FeedNeback
co-design for networked control systems

GRANT AGREEMENT N°223866

Deliverable	D02.1
Nature	Report
Dissemination	Public

D01.2 - PERFORMANCE METRICS

Report Preparation Date	15/08/2009
	Project month: 12
Authors	Sandro Zampieri, Federica Garin, University of Padova
Report Version	Vn 1
Doc ID Code	UNIPDP05_D02.1_15SEP09_v01
Contract Start Date	01/SEP/2008
Duration	36 months
Project Coordinator :	Carlos CANUDAS DE WIT, INRIA, France

SEVENTH FRAMEWORK PROGRAMME

Theme 3:

Information and Communication Technologies

SUMMARY

In this report we present the results of our investigations regarding the correct approach for the analysis of networked control systems. In fact, problems arising in networked control applications involve different disciplines, namely control, communication and computer science. The difficulties in finding a synthesis of these three areas of research are numerous, but one of the most critical issues arises from the difference on the performance measures that are typically considered in each of the areas. This is indeed the focus of this report. We start by summarizing the performance indexes that are typically considered in control, communication and computer science, which a particular emphasis in pointing out their differences and similarities. The outcome of our analysis is that there does not exist a general recipe for performance evaluation of a networked control system, since the application range for such systems is too wide. This suggests that the performance metrics must be selected based on the specific application which is addressed. Although this finding seems a trivial disappointing answer, it results in a source of interesting and inspiring novel research directions. Indeed, in the present document we show through examples how specific networked control or estimation problems can be inspiring starting points for the research activities of the present project, once the proper performance index is considered. In fact, in these examples we show how, in networked control and estimation problems, there are many opportunities for new relevant theoretical questions. In particular, the contribution of control, communication and computer science theory will play an important role in networked control systems and will foster the development of a common culture.

Contents

1 Introduction

The present project deals with problems whose solution requires methodologies which fuse together the instruments of different disciplines, namely control theory, communication, information theory, and computer science. Each of these disciplines has its own characteristics in terms of problems that are typically faced, in terms of the language in which the solutions are proposed, and in terms of the design requirements and constraints.

This deliverable focus on this last fact, namely on the different performance indices which are peculiar to each of these disciplines and on their relevance to the class of problems considered in the present project. In this document we first briefly recall the performance indices which are considered in pure control, communication and computing applications. This presentation is necessarily brief since its aim is only to provide a framework for the subsequent sections. Then we present a discussion about the guidelines for determining the performance indices which are suitable for networked systems and how these are related to the indexes which are peculiar to control, communication and computation. In this report we do not present technical results since the objective of the associated task, after the first year of the project, was not to present final answers to the questions raised in the project proposal, but only to make those questions more precise in order to try to find their answers more effectively during the rest of the project.

1.1 Performance indices in estimation and control

In control theory there are two problems that are classically considered. One is the estimation problem which yields to the theory of Kalman filtering. This consists in estimating an hidden quantity, called the state of the system, from the observation of some correlated signals. In this case the performance index is quite natural, namely it is the size of the estimation error.

The second problem, which is properly called the control problem, consists instead in trying to modify a plant dynamics making them suitable to desirable specifications. The main feature of the control is the heterogeneity of the plants to be controlled but especially of the specifications to be obtained.

In some cases the objective is simply to make one variable of the plant to follow a reference trajectory, so to obtain desired profiles, despite to the presence of noise and of errors in the system description. In this case the size of difference between the reference trajectory and the actual variable can be considered a significant performance index. In this scenario two different regimes are considered. One is the transient regime, which deals with the error signals characteristics in the time periods close to sudden variations of the reference trajectory. The second regime is called steady state regime and it considers the error signals characteristics in the time periods in which the reference is constant or at least slowly varying.

It is a classical fact in control that often the performance in these two regimes are conflicting and that the difficulty in solving a control problem stands in determining a trade-off between them.

1.2 Performance indices in communication

In classical communication theory the problem to be solved consists in sending information from a transmitter to a receiver. Information to be transmitted is modeled as a random variable. This is first transformed into a string of zeros and ones (bits) in which the redundancy is first reduced (source coding) so that the information is compressed. Then, in order to make the transmission robust to channels errors, some redundancy is added in an optimal way with respect to the channel characteristics (channel coding).

Both these two operations are typically performed with little consideration of:

1. the specific objective of the information transmission (voice, video, images, monitoring, control,...);
2. the delays introduced by the source and channel coding algorithms.

In case of information transmission for control, the presence of sensible delays is a factor of performance deterioration and for this reason delays must be kept as small as possible. The concept of anytime information theory [27, 26, 28] is a promising approach to deal with this kind of problems.

1.3 Performance indices in computation

In computer science the typical performance index is the computational complexity, which can be described simply as the number of computer iterations necessary complete a task. A very rough classification of the algorithmic computational complexity is the one which distinguishes the algorithms whose complexity is at most a polynomial function of the problem dimension and the algorithms whose complexity is at least an exponential function of the problem dimension.

This is only the first and probably the most important performance index in computation. More recently, for high complexity problems a different class of algorithms have been proposed. These are based on procedures which solve a problem only approximately, within a given accuracy, or with a certain probability. Typically these algorithms are such that, the more iterations you allow, the better the accuracy or the lower the error probability. A particular class of these algorithms are called anytime algorithms [33], which ensure an increasing accuracy in the result as far as the number of iterations grows. The main feature of anytime algorithms is that the user does not need to decide the accuracy before the algorithm starts to run, but, instead, the algorithm can be stop 'anytime', always providing a reasonable result, with the better quality the more the algorithm is allowed to operate. Such algorithms are particularly useful in case we need to run a task in real time, situation which is typical in control applications. Real time is a particularly critical requirement is case of computer units running several tasks at the same time so that a priority management becomes necessary.

Finally, another classical variation in the analysis computational performance has been proposed when in computer science the problem of distributed and parallel computation has been considered. In this case, instead of having several tasks running at the same time on one computer unit, we have one task to be distributed over several communicating computer units. In this case, two problems have to be considered. On the one hand, the possibility for an algorithm to be distributed over several units has been first analyzed. Then, the cost in terms of data communication has been analyzed, this case being important when the units are distributed geographically and with costly communication, as it happens in wireless sensor networks. A relevant theoretical framework able to provide interesting answers in this set-up is given by the theory of communication complexity [18].

2 Performance indices in networked control systems

The design of networked control systems require methodologies which use at the same time control theory, communication theory and computer science. However, these theories have been developed in the past for solving distinct problems which typically had very little to do with each other. In the development of interdisciplinary methodologies, it is natural to start from the characteristics of the specific problems we aim to solve. In other words, it will be the applications we have in mind which will suggest in general what is needed from control theory, communication theory and computer science and which are the performance indices which should be considered, in particular.

In order to illustrate this general concept, we will use some specific problems which we are considering in the present project. For those problems it will be clear how the choice of the performance indices is indeed the right point to start from in the developing co-design methodologies. At this preliminary stages we will be not very precise and general in the description of the problem solution, since the objective of this deliverable is not to propose solutions but instead to give an idea of the proposed investigation approach.

3 Consensus and distributed estimation

In recent years the analysis of the coordination mechanisms of multiagent systems is attracting an enormous attention in the engineering community. This is mainly due to the intrinsic robustness and to the degree of adaptation which these systems exhibit in nature which makes their structure very attractive also as an inspiring design paradigm for many engineering systems. This paradigm consists in the possibility of obtaining high performance levels through numerous simple and cheap cooperating units.

The analysis of information dynamics which allow these systems to work properly is a challenging problem for the information engineering community, since, while it is quite difficult to find a common feature exhibited by all cooperating systems, it is clear that cooperation needs communication and so efficient cooperation is surely related to efficient information diffusion.

One of the simplest instances of coordinated task is distributed estimation, by which a sensor network aims to have a good estimate of an unknown quantity through numerous noisy measurements. It is intuitive that, in order to obtain an estimate of this quantity, we need to compute the average of the measurements that each sensor is able to produce and this requires information diffusion in the sensor network. One solution to this problem comes from linear average-consensus algorithm [24], which allows to compute an average in a distributed way. Although it is not the most efficient method to reach this goal, this technique is attracting a lot of attention mainly because of its simplicity, which makes it intrinsically robust to node or to communication failures [11].

An average consensus algorithm is determined by a doubly-stochastic matrix P and the evolution of this algorithm is well described by the associated Markov chain. Indeed, through the theory of Markov chains, it is possible to prove that, under rather weak assumptions on P, the algorithm works properly. Traditionally, the index which is considered for determining the performance of a specific average consensus algorithm is given by the exponential rate of convergence of the algorithm which coincides with the second largest eigenvalue of P. In the literature devoted to Markov chains, much research has concerned estimations of this index [9] from the network architecture.

However, when this algorithm is used for specific applications requiring the distributed computation of an average, different performance indices become natural instruments for comparing different choices of the matrix P. This section will show some examples when this is the case. These examples will highlight that these new indices depend on all the eigenvalues of P and so they can provide different evaluations in the matrix selection. This remark calls for new results in spectral graph theory.

Moreover, in this section we will show that the study of performance indices different from second largest eigenvalue is essential in large-scale networks, namely in networks formed by a large number of cooperating agents. In this case the asymptotic analysis of these algorithms can be done both with respect to time and respect to the number of agents. In fact, in this case a trade-off can be expected, since an increasing number of agents will naturally be beneficial to the estimation precision, but on the other hand it will result in a more difficult communication among the agents. While this trade-off becomes quite clear when choosing correct performance indices, it is not highlighted by the study of the essential spectral radius, which only takes into account the slower information exchange. Our analysis allows instead to correctly highlight this trade-off.

3.1 Consensus algorithm and its standard performance index

Assume we have a finite set of nodes $V = \{1, 2, \dots, N\}$ and that each node $i \in V$ knows a real number z_i. Let $z \in \mathbb{R}^N$ be the vector having z_i as components. Assume a (directed) graph $G = (V, E)$ is given describing the allowed communications. More precisely, at each time $t \in \mathbb{Z}$ a node i can send a real number to a node j if and only if (i, j) belongs to E. We will assume that G is strongly connected and that $(i, i) \in E$ for all V. Consider the following iterative algorithm in which we have a sequence of vectors $x(t) \in \mathbb{R}^N$ for $t = 0, 1, 2, \dots$ and in which $x(0) = z$ and

$$x(t + 1) = Px(t)$$

where $P \in \mathbb{R}^{N\times N}$. If we choose as P a doubly-stochastic matrix consistent with the communication graph, i.e. such that $P_{ij} \neq 0$ iff $(i, j) \in E$, then by standard Markov chain theory [21] we have that

$$\lim_{t\to\infty} x(t) = \left(\frac{1}{N} \sum_{i=1}^{N} z_i \right) \mathbf{1}$$

where $\mathbf{1}$ denotes the column vector with all the entries equal to 1. Moreover it is well known that the convergence is exponential and that the exponential rate of convergence is given by the essential spectral radius of P, namely

$$\rho_{\text{ess}}(P) := \max\{|\lambda| \ : \ \lambda \in \sigma(P) \setminus \{1\}\}$$

where $\sigma(P)$ is the set of eigenvalues of P.

The relation between the network topology and the essential spectral radius has been the subject of extensive research in the community working on Markov chains and many results have been found. We will show now some alternative ways to evaluate the performance of this algorithm.

3.2 2-norm of the consensus error as a performance index

A first alternative way to evaluate how fast $x(t)$ converges to its limit $x(\infty)$ is by considering the $\mathcal{L}^2$ norm of the difference $x(t) - x(\infty)$, namely by considering the index

$$J(P, x(0)) := \frac{1}{N} \sum_{t=0}^{\infty} ||x(t) - x(\infty)||^2$$

where $|| \cdot ||$ denotes the 2-norm of a vector. Through some computations it can be seen that

$$J(P, x(0)) = x(0)^T Qx(0)$$

where T denotes transpose and where

$$Q = \frac{1}{N} \sum_{t=0}^{\infty} \left(P - \frac{1}{N}\mathbf{1}\mathbf{1}^T \right)^t \left(P^T - \frac{1}{N}\mathbf{1}\mathbf{1}^T \right)^t$$

In order to make this cost small, it is necessary to make the positive semidefinite Q small, by, for instance, making its trace small. In case the components $x_i(0)$ are random variables, then it makes sense to consider the cost $J(P) := \mathbb{E}[J(P, x(0))]$ and, in case $x_i(0)$ are i.i.d. random variables, we have that

$J(P) = r\,\mathrm{trace}\,Q$, where $r = \mathbb{E}[x_i(0)^2]$. Finally observe that, if P is a normal matrix, i.e. $P^T P = PP^T$ (e.g. symmetric matrices are normal), then we have that

$$J(P) = \frac{1}{N} \sum_{\substack{\lambda \in \sigma(P) \\ \lambda \neq 1}} \frac{1}{1 - |\lambda|^2} \tag{1}$$

which is a cost function depending on all the eigenvalues of P.

3.3 Performance of the consensus algorithm with noisy communications

Assume that, due to noise in the communication between nodes we have that the consensus algorithm becomes

$$x(t+1) = Px(t) + Wn(t)$$

where $W \in \mathbb{R}^{N\times N}$ and $n(t)$ is a i.i.d. random sequence with $\mathbb{E}[n(t)n(t)^T] = rI$. This variation of the consensus algorithm has been analyzed in [31] in case of $W = I$ and in [12] in case of $W = P - I$. In this case the consensus is not reached and the components of $x(t)$ will never become equal. It makes sense in this case to consider the displacement

$$e(t) := x(t) - \frac{1}{N} \sum_{i=1}^{N} x_i(t)$$

and then the cost function

$$J(P) := \frac{1}{N} \limsup_{t\to\infty} \mathbb{E}[e(t)^T e(t)]$$

which gives the asymptotic expected dimension of the displacement. It is shown in [31] that in case P is normal and $W = I$, then

$$J(P) = \frac{1}{N} \sum_{\substack{\lambda \in \sigma(P) \\ \lambda \neq 1}} \frac{1}{1 - |\lambda|^2}$$

which coincides with the cost function proposed in the previous section. It is shown in [12] that in case P is normal and $W = P - I$, then

$$J(P) = \frac{1}{N} \sum_{\substack{\lambda \in \sigma(P) \\ \lambda \neq 1}} \frac{|1-\lambda|^2}{1 - |\lambda|^2}$$

3.4 Performance of the consensus algorithm for distributed estimation

Consider the simple problem of distributed estimation in which N sensors measure the same real value z plus i.i.d. noises. Clearly, the best estimate for z would be the average of such measurements, but sensors can to communicate only along the graph G. The sensors' measurements form a vector $\boldsymbol{x}(0) \in \mathbb{R}^N$, with $x_i(0) = z + n_i$, where the noises $n_1, \ldots, n_N$ are i.i.d. random variables with zero mean and variance r.

In order to estimate z we run the average consensus algorithm and in this case the most natural performance measure is average quadratic error

$$J(P,t) = \tfrac{1}{N}\mathbb{E}\left[e^T(t)e(t)\right]$$

where $e(t)$ is defined as $e(t) = x(t) - z\mathbf{1}$. It can be shown that $J(P,t)$ can be re-written as

$$J(P,t) = \frac{r}{N} \operatorname{trace}\left((P^t)^T P^t\right) \tag{2}$$

If P is normal, then

$$J(P,t) = \frac{r}{N} \sum_{\lambda \in \sigma(P)} |\lambda|^{2t} \tag{3}$$

Notice that the best it can be done by the sensors is to obtain the exact average which is obtained only for $t = \infty$ when we get $J(P,\infty) = r/N$. Therefore the cost due to P is given by

$$\Delta J(P,t) := J(P,t) - r/N.$$

As an evaluation independent of t of this quantities can be obtained by taking

$$J(P) = \sum_{t=0}^{\infty} \Delta J(P,t)$$

It can be shown that

$$\Delta J(P,t) = \frac{1}{N} \sum_{\substack{\lambda \in \sigma(P) \\ \lambda \neq 1}} |\lambda|^{2t}$$

and that

$$J(P) = \frac{1}{N} \sum_{\substack{\lambda \in \sigma(P) \\ \lambda \neq 1}} \frac{1}{1 - |\lambda|^2}$$

which is the same cost function proposed in the previous section.

3.5 Performance of the consensus distributed Kalman filtering

Assume we have a scalar random process $x(t)$ described by the following model

$$z(t+1) = z(t) + w(t)$$

where $w(t)$ is an i.i.d. random sequence in which $w(t)$ is zero mean and variance equal to q. Assume that each node i can take a noisy measurement $y_i(t)$ of $z(t)$ at each time step according to the following model

$$y_i(t) = z(t) + n_i(t)$$

where $n_1(t), \dots, n_N(t)$ are i.i.d. random variables with zero mean and variance r which are independent over different time instants. A possible algorithm which allows the nodes to obtain an estimate of $z(t)$ has been proposed in [23] and it can be described as follows. Let $x_i(t)$ denote the estimate of $z(t)$ of the node i and let $x(t)$ be the vectors having $x_i(t)$ as components. Then we take

$$x(t+1) = (1-\ell)P^m x(t) + \ell y(t)$$

where ℓ is a real number in $]0,1[$ and P is a doubly stochastic matrix. This algorithm updates the estimates by first making m steps of the consensus algorithm, and then by making a convex combination

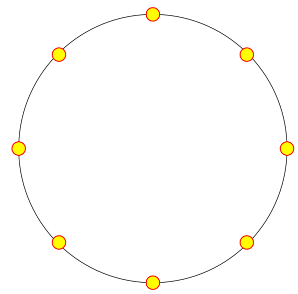

Figure 1. The circle graph

of the obtained vector and the measurements. It can be shown that these consensus steps improve the estimates. If we define the estimation error as $e(t) = x(t) - z(t)\mathbf{1}$ and we consider the cost

$$J(P, \ell, m) := \frac{1}{N} \limsup_{t\to\infty} \mathbb{E}[e(t)^T e(t)]$$

it can be seen that [3], if P is normal, then

$$J(P, \ell, m) = \frac{q(1-\ell)^2}{1-(1-\ell)^2} + \frac{r\ell^2}{N} \sum_{\lambda\in\sigma(P)} \frac{1}{1-(1-\ell)^2|\lambda|^m}$$

In the following examples we will concentrate to the performance indices (1) and (3). Through these examples we will show that these performance indices can suggest very different ranking of possible matrices P with respect to the ranking suggested by the essential spectral radius, which is standard performance index considered for evaluating consensus algorithms.

3.6 Examples

Example 1: The circle graph Consider the following $N \times N$ circulant matrix P

$$P_N = \begin{bmatrix} 1/3 & 1/3 & 0 & \cdots & \cdots & 1/3 \\ 1/3 & 1/3 & 1/3 & \cdots & \cdots & 0 \\ 0 & 1/3 & 1/3 & 1/3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 1/3 & 1/3 & 1/3 \\ 1/3 & \cdots & \cdots & 0 & 1/3 & 1/3 \end{bmatrix}$$

The graph which describes the communication topology between nodes required in this case is shown in figure 1. We assume that all self loops are edges of the graph even though they are not depicted in the figure. The matrix ideally represents communication over a one-dimensional space and in fact the diameter of this graph is proportional to N. It is shown in [4] that in this case $\rho_{\text{ess}}(P_N) \simeq 1 - C/N^2$ where C is a suitable constant. This shows that as N tends to infinity, the convergence of the algorithm

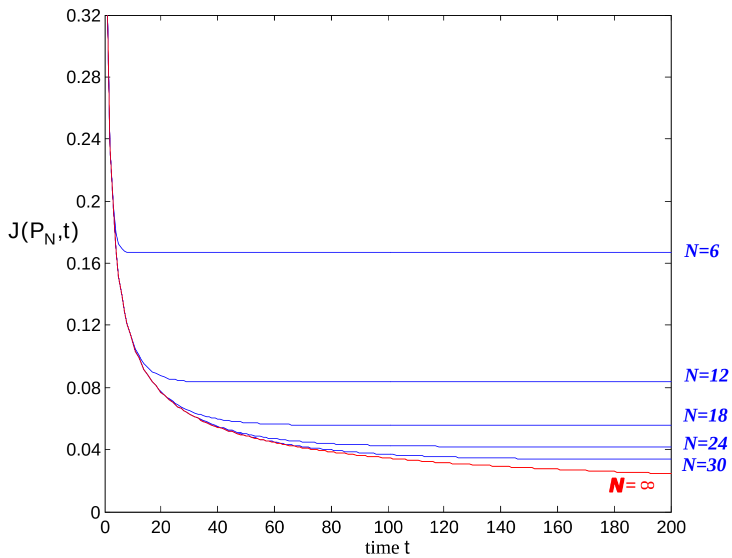

Figure 2. The time evolution of $J(P_N, t)$ for the matrix P_N introduced in Example 1.

tends to be very poor. Nonetheless we expect that, in case of distributed estimation, the presence of more sensors should instead increase the performance. It can be seen that the distributed estimation performance index

$$J(P_N, t) = \max\left\{\frac{1}{N}, \frac{1}{\sqrt{t}}\right\}$$

which shows that the bigger is N, the better is the estimation performance. Figure 2 depicts the time evolutions of $J(P_N, t)$. Notice that for fixed N we have evolutions which exponentially converge (with rate $1-C/N^2$) to a constant value r/N. Notice that $J(P_N, t) = J(P_\infty, t)$ if $t < N$, and that the envelope of the curves, which corresponds to the evolution with $N = \infty$, converges to zero as $1/\sqrt{t}$.

Example 2: The torus graph Let $N = M^2$ and consider the $N \times N$ matrix P_N which is the Kronecker product of two $N \times N$ circulant matrices like the one introduced in the previous example. The graph which describes the communication topology between nodes is shown in figure 3. We assume again that all self loops are edges of the graph even though they are not depicted in the figure. The matrix ideally represents communication over a two-dimensional space and in fact the diameter of this graph is proportional to $\sqrt{N}$.

It is shown in [4] that in this case $\rho_{\text{ess}}(P_N) \simeq 1 - C/N$ where C is a suitable constant. It can be shown conversely that $J(P_N) \simeq C\ \log N$, where C is again a suitable constant, different from the previous one.

Example 3: The two cluster graph Consider now a graph of N nodes. Assume that N is even and consider two complete graphs over $N/2$ nodes. Assume again that all self loops are edges of the graphs. Choose now p nodes in the first graph and other p nodes in the second. Delete the self loops of these $2p$ nodes and add p edges connecting one of the p selected nodes of the first graph with the other p

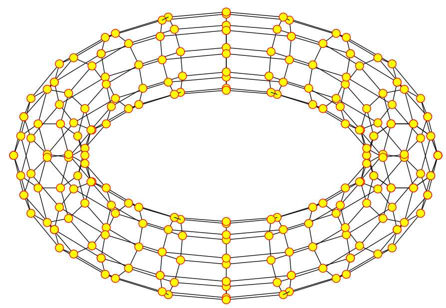

Figure 3. The torus graph

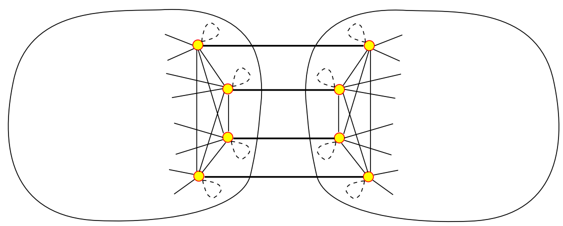

Figure 4. The two cluster graph

selected nodes of the second graph. In this way we obtain a coupling between p nodes of the two graphs. The graph which describes this communication topology is shown in figure 4. Consider the matrix $P_{N,p}$ obtained by multiplying the adjacency matrix of the obtained graph by the $2/N$. We obtain in this way that

$$P_{N,p} = \frac{2}{N} \left\{ \begin{bmatrix} \mathbf{1}\mathbf{1}^T & 0 \\ 0 & \mathbf{1}\mathbf{1}^T \end{bmatrix} - \begin{bmatrix} 0 \\ I \\ -I \\ 0 \end{bmatrix} \begin{bmatrix} 0 & I & -I & 0 \end{bmatrix} \right\}$$

Where $\mathbf{1}$ is the $N/2$ dimensional column vector with all ones, I is the $p \times p$ identity matrix and symbol 0 denotes matrices with zero entries of suitable dimensions. With some computations which we don't present here for space constraints it is possible to prove that $N - p - 2$ eigenvalues of $P_{N,p}$ are zero, one is equal to 1, p eigenvalues are equal to $-2/N$, one is $\simeq -1/2N$ and one is $\simeq 1 - p/N^2$. With this data we can argue that $\rho_{\text{ess}}(P_{N,p}) \simeq 1 - p/N^2$ and that $J(P_{N,p}) \simeq p/N + N/2p$. Taking $p \simeq \epsilon N$ with $\epsilon < 1/2$, we obtain that $\rho_{\text{ess}}(P_{N,p}) \simeq 1 - \epsilon/N$ and that $J(P_{N,p}) \simeq \epsilon + 1/2\epsilon$ which does not grow in N. Comparing this case with what obtained for the torus graph we realize that, while in the two cases the performance index ρ_{ess} has the same behavior has N grows, they present different behaviors with respect to the performance index $J(P_{N,p})$, since for the torus graph this grows logarithmically in N, while for the two cluster graph it does not diverge in N.

4 Anytime reliable transmission of real-valued information and state estimation over noisy digital channels

Reliable transmission of information among the nodes of a network is known to be a relevant problem in information engineering. It is indeed fundamental both when the network is designed for pure information transmission, as well as in scenarios in which the network is deputed to accomplish some specific tasks requiring information exchange. Important examples include: networks of processors performing parallel and distributed computation [30], wireless sensor networks, in which the final goal is estimation and decision making from distributed measurements [15, 17, 32, 10]; sensors/actuators networks, such as mobile multi-agent networks, in which the final goal is control [16, 24, 22]. Distributed algorithms to accomplish synchronization, estimation, or localization tasks, necessarily need to exchange quantities among the agents, which are often real-valued. Assuming that transmission links are digital, a fundamental problem is thus to transmit a continuous quantity, i.e. a real number or, possibly, a vector, through a digital noisy channel up to a certain degree of precision.

This section is concerned with the problem of computational efficient transmission of a a real vector through a noisy digital channel. We shall focus on anytime transmission algorithms, i.e. algorithms which can be stopped anytime while providing estimations of increasing precision.

As especially pointed out in a series of works by A. Sahai and S. Mitter [27, 26, 28], there is a specific feature distinguishing the problem of information transmission for control from the problem of pure information transmission. This is related to the different sensitivity to delay typically occurring in the two scenarios. Indeed, while the presence of sensible delays can often be tolerated in the communication performance evaluation, such delays can be detrimental for the system performance in several control applications. Here, the fundamental question is not only where, but also when the information is available. For this reason, it is often desirable to use transmission systems for control applications which are able to provide estimates whose precision increases with time, so as providing a reasonable partial information transmission anytime the process is stopped.

On the other hand, both in control and communication the computational complexity of the transmission schemes is a central issue. In fact, nodes in wireless networks are usually very simple devices with limited computational abilities and severe energy constraints. Applicable transmission systems should be designed performing a number of operations which remains bounded in time, both in the encoding and in the decoding. Hence, an analysis of the tradeoffs between performance and complexity of the transmission schemes is required.

A fundamental characteristic of digital communication for control applications concerns the nature of information bits. In the traditional communication theory, information bits are usually assumed to be equally valuable, and they are consequently given the same priority by the transmission system designer. In fact, design paradigms of modern low-complexity codes [25] –based on random sparse graphical models and iterative decoding algorithms– treat information bits as equally valuable. While such an assumption is typically justified by the source-channel separation principle, this principle does not generally hold when delay is a primary concern. For instance, it is known that separate source-channel coding is suboptimal in terms of the joint source-channel error exponent [7, 8]. In fact, in many problems of information transmission for control or estimation, different information bits typically require significantly different treatment. As an example, particularly relevant for the topics addressed in this paper, assume that a random parameter, uniformly distributed over a unitary interval, has to be reliably transmitted through a digital noisy channel (see [2] and references therein for the analysis of the information theoretic limits of this problem on the bandwidth-unlimited Gaussian channel). Such a parameter may be represented by its binary expansion, which is a stream of independent identically distributed bits. Clearly, such information bits are not equally valuable, since the first one is more significant than the second one, and so on. This motivates the study of unequal error protection codes [19, 1]. One of the challenges posed by information transmission for control/estimation applications is to come up with design paradigms for practical, low computational complexity, unequal error protection codes.

4.1 Notation and problem formulation

The symbol $\mathbb{R}$ and $\mathbb{N}$ will denote the sets of reals and naturals, respectively. A sequence x_t, $t = 1, 2, \ldots$ is sometimes denoted with the symbol $x = (x_t)_{t=1}^{\infty}$, while with the symbol $x = (x_t)_{t=1}^{T}$ we will mean its truncation to $t = 1, \ldots, T$.

We shall now provide a formal description of the problem. Let x be a random variable taking values on $\mathcal{X} \subseteq \mathbb{R}^d$ and assume we want to transmit an approximation of x through a digital noisy channel. Through this channel in a period of t seconds it can be transmitted $n = at$ bits $y_1, \ldots, y_n \in \{0, 1\}$. The parameter a together with the channel characteristics provide the communication capabilities of the system. The communication channel has output z_i taking values in another finite alphabet $\mathcal{Z}$. Transmission is assumed to be memoryless, i.e., given the current input y_i, the output z_i is assumed to be conditionally independent from the previous inputs $(y_s)_{s=1}^{i-1}$ and outputs $(z_s)_{s=1}^{i-1}$, as well as from the vector x. The conditional probability of $z_i = z$ given $y_i = y$ will be assumed stationary and denoted by $p(z|y)$.

The anytime transmission scheme consists of an encoder and a sequential decoder. The encoder consists of a family of maps $E_i : \mathcal{X} \to \mathcal{Y}$, specifying the i-th bit $y_i \in \{0, 1\}$ transmitted through the channel, namely $y_i = E_i(x)$. The decoder instead is given by a family of maps $\mathcal{D}_i : \mathcal{Z}^i \to \mathcal{X}$, describing the estimate $\hat{x}_i = \mathcal{D}_i((z_s)_{s=1}^{i})$ of x obtained from the string $(z_s)_{s=1}^{i}$ that has been received until time t.

4.2 Performance indices and communication and computation constraints

In order to evaluate the performance of a scheme, we introduce mean squared error after the transmission of n bits $y_1, \ldots, y_n$ by

$$\Delta_n := (\mathbb{E}||x - \hat{x}_n||^2)^{1/2} . \tag{4}$$

We are typically concerned with the rate of decay of Δ_n as a function of the number n of transmitted bits, for different anytime transmission schemes.

In this way we can compare the communication resources with the performance which can be obtained through different strategies. We can include into the picture also the computation resources as follows. Assume that the encoder and/or the decoder tasks are run by a computer unit which is able to do bt operations in a time interval of t seconds (the parameter b can be seen as the computation capability of the system). Now it makes sense to understand which is the precision we can get after t seconds, a time interval in which at most at bits can be sent and in which at most bt operations can be completed. In this way we will obtain the same mean squared error Δ_t, but now as a function of the time t.

In this way we can compare different strategies. In [27] a coding procedure has been proposed for general digital channels. This strategy uses the channel very efficiently, since the error after decoding is an exponentially decreasing function of the transmitted bits. However, this method is very computationally-inefficient, since it requires a number of operations which is an exponential function of the number of transmitted bits. Consequently, it can be shown that the error Δ_t, as a function of the time available, is only polynomially decreasing. In [6] other two procedures are proposed for the particular case of an erasure channel. One scheme, as for Sahai's scheme, provides an exponentially decreasing error, as a function of the number n of transmitted bits, and it requires $\simeq n^3$ operations from the computing unit. As a result we get an error Δ_t that, as a function of the time available, is of the form

$$\Delta_t \simeq 2^{-\alpha t^{1/3}}$$

which is subexponential in t. In the same paper [6] another simple (almost trivial) scheme is proposed, which seems very poor under a communication point of view, since it yields an error Δ_n as a function of the the number n transmitted bits which is subexponential, namely $\Delta_n \simeq 2^{-\beta n^{1/2}}$. However, since it requires only $\simeq n$ operation to process n bits, it yields an error Δ_t that, as a function of the time available, is of the form

$$\Delta_t \simeq 2^{-\gamma t^{1/2}}$$

which is the best of the three methodologies.

4.3 Application to state estimation under communication constraints

The problem illustrated in the previous paragraph is related to the state estimation problem under communication constraints (see [29, 20] and references therein). Assume we are given a discrete time stochastic linear system

$$x(t+1) = Ax(t) + v(t), \qquad x(0) = x_0, \tag{5}$$

where $x_0 \in \mathbb{R}^n$ is a random vector with zero mean, $v(t) \in \mathbb{R}^n$ is a zero-mean white noise, $x(t) \in \mathbb{R}^n$ is the state sequence, and $A \in \mathbb{R}^{n \times n}$ is a full rank, unstable matrix.

Suppose that a remotely positioned receiver is required to estimate the state of the system, while observing the output of a binary erasure channel only. Then, it is necessary to design a family of encoders E_t and of decoders D_t. At each time $t \geq 0$, the encoder E_t takes $x(0), \ldots, x(t)$ as input, and returns the symbol $y_t \in \{0, 1\}$, which is in turn fed as an input to the channel. The receiver observes the channel output symbols $z_0, \ldots, z_t$, from which the decoder D_t has to obtain an estimate $\hat{x}(t)$ of the current state.

If we have that $v(t) = 0$ for every $t \geq 0$, then the only source of uncertainty is due to the initial condition x_0. Hence, in this case, the encoder/decoder task reduces to obtaining good estimates of x_0 at the receiver side. Indeed, in order to obtain a good estimate $\hat{x}(t)$ of $x(t)$, the receiver has to obtain the best possible estimate $\hat{x}(0|t)$ of the initial condition $x(0)$ from the received data $y_0, \ldots, y_t$, and then it can define $\hat{x}(t) := A^t \hat{x}(0|t)$. In this way, one has $x(t) - \hat{x}(t) = A^t(x(0) - \hat{x}(0|t))$, so that the problem reduces to finding the best way of coding $x(0)$ in such a way that expansion of A^t is well dominated by the contraction of $x(0) - \hat{x}(0|t)$. The same technique can be applied if $v(t)$ is small with respect to x_0 as clarified by the following example.

Example 1: Consider the following unstable scalar discrete time linear system

$$x(t+1) = ax(t) + v(t), \qquad x(0) = x_0,$$

where $a > 1$ and where x_0 is a random variable with probability density $f(x)$ and $v(t)$ is a sequence of independent, identically distributed random variables with zero mean and variance σ_v^2, which are independent of x_0. Assume that a state estimation algorithm is run, based on the noiseless model $x(t+1) = ax(t)$ by estimating the initial condition x_0 from data transmitted until time t. As before, we shall denote this estimate by $\hat{x}(0|t)$. From $\hat{x}(0|t)$, we form the estimate $\hat{x}(t) := a^t\hat{x}(0|t)$ of $x(t)$. The estimation error at time t will be $e(t) := x(t) - \hat{x}(t) = a^t(x(0) - \hat{x}(0|t)) + \sum_{i=0}^{t-1} a^{t-1-i}v(i)$, so that $\mathbb{E}[e(t)^2] = a^{2t}\mathbb{E}[(x(0)-\hat{x}(0|t))^2]+\sigma_v^2\frac{1-a^{2t}}{1-a^2}$. This error depends both on the error in the estimation of the initial condition, and on the wrong model we used. As we shall see, our techniques yield an estimation error on $x(0)$ of the form $\mathbb{E}[(x(0) - \hat{x}(0|t))^2] = C\zeta(t)$, where C depends only on the probability density $f(x)$ and $\zeta(t)$ is a function converging to zero depending only on the communication channel characteristics and on the coding strategy. Therefore,

$$\mathbb{E}[e(t)^2] = a^{2t}\left[C\zeta(t) + \sigma_v^2\frac{1-a^{-2t}}{a^2-1}\right].$$

In case C is much larger than σ_v^2, there will be an initial time regime in which the error is not influenced by the model noise but only by the estimation of the initial condition $x(0)$.

5 Consensus and distributed estimation with noisy digital channels

The problem described in previous section, i.e., efficient transmission of real values over digital noisy channels, can be seen as a first step towards a more ambitious goal: a theory of in-network computation, which deals with the solution of distributed computing, estimation or control problems when local agents (e.g., sensors) are allowed to communicate only with a limited number of neighbors, and only through digital noisy channels (e.g., binary wireless broadcast channels). A tutorial overview of the challenges of such a research area can be found in [14].

Also in this more general framework, it is essential to have the correct tools to describe the complexity of an algorithm, by taking into account both the number of transmissions (channel uses) and the computation time required to achieve a desired precision, so as to choose the algorithms which give the best trade-off.

In order to clarify the issues that arise in this kind of scenario, in [13] we have considered the example of distributed averaging in the presence of erasures. Here, we will describe the problem, and discuss the performance metrics; then we will show one strategy that we proposed in [5] which produces a class of algorithms for solving this problem, and compare different algorithms according to the suitable performance index.

5.1 Problem statement

We shall consider a set $\mathcal{V}$ of N agents, possibly representing sensors in a wireless network, each having access to some partial information consisting in the observation of a scalar value, which for instance may represent the measurement of some physical quantity. The observation of agent v will be denoted by $\theta_v \in [0,1]$, while $\boldsymbol{\theta} = (\theta_v)$ will indicate the full vector of observations. The goal is for the network to compute the arithmetic average of such values, $\theta_{\text{ave}} = \frac{1}{N}\sum_v \theta_v$ by exchanging information among themselves and without a centralized computing system.

Communication among the agents takes place as follows. At each channel use $n = 1, 2, \ldots$, every agent v broadcasts a bit $a_v(n) \in \{0,1\}$ to a subset of agents, to be denoted by $\mathcal{N}_v^+ \subset V$. Every agent $w \in \mathcal{N}_v^+$ receives a possibly erased version $b_{v,w}(n) \in \{0,1,?\}$ of $a_v(n)$. The set of in-neighbors of agent

w will be denoted by $\mathcal{N}_w^- = \{v : w \in \mathcal{N}_v^+\}$, whereas $b_w(n) = (b_{v,w}(n))_{v\in\mathcal{N}_w^-}$ will denote the vector of signals received by agent w at n-th channel use. For simplicity, we consider the case of erasure channels: $b_{v,w}(n) = a_v(n)$ (i.e., the bit is correctly received) with probability $1-\varepsilon$, and $b_{v,w}(t) =?$ (i.e., an erasure occurred) with probability ε, where $\varepsilon > 0$ is some erasure probability which is assumed to remain constant in n, v and w; furthermore, erasures will be assumed independent in successive transmission (memoryless channel). Distributedness of the communication algorithm is then enforced by requiring that the bit $a_v(n)$ is a function of the local information available to agent v after n transmission rounds, i.e. of its initial observation θ_v, as well as the signals $\{b_v(s)\}_{1\le s<n}$ received by agent v up to $(n-1)$-th transmission round.

The communication setting outlined above can be conveniently described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ (the communication graph), whose vertices are the agents, and such that an ordered pair (i, j) with $i \neq j$ belongs to $\mathcal{E}$ if and only if $j \in \mathcal{N}_i^+$ (equivalently $i \in \mathcal{N}_j^-$), i.e., if i transmits to j with erasure probability $\varepsilon < 1$. We assume that the graph $\mathcal{G}$ is strongly connected, i.e. that any two nodes u, v are connected by a directed path. We will also assume that $\mathcal{G}$ has self-loops on each vertex; this represents the fact that each node has access to its own information, which is equivalent to assume a noiseless channel available from i to itself.

5.2 Performance measures

Given the communication scenario outlined above, an algorithm for solving the distributed averaging problem must specify how, at time t, each agents computes its current estimate of θ_{ave}, which can depend only on its initial observation and on the bits received from its neighbours up to time t. If you denote by $x(t)$ the vector of all agents' estimates, in order to analyze the performance of different distributed computation algorithms, we shall study the distance from the desired average $\theta_{\text{ave}} = \frac{1}{N}\boldsymbol{\theta}^T\mathbf{1}$:

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \theta_{\text{ave}}\mathbf{1} .$$

and in particular the average quadratic error $\mathbb{E}\left[||\boldsymbol{e}(t)||^2\right]$.

A measure of the global (transmission and computation) complexity of the algorithm is given by

$$T(\delta) := \inf\left\{t \ge 0 : \frac{1}{N}\mathbb{E}\left[||\boldsymbol{e}(s)||^2\right] \le \delta,\ \forall s \ge t\right\}, \qquad \delta \in]0,1] .$$

In other words, for $\delta \ge 0$, $T(\delta)$ denotes the minimum time which is needed to guarantee that the average mean squared estimation error does not exceed δ.

Clearly, in some applications it can be more interesting to consider separately the transmission and the computation time, for example because transmission is very energy-demanding and thus it can be useful to define a similar performance measure which involves transmission time only. It is easy to see in what follows that the comparison of the algorithms we propose gives a substantially different result after such a change of point of view in the definition of relevant time.

5.3 Proposed algorithm

Our idea is to use a traditional linear average-consensus algorithm, of the form $x(k+1) = Px(k)$, combined with a technique for transmission of real numbers over noisy digital broadcast channels, in particular we will consider the joint source and channel coding proposed in [6] and described in previous paragraph.

Consider the above-described scenario, specified by a set of N vertices $\mathcal{V}$, a strongly connected communication graph $\mathcal{G}$, and an erasure probability ε. The ingredients of our algorithm are:

- a consensus matrix P, i.e., a doubly-stochastic primitive matrix adapted to $\mathcal{G}$, with non-zero diagonal entries;
- an increasing sequence of positive integers $\{\tau(k)\}_{k\in\mathbb{N}}$, such that $\lim_{k\to\infty}\tau(k) = +\infty$; $\tau(k)$ represents the number of bits that each node transmits at k-th iteration of the algorithm;
- a sequence of encoders, i.e., maps $\phi^{(k)} : [0,1] \to \{0,1\}^{\tau(k)}$;
- a sequence of decoders, i.e., maps $\psi^{(k)} : \{0,1,?\}^{\tau(k)} \to [0,1]$.

Recall that linear average-consensus algorithm (see Sect. 3), after initialization $\boldsymbol{x}(0) = \boldsymbol{\theta}$, is: $\boldsymbol{x}(k+1) = P\boldsymbol{x}(k)$, i.e., at k-th iteration, node i receives from its in-neighbors the numbers $x_j(k)$, $j \in \mathcal{N}_i^-$, and updates its state:

$$x_i(k+1) = \sum_{j\in\mathcal{N}_i^-} P_{ij}x_j(k)\,.$$

We propose to adapt this algorithm, in a way that takes into account the necessity to transmit the real values $x_j(k)$ along digital noisy channels. The initialization of the algorithm is unchanged: $\boldsymbol{x}(0) = \boldsymbol{\theta}$. Between iterations k and $k+1$ of our consensus-like algorithm, we allow each node j to broadcast $\tau(k)$ bits to its neighbors:

- the bits transmitted by node j at iteration k are the message $a_j(k) = \phi^{(k)}(x_j(k))$, i.e., a suitably encoded version of the state $x_j(k)$;
- each $i \in \mathcal{N}_j^+$ will receive a corrupted version of $a_j(k)$, $b_{ij}(k)$, and will use the decoder $\psi^{(k)}$ to recover an estimate $\hat{x}_{ij}(k) = \psi^{(k)}(b_{ij}(k))$

Then, the next consensus iteration will take place, where node i will use $\hat{x}_{ij}(k)$ to replace the exact state $x_j(k)$ which he can not know exactly:

$$x_i(k+1) = P_{ii}x_i(k) + \sum_{j\in\mathcal{N}_i^-} P_{ij}\hat{x}_{ij}(k)$$

Clearly, we can write $\hat{x}_{ij}(k) = x_j(k) + v_{ij}(k)$, and we might think at $v_{ij}(k)$ as a residual noise which could not be removed by the error-correction performed by the decoder. A suitable choice of the encoder/decoder pairs and of the transmission phases allows to obtain a noise decreasing with respect to k, with a speed which depends on the computation complexity which we choose to allow for the encoding/decoding scheme (see Sect. 4.2). To this effect, the assumption that the transmission lengths $\tau(k)$ are increasing in k is essential, because the coding gain is asymptotic in the length of codewords. This remarks leads us to name our proposed algorithm ‘Increasing Precision Algorithm’ (IPA).

Notice that k takes into account the number of consensus-like updates. However, we must take into account the time that truly elapses. With a slight abuse of notation, we denote by $\boldsymbol{x}(t)$ the state at time t; by this we mean that $\boldsymbol{x}(t)$ is kept constant within the whole encoding/transmission/decoding phase, and then changes when the consensus-like update takes place. Notice that k-th transmission phase consists of $\tau(k)$ bits, and thus requires a time equal to $a\tau(k)$, for some coefficient a which is a property of the transmission devices. Then denote by $\kappa(k)$ the number of operations required at k-th phase: this is a function of the length $\tau(k)$, which depends on the choice of the encoding/decoding scheme. Clearly the computations required at k-th phase occupy a time $b\tau(k)$, for some constant b which describes the computing speed allowed for the agents. So, finally, k-th state update takes place at time $t(k) = \sum_{h\leq k}(\tau(k) + \kappa(k))$.

We consider in particular two instances of our IPA algorithm, differentiated by the choice of the encoding/decoding scheme and the choice of the transmission lengths $\{\tau(k)\}_{k\in\mathbb{N}}$. In both cases we consider anytime encoding/decoding schemes proposed in [6] and described in Sect. 4.2, so that each agent needs to memorize only one encoding and one decoding rule, which then can be applied for the desired transmission length; also, for simplicity, we assume all agents implement the same encoder and decoder. The two versions of the algorithm are the following:

(a) a high-performance code, based on linear random trees, which ensures an estimate error decay, after τ transmitted bits, which is exponential in τ: $\mathbb{E}[(x-\hat{x})^2] \leq C\beta^{2\tau}$ for some constants $C > 0$ and $\beta \in (0,1)$ depending on the erasure probability ε only. In this case, a suitable choice of transmission lengths is linear: $\tau(k) = Sk$ where $S > 0$ is a parameter to be designed. The good error-correcting capability comes at the price of computational complexity, particularly in the decoding process: the number of required operations grows cubically with the transmission length: $\kappa(k) = K\tau(k)^3 = KS^3k^3$ for some $K > 0$. Thus, k-th state update takes place at time

$$t(k) = \sum_{h\leq k}(aSh + bKS^3h^3) \simeq k^4$$

(b) a simple coding technique, with lower performance but also lower complexity. The error decay after τ transmitted bits is $\mathbb{E}[(x-\hat{x})^2] \leq C\beta^{2\sqrt{\tau}}$, and a suitable choice of the lengths is $\tau(k) = Sk^2$ so that the error decay is still exponential in k, even though this comes with a longer time devoted to transmission. The advantage is on the computational complexity, which is: $\kappa(k) = K\tau(k) = KSk^2$. Thus, k-th state update takes place at time

$$t(k) = \sum_{h\leq k}(aS^2h^2 + bKSk^2) \simeq k^3$$

5.4 Performance analysis

We study the distance from the correct average at k-th state iteration $\boldsymbol{e}(k) = \boldsymbol{x}(k) - \theta_{\text{ave}}\mathbf{1}$, because then it is easy to find results for the error at a given time t, $\boldsymbol{e}(t)$, using the different expressions for $t(k)$.

The main result of our analysis is that

$$\frac{1}{N}\mathbb{E}[\boldsymbol{e}(k)^2] \leq \rho^{2k}\frac{1}{(1-\alpha/\rho)^2} + \frac{\alpha^2}{(1-\alpha)^2}$$

The first term is due to the distance from the current average of their states (i.e., how far agents are from meeting in the nearest point, whichever it happens to be), and decays exponentially in k with a speed given by the graph topology (ρ being the essential spectral radius of P). The second term is due to the distance within the current average and the desired average of the initial states; this can be made arbitrarily small by choosing a larger length of the initial transmission phase, because $\alpha = \beta^S$, where $\beta \in (0,1)$ depends on the channel's erasure probability only, and S is a parameter which can be designed to fulfil the desired precision requirement.

Finally, this allows to say that the time required to achieve a desired precision δ is the following for the two versions of the IPA algorithm:

(a) $T(\delta) \leq C_1 + C_2 \dfrac{\log^4(\delta^{-1})}{\log^4(\rho^{-1})}$;

(b) $T(\delta) \leq C_1' + C_2' \dfrac{\log^3(\delta^{-1})}{\log^3(\rho^{-1})}$.

where C_1, C_2, C_1', C_2' are positive constants depending only on ε, a, b, i.e., on the given transmission and computation capabilities of the given network, while ρ is the essential spectral radius of P.

6 Performance metrics in camera networks

Other interesting performance metrics have been proposed within the WP7 case-study on camera networks. Regarding the surveillance and multiple target tracking, it is possible to utilize the following performance metrics:

- Trace reconstruction: it is the number of frames a target is effectively and continuously tracked by one or more sensors/cameras before being lost
- Information acquired from a target: this can be measured in terms of
  - target dimension on the image plane of a camera
  - the number of cameras (i.e., the number of different points of view) focusing on a target at the same time

As for the motion capture problem, the performance of the camera control algorithms can be formulated as the minimization of an error metric based on the reconstructions of the various markers: at the reconstruction stage a reconstruction and corresponding covariance matrix are produced for each visible marker, and the objective would be to minimize an index that depends on the overall quality of reconstruction based on all the information gathered from various cameras. The index will be minimized when all markers are clearly viewed by all cameras; under camera resource constraints this error will be minimized when cameras with a wide baseline are used together for a specific set of markers.

7 Conclusion

In this report we considered the problem of finding performance metrics which are suitable for the networked control systems. The difficulty of this task arises from the fact that networked control applications naturally concerns problems in which control theory, communication theory and computer science play a role. As a consequence, the heterogeneity of such disciplines, and in particular the great difference in the performance indices used in the various research communities, poses a challenge in selecting a universal performance criterion. The proposed strategy is to start from the specific networked control or estimation problem that we want to solve, since it will naturally suggest the performance index to be considered. Through the aid of some examples we showed that this simple strategy can yield to new interesting and relevant theoretical questions whose solution will indeed require a synthesis of control, communication and computer science methodologies.

References

[1] S. Borade, B. Nakiboğlu, and L. Zheng, Unequal error protection: some fundamental limits, IEEE Trans. Inform. Theory, (2009), to appear.

[2] M.V. Burnashev, A new lower bound for the α-mean error of parameter transmission over the white Gaussian channel, IEEE Trans. Inform. Theory, 30(1984), pp. 23-34.

[3] R. Carli, A. Chiuso, L. Schenato and S. Zampieri (2008), "Distributed Kalman filtering using consensus strategies", IEEE Journal on Selected Areas in Communications, 26(4), pp. 622-633.

[4] R. Carli, F. Fagnani, A. Speranzon, S. Zampieri, Communication constraints in the average consensus problem, Automatica 44 (3), pp. 671-684, 2008.

[5] R. Carli, G. Como, P. Frasca, and F. Garin, Average consensus on digital noisy networks, 1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys'09), Venice (Italy), 2009

[6] G. Como, F. Fagnani, and S. Zampieri. Anytime reliable transmission of real-valued information through digital noisy channels. *SIAM Journal on Control and Optimization*, December 2008. Submitted.

[7] I. Csiszàr, Joint source-channel error exponent, Probl. Control Inf. Theory, 9(1980), pp. 315-328.

[8] I. Csiszàr, On the error exponent of source-channel transmission with a distortion threshold, IEEE Trans. Inform. Theory, 28(1982), pp. 823-828.

[9] P. Diaconis, and D. Stroock, Geometric bounds for eigenvalues of Markov chains The Annals of Applied Probability, pp. 36-61, 1991

[10] A.G. Dimakis, A.D. Sarwate, M.J. Wainwright, Geographic Gossip: Efficient Averaging for Sensor Networks, IEEE Transactions on Signal Processing, 2008.

[11] F. Fagnani, and S. Zampieri. Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, Vol. 48, pp. 102-133, 2009.

[12] Paolo Frasca, Ruggero Carli, Fabio Fagnani and Sandro Zampieri , Average consensus on networks with quantized communication, To appear on International Journal of Robust and Non-Linear Control, 2009.

[13] Federica Garin and Sandro Zampieri , Performance of consensus algorithms in large-scale distributed estimation, European Control Conference 2009, Budapest.

[14] A. Giridhar and P. R. Kumar. Towards a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4):98-107, 2006.

[15] C. Intanagonwiwat, R. Govindan, and D. Estrin, Directed diffusion: a scalable and robust communication paradigm for sensor networks, In Proc. ACM/IEEE Conf. Mobile Computing and Networking, 2000, pp. 56-67.

[16] A. Jadbabaie, J. Lin, and A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, IEEE Trans. Automat. Control, 48(2003), pp. 988-1001.

[17] D. Kempe, A. Dobra, and J. Gehrke, Gossip-based computation of aggregate information, Proc. IEEE FOCS 2003, pp. 1-10.

[18] Eyal Kushilevitz, and Noam Nisan, Communication complexity, Cambridge University Press, 1997.

[19] B. Masnick and J. Wolf, On linear unequal error protection, IEEE Trans. Inf. Theory, 13(1967), pp. 600-607.

[20] A.S. Matveev and A.V. Savkin, *Estimation and control over communication networks*, Birkhauser, Boston, MA, 2009.

[21] Sean Meyn, and Richard L. Tweedie, Markov Chains and Stochastic Stability, Cambridge University Press, 2009.

[22] L. Moreau, Stability of multiagent systems with time-dependent communication links, IEEE Trans. Automat. Control, 50(2005), pp. 169-182.

[23] R. Olfati-Saber. "Distributed Kalman Filtering for Sensor Networks," Proc. of the 46th IEEE Conference on Decision and Control, Dec. 2007

[24] R. Olfati-Saber and R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, IEEE Trans. Automat. Control, 49(2004), pp. 1520-1533.

[25] T.J. Richardson and R. Urbanke, Modern Coding Theory, Cambridge Univeristy Press, Cambridge, UK, 2007.

[26] A. Sahai, Why do block length and delay behave differently if feedback is present?, IEEE Trans. Inform. Theory, 54(2008), pp. 1860-1886.

[27] A. Sahai and S. Mitter, The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link–Part I: scalar systems, IEEE Trans. Inform. Theory, 52(2006), pp. 3369-3395.

[28] A. Sahai and S. Mitter, Source coding and channel requirements for unstable processes, submitted, 2006.

[29] S. Tatikonda and S. Mitter, Control under communication constraints, IEEE Trans. Automat. Control, 49(2004), pp. 1056-1068.

[30] D. Bertsekas, and J. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, Belmont, MA, 1997.

[31] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. J. Parallel Distrib. Comput., 67(1):33-46, 2007.

[32] J. Zhao, R. Govindan and D. Estrin, Computing aggregates for monitoring wireless sensor networks, Proc. SNPA 2003, pp. 139-148.

[33] S. Zilberstein, Using Anytime Algorithms in Intelligent Systems, AI Magazine, 17(1996), pp. 73-83.