December, 1983

LIDS-P-1342

CONVERGENCE THEORIES OF DISTRIBUTED ITERATIVE PROCESSES: A SURVEY

by

Dimitri P. Bertsekas*
John N. Tsitsiklis**
Michael Athans*

ABSTRACT

We consider a model of distributed iterative algorithms whereby several processors participate in the computation while collecting, possibly stochastic information from the environment or other processors via communication links. Several applications in distributed optimization, parameter estimation, and communication networks are described. Issues of asymptotic convergence and agreement are explored under very weak assumptions on the ordering of computations and the timing of information reception. Progress towards constructing a broadly applicable theory is surveyed.

---

* The research of D.P. Bertsekas was supported by NSF-ECS-8217668 and under DARPA Grant ONR-N00014-75-C-1183. The research of J.N. Tsitsiklis and M. Athans was supported by ONR-N00014-77-C-0532(NR- 041-519).

* Dept. of Electrical Engineering and Computer Science, Laboratory for Information and Decision Systems, M.I.T., Cambridge, Mass. 02139.

** Dept. of Electrical Engineering, Stanford University, Stanford, California.

2. A Distributed Iterative Computation Model

In our model we are given a set of feasible decisions X and we are interested in finding an element of a special subset X* called the solution set. We do not specify X^* further for the time being. An element of X* will be referred to as a solution. Without loss of generality we index all events of interest (message transmissions and receptions, obtaining measurements, performing computations) by an integer time variable t. There is a finite collection of processors i=1,...,n each of which maintains an estimate $x^i(t) \in X$ of a solution and updates it once in a while according to a scheme to be described shortly. The ith processor receives also from time to time m_i different types of measurements and maintains the latest values $z_1^i, z_2^i, \ldots, z_{m_i}^i$ of these measurements. (That is, if no measurement of type j is received at time t, then $z_j^i(t) = z_j^i(t-1)$). The measurement z_j^i is an element of a set Z_j^i. Each time a measurement z_j^i of type j is received by processor i the old value z_j^i is replaced by the new value and the estimate x^i is updated according to

$$x^i(t+1) = M_{ij}(x^i(t), z_1^i(t), \ldots, z_{m_i}^i(t)) , \tag{2.1}$$

where M_{ij} is a given function. Each node i also updates from time to time the estimate x^i according to

$$x^i(t+1) = C_i(x^i(t), z_1^i(t), \ldots, z_{m_i}^i(t)) \tag{2.2}$$

where C_i is a given function. Thus at each time t each processor i either receives a new measurement of type j and updates x^i according to (2.1), or

updates x^i according to (2.2), or remains idle in which case $x^i(t+1) = x^i(t)$ and $z_j^i(t) = z_j^i(t-1)$ for all j. The sequence according to which a processor executes (2.1) or (2.2) or remains idle is left unspecified and indeed much of the analysis in this paper is oriented towards the case where there is considerable a priori uncertainty regarding this sequence. Note that neither mapping M_{ij} or C_i involves a dependence on the time argument t. This is appropriate since it would be too restrictive to assume that all processors have access to a global clock that records the current time index t. On the other hand the mappings M_{ij} and C_i may include dependences on local clocks (or counters) that record the number of times iterations (2.1) or (2.2) are executed at processor i. The value of the local counter of processor i may be artificially lumped as an additional component into the estimate x^i and incremented each time (2.1) or (2.2) are executed.

Note that there is redundancy in introducing the update formula (2.2) in addition to (2.1). We could view (2.2) as a special case of (2.1) corresponding to an update in response to a "self-generated" measurement at node i. Indeed such a formulation may be appropriate in some problems. On the other hand there is often some conceptual value in separating the types of updates at a processor in updates that incorporate new exogenous information (cf. (2.1)), and updates that utilize the existing information to improve the processor's estimate (cf. (2.2)).

The measurement $z_j^i(t)$, received by processor i at time t, is related to the processor estimates $x^1, x^2, \ldots, x^n$ according to an equation of the form

$$z_j^i(t) = \phi_{ij}(x^1(\tau_j^{i1}(t)), x^2(\tau_j^{i2}(t)), \ldots, x^n(\tau_j^{in}(t)), \omega), \qquad (2.3)$$

where ω belongs to the sample space Ω corresponding to a probability space (Ω, F, P), and $\tau_j^{ik}(t) \leq t$, for every i,j,k.

We allow the presence of delays in equation (2.3) in the sense that the estimates $x^1,\ldots,x^n$ may be the ones generated via (2.1) or (2.2) at the corresponding processors at some times $\tau_j^{ik}(t)$, prior to the time t that $z_j^i(t)$ was received at processor i. Furthermore the delays may be different for different processors. We place the following restriction on these delays which essentially says that successive measurements of the same type depend on successive processor estimates.

Assumption 2.1: If $t \geq t'$, then

$$\tau_j^{ik}(t) \geq \tau_j^{ik}(t'), \quad \forall i,j,k \quad .$$

For the time being, the only other assumption regarding the timing and sequencing of measurement reception and estimate generation is the following:

Assumption 2.2 (Continuing Update Assumption): For any i and j and any time t there exists a time $t'>t$ at which a measurement z_j^i of the form (2.3) with $\tau_j^{ik}(t') > t$, $k = 1,\ldots,n$ will be received at i and the estimate x^i will be updated according to (2.1). Also for any i and time t there exists a time $t''>t$ at which the estimate x^i will be updated according to (2.2).

The assumption essentially states that each processor will continue to receive measurements in the future and update his estimate according to (2.1) and (2.2). Given that we are interested in asymptotic results there isn't much we can hope to prove without an assumption of this type. In order to formulate substantive convergence results we will also need further assumptions on the nature of the mappings M_{ij}, C_{ij}, and ϕ_{ij} and possibly on the relative timing of measurement

4. Convergence of Contracting Processes

In our effort to develop a general convergence result for the distributed algorithmic model of Section 2 we draw motivation from existing convergence theories for (centralized) iterative algorithms. There are several theories of this type (Zangwill [15], Luenberger [16], Ortega and Rheinboldt [17]--the most general are due to Polak [18] and Poljak [19]). Most of these theories have their origin in Lyapunov's stability theory for differential and difference equations. The main idea is to consider a generalized distance function (or Lyapunov function) of the typical iterate to the solution set. In optimization methods the objective function is often suitable for this purpose while in equation solving methods a norm of the difference between the current iterate and the solution is usually employed. The idea is typically to show that at each iteration the value of the distance function is reduced and reaches its minimum value in the limit.

The result of this section is based on a similar idea. However instead of working with a generalized distance function we prefer to work (essentially) with the level sets of such a function; and instead of working with a single processor iterate (as in centralized processes) we work with what may be viewed as a state of computation of the distributed process which includes all current processor iterates and all latest information available at the processors.

The subsequent result is reminiscent of convergence results for successive approximation methods associated with contraction mappings. For this reason we refer to processes satisfying the following assumption as contracting processes. In what follows in this section we assume that the feasible set X in the model of Section 2 is a topological space so we can talk about convergence of sequences in X.

Assumption 3.1: There exists a sequence of sets X(k) with the following properties:

a) $X^* \subset X(k+1) \subset X(k) \subset \ldots \subset X$

b) If $\{x_k\}$ is a sequence in X such that $x_k \in X(k)$ for all k, then every limit point of $\{x_k\}$ is a solution.

c) For all i, j and k denote $z^i = (z_1^i, \ldots, z_{m_i}^i)$ and

$$Z_j^i(k) = \{\phi_{ij}(x^1, \ldots, x^n, \omega) \mid x^1 \in X(k) \ldots, x^n \in X(k),\ \omega \in \Omega\} \tag{4.1}$$

$$Z^i(k) = Z_1^i(k) x Z_2^i(k) x, \ldots, x Z_{m_i}^i(k),$$

$$\bar{X}^i(k) = \{C_i(x^i, z^i) \mid x^i \in X(k),\ z^i \in Z^i(k)\} \tag{4.2}$$

$$\bar{Z}_j^i(k) = \{\phi_{ij}(x^1, \ldots, x^n, \omega) \mid x^1 \in \bar{X}^1(k), \ldots, x^n \in \bar{X}^n(k),\ \omega \in \Omega\}, \tag{4.3}$$

$$\bar{Z}^i(k) = \bar{Z}_1^i(k) x \bar{Z}_2^i(k) x, \ldots, x \bar{Z}_{m_i}^i(k).$$

The sets X(k) and the mappings ϕ_{ij}, M_{ij}, and C_i are such that for all i,j and k

$$\bar{X}^i(k) \subset X(k) \tag{4.4}$$

$$M_{ij}(x^i, z^i) \in X(k), \qquad \forall x^i \in X(k),\ z^i \in Z^i(k), \tag{4.5}$$

$$M_{ij}(x^i, z^i) \in \bar{X}^i(k), \qquad \forall x^i \in \bar{X}^i(k),\ z^i \in Z^i(k), \tag{4.6}$$

$$M_{ij}(x^i, z^i) \in X(k+1), \qquad \forall x^i \in \bar{X}^i(k),\ z^i \in \bar{Z}^i(k), \tag{4.7}$$

$$C_i(x^i, z^i) \in X(k+1), \qquad \forall x^i \in X(k+1),\ z_i^i \in \bar{Z}^i(k), \tag{4.8}$$

Assumption 3.1 is a generalized version of a similar assumption in reference [13]. Broad classes of deterministic specialized processes satisfying the assumption are given in that reference. The main idea is that membership in the set X(k) is

representative in some sense of the proximity of a processor estimate to a solution. By part b), if we can show that a processor estimate successively moves from $X(0)$ to $X(1)$, then to $X(2)$ and so on, then convergence to a solution is guaranteed. Part c) assures us that once all processor estimates enter the set $X(k)$ then they remain in the set $X(k)$[cf. (4.4),(4.5),(4.8)] and (assuming all processors keep on computing and receiving measurements) eventually enter the set $X(k+1)$[cf. (4.6),(4.7)]. In view of these remarks the proof of the following result is rather easy. Note that the assumption does not differentiate the effects of two different members of the probability space [cf. part c)] so it applies to situations where the process is either deterministic (Ω consists of a single element), or else stochastic variations are not sufficiently pronounced to affect the membership relations in part c).

It is interesting to note that in the case where all functions M_{ij} are the identity, i.e. measurement reception does not prompt any update of the processor's estimate, the conditions (4.4)-(4.8) can be replaced by the simpler conditions

$$X(k+1) \subset \overline{X}^i(k) \subset X(k)$$

$$C_i(x^i, z^i) \in X(k+1), \qquad \forall\, x^i \in \overline{X}^i(k),\ z^i \in \overline{Z}^i(k).$$

Proposition 3.1: Let Assumptions 2.1, 2.2, 3.1, hold and assume that all initial processor estimates $x^i(0)$, $i=1,\ldots,n$ belong to $X(0)$, while all initial measurements $z^i_j(0)$ available at the processors belong to the corresponding sets $Z^i_j(0)$. Then every limit point of the sequences $\{x^i(t)\}$ is almost surely a solution.

The proof will not be given since it is very similar to the one given in [13]. Note that the proposition does not guarantee asymptotic agreement of the processor estimates but in situations where Assumption 3.1 is satisfied one can typically also show agreement.

Example 2 (continued): As an illustration consider the specialized process for computing a fixed point of a mapping F in example 2. There X is a Cartesian product $X_1 \times X_2 \times \ldots \times X_n$, and each processor i is responsible for updating the

ith "coordinate" x_i of $x = (x_1, x_2, \ldots, x_n)$ while relying on essentially direct communications from other processors to obtain estimates of the other coordinates. Suppose that each set X_i is a Banach space with norm $||\cdot||_i$ and X is endowed with the sup norm

$$||x|| = \max\{||x_1||_1, \ldots, ||x_n||_n\}, \qquad \forall x \in X \tag{4.9}$$

Assume further that F is a contraction mapping with respect to this norm, i.e., for some $\alpha \in (0,1)$

$$||F(x)-F(y)|| \leq \alpha ||x-y||, \qquad \forall x,y \in X . \tag{4.10}$$

Then the solution set consists of the unique fixed point x^* of F. For some positive constant B let us consider the sequence of sets

$$X(k) = \{x \in X \mid ||x-x^*|| \leq B\alpha^k\} , \qquad k=0,1,\ldots,$$

The sets defined by (4.1)-(4.3) are then given by

$$Z_j^i(k) = \{x_j \in X_j \mid ||x_j - x_j^*||_j \leq B\alpha^k\}$$

$$\overline{X}^i(k) = \{x \in X(k) \mid ||x_i - x_i^*||_i \leq B\alpha^{k+1}\}$$

$$\overline{Z}_j^i(k) = \{x_j \in X_j \mid ||x_j - x_j^*||_j \leq B\alpha^{k+1}\} .$$

It is straightforward to show that the sequence $\{X(k)\}$ satisfies Assumption 3.1. Further illustrations related to this example are given in [13]. Note however that the use of the sup norm (4.10) is essential for the verification of Assumption 3.

Similarly Assumption 3 can be verified in the preceding example if the contraction assumption is substituted by a monotonicity assumption (see [13]). This monotonicity assumption is satisfied by most of the dynamic programming problems of interest including the shortest path problem of example 1 (see also [12]). An important exception is the infinite horizon average cost Markovian decision problem (see [12], p. 616).

An important special case for which the contraction mapping assumption (4.9) is satisfied arises when $X=R^n$ and $x_1,x_2,\ldots,x_n$ are the coordinates of x. Suppose that F satisfies

$$|F(x)-F(y)| \leq P|x-y|, \qquad \forall x,y \in R^n$$

where P is an nxn matrix with nonnegative elements and spectral radius strictly less than unity, and for any $z=(z_1,z_2,\ldots,z_n)$ we denote by $|z|$ the column vector with coordinates $|z_1|,|z_2|,\ldots,|z_n|$. Then F is called a P-contraction mapping. Fixed point problems involving such mappings arise in dynamic programming ([20], p.374), and solution of systems of nonlinear equations ([17], Section 13.1). It can been shown ([11], p.231) that if F is a P-contraction then it is a contraction mapping with respect to some norm of the form (4.9). Therefore Proposition 3.1 applies.

We finally note that it is possible to use Proposition 3.1 to show convergence of similar fixed point distributed processes involving partial or total overlaps between the processors (compare with example 6).

Example 3 (continued): Consider the special case of the deterministic gradient algorithm of example 3 corresponding to the mapping

$$F(x) = x - \alpha \nabla f(x) \ . \tag{4.11}$$

Assume that $f:R^n \to R$ is a twice continuously differentiable convex function with Hessian matrix $\nabla^2 f(x)$ which is positive definite for all x. Assume also that there exists a unique minimizing point x^* of f over R^n. Consider the matrix

$$H^* = \begin{bmatrix} \left|\frac{\partial^2 f}{(\partial x_1)^2}\right| & -\left|\frac{\partial^2 f}{\partial x_1 \partial x_2}\right| & \cdots & -\left|\frac{\partial^2 f}{\partial x_1 \partial x_n}\right| \\ \vdots & \vdots & & \vdots \\ \left|\frac{\partial^2 f}{\partial x_n \partial x_1}\right| & -\left|\frac{\partial^2 f}{\partial x_n \partial x_2}\right| & \cdots & \left|\frac{\partial^2 f}{(\partial x_n)^2}\right| \end{bmatrix} \tag{4.12}$$

obtained from the Hessian matrix $\nabla^2 f(x^*)$ by replacing the off-diagonal terms by their negative absolute values. It is shown in [13] that if the matrix H^* is positive definite then the mapping F of (4.11) is a P-contraction within some open sphere centered at x^* provided the stepsize α in (4.11) is sufficiently small. Under these circumstances the distributed asynchronous gradient method of this example is convergent to x^* provided all initial processor estimates are sufficiently close to x^* and the stepsize α is sufficiently small. The neighborhood of local convergence will be larger if the matrix (4.12) is positive definite within an accordingly larger neighborhood of x^*. For example if f is positive definite quadratic with the corresponding matrix (4.12) positive definite a global convergence result can be shown.

One condition that guarantees that H^* is positive definite is strict diagonal dominance ([17], p.48-51).

$$\frac{\partial^2 f}{(\partial x_i)^2} > \sum_{\substack{j=1 \\ j\neq i}}^{n} \left| \frac{\partial^2 f}{\partial x_i \partial x_j} \right| , \qquad \forall\, i=1,\ldots,n,$$

where the derivatives above are evaluated at x^*. This type of condition is typically associated with situations where the coordinates of x are weakly coupled in the sense that changes in one coordinate have small effects on the first partial derivatives of f with respect to the other coordinates. This result can be generalized to the case of weakly coupled systems (as opposed to weakly coupled coordinates). Assume that x is partitioned as $x=(x_1,x_2,\ldots,x_n)$ where now $x_i \in R^{m_i}$ (m_i may be greater than one but all other assumptions made earlier regarding f are in effect). Assume that there are n processors and the ith processor asynchronously updates the subvector x_i according to an approximate form of Newton's method where the second

order submatrices of the Hessian $\nabla^2_{x_i x_j} f$, $i \neq j$ are neglected, i.e.

$$x_i \leftarrow x_i - (\nabla^2_{x_i x_i} f)^{-1} \nabla_{x_i} f \ . \tag{4.13}$$

In calculating the partial derivatives above processor i uses the values x_j latest communicated from the other processors $j \neq i$ similarly as in the distributed gradient method. It can be shown that if the cross-Hessians $\nabla^2_{x_i x_j} f$, $i \neq j$ have sufficiently small norm relative to $\nabla^2_{x_i x_i} f$, then the totally asynchronous version of the approximate Newton method (4.13) converges to x^* if all initial processor estimates are sufficiently close to x^*. The same type of result may also be shown if (4.13) is replaced by

$$x_i \leftarrow \arg \min_{x_i \in R^{m_i}} f(x_1, x_2, \ldots, x_n) \ . \tag{4.14}$$

Unfortunately it is not true always that the matrix (4.12) is positive definite, and there are problems where the totally asynchronous version of the distributed gradient method is not guaranteed to converge regardless of how small the stepsize α is chosen. As an example consider the function $f: R^3 \rightarrow R$

$$f(x_1, x_2, x_3) = (x_1 + x_2 + x_3)^2 + (x_1 + x_2 + x_3 - 3)^2 + \varepsilon(x_1^2 + x_2^2 + x_3^2)$$

where $0 < \varepsilon << 1$. The optimal solution is close to $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ for ε: small. The scalar ε plays no essential role in this example. It is introduced merely for the purpose of making the Hessian of f positive definite. Assume that all initial processor estimates are equal to some common value $\bar{x}$, and that processors execute many gradient

iterations with a small stepsize before communicating the current values of their respective coordinates to other processors. Then (neglecting the terms that depend on ε) the ith processor tries in effect to solve the problem

$$\min_{x_i} \{(x_i+2\bar{x})^2 + (x_i+2\bar{x}-3)^2\}$$

thereby obtaining a value close to $\frac{3}{2} - 2\bar{x}$. After the processor estimates of the respective coordinates are exchanged each processor coordinate will have been updated approximately according to

$$\bar{x} \leftarrow \frac{3}{2} - 2\bar{x} \tag{4.15}$$

and the process will be repeated. Since (4.15) is a divergent iterative process we see that, regardless of the stepsize chosen and the proximity of the initial processor estimates to the optimal solution, by choosing the delays between successive communications sufficiently large the distributed gradient method can be made to diverge when the matrix H^* of (4.12) is not positive definite.

5. Convergence of Descent Processes

We saw in the last section that the distributed gradient algorithm converges appropriately when the matrix (4.12) is positive definite. This assumption is not always satisfied, but convergence can be still shown (for a far wider class of algorithms) if a few additional conditions are imposed on the frequency of obtaining measurements and on the magnitude of the delays in equation (2.3). The main idea behind the results described in this section is that if delays are not too large, if certain processors do not obtain measurements and do not update much more frequently than others, then the effects of asynchronism are relatively small and the algorithm behaves approximately as a centralized algorithm, similar to the class of centralized pseudo-gradient algorithms considered in [40].

Let $X = X_1 x X_2 x \ldots x X_L$ be the feasible set, where X_ℓ ($\ell=1,\ldots,L$) is a Banach space. If $x=(x_1,\ldots,x_L)$, $x_\ell \in X_\ell$, we refer to x_ℓ as the ℓ-th component of x. We endow X with the sup norm, as in (4.8). Let $f:x \to [0,\infty)$ be a cost function to be minimized. We assume that f is Frechet differentiable and its derivative is Lipschitz continuous.

Each processor i keeps in his memory an estimate $x^i(t) = (x_1^i(t),\ldots,x_L^i(t)) \in X$ and receives measurements $z_{j,\ell}^i \in X_\ell$, $i \neq j$, with the value of the ℓ-th component of x^j, evaluated by processor j at some earlier time $\tau_{j,\ell}^i(t) \leq t$; that is, $z_{j,\ell}^i(t) = x_\ell^j(\tau_{j,\ell}^i(t))$. He also receives from the environment exogenous, possibly stochastic measurements $z_i^i \in X$, which are in a direction of descent with respect to the cost function f, in a sense to be made precise later. We denote by $z_{i,\ell}^i$ the ℓ-th component of z_i^i.

Whenever processor i receives measurements $z_{j,\ell}^i$, he updates his estimate vector x^i componentwise, according to:

$$x_\ell^i(t+1) = \beta_{i,\ell}^i(t)x_\ell^i(t) + \sum_{j\neq i} \beta_{j,\ell}^i(t)z_{j,\ell}^i(t) + \alpha^i(t)z_{i,\ell}^i(t) \ . \qquad (5.1)$$

The coefficients $\beta_{j,\ell}^i(t)$ are nonnegative scalars satisfying

$$\sum_{j=1}^{n} \beta_{j,\ell}^i(t)=1, \qquad \forall i,\ell,t,$$

and such that: if no measurement $z_{j,\ell}^i$ was received by processor i ($i\neq j$) at time t, then $\beta_{j,\ell}^i(t)=0$. That is, processor i combines his estimate of the ℓ-th component of the solution with the estimates (possibly outdated) of other processors that he has just received, by forming a convex combination. Also, if no new measurement z_i^i was obtained at time t, we should set $z_{i,\ell}^i(t)=0$ in equation (5.1). The coefficient

$\alpha^i(t)$ is a nonnegative stepsize. It can be either independent of t or it may depend on the number of times up to t that a new measurement (of any type) was received at processor i.

Equation (5.1) which essentially defines the algorithm, is a linear system driven by the exogenous measurements $z_i^i(t)$. Therefore, there exist linear operators $\Phi^{ij}(t|s),(t\geq s)$, such that

$$x^i(t) = \sum_{j=1}^{n} \Phi^{ij}(t|0)x^j(1) + \sum_{s=1}^{t-1} \sum_{j=1}^{n} \alpha^j(s)\Phi^{ij}(t|s)z_j^j(s) \ .$$

We now impose an assumption which states that if the processors cease obtaining exogenous measurements from some time on (that is, if they set $z_i^i=0$), they will asymptotically agree on a common limit (this common limit is the vector y(t) to be defined shortly):

Assumption 5.1: For any i,j,s, $\lim_{t\to\infty} \Phi^{ij}(t|s)$ exists (with respect to the induced operator norm) and is the same for all i. The common limit is denoted by $\Phi^j(s)$.

Assumption 5.1 is very weak. Roughly speaking it requires that for every component $\ell\in\{1,\ldots,L\}$ there exists a directed graph $G=(N,A)$, where the set N of nodes is the set $\{1,\ldots,n\}$ of processors, and such that there exists a path from every processor to every other processor and such that if $(i,j)\in A$ then processor j receives an infinite number of measurements (messages) of type $z_{i,\ell}^j$. Also the coefficients $\beta_{i,\ell}^j(t)$ must be such that "sufficient combining" takes place and the processors tend to agree.

We can now define a vector $y(t)\in X$ by

$$y(t) = \sum_{j=1}^{n} \Phi^j(0)x^j(1) + \sum_{s=1}^{t-1} \sum_{j=1}^{n} \alpha^j(s)\Phi^j(s)z_j^j(s)$$

and observe that $y(t)$ is recursively generated by

$$y(t+1) = y(t) + \sum_{i=1}^{n} \alpha^i(t)\Phi^i(t)z_i^i(t) \ . \tag{5.2}$$

We can now explain the main idea behind the results to be described: if $\Phi^{ij}(t|s)$ converges to $\Phi^j(s)$ fast enough, if $\alpha^i(t)$ is small enough, and if $z_i^i(t)$ is not too large, then $x^i(t)$, for each i, will evolve approximately as $y(t)$. We may then study the behavior of the recursion (5.2) and make inferences about the behavior of $x^i(t)$.

The above framework covers both specialized processes, in which case we have L=n, as well as the case of total overlap where we have L=1 and we do not distinguish between components of the estimates. For specialized processes (e.g. example 3) it is easy to see that $y(t) = (x_1^1(t), x_2^2(t), \ldots, x_n^n(t))$.

We now proceed to present some general convergence results. We allow the exogenous measurements z_i^i of each processor, as well as the initialization $x^i(1)$ of the algorithm to be random (with finite variance). We assume that they are all defined on a probability space (Ω, F, P) and we denote by F_t the σ-algebra generated by $\{x^i(1), z_i^i(s); i=1,\ldots,n; s=1,\ldots,t-1\}$. For simplicity we also assume that the sequence of times at which measurements are obtained, computations are performed, the times $\tau_{j,\ell}^i(t)$, as well as the combining coefficients $\beta_{j,\ell}^i(t)$ are deterministic. (In fact, this assumption may be relaxed [35], [43]). In order to quantify the speed of convergence of $\Phi^{ij}(t|s)$ we introduce

$$c(t|s) = \max_{i,j} \| \Phi^{ij}(t|s) - \Phi^j(s) \| .$$

By Assumption 5.1 $\lim_{t\to\infty} c(t|s)=0$ and it may be shown that $c(t|s) \leq 1$, $\forall t,s$. Consider the following assumptions:

Assumption 5.2:

$$E\left[\left\langle \frac{\partial f}{\partial x}\,(x^i(t)),\ \Phi^i(t)z_i^i(t)\right\rangle \,|F_t\right] \leq 0, \qquad \forall\, t,i, \quad \text{a.s.}$$

Assumption 5.3:

a) For some $K_o \geq 0$

$$E\left[\left|\left|z_i^i(t)\right|\right|^2\right] \leq -\,K_o E\left[\left\langle \frac{\partial f}{\partial x}\,(x^i(t)), \Phi^i(t)z_i^i(t)\right\rangle\right], \qquad \forall\, i,t\ .$$

b) For some $B\geq 0$, $d\in[0,1)$, $c(t|s)\leq Bd^{t-s}$, $\forall t\geq s$, $\forall s$.

Assumption 5.2 states that $\Phi^i(t)z_i^i(t)$ (which is the "effective update direction" of processor i, see (5.2)) is a descent direction with respect to f. Assumption 5.3a requires that $z_i^i(t)$ is not too large. In particular any noise present in $z_i^i(t)$ can only be "multiplicative-like": its variance must decrease to zero as a stationary point of f is approached. For example, we may have

$$z_i^i(t) = -\left[\frac{\partial f}{\partial x}\,(x^i(t))\right](1+w^i(t)),$$

where $w^i(t)$ is scalar white noise. Finally, Assumption 5.3b requires that the processors tend to agree exponentially fast. Effectively, this requires that the time between consecutive measurements of the type $z_{j,\ell}^i$, $i\neq j$, as well as the delays $t-\tau_{j,\ell}^i(t)$ are bounded together with some minor restriction of the coefficients $\beta_{j,\ell}^i(t)$ for those times that a measurement of type $z_{j,\ell}^i$ is obtained.

Letting

$$\alpha_o = \sup_{t,i} \alpha^i(t) \ ,$$

we may use Assumptions 5.3a and 5.3b to show that $||x^i(t)-y(t)||$ is of the order of α_o. Using the Lipschitz continuity of $\frac{\partial f}{\partial x}$ it follows that $||\frac{\partial f}{\partial x}(y(t))-\frac{\partial f}{\partial x}(x^i(t))||$ is also of the order of α_o; then, using Assumption 5.2, it follows that (5.2) corresponds to a descent algorithm, up to first order in α_o. Choosing α_o small enough, convergence may be shown by invoking the supermartingale convergence theorem. The above argument can be made rigorous and yields the following proposition (the proof may be found in [35] and [43]):

Proposition 5.1: If Assumptions 5.1, 5.2, 5.3, hold and if α_o is small enough, then:

a) $f(x^i(t))$, $i=1,\ldots,n$, as well as $f(y(t))$ converge, almost surely, and to the same limit.

b) $\lim_{t\to\infty} (x^i(t)-y(t))=0$, $\forall i$, almost surely and in the mean square.

c) $$\sum_{t=1}^{\infty} \sum_{i=1}^{n} \alpha^i(t) E\left[< \frac{\partial f}{\partial x}(x^i(t)), \Phi^i(t) z_i^i(t) > | F_t \right] > -\infty \ , \qquad (5.3)$$

almost surely. The expectation of the above expression is also finite.

A related class of algorithms arises if the noise in $z_i^i(t)$ is allowed to be additive, e.g.

$$z_i^i(t) = -\frac{\partial f}{\partial x}(x^i(t)) + w^i(t) \ ,$$

where $w^i(t)$ is zero-mean white. In such a case, an algorithm may be convergent only if $\lim_{t\to\infty} \alpha^i(t)=0$. In fact, $\alpha^i(t)=1/t_i$, where t_i is the number of times up to time t that a new measurement was received at i, is the most convenient choice, and this is what we assume here. However, this choice of stepsize implies that the algorithm becomes progressively slower, as $t\to\infty$. We may therefore allow the agreement process to become progressively slower as well, and still retain convergence. In physical terms, the time between consecutive measurements $z^i_{j,\ell}(i\neq j)$ may increase to infinity, as $t\to\infty$. In mathematical terms:

Assumption 5.4: a) For some K_o, K_1, $K_2 \geq 0$,

$$E\left[\left|\left|z_i^i(t)\right|\right|^2\right] \leq - K_o E\left[\left\langle \frac{\partial f}{\partial x}(x^i(t)), \Phi^i(t) z_i^i(t)\right\rangle\right] + \\ + K_1 E\left[f(x^i(t))\right] + K_2$$

b) For some $B\geq 0$, $\delta\in(0,1]$, $d\in[0,1)$

$$c(t|s) \leq B d^{t^\delta - s^\delta}, \quad \forall t\geq s, \ \forall s.$$

We then have [35], [43]:

Proposition 5.2: Let $\alpha^i(t)=1/t_i$, where t_i is the number of times up to time t that a new measurement was received at i, and assume that for some $\varepsilon>0$, $t_i \geq \varepsilon\cdot t$ for all i,t. Asumme also that Assumptions 5.1, 5.2, 5.4 hold. Then the conclusions (a),(b),(c) of Proposition 5.1 remain valid.

Propositions 5.1,5.2 do not prove yet convergence to the optimum (suppose, for example, that $z_i^i(t)\equiv 0$, $\forall i,t$). However, (5.3) may be exploited to yield optimality under a few additional assumptions:

Corollary: Let the assumptions of either Proposition 5.1 or 5.2 hold. Let T^i be the set of times that processor i obtains a measurement of type z_i^i. Suppose that there exists some $B \geq 0$ and, for each i, a sequence $\{t_k^i\}$ of distinct elements of T^i such that

$$\max_{i,j} |t_k^i - t_k^j| \leq B \ , \tag{5.4}$$

$$\sum_{k=1}^{\infty} \min_i \{\alpha^i(t_k^i)\} = \infty \ .$$

Finally, assume that f has compact level sets and that there exist continuous functions $g^i: x \to [0,\infty)$ satisfying

a) $$E\left[\left\langle \frac{\partial f}{\partial x}(x^i(t)), \Phi^i(t) z_i^i(t) \right\rangle | F_t \right] \leq -g^i(x^i(t)), \quad \forall t \in T^i, \forall i, \text{ almost surely.}$$

b) $$\sum_{i=1}^{n} g^i(x^*) = 0 \Rightarrow x^* \in X^* \triangleq \{x \in X \mid f(x^*) = \inf_x f(x)\}$$

Then, $\lim_{t\to\infty} f(x^i(t)) = \inf_x f(x)$, $\forall i$, almost surely.

Example 3: (continued): It follows from the above results that the distributed deterministic gradient algorithm applied to a convex function converges provided that a) The stepsize α is small enough, b) Assumption 5.3(b) holds and c) The processors update, using (3.7), regularly enough, i.e. condition (5.4) is satisfied. Similarly, convergence for the distributed stochastic gradient algorithm follows if we choose a stepsize $\alpha^i(t) = 1/t_i$, if Assumption 5.4 and condition (5.4) hold.

Example 4: (continued) Similarly with the previous example, convergence to stationary points of f may be shown, provided that α_i is not too large, that the delays $t-\tau_j^{im}(t)$ are not too large and that the processors do not update too irregularly. It should be pointed out that a more refined set of sufficient conditions for convergence may be obtained, which links the "coupling constants" $K_{j,m}^i$ with bounds on the delays $t-\tau_j^{im}(t)$ [35]. These conditions effectively quantify the notion that the time between consecutive communications and communication delays between decision makers should be inversely proportional to the strength of coupling between their respective divisions.

Example 7: (continued) Several common algorithms for identification of a moving average process satisfy the conditional descent Assumption 5.2. (e.g. the Least Mean Squares algorithm, or its normalized version-NLMS). Consequently, Proposition 5.2 may be invoked. Using part (c) of the Proposition, assuming that the input is sufficiently rich and that enough messages are exchanged, it follows that the distributed algorithm will correctly identify the system. A detailed analysis is given in [35].

A similar approach may be taken to analyze distributed stochastic algorithms in which the noises are correlated and Assumption 5.2 fails to hold. Very few global convergence results are available even for centralized such algorithms [34,36] and it is an open question whether some distributed versions of them also converge. However, as in the centralized case one may associate an ordinary differential equation with such an algorithm (compare with [37,38]), and prove local convergence subject to an assumption that the algorithm returns infinitely often to a bounded region (see [35]). Such results may be used, for example, to demonstrate local convergence of a distributed extended least squares (ELS) algorithm, applied to the ARMAX identification problem in Example 7.