Fast Clock Synchronization in Wireless Sensor Networks via ADMM-based Consensus

Davide Zennaro*, Emiliano Dall’Anese†, Tomaso Erseghe*, and Lorenzo Vangelista*

*Department of Information Engineering, University of Padova, via G. Gradenigo 6/b, 35131 Padova, Italy
†Department of Electrical Engineering, University of Minnesota, 200 Union St., MN 55414, Minneapolis, USA

E-mails: {zennrod, erseghe}@dei.unipd.it, lorenzo.vangelista@unipd.it, emiliano@umn.edu

Abstract—In this paper, a consensus-based clock synchronization algorithm is presented and its resilience to frequency displacement estimation errors as well as communication noise is analyzed. The synchronization-enforcing control that is to be applied to clock periods is derived upon re-casting the consensus problem into a convex optimization problem and solving it in a distributed fashion via alternating direction method of multipliers. Conversely, plain consensus is used for reaching agreement on clock values. The proposed algorithm achieves higher convergence rates and equivalent noise resilience with respect to algorithmic solutions based on the only plain average multipliers. Plain consensus was applied to clock periods in Section IV. Consensus on the clock counters is introduced in Section V. The ADMM-based consensus-enforcing control is applied to the (local) clock periods, assuring coherent values across the network for both quantities. As inputs, frequency displacement measurements are used, which can be obtained, for example, by using methods that rely on time differences [4], [8], [14], [15], [16]. Robustness of the algorithm to both frequency displacement measurements errors and communication and/or quantization errors is analytically carried out and numerically corroborated. Compared to the classical average consensus, the ADMM-based synchronization algorithm shows superior noise resilience and faster convergence.

The rest of the paper is organized as follows. Section II introduces the system model and poses the clock synchronization problem. Section III briefly introduce the basics on the ADMM-based consensus, which is to be applied to the clock periods in Section IV. Consensus on the clock counters is faced in Section V. Numerical results are presented in Section VI and, finally, the paper is wrapped up in Section VII.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a sensor network comprising \(N\) nodes deployed in an ad hoc fashion. Denote as \(c_i(k)\) and \(T_i(k)\) the values of clock counter (also called clock phase) and clock
period, respectively, at sensor \( i = 1, \ldots, N \) at time instant \( k \in \mathbb{N} \). Collect in the \( N \times 1 \) vector \( c(k) \) the clock counters, i.e. \( c(k) \triangleq [c_1(k), \ldots, c_N(k)]^T \). Likewise, define \( T(k) \triangleq [T_1(k), \ldots, T_N(k)]^T \). Then, the well-established clock synchronization problem essentially entails sensors to reach consensus on common values for \( c(k) \) and \( T(k) \) as time goes by, starting from initial non-coherent values \( c(0), T(0) \).

The ensuing analysis as well as the algorithmic solutions are based on the reference model \([7]-[9], [17]\)

\[
e(k + 1) = c(k) + T(k) + u(k) , \\
T(k + 1) = (T(k) + v(k)) ,
\]

(1)

(2)

where \( u(k) \) and \( v(k) \) are \( N \times 1 \) real vectors collecting control signals, which are to be designed in order to enforce clock synchronization. A handful of algorithms have been recently devised for computing \( u(k) \) and \( v(k) \), comprising:

i) \( u(k) = Hc(k) \) \([7]\), with \( H + I \) a row-stochastic matrix that adheres to the network configuration (hence \( H1 = 0 \), and \( I \) the \( N \times N \) identity matrix. Conversely, no control is applied to \( T(k) \). This solution ensures clock frequencies to line up while going ahead with the iterations \( k \), but timing phases remain generally mismatched.

ii) \( u(k) = Hc(k) \) and \( v(k) = HT(k) \) \([8]\), i.e., average consensus is applied to clock counters as well as periods. Not that an estimate of the clock periods (equivalently, of the clock frequencies) is required here. Feasibility of this method is assured by the capability of sensors in measuring the period ratios \( T_i(k)/T_j(k) \).

iii) \( u(k) = Hc(k) \) and \( v(k) = \alpha Hc(k) \) \([9]\). However, the possibility to practically apply the control signal \( \alpha Hc(k) \) to \( T(k) \) is arguable, as \( T(k) \) is not a quantity directly measurable.

The problem of designing the synchronization-enforcing controls is tackled here by adhering to the classical consensus approach, but applying the ADMM to derive the equation governing the update of \( v(k) \). Upon recasting the consensus problem into a constrained convex optimization problem \([11], [13]\), the application of the ADMM naturally leads to a distributed (in the sense that only one-hop message passing is required) algorithmic implementation. As shown later on, the proposed solution will comprise a plain consensus algorithm running in parallel to an ADMM-based consensus algorithm, ensuring fast convergence to commons values for \( c(k) \) and \( T(k) \).

III. ADMM-BASED CONSENSUS

Consider computing the quantity \( \hat{\theta} \triangleq (1/N) \sum_{i=1}^{N} \theta_i \), with \( \theta_i \in \mathbb{R} \) a given parameter collected at sensor \( i \). Equivalently, \( \hat{\theta} \) can be found by solving the following constrained convex optimization problem \([11], [13], [14]\)

\[
\arg\min_{\{\hat{\theta}_i, z_i\}_{i=1}} \sum_{i=1}^{N} (\hat{\theta}_i - \theta_i)^2 \\
\text{s.t. } \hat{\theta}_i = z_j , \quad i = 1, \ldots, N \quad j \in N_i ,
\]

(3)

where \( \hat{\theta}_i \) is a copy of \( \hat{\theta} \) at node \( i \), \( N_i \) denotes the set of one-hop neighboring nodes of sensor \( i \), and \( \{z_j\} \) are real auxiliary variables. Under the assumption of connected network, i.e., there always exists a (possibly multi-hop) path connecting two nodes of the network, \( 4 \) ensures local copies \( \{\hat{\theta}_i\} \) to be coherent across the network. Differently from \([11]\), here election of the so called “bridge” nodes is not required. See, also, \([13]\).

By employing the ADMM to solve problem (3)-(4), and defining the following real (and known) quantities

\[
a_{i,j} = \frac{c_{i,j}}{\sum_{\ell \in N_i} c_{\ell,j}} , \quad b_{i,j} = \frac{c_{i,j}}{1 + \sum_{\ell \in N_i} c_{i,\ell}} , \quad d_i = \sum_{j \in N_i} b_{i,j} ,
\]

where \( \{c_{i,j} > 0\} \) are arbitrary augmentation constants and \( \sum_i a_{i,j} = 1 \), one can reduce computation of the estimand in (3) to be obtained upon iteratively performing the following updates (see \([14]\) for a detailed derivation)

\[
\hat{\theta}_i(k+1) = (1 - d_i)\hat{\theta}_i + d_i \bar{\theta}_i(k) + \bar{\theta}_i(k) + 2u_i(k) , \quad \bar{\theta}_i(k) = \bar{\theta}_i(k) + u_i(k) ,
\]

(5)

(6)

where \( k \in \mathbb{N} \) is the iteration index and

\[
u_i(k) = \sum_{j \in N_i} b_{i,j} \left( \sum_{n \in N_j} a_{n,j} \hat{\theta}_n(k) \right) - d_i \hat{\theta}_i(k) ,
\]

(7)

is a quantity that can be evaluated in a distributed fashion by utilizing \( n \) message exchanges between one-hop neighboring nodes: the first for the summation in \( a_{i,j} \), the second for the summation in \( b_{i,j} \).

It is convenient to rewrite iterations (5)-(7) in a compact vector-matrix form. To this end, collect constants \( \{c_{i,j}\} \) in the \( N \times N \) matrix \( C \), with entry \( c_{i,j} \) in \( i,j \)-th position for \( j \in N_i \), and a zero entry for \( j \notin N_i \). Define \( \Delta_1 \triangleq \text{diag}(C^1) \) and \( \Delta_2 \triangleq \text{diag}(C1) \), where \( 1 \) is the all-ones column vector. Let \( \Lambda \triangleq \Delta_1^{-1} C^1 \) and \( B \triangleq (I + \Delta_2^{-1}) \Lambda \) the \( N \times N \) matrices collecting parameters \( a_{i,j} \) and \( b_{i,j} \), respectively. Then, upon defining \( D \triangleq (I + \Delta_2^{-1}) \Delta_2 \) the diagonal matrix collecting parameters \( d_i \) and, finally, \( U \triangleq BA - D \), (5)-(7) can be compactly rewritten as

\[
\hat{\theta}(k+1) = (I + D + 2U)\hat{\theta}(k) - (D + U)\hat{\theta}(k-1) ,
\]

(8)

where \( \hat{\theta}(k) \triangleq [\hat{\theta}_1(k), \ldots, \hat{\theta}_N(k)]^T \).

Parameters \( \{c_{i,j}\} \) can now be optimized in order to speed up the convergence toward \( \theta \). To this end, and to reveal the Jordan-like properties of the consensus updates (8), it is convenient to pick matrices \( C \) and \( D \) as \([14]\)

\[
C = \epsilon S , \quad D = d I , \quad d = \frac{\epsilon}{\epsilon + 1} ,
\]

(9)

with \( \epsilon > 0 \) a tuning parameter and \( S \) a row-stochastic matrix that are to be optimized in order to speed up the rate of convergence.

Using (9), (8) can be finally re-expressed as

\[
\hat{\theta}(k+1) = \hat{\theta}(k) + d \left[ (L - I)\hat{\theta}(k) + L(\hat{\theta}(k) - \hat{\theta}(k-1)) \right] ,
\]

(10)
with the initial conditions $\theta(1) = \hat{\theta}(0) = [\theta_1, \ldots, \theta_N]^T$, and $L$ is a symmetric doubly stochastic ($L = L^T$, $L1 = 1$) and positive semi-definite matrix given by
\[
L = SQ, \quad Q = [\text{diag}(S^T1)]^{-1} S^T, \quad (11)
\]
where both $S$ and $Q$ are (row stochastic) one-step communication matrices. Critical to the speed of convergence of the ADMM-based consensus is $\epsilon$; matrix $S$ can instead be set to any consensus update matrix, as this provides a very small loss compared to the optimum matrix choice. In general, (10) provides fast convergence (much faster than the optimal Boyd’s solution) and resilience to noise impairments.

The choice of the value $\epsilon_{\text{opt}}$ that assures maximum convergence speed is given by the following result, that holds from [14]:

**Theorem 1:** Let $1 = \phi_1 \geq \phi_2 \geq \ldots \geq \phi_N \geq \frac{1}{2}$ be the ordered eigenvalues of matrix $\Phi \triangleq (L + I)/2$. The convergence speed of (10) is maximized when $\epsilon$ is chosen equal to
\[
\epsilon_{\text{opt}} = \begin{cases} 
\frac{1}{1-(4\phi_2-3)^2}, & \phi_2 > \frac{3}{4} \\
1, & \phi_2 = \frac{3}{4} \\
\frac{\phi_2 - \frac{1}{2}}{(\phi_2 - \phi_N)^2 + (\phi_2 + \phi_N - 1)(\phi_2 - \phi_N)^2 + 4(\phi_2 - \frac{1}{2})(1-\phi_N)}, & \phi_2 < \frac{3}{4}
\end{cases},
\]
(12)

### IV. ADMM-BASED CONSENSUS ON CLOCK PERIODS

#### A. Chosen approach

The control signal $v(k)$ is envisioned here to be only dependent upon the (current) vector $T(k)$. Consider computing the average quantity $T \triangleq (1/N) \sum_{i=1}^{N} T_i(0)$. Retracing the procedure in Section III reveals that the ADMM-based consensus-enforcing control $v(k)$ at time $k$ can be expressed as [cf. (10)]
\[
v(k) = d \left[ (L - I)T(k) + L(T(k) - T(k-1)) \right]. \quad (13)
\]

#### B. Implementation aspects

It can be observed that the control $v(k)$ in (13) can be applied through standard phase-locked loop (PLL) [18] or delay-locked loop (DLL) structures. As already mentioned, $T(k)$ is not directly measurable. However, what is available for computing (13) is the set of raw measurements of the (relative) frequency displacement between node pairs
\[
\Delta f_{i,j}(k) \triangleq \frac{T_j(k) - T_i(k)}{T_i(k)} - 1, \quad \forall j \in N_i, \quad (14)
\]
that are sequentially gathered at each node $i$. Conceivably, the foregoing frequency displacement can be measured in many different ways. Broadly-used methods rely on time differences, which either calculate packet length means using both the transmit and the receive clock time bases or employ a two-way handshake to evaluate the round trip time [4], [8], [15], [16]. Nevertheless, any modern digital receiver is endowed with reliable frequency estimation capabilities and, thus, estimation methods such as the Schmidl-Cox one [19] can be effectively employed. In particular, [19] can be employed in a cross layer fashion.

Upon defining the frequency correction factor for sensor $i$ at time $k$ as
\[
\Delta F_i(k) \triangleq \frac{T_j(k-1)}{T_i(k)} - 1, \quad (15)
\]
(13) can be re-written as a function of (14) as
\[
\Delta F_i(k+1) \simeq d \sum_{j \in N_i^{(2)}} L_{i,j} \left[ \Delta F_j(k) - \Delta f_{i,j}(k) \right], \quad (16)
\]
where the approximations $[1, \Delta F_j(k+1)]^{-1} \simeq 1 - \Delta F_j(k+1)$ and $\Delta F_j(k+1) \Delta f_{i,j}(k) \simeq 0$ have been used, as the frequency offset of practical systems is small - usually it ranges from 10 to 100 ppm [20]. Also, $N_i^{(2)}$ denotes the set of two-hop neighboring nodes of sensor $i$.

Because of the two-hop structure of $L$ in (11), update (16) requires that the two-hop correction factors $\Delta F_j(k)$ and the two-hop displacement estimates $\Delta f_{i,j}(k)$ are available at node $i$. This requires an information exchange between nodes. Specifically, let $n$ the intermediate node, so that $n \in N_i$ and $j \in N_n$, and let $\Delta F_{i,n}(k) \simeq \Delta f_{i,n}(k) + \Delta f_{n,j}(k)$ because of the assumption of small frequency offsets. Then (16) can be rewritten as
\[
\Delta F_i(k+1) \simeq d \sum_{n \in N_i} S_{i,n} \left[ \Delta G_n(k) - \Delta f_{i,n}(k) \right], \quad (17)
\]
where
\[
\Delta G_n(k) = \sum_{j \in N_n} Q_{n,j} \left[ \Delta F_j(k) - \Delta f_{n,j}(k) \right]. \quad (18)
\]
The above imply that $\Delta F_j(k)$ is transmitted to neighbors during the first message exchange, and $\Delta G_n(k)$ during the second message exchange. The quantities $\Delta f_{i,n}(k)$ and $\Delta f_{n,j}(k)$ are instead directly observable at node $i$ and $n$, respectively.

#### C. Accounting for noise

From a practical perspective, equations (17)–(18) constitute the main iterates of the consensus-enforcing algorithm; hence, they serve as a reference point for introducing a realistic noise model. Noise in (17)–(18) is due to either the effect of errors in estimating $\{\Delta f_{i,n}(k), \Delta f_{n,j}(k)\}$, or quantization and/or communication errors that may happen due to the low-rate transmission of $\{\Delta F_j(k), \Delta G_n(k)\}$.

Here it is assumed an additive white Gaussian noise (AWGN) model for noise sources; moreover, noises associated to $\Delta f_{i,n}(k)$, $\Delta f_{n,j}(k)$, $\Delta F_j(k)$, and $\Delta G_n(k)$, are also assumed mutually independent. This implies that $\Delta f_{i,n}(k)$ and $\Delta f_{n,j}(k)$ are estimated independently. Information reuse of frequency deviation estimates is also possible, but not pursued as it implies larger errors.

By incorporating all the sources of uncertainty in the $N \times 1$ noise vector $w_n(k)$, which is to be associated to the ADMM step (13), one obtains the following noise-corrupted control
\[
v(k) = d \left[ (L - I)T(k) + L(T(k) - T(k-1)) \right] + w_n(k). \quad (19)
\]
Note that \( u_w(k) \) is AWGN and temporally white. Also, its covariance matrix can be obtained after some manipulations as

\[
R_u = \mathbb{E}[w_u(k)w_u^*(k)] 
\approx d^2T_{\text{nom}}^2 \left[ S \Sigma G S^T + ((S \circ \Sigma_f)S^T) \circ I \right] 
+ L \Sigma_F L^T + S[(Q \circ \Sigma_f)Q^T \circ I]S^T
\]

(20) where \( \circ \) denotes the Hadamard matrix product (entry-wise product), \( \Sigma_F \) denotes the nominal clock period, and

\[
\Sigma_F \equiv \text{diag}(\sigma^2_{f_1}, \ldots, \sigma^2_{f_N}) 
\]

(21)

\[
\Sigma_f \equiv \text{diag}(\sigma^2_{c_1}, \ldots, \sigma^2_{c_N}) 
\]

(22)

\[
\sigma^2_c \equiv [\sigma^2_{c_{i,j}}]_{i,j=1,\ldots,N}
\]

(23)

aggregate estimation and transmission noise variances.

Because of uncorrelated noise, the following result holds from [14]:

**Theorem 2:** The noisy control signal (19) assures consensus on clock periods \( T(k) \). Letting the deviation from the average value of \( T(k) \) be written as \( \tilde{T}(k) \equiv K T(k) \), with \( K = I - \frac{1}{N} 11^T \), upon defining the mean squared error on estimating \( T \),

\[
\text{MSE}_{T}(k) \equiv \mathbb{E} \left[ \| \tilde{T}(k) \|^2 \right]
\]

(24) the asymptotic convergence can be expressed by the norm-2 result

\[
\lim_{k \to \infty} \text{MSE}_{T}(k) \leq \sum_{i=2}^{N} \frac{(1 + d \ell_i) \sigma^2_{\max}}{d(1 - \ell_i)(1 - d \ell_i)(2 + 3d \ell_i - d)}
\]

(25)

where \( \ell_i, i = 2, \ldots, N \) are the eigenvalues of \( L \) excluding \( \ell_1 = 1 \), and where \( \sigma^2_{\max} \) is the largest eigenvalue of matrix \( K R_u K \).

**V. CONSENSUS ON CLOCK PHASES**

Differently from (13), ADMM-based consensus can be directly applied to enforce coherent clock phases, as \( c(k) \) are observable variables. However, a naive application of ADMM to (1) causes a bias on the control signal \( u(k) \) when convergence is reached. To overcome this issue, one can apply a plain consensus approach. In fact, the application of ADMM on \( T(k) \) allows a faster convergence to the common \( \bar{T} \) and this speed up consistently also the agreement on a common clock phase, no matter what algorithm is used for the latter. The control signal at time \( k \) can then be expressed as

\[
u(k) = (H_A - I)c(k) + w_u(k),
\]

(26)

where \( H_A \) is any plain consensus matrix, i.e. the Metropolis weights matrix [21], the Laplacian matrix [7] or the consensus average matrix [14]. Additive noise in (26) is taken into account by \( w_u(k) \). The noise model can easily be derived if \( c(k) \) is assumed to be affected by noise with covariance \( \Sigma_c \). Therefore, in an AWGN context, the noise covariance matrix has the form

\[
R_u = (H_A - I) \Sigma_c (H_A - I)^T, \quad \Sigma_c \equiv \text{diag}(\sigma^2_{c_1}, \ldots, \sigma^2_{c_N})
\]

(27)

where \( \sigma^2_c \) are communication and/or quantization noise variances.

As a standard performance metric, the mean squared error for the convergence on a common clock phase is defined as

\[
\text{MSE}_{c}(k) \equiv \mathbb{E} \left[ \| \hat{c}(k) \|^2 \right]
\]

(28)

where \( \hat{c}(k) \equiv Kc(k) \).

**VI. SIMULATION RESULTS**

In this Section, performance of clock synchronization via ADMM is evaluated an compared with existing approaches via numerical simulations. Considered the random geometric graph (RGG) depicted in Fig. 1; in particular, \( N = 50 \) nodes, with an average number of 6 neighbors per node. The average consensus matrix \( S_{\text{AC}} \) [14] is used in (11) and optimum values for \( \epsilon \) are selected as indicated in (12). Matrix \( H_A \) in (26) is also set to the consensus average matrix (ADMM+AC), while the covariance matrices of the noise corrupting \( u(k) \) and \( v(k) \) are set to \( R_u = \sigma^2_u I \) with \( \sigma^2_u = 10^{-2} \), and \( R_v = \sigma^2_v I \) with \( \sigma^2_v = 10^{-6} \), respectively. The nominal oscillation frequency is set to \( f_{\text{nom}} = 1/T_{\text{nom}} = 32768 \) Hz. The initial values \( T(0) \) are chosen as independent Gaussian random variables, each having mean \( T_{\text{nom}} \) and standard deviation 50 ppm, while the initial offset values \( c(0) \) are chosen as independent Gaussian random variables with zero mean and variance 1. Finally, the time between two successive iterations of the algorithm is set to 10 seconds.

**A. Convergence Speed and Noise Resilience**

The convergence speed and noise resilience of (1) and (2) are considered in this section, through an investigation of the error on the clock phases and periods.

Fig. 2 shows the phase errors \( \hat{c}_i(k) \) for five nodes of the network. It is evident that the error decreases to the order of the unity after less than 500 iterations of the algorithm, after which it remains under this threshold and, thus, network synchronization is kept. The sawtooth shape of the trajectory happens due to the fact that between two successive iterations (when the
corrections (26) and (19) are applied) the synchronization error increases linearly, since the frequency drift remains constant. On the other hand, it is observable that the slope of such an error decreases as the number of iterations increases, denoting an improvement in the oscillation frequency agreement as time goes by.

The average mean squared error for clock periods and counters, namely $\text{MSE}_c(k)$ and $\text{MSE}_T(k)$ respectively, are shown in Fig. 3; also, bound (25) for $\text{MSE}_T(k)$ is reported. It can be noted that consensus on clock phases comes obviously after agreement on the oscillation frequency has been reached. Moreover, the difference between oscillation periods at the very beginning causes an initial increment for the phase error. This trend is then inverted once consensus over clock periods is going to be reached. Finally, it can be observed that the noise bound (25) for the estimation of $\bar{T}$ is tight; in fact, the error curve overlaps the trajectory corresponding to the bound for large $k$.

**B. Performance Comparison with Plain Consensus**

Fig. 4 shows $\text{MSE}_c(k)$ and $\text{MSE}_T(k)$ for the proposed ADMM-based synchronization algorithm, compared to the case in which the standard consensus is applied to both clock periods and clock phases; in particular, the standard average consensus is considered when a) the update matrix is given by $H = \left(\text{diag}(E_1)\right)^{-1} E$, where $E$ is the adjacency matrix associated to the underlying graph (consensus average matrix, AC) and b) the update matrix is designed as in [21], which ensures the fastest convergence speed among standard consensus approaches (Boyd’s consensus, BO). Although its high computational cost prevents its practical implementation, Boyd’s consensus is reported as a benchmark. It is noted that clock synchronization via ADMM significantly outperforms the “double” application of the Boyd’s consensus, which in turn has clearly better performance with respect to the application of the consensus average matrix. As mentioned in Section V, thanks to the dramatic convergence speed improvement in achieving consensus with ADMM for oscillation periods, the agreement on clock phases is reached earlier: in fact, while for standard consensus more than 340 iterations are needed in order to get the network synchronized (i.e., $\text{MSE}_c < 1$), the same performance is obtained by waiting almost 270 iterations if using the ADMM based consensus. Finally, both for periods and clock values, the proposed algorithm results to be as robust as the standard consensus with respect to additive noise.

**VII. CONCLUSIONS**

In this paper a new distributed algorithm has been proposed in order to synchronize nodes in a wireless sensor network. The proposed approach is based on the application of ADMM for reaching consensus on the oscillation periods. ADMM has been observed in the recent past to bring substantial improvements in convergence speed with respect to standard
consensus approaches, and its implementation in parallel to plain consensus for correcting the phase error on oscillators allows the clock synchronization between nodes to be reached well in advance with respect to existing methods.

REFERENCES