ALMOST SURE CONVERGENCE OF CONSENSUS ALGORITHMS
BY RELAXED PROJECTION MAPPINGS

Ondrej Slučiak and Markus Rupp
Institute of Telecommunications, Vienna University of Technology, Austria
{osluciak,mrupp}@tuwien.ac.at

ABSTRACT
In this contribution we present a stronger notion of almost sure convergence for a large class of consensus algorithms including also asynchronous updates. We introduce the concept of the so-called relaxed projection algorithms and show that many consensus algorithms can be interpreted as such relaxed projection updates. It is well known that such algorithms converge to a solution lying in the intersection of the projections. The convergence of such algorithms is, however, guaranteed only for deterministic ordering of the projections. Since we are interested in random data exchanges, we analyze the convergence in case of random orderings of the projections and show that the algorithms converge in the underrelaxed case even for time-varying and individual mixing parameters.

Index Terms — relaxed projection algorithm, average consensus, asynchronous updates, wireless sensor network

1. INTRODUCTION
Learning algorithms for wireless sensor networks (WSN) have attracted much attention over the past decade. While most literature on consensus algorithms in WSNs, e.g., [1–4], and thus, also the methods based on these algorithms, e.g., [5–7], assume a certain synchronism for the updates of their sensor nodes, few have also analyzed more realistic asynchronous updates [2,5]. In the more classical synchronous case, the modeling assumption is that at a certain time instant \( t_{k-1} \) all \( N \) nodes of the WSN are transmitting, so that a specific node \( i \) receives from all its neighbors in its neighborhood \( N_i \) new information

\[
x_i(t_k) = f(x_{N_i}(t_{k-1})), \text{ for } i = 1 \ldots N.
\]

The notation here indicates that the state \( x_i \) of node \( i \) is changed by the state information collected in a vector \( x_{N_i} \) from its neighborhood \( N_i \).

Realistic modelling of sensor networks assumes that the communication between the sensor nodes occurs asynchronously, mostly due to the fact that the sensors may wake up from their idle states only in need of operations [8]; there is no centralized clock; and possible retransmissions may delay the transmission of the more current data. Although, some synchronization may be possible [2], transmissions at the same time instance may lead to big inter-signal interference and channel congestion.

Therefore, it is reasonable to prefer asynchronous updates, with sensors transmitting independently at a certain time instant, say \( t_k \), its data to its neighbors, whoever may receive the information. This shifts the model from the perspective of the receiving node to the transmitting node.

The paper is structured as follows. In Sec. 2 we review relaxed projection mapping algorithm (RPMA), well known in SP literature. Extending the analysis of the projection algorithms from deterministic to random sequences, we show under which conditions the algorithm converges with probability one. In Sec. 3 we introduce three algorithms for reaching a consensus in a WSN. A relatively simple structured so-called state transition matrix of row-stochastic type is introduced as the key element. With some linear transformations, it is possible to describe the algorithms in a very general form, including even time-variant updates in the concept of the RPMA. Almost sure convergence results follow immediately. Simulation Sec. 4 concludes the paper.

2. CONCEPT OF PROJECTIONS
Projection algorithms have attracted a lot of attention recently [10,14]. Consider the so-called relaxed projection mapping algorithm (RPMA):

\[
x[k+1] = x[k] + \mu_k (P_k x[k] - x[k]),
\]

that, in general, iteratively maps a vector \( x_k \) onto a closed linear subspace \( P_k \) defined by the projection \( P_k \) with \( 0 < \mu_k < 2 \). Moreover, following the terminology of RPMA, if \( 0 < \mu_k < 1 \), we say that the projection is underrelaxed, and if \( 0 < \mu_k < 2 \), the projection is overrelaxed. It is well known that if a finite set of projections, say \( P_i, i = 1, 2 \ldots M \), with corresponding subspaces \( P_i \) is applied, typically in a fixed order, the algorithm converges to a solution lying in the intersection of all such subspaces [15–17], i.e.,

\[
\lim_{k \to \infty} x[k] = x^* \in P^* := \bigcap_i P_i.
\]

We are, however, interested in the behavior of the algorithm if the projections \( P_i \) are randomly chosen. To describe this, we will prove the following theorem.
Theorem 2.1 If the projections $P_i$ onto linear subspaces $\mathcal{P}_i$ are from a finite set and selected randomly each with probabilities $p_i > 0$, then the steady-state solution of the relaxed projection algorithm in (1) lies with probability one in the intersection of the subspaces $\cap_i \mathcal{P}_i$.

Proof: To revise the theory, it was shown by von Neumann [16] and by Prager [18] that if there is a set of $M = 2$ projections $P_i$ ($i = 1, 2$), the alternating problem of the projections converges strongly, i.e.,

$$\lim_{n \to \infty} (P_1 P_2)^n \to P^* := \{P^* x \in \bigcap_i \mathcal{P}_i : \forall x \in \mathcal{H}\}.$$ 

The proof was extended to a cyclic iteration of a finite number $M$ of projections by Halperin [17], i.e.,

$$\lim_{n \to \infty} (P_1 P_2 \ldots P_M)^n \to P^* := \{P^* x \in \bigcap_i \mathcal{P}_i : \forall x \in \mathcal{H}\}.$$ 

Amemiya and Ando [19] later showed that a sequence $x_n$ (in a real Hilbert space $\mathcal{H}$) defined by general iterations $x_n = P_n x_{n-1}$ converges weakly. Bauschke [20] provided a proof of strong convergence as long as all projections from a finite set are selected infinitely often times.

However, a proof of strong convergence for the case of randomly selected projections $P_i$ remains an open issue.3

Let us now randomly (independently) select the projection $P_i$ with probability $p_i > 0$ and construct $L$ sufficiently long chains $C^{(m)}$ of $N_m > M$ elements ($m = 1 \ldots L$), i.e.,

$$C^{(m)} = \prod_{k=1}^{N_m} P_{m_k}$$

with $m_k \in \{1, 2 \ldots M\}$. The probability that in the $m$-th chain of $N_m$ elements there is at least one specific projection $P_i$, $i = 1 \ldots M$, is given by

$$\mathbb{P}(\text{exists at least one } P_i) = 1 - (1 - p_i)^{N_m}.$$ 

The probability that all $M$ projections appear at least once is given by

$$\mathbb{P}(\text{exists at least one } P_i; \forall i = 1 \ldots M) = (1 - (1 - p_i)^{N_m})^M.$$ 

If the number $N_m$ grows, the probability that all projections occur at least once, tends to one, i.e.,

$$\lim_{N_m \to \infty} \mathbb{P}(\text{exists at least one } P_i; \forall i = 1 \ldots M) = 1.$$ 

Applying such chains many times, thus assuring (cf. 5) that the given projection appeared in the chain, we find that

$$\lim_{L \to \infty} \lim_{N_m \to \infty} \lim_{m \to \infty} \prod_{k=1}^{N_m} P_{m_k} \to P^* := \{P^* x \in \bigcap_i \mathcal{P}_i : \forall x \in \mathcal{H}\}.$$ 

Thus, with probability one (almost sure) the solution of the RPMA belongs to the intersection of the subspaces.

3For completeness, we note that a novel insight for proving the strong convergence in the particular case of orthogonal projections (not considered here) was proposed by Baillon and Bruck [21, 22].

3From now on, we abuse the notation and we drop time index $\{k\}$ in $S^{(A)}_{im}$ and all matrices dependent on $S^{(A)}_{ij}$.

3. APPLICATION ON VARIOUS CONSENSUS ALGORITHMS

We now discuss the consequences of Theorem 2.1 in the case of three simple consensus algorithms.

3.1. Asynchronous Approach

We will first consider asynchronous updates in which every time an information is received in the neighborhood of a transmitting node $i$,

$$x_N^i[k] = f(x_i[k - 1]),$$

where we changed the notation for time to a simple event counter $k$ as time itself is of no further importance, just the events count. On the example of the average consensus algorithm [1] we will show the mechanics of such network.

Take, for example, node $i$ that receives from its neighbor node $m$ a value $x_m$ and combines it with its internal state $x_i$, i.e.,

$$x_i[k] = \alpha_i x_i[k - 1] + (1 - \alpha_i) x_m[k - 1] = x_i[k - 1] + (1 - \alpha_i)(x_m[k - 1] - x_i[k - 1]),$$

where $\alpha_i$ is a so-called mixing parameter.

All other states remain unchanged by such operation. We can describe such state transmission by a matrix operation

$$x[k] = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & \ldots \\ \vdots & \ddots & 0 & \alpha_i \\ \vdots & \ddots & 0 & 1 \\ 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \alpha_i & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \ldots & 0 \end{pmatrix} x[k - 1]$$

$$= S^{(A)}_{im} x[k - 1].$$

The state transition matrix $S^{(A)}_{im} \in \mathbb{R}^{N \times N}$ has a specific form: essentially a unit matrix with ones on its diagonal, except $\{S^{(A)}_{im}\}_{i \neq i} = 1 - \alpha_i$, respectively. We can thus define a matrix of transitions at time $k$, i.e.,

$$S[k] \in \mathcal{S} = \left\{ S^{(A)}_{im} \right\},$$

where $\mathcal{S}$ is the set of all allowed transitions in the network. The number of possible transition matrices is equal to the number of links in the network $|\mathcal{E}|$.

Note that we assume individual values of $\alpha_i$ for each node and the values could even vary over time (events $\mathcal{E}$). Such variations on the updates clearly include link failures [3], for which $\alpha_i$ simply turns to one for a certain time, and quantization effects [23]. To each of the existing transition matrices we associate a probability of its occurrence $p_{ij}$ such that $\sum_i p_{ij} = 1$. As the transition matrices occur randomly, the typical approach to analyze such network is to study the convergence in the mean and mean square sense [6].
We now describe the learning process of the network in terms of the relaxed projection mapping \( \Pi \). For this we define a projection matrix \( T_{ij} \) associated to the transition matrix \( S_{ij} \), such that \( S_{ij} = I - (1 - \alpha_i)T_{ij} \). We now identify the projections \( P_k \) to be \( P_k = P_{ij} = I - T_{ij} \) and \( \mu_t = 1 - \alpha_i \).

Thus, \( x[k + 1] = x[k] + \mu_k (P_k(x[k]) - x[k]) \)
\( = x[k] + \mu_k ((I - T_{ij})x[k] - x[k]) = (I - \mu_k T_{ij})x[k] \)

The conditions of Theorem 2.3 are, thus, satisfied.

**Lemma 3.1** The asynchronous update algorithm \((\Pi)\) converges almost surely to a consensus, i.e., \( \lim_{k \to \infty} x[k] \to c1 \), if \( 0 < \alpha_i < 1 \), the probabilities of the established connections appear with \( p_{ij} > 0 \) and the intersection of all transition matrices contains only \( \text{span}\{1\} \).

**Conjecture 3.1** Under some circumstances and conditions on the probability distribution and the network topology, the algorithm \((\Pi)\) converges to a consensus also in the overrelaxed case, i.e. if \(-1 < \alpha_{min} \leq \alpha_i < 0\), as expected from the conditions on general relaxed projection mapping algorithm.

Note that, the Conjecture 3.1 is supported, for example, also by the convex weight selection proposed in [1], where \(\alpha_i\) must also hold. Therefore, also for \(-1 < \alpha_i < 0\), all elements of \(S_{ij}^{(S)}\) are in absolute value smaller than 1, and thus, the contraction property \((\Pi)\) holds.

3.2. Asynchronous Bidirectional Approach

If we allow simultaneous bidirectional connections between nodes (pair-wise), the learning is assumed to be more efficient. In this case the transition matrix takes on the form

\[
x[k] = (1 - \alpha_i)(I - T_{ij})x[k] = (I - \mu_k T_{ij})x[k] = S_{ij}^{(B)}x[k]
\]

Similarly to the case before, Sec. 3.1, we can define \(S_{ij}^{(B)} = I - (1 - \alpha_i)T_{ij}\), where \(T_{ij}\) describes a projection matrix. We further identify \(\mu_i = 2(1 - \alpha_i)\) and thus:

**Lemma 3.2** The asynchronous bidirectional update algorithm \((\Pi)\) converges almost surely to a consensus, i.e., \( \lim_{k \to \infty} x[k] \to c1 \), if \( 0 < \alpha_i < 1 \), the probabilities of the established connections appear with \( p_{ij} > 0 \) and the intersection of all transition matrices contains only \( \text{span}\{1\} \).

We note here, that since the matrices \(S_{ij}^{(B)}\) satisfy so-called conservation property \((\Pi)\), \(c = \frac{1}{N} \sum_i x_i(0)\). This means that the algorithm converges to the average of the initial values.

3.3. Synchronous Approach

We finally extend our analysis to the case in which each node interacts in one time instance with all is neighbor nodes and updates its weights simultaneously. Let us assume that node \(i\) has three neighbors \(m, n, l\). Its transition matrix would then exhibit the following shape:

\[
x[k] = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & \ddots & 0 & \cdots \\
0 & 0 & 1 & 0 & \cdots \\
0 & \cdots & 0 & 1 & 0 \\
0 & \cdots & 0 & 0 & 1
\end{bmatrix}
\]

for which we assume that \(\beta_{im}, \beta_{in}, \beta_{il} > 0\), and satisfying the condition on having the matrix \(S_{im}^{(S)}\) row-stochastic, the condition \(\sum_j \beta_{ij} = 1 - \alpha_i\) must also hold. Therefore, also for \(-1 < \alpha_i < 0\), all elements of \(S_{ij}^{(S)}\) are in absolute value smaller than 1, and thus, the contraction property \((\Pi)\) holds.

**Lemma 3.3** The synchronous update algorithm \((\Pi)\) converges almost surely to a consensus, i.e., \( \lim_{k \to \infty} x[k] \to c1 \), if \(-1 < \alpha_i < 1\), the probabilities of the established connections appear with \( p_{ij} > 0 \) and the intersection of all transition matrices contains only \( \text{span}\{1\} \).

4. SIMULATIONS

In this section, we show simulation results for the algorithms \((\Pi)\), \((\Pi)\), \((\Pi)\), as functions of the mixing parameter \(\alpha_i\). We assume a static WSN with random geometric topology, i.e., randomly placed sensors transmitting in some radius \(r\), of \(N = 40\) nodes. Transmissions are selected randomly with uniform distribution. The showed results are averaged variance of values across the nodes (disagreement \(\sigma^2\)) through 100 averaging cycles with different initializations.

We observe (Fig. 1(a)) that in some cases the algorithm \((\Pi)\) converges to a consensus also in overrelaxed case (cf., \(\alpha_i = -0.2\)), as we conjecture in Conjecture 3.1 which is based on the conditions from the RPMA. The conditions on the network topology and initial values for which this conjecture holds remain, however, an open issue. We also see (Fig. 1(b)), as expected in Lemma 3.2 that for \(\alpha_i < 0\) algorithm \((\Pi)\) diverges. Next, since the algorithm \((\Pi)\) satisfies the contraction property \((\Pi)\), even for \(\alpha_i < 0\), for all \(S_{ij}^{(S)}\), convergence is guaranteed for all \(-1 < \alpha_i < 1\) (See Fig. 1(c)), which is also in agreement with conditions from the RPMA \((0 < \mu_k < 2)\). Furthermore, since the algorithm \((\Pi)\) mixes the information from neighbors most rapidly (in one iteration with all neighbors), the convergence rate is the highest in comparison to the other algorithms.

Note that the error floors at each algorithm is caused only by the numerical computation accuracy.
5. CONCLUSION

In this paper we first proved the almost sure convergence of general random projection mapping algorithm onto which we mapped three distributed linear consensus algorithms. By formulating the consensus algorithms in terms of the relaxed projection algorithms, we were able to provide convergence conditions in terms of the mixing parameter $\alpha_i$. In case of overrelaxed projections, we conjectured that the consensus algorithms may still converge, as the theory of relaxed projections implies. This conjecture is supported also by the simulation results.

6. ACKNOWLEDGMENTS

The authors would like to thank Prof. S. Theodoridis for helpful comments and suggestions.

7. REFERENCES


