

NONLINEAR AVERAGE CONSENSUS BASED ON WEIGHT MORPHING

Valentin Schwarz and Gerald Matz

Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology
Gußhausstraße 25/389, A-1040 Vienna, Austria; email: {vschwarz,gmatz}@nt.tuwien.ac.at

ABSTRACT

We present a novel weight design for average consensus that improves its transient and steady-state performance. The idea is to blend Metropolis-Hastings weights and convex-optimization weights via signal-adaptive morphing coefficients. The resulting scheme is shown to be particularly useful in dynamic scenarios where the measurements feature abrupt changes or unknown noise levels.

Index Terms— Average consensus, wireless sensor networks, distributed inference

1. INTRODUCTION

Distributed computing, in particular distributed inference, has developed into a very attractive research area during the last decade, due to the numerous applications, e.g., in tele-medicine, environmental monitoring, and military surveillance (cf. [1]). We here deal with the problem of distributed averaging based on average consensus (AC) which has its origin in the thesis of Tsitsiklis [2]. Since then, numerous extensions and modifications of the AC method have been proposed (see [3] for an overview). In this paper, we consider discrete-time AC because we are interested in applications of AC in wireless sensor networks (WSN) where the communication between sensors requires temporal discretization (sampling). For simplicity we will restrict to the synchronous case even though asynchronous AC (see e.g. [4] and [5]) has certain advantages in the context of WSN.

An important practical problem with AC is the choice of the weights (we will consider this in the next sections in more detail). In [6], several weight designs have been proposed with the goal of fast convergence. A different method to improve convergence was proposed in [7] based on local state prediction in the network nodes. An alternative to AC for distributed averaging is provided by consensus propagation [8–10]. Consensus propagation is based on a completely different approach and cannot be transformed into the form of AC, even though it is sometimes incorrectly described as a dynamic weight adaptation method for AC.

An extension of continuous-time AC to time-varying (dynamic) scenarios was introduced in [11]. A discrete-time version of dynamic AC based on first-order differences was proposed by Zhu and Martínez [12] along with an upper bound on the steady state error. This paper also introduces another variant which uses higher-order differences in order to improve the averaging performance at the price of increased communication overhead.

In this paper, we propose a novel nonlinear method for designing the AC weights; our scheme is motivated by nonlinear AC introduced in [13]. Our method involves a local linear combination of differently designed weights where the coefficients in the linear combination depend nonlinearly on the previous states/measurements,

Funded by WWTF Grant ICT-044 and FWF Grant S10606. The authors thank Raman Jafroudi for his valuable input.

thereby rendering the whole design nonlinear. We develop this approach both for the static and the dynamic case and we demonstrate by means of numerical simulations that our method outperforms existing weight designs in terms of convergence speed and residual error.

2. PRELIMINARIES

We consider a WSN where nodes/sensors are distributed randomly. Each node can exchange messages with other nodes that lie within a prescribed communication range. With the assumption of bidirectional communication, such a WSN can be modeled by an undirected random geometric graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where \mathcal{V} denotes the set of nodes ($I = |\mathcal{V}|$ is the number of nodes) and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ the set of undirected edges. Node i measures the signal $s_i[n]$ (n denotes discrete time). The goal of distributed averaging is to estimate the (time-varying) mean $\bar{s}[n] = \frac{1}{I} \sum_{i=1}^I s_i[n]$ in a non-centralized fashion. For the static case the measurements are time-invariant and we omit the variable n .

In static scenarios, AC is applied to the sensor measurements s_i , $i = 1, \dots, I$, resulting in a length- I state vector $\mathbf{x}[k]$ that serves as estimate of the true mean $\bar{\mathbf{s}} = \bar{s}\mathbf{1}$; here, k denotes the iteration index, $\bar{\mathbf{s}} = \frac{1}{I} \sum_{i=1}^I s_i$, and $\mathbf{1}$ is the all-ones vector. Static linear AC works by choosing the initial states equal to the measurements, i.e., $\mathbf{x}[0] = \mathbf{s} = (s_1 \dots s_I)^T$, and subsequently exchanging and averaging the states of neighboring nodes in an iterative manner, i.e.,

$$x_i[k+1] = w_{ii} x_i[k] + \sum_{j:(i,j) \in \mathcal{E}} w_{ij} x_j[k],$$

where w_{ij} denotes suitably chosen weights. This procedure can be compactly characterized by the state update

$$\mathbf{x}[k+1] = \mathbf{W}\mathbf{x}[k], \quad (1)$$

where the weight matrix is defined by $[\mathbf{W}]_{ij} = w_{ij}$. Note that $w_{ij} = 0$ unless $(i, j) \in \mathcal{E}$ or $i = j$. Since we consider undirected graphs, the weight matrix is assumed to be symmetric. It is known that in this case $\mathbf{x}[k]$ converges asymptotically to $\bar{\mathbf{s}}$ if the weight matrix satisfies the conditions $\mathbf{W}\mathbf{1} = \mathbf{1}$ and $\rho(\mathbf{W} - \frac{1}{I}\mathbf{1}\mathbf{1}^T) < 1$, where $\rho(\cdot)$ denotes the spectral radius.

Several different methods to design the weight matrix \mathbf{W} have been proposed. An overview of methods to optimize the asymptotic convergence speed is given in [6]. Here, we use the weight matrix \mathbf{W}^{CVX} that provides the fastest asymptotic convergence for undirected graphs; this weight matrix can be efficiently obtained by solving a convex optimization problem. A drawback of this weight design (termed CVX in what follows) is the requirement for global knowledge of the network structure. An alternative to convex optimization weight design that is extremely simple and requires only local information about the network structure is based on the

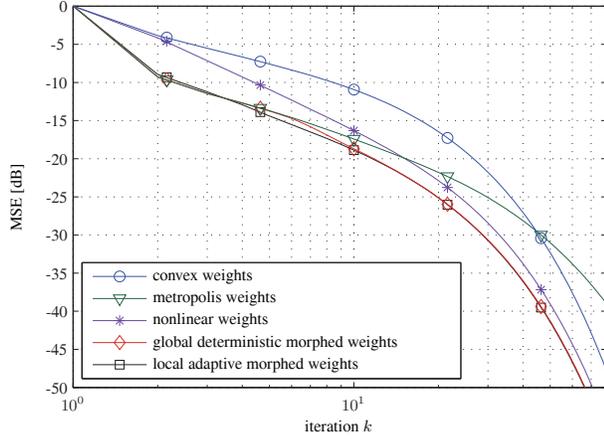


Fig. 1. Illustration of AC behavior for different weight designs in terms of mean-square error versus iteration index k .

Metropolis-Hastings (MH) algorithm coming from Markov chain theory. The MH weight matrix \mathbf{W}^{MH} is designed according to

$$w_{ij}^{\text{MH}} = \begin{cases} \frac{1}{\max\{d_i, d_j\}}, & \text{for } (i, j) \in \mathcal{E}, \\ 0, & \text{for } (i, j) \notin \mathcal{E}, \\ 1 - \sum_{j \neq i} w_{ij}^{\text{MH}}, & \text{for } i = j. \end{cases}$$

Here, d_i denotes the degree (the number of neighbors) of node i .

Fig. 1 illustrates the convergence behavior of static AC with \mathbf{W}^{CVX} and with \mathbf{W}^{MH} (and with other weight designs described below) for the case of a WSN with $I = 100$ nodes; the measurements here were obtained by sampling from a Gaussian distribution $\mathcal{N}\{0, 1\}$. It is seen that the CVX weights provide excellent asymptotic convergence rate, but the MH weights are clearly superior in the transient phase (i.e., for $k < 40$). To improve the transient behavior of CVX, a nonlinear AC algorithm was proposed in [13]. With this method, the nonlinear (NL) state update is given by

$$\mathbf{x}[k+1] = \mathbf{W}^{\text{NL}}[k] \mathbf{x}[k], \quad (2)$$

where the weight $\mathbf{W}^{\text{NL}}[k]$ depends on the current states and hence renders the updates time-varying and nonlinear. More specifically, the edge weights are given by

$$w_{ij}^{\text{NL}}[k] = \begin{cases} w_{ij}^{\text{CVX}} f(x_i[k] - x_j[k]), & \text{for } i \neq j, \\ 1 - \sum_{j \neq i} w_{ij}^{\text{NL}}[k], & \text{for } i = j. \end{cases}$$

Here $f(\cdot)$ is a suitable nonlinear function, chosen as $f(u) = \frac{\tanh(\theta_1 u) \theta_2}{u}$ with $\theta_1 \theta_2 \leq 1$ in [13]. The behavior of this NL weight design is also shown in Fig. 1. It is seen that the transient performance of NL outperforms CVX (with appropriately chosen θ_1 and θ_2). However, NL is still inferior to MH in the initial phase; furthermore, the parameters θ_1 and θ_2 have to be manually adapted to the specific scenario.

3. PROPOSED SCHEME

3.1. Static Case

Motivated by the observation that MH performs best in the transient phase while CVX is superior in the asymptotic phase, we aim at

combining the advantages of MH and CVX to obtain a weight design that performs uniformly best for all k . To this end, we propose to morph MH into CVX via local convex combinations in which the coefficients depend nonlinearly on the current states. This results in a time-varying non-linear update as in (2), with the morphed weights given by

$$w_{ij}^{\text{M}}[k] = \begin{cases} (1 - \alpha_{ij}[k])w_{ij}^{\text{MH}} + \alpha_{ij}[k]w_{ij}^{\text{CVX}}, & \text{for } i \neq j, \\ 1 - \sum_{j \neq i} w_{ij}^{\text{M}}[k], & \text{for } i = j. \end{cases} \quad (3)$$

where $0 < \alpha_{ij}[k] < 1$. Note that the coefficients $\alpha_{ij}[k]$ are time-varying and different for each edge. MH and CVX are obtained as special cases of (3) with $\alpha_{ij}[k] = 0$ and $\alpha_{ij}[k] = 1$, respectively. It is straightforward to check that $\mathbf{W}^{\text{M}}\mathbf{1} = \mathbf{1}$, i.e., (3) preserves the true mean and satisfies the property that the mean is a fixed point. The condition $\rho(\mathbf{W}^{\text{M}}[k] - \frac{1}{I}\mathbf{1}\mathbf{1}^T) < 1$ may be violated in the transition phase between MH and CVX. However, if our scheme should start to diverge, the weight adaptation described below would switch back to MH and thereby bring the algorithm back on track.

Global deterministic morphing. A very simple and intuitive choice for the coefficients $\alpha_{ij}[k]$ amounts basically to using MH initially (i.e., for $k < k_0$, with k_0 a precomputed number of iterations), and afterwards switch to CVX. This means $\alpha_{ij}[k] = 0$ for $k < k_0$ and $\alpha_{ij}[k] = 1$ for $k \geq k_0$. In practice, it is advantageous to use a gradual transition rather than an abrupt change, i.e., $\alpha_{ij}[k] = g[k]$, where $g[k]$ is a sigmoid-type function with $0 \leq g[0]$ and $\lim_{k \rightarrow \infty} g[k] = 1$. Here, the coefficients for all edges are identical and (3) simplifies to $\mathbf{W}^{\text{M}}[k] = (1 - g[k])\mathbf{W}^{\text{MH}} + g[k]\mathbf{W}^{\text{CVX}}$. Since the weights do not depend on the current state, the AC update is time-varying but remains linear. Due to the triangle equality $\rho((1-a)\mathbf{A} + a\mathbf{B}) \leq |1-a|\rho(\mathbf{A}) + |a|\rho(\mathbf{B})$, it follows directly that

$$\rho\left(\mathbf{W}^{\text{M}}[k] - \frac{1}{I}\mathbf{1}\mathbf{1}^T\right) \leq 1. \quad (4)$$

The performance of the globally morphed weights with $g[k] = \frac{1}{1 + \exp(-2(k-6))}$ is also shown in Fig. 1. Clearly, this scheme outperforms both MH and CVX. While it is desirable that the transition between MH and CVX is optimized for the specific scenario at hand, the performance of the algorithm turns out to be rather insensitive to the precise shape of $g[k]$.

Local adaptive morphing. We next propose an alternative to the global morphing described above that chooses the coefficients $\alpha_{ij}[k]$ locally in an adaptive manner, i.e., depending on the current states. The basic idea is that if the states $x_i[k]$, $i = 1, \dots, I$, differ “strongly,” we are still in the transient phase and should rather use MH; in contrast, if all states are “close” to each other, we are in the asymptotic phase and should use CVX. To prevent additional communication between nodes, we propose to assess the current discrepancy of the states via the following two metrics that depend only on the states of two neighboring nodes:

$$m_{ij}^{(1)}[k] = \frac{(x_i[k] - x_j[k])^2}{2}, \quad (5)$$

$$m_{ij}^{(2)}[k] = \frac{(x_i[k] - x_j[k])^2}{x_i^2[k] + x_j^2[k]}. \quad (6)$$

The morphing coefficients are then chosen in a local and adaptive fashion as

$$\alpha_{ij}[k] = f(m_{ij}^{(l)}[k]), \quad l = 1, 2. \quad (7)$$

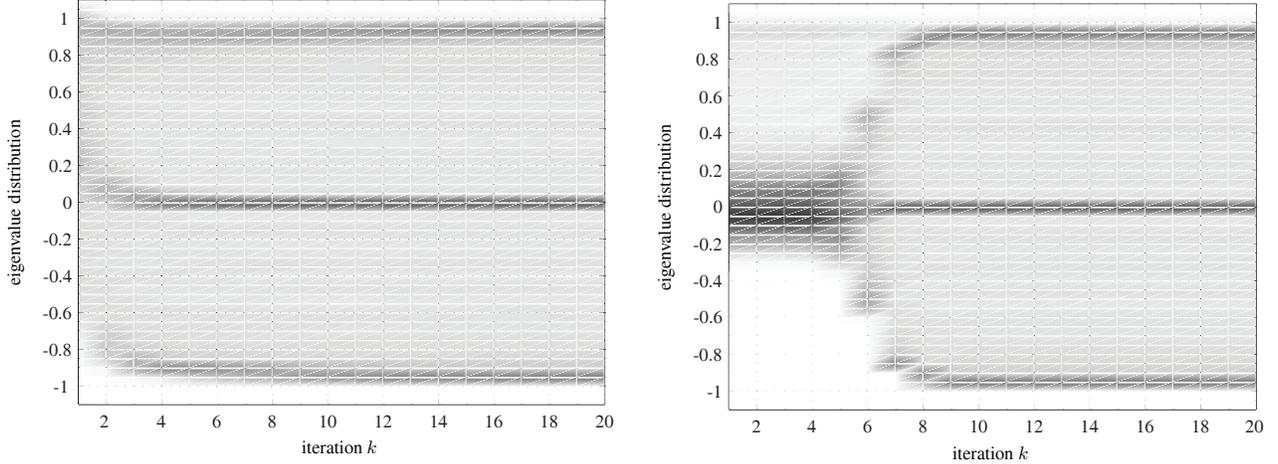


Fig. 2. Distribution of the eigenvalues of the weight matrix $\mathbf{W}[k]$ versus iteration index k : nonlinear AC [13] (left) and global deterministic morphing (right). In the right-hand side, the eigenvalue distribution for $k = 0$ and $k = 20$ corresponds to MH and CVX, respectively.

where $f(\cdot)$ is a sigmoid-like monotonic decreasing function satisfying $f(0) = 1$. Since the morphing coefficients and thus the AC weights depend on the current states, the overall AC update here becomes nonlinear. The metric $m_{ij}^{(1)}[k]$ equals the sample variance of the pair $\{x_i[k], x_j[k]\}$ and has the advantage of being translation invariant, i.e., an additive offset of both states leaves $m_{ij}^{(1)}[k]$ unchanged. In contrast, $m_{ij}^{(2)}[k]$ is scale invariant but not translation invariant, i.e., a multiplicative scaling of both states does not change $m_{ij}^{(2)}[k]$. A metric that is both translation invariant and scale invariant does not exist.

An illustration of the performance of AC with locally morphed weights based on the metric $m_{ij}^{(1)}[k]$ is given in Fig. 1. It is seen that this scheme performs best among all weight design even though the transition between MH and CVX is completely adaptive.

Interpretation. Fig. 1 reveals that our proposed method using morphed AC weights combines the good transient performance of MH and the optimal asymptotic performance of CVX. This excellent performance can be supported by Fig. 2, which shows the average eigenvalue distribution of the weight matrix for the case of nonlinear AC [13] and our globally morphed weight matrix. It is seen that—in contrast to nonlinear AC—our method initially has an eigenvalue distribution that is strongly concentrated about zero. This implies that the corresponding components of the state vector (i.e., which are orthogonal to $\mathbf{1}$) are quickly attenuated. The transition to CVX weights then minimizes the spectral radius, which in turn attenuates the remaining components as fast as possible.

3.2. Dynamic Case

In this section we show how to apply our weight morphing scheme to dynamic AC. In this scenario each sensor obtains a time-varying measurement $s_i[n]$, $i = 1, \dots, I$. The discrete-time dynamic AC proposed in [12] offers the potential to track the mean $\bar{s}[n]$ of the measurements when the latter vary so fast that performing numerous AC iterations per sampling interval becomes infeasible. With dynamic AC, the update (1) is modified by incorporating the first order difference $(\Delta \mathbf{s})[n] = \mathbf{s}[n] - \mathbf{s}[n-1]$ and the resulting modified update is performed once per sampling interval, i.e.,

$$\mathbf{x}[n+1] = \mathbf{W}\mathbf{x}[n] + (\Delta \mathbf{s})[n]. \quad (8)$$

Note that the term $(\Delta s_i)[n]$ basically serves as a simple means to predict the measurement at time $n+1$. For the case of temporally constant measurements, $(\Delta \mathbf{s})[n] = \mathbf{0}$ and (8) reduces to the static AC (1). For more details regarding dynamic AC we refer to [12]. An extension of this algorithm uses higher-order differences $(\Delta^m s_i)[n]$ but entails an increased communication overhead between the sensors. The optimal order m depends on the amount of time-variation in the measurements.

The conditions imposed on the weight matrix in the static case (see Section 2) ensure

$$\mathbf{1}^T \mathbf{x}[n] = \mathbf{1}^T \mathbf{s}[n],$$

which in turn implies that dynamic AC achieves an asymptotically vanishing error provided that $(\Delta s_i)[n]$ is independent of i (cf. [12]). Unfortunately, the latter condition is not satisfied in the case of noisy measurements and consequently the mean-square error (MSE) achieved with dynamic AC is rather high.

We propose to cope with the noise and fast time-variations of the measurements by applying our weight morphing method to dynamic AC, i.e., by replacing \mathbf{W} in (8) with the time-varying morphed weights in (3):

$$\mathbf{x}[n+1] = \mathbf{W}^M[n] \mathbf{x}[n] + (\Delta \mathbf{s})[n]. \quad (9)$$

For the morphing coefficients, we use local adaptation according to (7). The resulting AC algorithm tends to prefer MH weights for fast varying signal portions and low SNR while for slowly varying signal parts and high SNR the CVX weights will be used.

4. NUMERICAL RESULTS

In our simulations, we considered a WSN with $I = 100$ sensors randomly deployed within the unit square. The communication range was $r = \frac{\lambda}{\sqrt{I}}$ with $\lambda = 2$ unless stated otherwise. The performance of the various dynamic AC algorithms is assessed by the MSE

$$\epsilon^2[n] = \frac{\sum_{i=1}^I (x_i[n] - \bar{s}[n])^2}{\mathbb{E}\{s_i^2[n]\}},$$

possibly averaged over time.

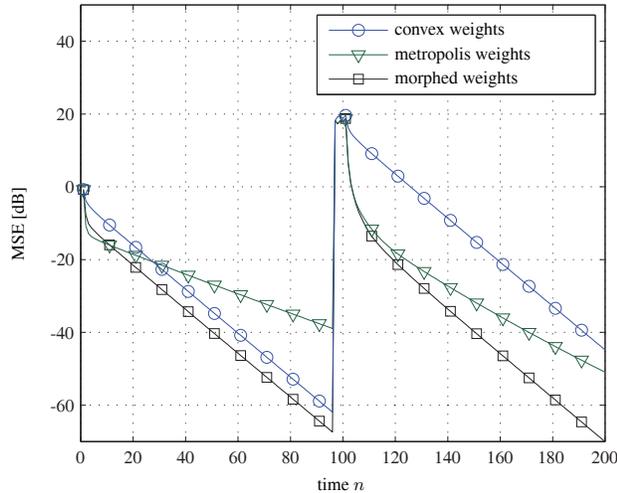


Fig. 3. MSE versus time for dynamic AC with different weight designs and an abrupt change in the measurements.

We first consider a pseudo-static scenario in which the measurements are constant for some time and then change abruptly to different values that again remain constant afterwards. Fig. 3 shows the MSE $\epsilon^2[n]$ achieved in this scenario with MH, CVX, and local adaptive weight morphing (based on $m_{ij}^{(1)}[n]$). It is seen that our proposed scheme outperforms MH and CVX and is able to cope both with the transient and the quasi-stationary parts of the measurements. In particular, while CVX weights are used after $n = 30$, the adaptation underlying the morphing coefficients recognizes the abrupt change at $n = 95$ and switches back from CVX to MH.

We next examine a scenario in which the sensor measurements are noisy samples from a spatio-temporal field which is constructed via a spatial Fourier series in which the coefficients are low-pass processes with normalized bandwidth $\theta_c = 6 \cdot 10^{-4}$. Fig. 4 displays the (time-averaged) MSE versus the SNR (averaged over 100 scenarios) for MH, CVX, and local adaptive weight morphing (based on $m_{ij}^{(2)}[n]$). Again our proposed scheme overall performs best by combining in a non-supervised manner the advantages of MH at low SNR and of CVX at high SNR. Hence, in the context of distributed averaging in unknown noise levels, our weight morphing scheme picks the appropriate weights in an automated manner.

5. CONCLUSION

The performance of AC algorithms for distributed averaging depends strongly on the weight design. This motivated us to propose an adaptive weight morphing scheme which leads to a nonlinear AC scheme and combines the favorable transient behavior of MH weight design with the excellent asymptotic performance of CVX weights. Morphed weights are advantageous also in the dynamic case to deal with different noise levels and signal variations. Numerical simulations confirmed that the AC algorithm with morphed weights leads to the best convergence and tracking behavior both in static and dynamic scenarios.

REFERENCES

[1] I. F. Akyildiz, *Wireless Sensor Networks*, Wiley & Sons, 2010.

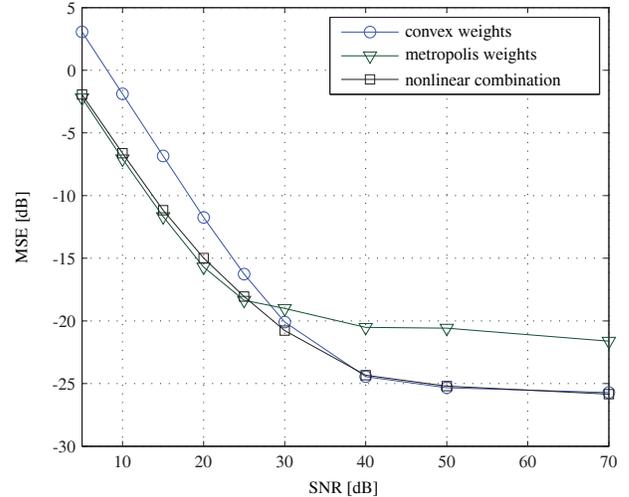


Fig. 4. Influence of the SNR on the MSE achieved with dynamic AC using different weight designs.

- [2] J.N. Tsitsiklis, *Problems in Decentralized Decision making and Computation*, Ph.D. thesis, Massachusetts Institute of Technology, Dec. 1984.
- [3] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
- [4] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*, Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [5] M. Mehyar, D. Spanos, J. Pongsajapan, S.H. Low, and R.M. Murray, "Asynchronous distributed averaging on communication networks," *IEEE/ACM Trans. on Networking*, vol. 15, no. 3, pp. 512–520, June 2007.
- [6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [7] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, "Accelerated distributed average consensus via localized node state prediction," *IEEE Trans. Signal Processing*, vol. 57, pp. 1563–1576, April 2009.
- [8] C. C. Moallemi, *A Message-Passing Paradigm For Optimization*, Ph.D. thesis, Stanford University, Sept. 2007.
- [9] C. C. Moallemi and B. Van Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.
- [10] V. Schwarz and G. Matz, "System-theoretic formulation and analysis of dynamic consensus propagation," in *Proc. IEEE ICASSP-2011*, Prague, CZ, May 2011, pp. 3020–3023.
- [11] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *IFAC world congress*, 2005.
- [12] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, issue 2, pp. 322–329, Feb. 2009.
- [13] L. Georgopoulos and M. Hasler, "Nonlinear average consensus," in *Proceedings of the 2009 International Symposium on Nonlinear Theory and its Applications*, Oct. 2009, p. 10.