Distributed Calculation of Linear Functions in Noisy Networks via Linear Iterations

Shreyas Sundaram and Christoforos N. Hadjicostis
Consider a network with nodes \{x_1, x_2, ..., x_N\}
- e.g., sensors, robots, unmanned vehicles, computers, etc.

Each node \( x_i \) has some initial value \( x_i[0] \)
- e.g., temperature measurement, position, vote, etc.

**Objective:** Some nodes must calculate certain functions of initial values

![Diagram of a network with nodes connected by lines](image.png)
Consider a network with nodes \( \{x_1, x_2, \ldots, x_N\} \)
- e.g., sensors, robots, unmanned vehicles, computers, etc.

Each node \( x_i \) has some initial value \( x_i[0] \)
- e.g., temperature measurement, position, vote, etc.

**Objective:** Some nodes must calculate certain functions of initial values

\[
x_1[0] + x_7[0] = ?
\]

\[
x_6[0] = ?
\]

\[
4x_3[0] - x_5[0] + 5x_7[0] = ?
\]
Problem Formulation

Consider a network with nodes \{x_1, x_2, ..., x_N\}

- e.g., sensors, robots, unmanned vehicles, computers, etc.

Each node \( x_i \) has some initial value \( x_i[0] \)

- e.g., temperature measurement, position, vote, etc.

**Objective**: Some nodes must calculate certain functions of initial values

- **Consensus**: All nodes calculate the same function

\[ g(x_1[0], \ldots, x_N[0]) = ? \]
Previous Work

- Distributed function calculation schemes have been well studied over past few decades
  - Issues of communication complexity, computational complexity, time complexity, fault tolerance, ...

- Many excellent books on this topic
  - *Dissemination of Information in Communication Networks*, Hromkovic et. al., 2005
  - *Communication Complexity*, Kushilevitz and Nisan, 1997
  - *Elements of Distributed Computing*, Garg, 2002
  - *Parallel and Distributed Computation*, Bertsekas and Tsitsiklis, 1997
  - ...
Linear Iterative Schemes

- Investigate **linear iterative schemes** for distributed function calculation

- At each time-step $k$, every node updates its value as

$$x_i[k + 1] = w_{ii} x_i[k] + \sum_{j \in \text{nbr}(i)} w_{ij} x_j[k]$$

- Update equation for entire system:

$$\begin{bmatrix}
  x_1[k + 1] \\
  \vdots \\
  x_N[k + 1]
\end{bmatrix} =
\begin{bmatrix}
  w_{11} & \cdots & w_{1N} \\
  \vdots & \ddots & \vdots \\
  w_{N1} & \cdots & w_{NN}
\end{bmatrix}
\begin{bmatrix}
  x_1[k] \\
  \vdots \\
  x_N[k]
\end{bmatrix}$$

- Weight $w_{ij} = 0$ if $x_j$ is not a neighbor of $x_i$
Linear Iterative Schemes

- Extensively studied for **asymptotic consensus**: For all \( i \), and some function \( g(\cdot) \), \( \lim_{k \to \infty} x_i[k] = g(x_i[0], \ldots, x_N[0]) \)

- **Theorem (Xiao & Boyd, 2005)**: Iteration achieves asymptotic consensus on \( f^T x[0] \) (for some vector \( f \)) if and only if:
  - \( W1 = 1, \ f^T W = f^T \) (where \( 1 \) is vector of all 1’s)
  - All other eigenvalues have magnitude less than 1

- Survey papers:
  - **Ren, Beard & Atkins**, *Proc. ACC*, 2005
Noisy Networks

What if transmissions/updates are noisy?

Examples:

- Noisy measurement of neighbor’s value
- Quantized transmissions between neighbors

At each time-step $k$, node $x_i$ updates value as:

$$x_i[k + 1] = w_{ii}x_i[k] + \sum_{j \in \text{nbr}(i)} w_{ij}x_j[k] + n_i[k]$$

What happens as $k \to \infty$?
Asymptotic Behavior of Noisy Linear Iterations

Theorem ([Xiao, Boyd & Kim, 2007]):

For the noisy linear iteration

\[
\begin{bmatrix}
    x_1[k + 1] \\
    \vdots \\
    x_N[k + 1]
\end{bmatrix}
= 
\begin{bmatrix}
    w_{11} & \cdots & w_{1N} \\
    \vdots & \ddots & \vdots \\
    w_{N1} & \cdots & w_{NN}
\end{bmatrix}
\begin{bmatrix}
    x_1[k] \\
    \vdots \\
    x_N[k]
\end{bmatrix}
+ 
\begin{bmatrix}
    n_1[k] \\
    \vdots \\
    n_N[k]
\end{bmatrix}
\]

with \( W1 = 1, \ 1^T W = 1^T, \ E[n[k]] = 0, \ E[n[k] n^T[j]] = I \delta_{k,j} \),

\[
E[x_i[k]] \xrightarrow{k \to \infty} \frac{1}{N} 1^T x[0] \quad \text{Asymptotically Unbiased}
\]

\[
E\left[ \left( x_i[k] - \frac{1}{N} 1^T x[0] \right)^2 \right] \xrightarrow{k \to \infty} \infty \quad \text{Unbounded Variance!}
\]
How to Handle Unbounded Variance?

- Various techniques to prevent unbounded variance
  - Predictive Coding (Yildiz & Scaglione, 2007)
  - Time-varying weights (Huang & Manton, 2007)
  - Second order recursions (Schizas, Ribeiro & Giannakis, 2008)

- However, all existing work focuses on obtaining convergence asymptotically

- Our contribution:
  - Each node obtains an unbiased, bounded variance estimate of any linear function of initial values after a finite number of iterations
  - For given weights and number of iterations, can minimize the variance of the estimate
Modeling The Noisy Linear Iteration

- Noisy linear iteration model:

\[
x[k + 1] = Wx[k] + n[k] \\
y_i[k] = C_i x[k], \quad i \in \{1, 2, \ldots, N\}
\]

\[
E[n[k]] = 0 \\
E[n[k]n^T[l]] = Q_{kl}
\]

- \(y_i[k] = C_i x[k]\) denotes values seen by node \(x_i\) at time-step \(k\)
  - Rows of \(C_i\) index portions of \(x[k]\) available to \(x_i\)

For node \(x_1\):

\[
y_1[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \\ x_4[k] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x[k] \\
C_i[k]
\]
Modeling The Noisy Linear Iteration (2)

- Noisy linear iteration model:

\[
x[k + 1] = Wx[k] + n[k] \quad \quad E[n[k]] = 0
\]
\[
y_i[k] = C_i x[k], \quad i \in \{1, 2, \ldots, N\} \quad \quad E[n[k]n^T[l]] = Q_{kl}
\]

Note:

- Above model corresponds to update noise, with perfect measurements
- Paper analyzes more general model

\[
x[k + 1] = Wx[k] + Bn[k]
\]
\[
y_i[k] = C_i x[k] + D_i n[k]
\]
- Encapsulates wide range of noisy scenarios, including measurement noise
- Simplified model will be sufficient to explain theory
Values Seen Over Several Time-Steps

- Noisy linear iteration:
  \[ x[k + 1] = Wx[k] + n[k] \]
  \[ y_i[k] = C_i x[k], \quad i \in \{1, 2, \ldots, N\} \]
  \[ E[n[k]] = 0 \]
  \[ E[n[k]n^T[l]] = Q_{kl} \]

- Values seen by node \( x_i \) over \( L+1 \) time-steps:

  \[
  \begin{bmatrix}
  y_i[0] \\
  y_i[1] \\
  y_i[2] \\
  \vdots \\
  y_i[L]
  \end{bmatrix}
  \begin{bmatrix}
  C_i \\
  C_i W \\
  C_i W^2 \\
  \vdots \\
  C_i W^L
  \end{bmatrix}
  \begin{bmatrix}
  x[0] \\
  n[0] \\
  n[1] \\
  \vdots \\
  n[L]
  \end{bmatrix}
  \begin{bmatrix}
  0 & 0 & \cdots & 0 \\
  C_i & 0 & \cdots & 0 \\
  C_i W & C_i & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  C_i W^{L-1} & C_i W^{L-2} & \cdots & 0
  \end{bmatrix}
  \]

  - \( O_{i,N-1} \) is the **observability matrix** for the pair \((W,C_i)\)
  - \( M_{i,L} \) is the **noise matrix**
Unbiased Estimation

- Values seen by node $x_i$:
  \[
y_i[0:L] = O_{i,L} x[0] + M_{i,L} n[0:L]
  \]

- Node $x_i$ wants to calculate **unbiased** estimate of $f^T_i x[0]$, for some vector $f^T_i$
  - **Unbiased**: Expected value of estimate is equal to $f^T_i x[0]$

- Suppose row-space of $O_{i,L}$ contains $f^T_i$
  - Find vector $\Gamma_i$ such that $\Gamma_i O_{i,L} = f^T_i$

**Theorem:** $\Gamma_i y_i[0:L]$ is an **unbiased estimate** of $f^T_i x[0]$

\[
E[\Gamma_i y_i[0:L]] = E[\Gamma_i O_{i,L} x[0] + \Gamma_i M_{i,L} n[0:L]] = f^T_i x[0]
\]
Row-Space of Observability Matrix

- How to choose $W$ so that row-space of $O_{i,L}$ contains $f_i^T$?

- **Theorem ([1]):** If network is strongly connected, then for *every* $i$ and *almost any choice of weights*, $O_{i,L}$ will have *rank N* for some $L \leq N$.
  - "Almost any": for all but a set of measure zero
  - Result obtained by using *structural observability theory*

- Application to noisy networks:

  **Theorem:** If network is strongly connected, *every* node $x_i$ can calculate an unbiased estimate of *any linear function* of initial values after running linear iteration with *almost any weights* for at most $N$ time-steps

---

Row-Space of Observability Matrix

- How to choose $W$ so that row-space of $O_{i,L}$ contains $f_i^T$?

- **Theorem ([1]):** If network is strongly connected, then for every $i$ and almost any choice of weights, $O_{i,L}$ will have rank $N$ for some $L \leq N$.
  - “Almost any”: for all but a set of measure zero
  - Result obtained by using structural observability theory

- Application to noisy networks:

  **Theorem:** If network is strongly connected, every node $x_i$ can calculate an unbiased estimate of any linear function of initial values after running linear iteration with almost any weights for at most $N$ time-steps

*Paper contains generalization to networks that are not strongly connected*

---

Minimizing Variance of Estimation Error

- **Multiple choices** of $\Gamma_i$ may satisfy $\Gamma_i \mathbf{O}_{i,L} = \mathbf{f}_i^T$
  - Choose $\Gamma_i$ that satisfies $\Gamma_i \mathbf{O}_{i,L} = \mathbf{f}_i^T$, and **minimizes mean square estimation error** (variance)

- **Estimation error:**
  \[ \varepsilon = \Gamma_i \mathbf{y}_i[0 : L] - \mathbf{f}_i^T \mathbf{x}[0] \]
  \[ = \Gamma_i \mathbf{O}_{i,L} \mathbf{x}[0] + \Gamma_i \mathbf{M}_{i,L} \mathbf{n}[0 : L] - \mathbf{f}_i^T \mathbf{x}[0] \]
  \[ = \Gamma_i \mathbf{M}_{i,L} \mathbf{n}[0 : L] \]

- **Mean Square Estimation Error:**
  \[ E[\varepsilon \varepsilon^T] = \Gamma_i \mathbf{M}_{i,L} \left( E[\mathbf{n}[0 : L] \mathbf{n}^T[0 : L]] \right) \mathbf{M}_{i,L}^T \Gamma_i^T \]
Minimizing Variance of Estimation Error

- **Optimization problem:** Find $\Gamma_i$ to minimize
  \[ E[\varepsilon^T] = \Gamma_i M_{i,L} \prod_L M_{i,L}^T \Gamma_i^T \]
  subject to $\Gamma_i O_{i,L} = f_i^T$

- **Convex (Quadratic) cost function, with linear constraints**
  - Parameterize $\Gamma_i$ to satisfy constraints, and use remaining freedom to minimize cost

**Solution:**
\[
\Gamma_i = a_i^T \Lambda_i^{-1} \left[ I - \Phi_i \prod_L \Psi_i^T (\Psi_i \prod_L \Psi_i^T)^{-1} \right] U_i^T
\]
where
\[
O_{i,L} = U_i \begin{bmatrix} \Lambda_i & 0 \\ 0 & 0 \end{bmatrix} V_i^T, \quad [a_i^T \ 0] = f_i^T V_i, \quad \begin{bmatrix} \Phi_i \\ \Psi_i \end{bmatrix} = U_i^T M_{i,L}
\]
**Objective:** Every node has to calculate unbiased estimate of $1^T x[0]/4$

Consider node $x_1$:

$$ y_1[k] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x[k] = C_1 x[k] $$

Update noise

$$ E[n[k]] = 0 $$

$$ E[n[k] n^T [l]] = \mathbf{I} \delta_{k-l} $$
Example (cont.)

- Find smallest delay $L$ so row-space of $O_{1,L}$ contains $1^T/4$
  - For $L = 0$:
    $$O_{1,0} = C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
    ![Red X]
  - For $L = 1$:
    $$O_{1,1} = \begin{bmatrix} C_1 \\ C_1 W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$
    ![Green Check]
  - Node $x_1$ can calculate its function after $L+1 = 2$ time-steps
Example (cont.)

- Find vector $\Gamma_1$ satisfying $\Gamma_1 O_{1,1} = 1^T/4$ and minimizing

$$MSE = \Gamma_1 M_{1,1} \Pi_1 M_{1,1}^T \Gamma_1^T = \Gamma_1 \begin{bmatrix} 0 & 0 \\ C_1 & 0 & 0 \\ 1 & 0 & 0 \\ C_1^T \end{bmatrix} \Gamma_1^T$$

- Optimal Solution:

$$\Gamma_1 = \begin{bmatrix} 0 & \frac{1}{8} & \frac{1}{8} & 0 & \frac{3}{8} & \frac{3}{8} \end{bmatrix}$$

- Minimum Mean Square Error $= 9/32$
Linear Iterative Scheme With Noise

Time-Step k = 0

\[
\begin{align*}
(1+2-2)/3 &= 0.61 = 0.95 \\
(2+1-1)/3 &= 1.01 = 0.34 \\
(-1+2-2)/3 &= -0.06 = -0.39 \\
(-2+1-1)/3 &= -2.17 = -2.84 \\
\end{align*}
\]

Initial values: -2, -1, 2, 1
Average = 0
Additive noise: \( N(0,1) \)

Values seen by \( x_1 \):
\[
y_1[k] = [x_1[k] \ x_2[k] \ x_4[k]]^T
\]
\[
y_1[0] = [-2 -1 1]^T \\
y_1[1] = [-2.84 -0.39 0.95]^T
\]

\( x_1 \) calculates unbiased estimate of average as \( \Gamma_1 y_1[0:1] = 0.2082 \)
Summary and Future Work

Summary

- Linear iterative strategy allows every node to obtain unbiased estimate of any linear function of initial values after finite number of time-steps
- Can be done with almost any choice of weights

Future Work

- Choose weight matrix $W$ to further reduce variance of estimation error
- Obtain quantitative relationship between delay $L$ and mean square estimation error