion of n samples $D = E[d(\mathbf{X}_n, \hat{\mathbf{X}}_n)] = E[d(x_k, \hat{x}_k)]$.

Given a memoryless and continuous random source $\mathbf{X}$ with a *pdf* $p(x)$ and a corresponding quantized amplitude alphabet $\hat{\mathbf{X}}$ in which $x \in \mathbf{X}$ and $\hat{x} \in \hat{\mathbf{X}}$ we define the *rate distortion function* $R(D)$ as

$$R(D) = \min_{p(\hat{x}|x): E[d(\mathbf{X}, \hat{\mathbf{X}})] \leq D} I(\mathbf{X}; \hat{\mathbf{X}}) \tag{10}$$

in which $I(\mathbf{X}; \hat{\mathbf{X}})$ is denoted as the *mutual information* between $\mathbf{X}$ and $\hat{\mathbf{X}}$ [7].

It has been shown that the *rate distortion function* for any memoryless source with zero mean and finite variance σ_x^2 can be bounded as follows:

$$H(\mathbf{X}) - \frac{1}{2}\log_2 2\pi e D \leq R(D) \leq \frac{1}{2}\log_2(\frac{\sigma_x^2}{D}),\ 0 \leq D \leq \sigma_x^2 \tag{11}$$

$H(\mathbf{X}) = \int_{-\infty}^{\infty} p(x) \log p(x) dx$ is denoted as the *differential entropy*. Note that the upper bound is the *rate distortion function* for a Gaussian source $H_g(\mathbf{X})$. Similarly, the bounds on the corresponding distortion-rate function are:

$$\frac{1}{2\pi e} 2^{-2[R - H(X)]} \leq D(R) \leq 2^{-2R}\sigma_x^2 \tag{12}$$

The *rate distortion function* for a band-limited Gaussian channel of width W normalized by σ_X^2 can be expressed in decibels as

$$10 \log \frac{D_g(R)}{\sigma_x^2} = -\frac{3R}{W} \tag{13}$$

[6, p. 104-108]. Thus, decreasing the bandwidth of the source of information results in an exponential decrease in the *rate distortion function* for a given data rate R.

Similar to the grouped Huffman Encoding Algorithm, significant gains can be made by designing a quantizer $\hat{\mathbf{X}} = Q(\cdot)$ for a vector $\mathbf{X}$ of individual scalar components $\{x_k,\ 1 \leq k \leq n\}$ which are described by the joint *pdf* $p(x_1, x_2, \ldots, x_n)$. The optimum quantizer is the one which can achieve the minimum distortion $D_n(R)$.

$$D_n(R) = \min_{Q(\mathbf{X})} E[d(\mathbf{X}, \hat{\mathbf{X}})] \tag{14}$$

As the dimension $n \to \infty$ it can be shown that $D(R) = D_n(R)$ in the limit [6, p. 116-117]. One method to implement such a vector quantization is the K-means algorithm [6, p.117].

3 Networked Systems Communication Limitations

As we have seen in our review of communication theory, there is no mathematical framework that guarantees a bounded deterministic fixed delay in transmitting information through a wireless or a wired medium. All digital representations of an analog waveform are transmitted with an average delay and variance, which is typically captured by its distortion measure. Clearly wired media tend to have a relative low degree of distortion when delivering information from a certain source to destination. For example, receiving digitally encoded data from a wired analog to digital converter, sent to a single digital controller at a fixed rate of 8 kbits/second, occurs with little data loss and distortion (i.e. only the least significant bits tend to have errors). When sending digital information over a shared network, the problem becomes much more complex, in which the communication channel, medium access control (*MAC*) mechanism, and the data rate of each source on the network come into play [8]. Even to determine the average delay of a relatively simple *MAC* mechanism such as time-division multiple access (TDMA) is a fairly complex task [9]. In practice there are wired networking protocols which attempt to achieve a relatively constant delay profile by using a token to control access to the network such as ControlNet and PROFIBUS-DP. Note that Control Area Network (CAN) offers a fixed priority scheme in which the highest priority device will always gain access to the network, therefore, allowing it to transmit data with the lowest average delay while the lower priority devices will have a corresponding increase in average delay [10, Figure 4]. Protocols such as ControlNet and PROFIBUS-DP, however, allow each member on the network an equal opportunity to transmit data within a given slot and can guarantee the same average delay for each node on a network for a given data rate. Usually the main source of variance in these delays is governed by the processing delays associated with the processors used on the network, and the additional higher layer protocols which are built on top of these lower layer protocols.

Wireless networks can perform as well as a wired network if the environmental conditions are ideal. For example, when devices have clear line of sight for transmission, and are not subject to interference (high gain microwave transmission stations). Unfortunately, devices which are used on a factory floor, are more closely spaced and typically have isotropic antennas, which will lead to greater interference and variance of delays as compared to a wired network. Wireless token passing protocols such as that described by [11] are a good choice to implement for control systems, since they limits interference in the network, which limits variance in delays, while providing a reasonable data throughput.

4 Networked Control Systems.

One of the main advantages of using communication networks, instead of point to point wired connections, is the significantly reduced wiring together with the reduced failure rates of much lower connector numbers, which have significant cost implications on automated systems. Additional advantages include easier troubleshooting, maintenance, interoperability of devices and easy integration of new devices added to the network [10]. Automated systems utilize digital shared communication networks. A number of communication protocols are used including Ethernet TCP/IP, DeviceNet, ControlNet, WiFi, Bluetooth. Each one has different characteristics such as data speed and delays. Data are typically transmitted in packets of bits, for example an Ethernet IEEE 802.3 frame has a 112- or 176- bit header and a data field that must be at least 368-bit long.

Any automated system that uses shared digital wired or wireless communication networks must address certain concerns including:

1. Bandwidth limitations, since any communication network can only carry a finite amount of information per unit of time,
2. Delay jitter, since uncertainties in network access delay, or delay jitter, is commonly present, and
3. Packet dropouts, since transmission errors, buffer overflows due to congestion, or long transmission delays may cause packets to be dropped by the communication system.

All these issues are currently being addressed in ongoing research on Networked Control Systems (NCS) [12].

4.1 Networked Control Systems

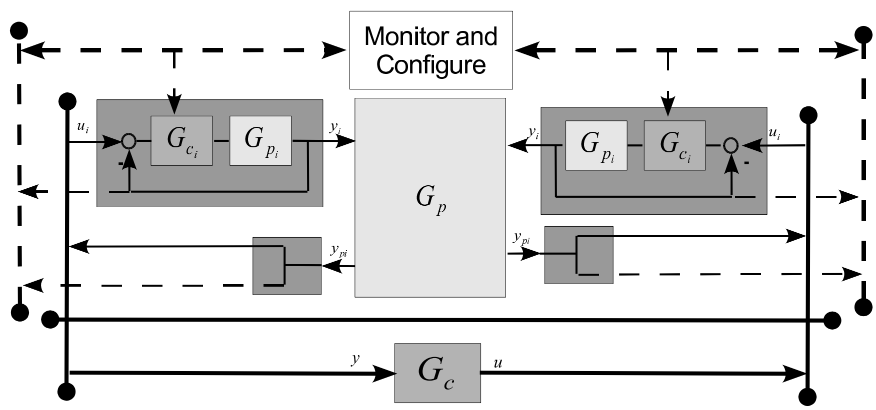

Fig. 4. Typical automation network.

Figure 4 depicts a typical automation network in which two dedicated communication buses are used in order to control an overall process G_p with a dedicated controller G_c. The heavy solid line represents the control data network which provides timely sensor information y to G_c and distributes the appropriate control command u to the distributed controllers G_{c_i}. The heavy dashed solid line represents the monitor and configure data network which allows the various controllers and sensors to be configured and monitored while G_p is being controlled. The control network usually has a lower data capacity but provides a fairly constant data delay with little variance in which field buses such as CAN, ControlNet, and PROFIBUS-DP are appropriate candidates. The monitoring and configuring network should have a higher data capacity but can tolerate more variance in its delays such that standard Ethernet or wireless networks using TCP/IP would be suitable. Sometimes the entire control network is monitored by a programmable logic controller (PLC) which acts as a gateway to the monitoring network as depicted in [10, Figure 12]. However, there are advanced distributed controllers G_{c_i} which can both receive and deliver timely data over a control field bus such as CAN, yet still provide an Ethernet interface for configuration and monitoring. One such example is the π*MFC*, which is an advanced pressure insensitive mass flow controller that provides both communication interfaces in which a low-cost and low-power dual processor architecture provides dedicated real-time control with advanced monitoring and diagnostic capabilities offloaded to the communications processor [13]. Although, not illustrated in this figure there is current research in establishing digital safety networks as discussed in [10]. In particular the safety networks discussed are implemented over a serial-parallel line interface and implement the SafetyBUS p protocol.

Automated Control systems with spatially distributed components have existed for several decades. Examples include chemical processes, refineries, power plants, and airplanes. In the past, in such systems the components were connected via hard-wired connections and the systems were designed to bring all the information from the sensors to a central location where the conditions were being monitored and decisions were taken on how to act. The control policies then were implemented via the actuators, which could be valves, motors etc. Today's technology can put low cost processing power at remote locations via microprocessors and that information can be transmitted reliably via shared digital networks or even wireless connections. These technology driven changes are fueled by the high costs of wiring and the difficulty in introducing additional components into the systems as the needs change.

In 1983, Bosch GmbH began a feasibility study of using networked devices to control different functions in passenger cars. This appears to be one of the earliest efforts along the lines of modern networked control. The study bore fruit, and in February 1986 the innovative communications protocol of the Control Area Network (CAN) was announced. By mid 1987, CAN hardware in the form of Intel's 82526 chip had been introduced, and today virtually all cars manufactured in Europe include embedded systems integrated through CAN. Networked control systems are found in abundance in many technologies, and all levels of industrial systems are now being integrated through various types of data networks. Although networked control system technologies are now fairly mature in a variety of industrial applications, the recent trend toward integrating devices through wireless rather than wired communication channels has highlighted important potential application advantages as well as several challenging problems for current research.

These challenges involve the optimization of performance in the face of constraints on communication bandwidth, congestion, and contention for communication resources, delay, jitter, noise, fading, and the management of signal transmission power. While the greatest commercial impact of networked control systems to date has undoubtedly been in industrial implementations, recent research suggests great potential together with significant technical challenges in new applications to distributed sensing, reconnaissance and other military operations, and a variety of coordinated activities of groups of mobile robot agents. Taking a broad view of networked control systems, we find that in addition to the challenges of meeting real-time demands in controlling data flow through various feedback paths in the network, there are complexities associated with mobility and the constantly changing relative positions of agents in the network.

Networked control systems research lies primarily at the intersection of three research areas: control systems, communication networks and information theory, and computer science. Networked control systems research can greatly benefit from theoretical developments in information theory and computer science. The main difficulties in merging results from these different fields of study have been the differences in emphasis in research so far. In information theory, delays in the transmitted information are not of central concern, as it is more important to transmit the message accurately even though this may involve sometimes significant delays in transmission. In contrast, in control systems delays are of primary concern. Delays are much more important than the accuracy of the transmitted information due to the fact that feedback control systems are quite robust to such inaccuracies. Similarly, in traditional computer science research, time has not been a central issue since typical computer systems were interacting with other computer systems or a human operator and not directly with the physical world. Only recently, areas such as real-time systems have started addressing the issues of hard time constraints where the computer system must react within specific time bounds, which is essential for embedded processing systems that deal directly with the physical world.

So far, researchers have focused primarily on a single loop and stability. Some fundamental results have been derived that involve the minimum average bit rate necessary to stabilize a linear, time-invariant system.

An important result relates the minimum bit rate R of feedback information needed for stability (for a single input, linear system) to the fastest unstable mode of the system via

$$R > \log_2 \exp(\sum \mathcal{R}(a_i)). \tag{15}$$

Although progress has been made, much work remains to be done. In the case of a digital network where information is typically sent in packets, the minimum average rate is not the only guide to control design. A transmitted packet typically contains a payload of tens of bytes, and so blocks of control data are typically grouped together. This enters into the broader set of research questions on the comparative value of sending 1 bit per second or 1000 bits every 1000 seconds-for the same average data rate. In view of the typical actuator constraints, an unstable system may not be able to recover after 1000 seconds.

An alternative measure is to see how infrequent feedback information is needed to guarantee that the system remains stable. See, for example, [14] and [15], where this scheme has been combined with model-based ideas for significant increases in the periods where the system is operating in an open-loop fashion. Intermittent feedback is another way to avoid taxing the networks for sensor information. In this case, every so often the loop is closed for a certain-fixed or time-varying period of time [16]. This may correspond to opportunistic, bursty situations where the sensor sends up bursts of information when the network is available. The original idea of intermittent feedback was motivated by human motor control considerations. There are strong connections with cooperative control. There, researchers have used spatial invariance ideas to describe results on stability and performance [17]. If spatial invariance is not present, then one may use the mathematical machinery of graph theory to describe the interaction of systems/units and to develop detailed models of groups of agents flying in formation, foraging, cooperation in search of targets or food, etc. An additional dimension in the wireless case is to consider channels that vary with time, fade, or disappear and reappear. The problem, of course, in this case becomes significantly more challenging. Consensus approaches have also been used which typically assume rather simple dynamics for the agents and focus on the topology considering fixed or time-varying links in synchronous or asynchronous settings. Implementation issues in both hardware and software are at the center of successful deployment of networked control systems. Data integrity and security are also very important and may lead to special considerations in control system design even at early stages.

Overall, single loop and stability have been emphasized and studied under quantization of sensor measurements and actuator levels. Research is needed to understand multiple interconnected systems over realistic channels that work together in a distributed fashion towards common goals with performance guarantees. Note that limits to performance in networked control systems appear to be caused primarily by delays and dropped packets. Other issues being addressed by current research are actuator constraints, reliability, fault detection and isolation, graceful degradation under failure, reconfigurable control, and ways to build increased degrees of autonomy into networked control systems.

4.2 *Teleoperation.*

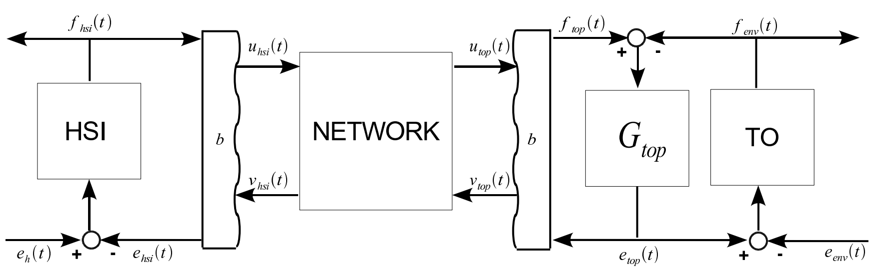

Fig. 5. Typical *teleoperation* network.

An important area of networked control is *teleoperation*. *Teleoperation* is the process of a human performing a remote task over a network with a *teleoperator* (TO). Ideally, the TO's velocity ($f_{top}(t)$) should follow the human velocity commands ($f_{hsi}(t) = f_{top}(t - T)$) through a *human system interface* (HSI) [18]. Force feedback from the TO ($e_{top}(t)$) is sent back to the HSI ($e_{hsi}(t) = e_{top}(t - T)$) in order for the operator to feel immersed in the remote environment. The controller (G_{top}) depicted in Figure 5 is typically a proportional derivative controller which maintains $f_{top}(t) = f_{env}(t)$ over a reasonably large bandwidth. The use of force feedback can lead to instabilities in the system due to small delays T in data transfer over the network. In order to recover stability the HSI velocity f_{hsi} and TO force e_{top} are encoded into *wave variables* [19], based on the wave port impedance b such that

$$u_{hsi}(t) = \frac{1}{\sqrt{2b}}(bf_{hsi}(t) + e_{hsi}(t)) \tag{16}$$

$$v_{top}(t) = \frac{1}{\sqrt{2b}}(bf_{top}(t) - e_{top}(t)) \tag{17}$$

are transmitted over the network from the corresponding HSI, and TO. As the delayed *wave variables* are received ($u_{top}(t) = u_{hsi}(t - T)$, $v_{hsi}(t) = v_{top}(t - T)$), they are transformed back into the corresponding velocity and force variables ($f_{top}(t),\ e_{hsi}(t)$) as follows

$$f_{top}(t) = \sqrt{\frac{2}{b}}u_{top}(t) - \frac{1}{b}e_{top}(t) \tag{18}$$

$$e_{hsi}(t) = bf_{hsi}(t) - \sqrt{2b}v_{hsi}(t). \tag{19}$$

Such a transformation allows the communication channel to remain *passive* for fixed time delays T and allows the *teleoperation* network to remain stable. The study of *teleoperation* continues to evolve for both the continuous and discrete time cases as surveyed in [20].

5 Appendix

5.1 Channel Encoder/Decoder Design

Denoting T (seconds) as the signal period, and W (Hz) as the bandwidth of a communication channel, we will use the ideal Nyquist rate assumption that $2TW$ symbols of $\{a_n\}$ can be transmitted with the analog wave forms $s_m(t)$ over the channel depicted in Figure 1. We further assume that *independent* noise $n(t)$ is added to create the received signal $r(t)$. Then we can state the following:

1. [2, Theorem 16] The actual rate of transmission is

$$R = H(s) - H(n), \tag{20}$$

   in which the channel capacity is the best signaling scheme which satisfies

$$C = \max_{P(s_m)} H(s) - H(n). \tag{21}$$

2. [2, Theorem 17] if we further assume the noise is white with power N and the signals are transmitted at power P then the channel capacity C (bits per second) is
$$C = W \log_2 \frac{P+N}{N}. \tag{22}$$

Various channel coding techniques have been devised in order to transmit digital information to achieve rates R which approach this channel capacity C with a correspondingly low bit error rate. Among these *bit error correcting* codes are block and convolutional codes in which the *Hamming Code* [6, p.423-425] and the *Viterbi Algorithm*[6, p. 482-492] are classic examples for the respective implementations.

5.2 Digital Modulation

A linear filter can be described by its frequency response $H(f)$ and *real* impulse response $h(t)$ ($H^*(-f) = H(f)$). It can be represented in an equivalent low-pass form $H_l(f)$ in which:

$$H_l(f - f_c) = \begin{cases} H(f),\ f > 0 \\ 0,\ f < 0 \end{cases} \tag{23}$$

$$H_l^*(-f - f_c) = \begin{cases} 0,\ f > 0 \\ H^*(-f),\ f < 0 \end{cases} \tag{24}$$

Therefore, with $H(f) = H_l(f - f_c) + H_l^*(f - f_c)$ the impulse response $h(t)$ can be written in terms of the *complex* valued inverse transform of $H_l(f)$ ($h_l(t)$) [6, p. 153].
$$h(t) = 2\mathsf{Re}[h_l(t)e^{j2\pi f_c t}] \tag{25}$$

Similarly the signal response $r(t)$ of a filtered input signal $s(t)$ through a linear filter $H(f)$ can be represented in terms of their low-pass equivalents (26).

$$R_l(f) = S_l(f)H_l(f) \tag{26}$$

Therefore it mathematically convenient to discuss the transmission of equivalent low-pass signals through equivalent low-pass channels [6, p. 154].

Digital signals $s_m(t)$ consist of a set of analog waveforms which can be described by an *orthonormal* set of waveforms $f_n(t)$. An *orthonormal* waveform satisfies the following:
$$\langle f_i(t), f_j(t)\rangle_T = \begin{cases} 0,\ i \neq j \\ 1,\ i = j \end{cases} \tag{27}$$

in which $\langle f(t), g(t)\rangle_T = \int_0^T f(t)g(t)dt$. The *Gram-Schmidt procedure* is a straight forward method to generate a set of *orthonormal* wave forms from a basis set of signals [6, p. 163].

Table 1 provides the corresponding orthonormal wave forms and minimum signal distances ($d_{min}^{(e)}$) for *pulse-amplitude-modulated* (*PAM*), *phase-shift keying* (*PSK*), and *quadrature amplitude modulation* (*QAM*). Note that *QAM* is a combination of *PAM* and *PSK* in which $d_{min}^{(e)}$ is a special case of amplitude selection where $2d$ is the distance between adjacent signal amplitudes. Signaling amplitudes are in terms of the low-pass signal pulse shape $g(t)$ energy $\mathcal{E}_g = \langle g(t), g(t) \rangle_T$. The pulse shape is determined by the transmitting filter which typically has a *raised cosine* spectrum in order to minimize inter-symbol interference at the cost of increased bandwidth [6, p. 559]. Each modulation scheme allows for M symbols in which $k = \log_2 M$ and N_o

Table 1. Summary of *PAM*, *PSK* and *QAM*

Modulation	s_m(t)	$f_1(t)$	$f_2(t)$
PAM	$s_m f_1(t)$	$\sqrt{\frac{2}{\mathcal{E}_g}} g(t) \cos 2\pi f_c t$	
PSK	$s_{m1} f_1(t) + s_{m2} f_2(t)$	$\sqrt{\frac{2}{\mathcal{E}_g}} g(t) \cos 2\pi f_c t$	$-\sqrt{\frac{2}{\mathcal{E}_g}} g(t) \sin 2\pi f_c t$
QAM	$s_{m1} f_1(t) + s_{m2} f_2(t)$	$\sqrt{\frac{2}{\mathcal{E}_g}} g(t) \cos 2\pi f_c t$	$-\sqrt{\frac{2}{\mathcal{E}_g}} g(t) \sin 2\pi f_c t$

Modulation	$\mathbf{s}_m$	$d_{min}^{(e)}$
PAM	$(2m - 1 - M) d \sqrt{\frac{\mathcal{E}_g}{2}}$	$d\sqrt{2\mathcal{E}_g}$
PSK	$\sqrt{\frac{\mathcal{E}_g}{2}} [\cos \frac{2\pi}{M}(m-1) \ \sin \frac{2\pi}{M}(m-1)]$	$\sqrt{\mathcal{E}_g (1 - \cos \frac{2\pi}{M})}$
QAM	$\sqrt{\frac{\mathcal{E}_g}{2}} [(2m_c - 1 - M) d \ (2m_s - 1 - M) d]$	$d\sqrt{2\mathcal{E}_g}$

is the average noise power per symbol transmission. Denote P_M as the probability of a symbol error, and assume we use a Gray code, then we approximate the average bit error $P_b \approx \frac{P_M}{k}$. The corresponding symbol errors are:

1. [6, p. 265] for M-ary *PAM*

$$P_M = \frac{2(M-1)}{M} Q(\sqrt{\frac{d^2 \mathcal{E}_g}{N_o}}), \tag{28}$$

2. [6, p. 270] for M-ary *PSK*

$$P_M \approx 2Q(\sqrt{\frac{\mathcal{E}_g}{N_o}} \sin \frac{\pi}{M}), \tag{29}$$

3. [6, p. 279] for *QAM*

$$P_M < (M-1) Q(\sqrt{\frac{[d_{min}^{(e)}]^2}{2N_o}}). \tag{30}$$

References

1. P. R. Gallager, “Mit opencourseware — electrical engineering and computer science — 6.45 principles of digital communication - i, fall 2002,”

2. C. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379 – 423, 1948/07/.
3. S. Vembu, S. Verdu, and Y. Steinberg, “The source-channel separation theorem revisited,” *IEEE Transactions on Information Theory*, vol. 41, no. 1, pp. 44 – 54, 1995/01/.
4. M. Gasfpar, B. Rimoldi, and M. Vetterli, “To code, or not to code: Lossy source-channel communication revisited,” *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1147 – 1158, 2003.
5. H. El Gamal, “On the scaling laws of dense wireless sensor networks: The data gathering channel,” *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 1229 – 1234, 2005.
6. J. Proakis, *Digital Communications*. McGraw-Hill Science/Engineering/Math, August 2000.
7. T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.
8. M. Xie and M. Haenggi, “Delay-reliability tradeoffs in wireless networked control systems,” in *Workshop on Networked Embedded Sensing and Control*, Lecture Notes in Control and Information Sciences, Springer, October 2005.
9. K. K. Lee and S. T. Chanson, “Packet loss probability for bursty wireless real-time traffic through delay model,” *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 929 – 938, 2004.
10. J. Moyne and D. Tilbury, “The emergence of industrial control networks for manufacturing control, diagnostics, and safety data,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 29 – 47, 2007.
11. M. Ergen, D. Lee, R. Sengupta, and P. Varaiya, “Wtrp - wireless token ring protocol,” *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1863 – 81, 2004.
12. P. Antsaklis and J. Baillieul, eds., *Special Issue: Technology of Networked Control Systems*, vol. 95 number 1 of *Proceedings of the IEEE*. IEEE, 2007.
13. A. Shajii, N. Kottenstette, and J. Ambrosina, “Apparatus and method for mass flow controller with network access to diagnostics,” 2004. U.S. Patent #6810308.
14. L. A. Montestruque and P. J. Antsaklis, “On the model-based control of networked systems,” *Automatica*, vol. 39, no. 10, pp. 1837 – 1843, 2003.
15. L. A. Montestruque and P. Antsaklis, “Stability of model-based networked control systems with time-varying transmission times,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562 – 1572, 2004.
16. T. Estrada, H. Lin, and P. Antsaklis, “Model-based control with intermittent feedback,” in *Proceedings of the 14th Mediterranean Conference on Control and Automation*, (Ancona, Italy), pp. 6 pp. –, 2006.
17. B. Recht and R. D’Andrea, “Distributed control of systems over discrete groups,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1446 – 1452, 2004.
18. M. Kuschel, P. Kremer, S. Hirche, and M. Buss, “Lossy data reduction methods for haptic telepresence systems,” in *Proceedings. 2006 Conference on International Robotics and Automation (IEEE Cat. No. 06CH37729D)*, (Orlando, FL, USA), pp. 2933 – 8, 2006.
19. G. Niemeyer and J.-J. E. Slotine, “Telemanipulation with time delays,” *International Journal of Robotics Research*, vol. 23, no. 9, pp. 873 – 890, 2004.
20. P. Hokayem and M. Spong, “Bilateral teleoperation: An historical survey,” *Automatica*, vol. 42, no. 12, pp. 2035 – 57, 2006.