Learning Conserved Networks from Flows

Satya Jayadev P. a,d, Shankar Narasimhan b,d,2, and Nirav Bhatt c,d,2

aDepartment of Electrical Engineering, Indian Institute of Technology Madras, Chennai-600036, India; bDepartment of Chemical Engineering, Indian Institute of Technology Madras, Chennai-600036, India; cDepartment of Biotechnology, Indian Institute of Technology Madras, Chennai-600036, India; dRobert Bosch Centre for Data Science and Artificial Intelligence, Indian Institute of Technology Madras, Chennai-600036, India

The network reconstruction problem is one of the challenging problems in network science. This work deals with reconstructing networks in which the flows are conserved around the nodes. These networks are referred to as conserved networks. We propose a novel concept of conservation graph for describing conserved networks. The properties of conservation graph are investigated. We develop a methodology to reconstruct conserved networks from flows by combining these graph properties with learning techniques, with polynomial time complexity. We show that exact network reconstruction is possible for radial networks. Further, we extend the methodology for reconstructing networks from noisy data. We demonstrate the proposed methods on different types of radial networks.

Network reconstruction | Low rank approximation | PCA | Graph realization

Network science viewpoint of complex systems in life science, engineering, and physics has shown that the connectivity between the different dynamic units of complex systems has an important role to play in understanding the behaviour of these systems (1, 2). The existing literature related to network science deals with characterizing networks and understanding different properties of the underlying complex processes (3–6). Typically, in network science, complex systems are modelled as a graph where the dynamic units are modelled as nodes while the interactions between the dynamic units are modelled as edges (3, 6). Then, tools from graph theory and dynamical systems have been applied to unravel the organisational principles of the underlying complex processes (4, 6). However, in practice, information related to edge connectivity between the different nodes may not be available at all times. Then, the edge connectivity information has to be inferred from data. This problem of inferring the edge connectivity from data is one of the important problems in network science and labelled as network reconstruction problem.

Several methods based on machine learning and time series analysis in the existing literature have been proposed to infer the underlying network connectivity using temporal (7–9) and static data (10–12) arising in different fields. On the one hand, steady-state data have been used to discover biological networks (10), social science networks (11) and engineering networks (13). On the other hand, time series data have been used to infer networks in financial networks (14), biological networks (9, 15), and social networks (9). These methods have used tools from graph theory, Bayesian theory, systems and control theory, and learning theory to solve the problem of network reconstruction.

Different networks such as biological networks, economic networks, engineering networks etc. can be visualized as a chain of interconnected physical or abstract lines through which different entities flow via an edge from one node to other nodes. Furthermore, flows through the edges in such networks are conserved at each incident node. This property of conservation leads to a set of networks which is referred to as conserved networks in this work. Flow of reaction flux from one metabolites to another in metabolic biological networks, flow of commodity (such as power, gas, water) flows from a source to a set of consumers in distribution networks, and flow of information or money flow from a sender to a set of receivers in communication or economical networks are examples of conserved networks.

A conserved network can be represented as a digraph $G(N, E)$ where $N$ is the set of nodes, and $E$ is the set of directed edges indicating the flow direction. In addition, A conservation graph $G_c(N_c, E)$, with $N_c < N$ is the set of conserved nodes, of the digraph $G(N, E)$ can be defined such that the flows in the edges are conserved at every node in $N_c$. Then, conservation equations in terms of flow variables can be written at every node in the conservation graph $G_c$. If the flow data in each edge at different steady states are available, then the question is: "is it possible to reconstruct the underlying conserved network from flow data?" In this work, we show that a conserved network can be reconstructed from flow data by combining the concept of conservation graph and linear learning technique. Particularly, the network reconstruction problem is connected with the graph realization problem by exploiting the conservation property of the
underlying networks.

The main contributions of this work are as follows: (i) a novel concept of conservation graphs is introduced and their properties related to the network structure are established, (ii) the properties of conservation graphs are exploited for developing methodology to reconstruct conserved networks from flow data, (iii) we show that the developed methodology can reconstruct the underlying conserved network exactly for single-source non-looped networks (here onwards referred to as radial or rooted tree-type networks), (iv) we extend the proposed methodology to noisy measurements under different scenarios, and (v) the proposed methodology has a polynomial time complexity.

1. Conservation Graphs

The core idea here is to leverage the conservation in the network to identify the topology. To this effect, we consider the conservation graph of the network instead of the original topology. A conservation graph $G_c$ is one in which the flows in the lines are conserved at ever node and therefore, conservation equations in terms of flow variables can be written at every node. Topology graph is not a conservation graph because conservation equations cannot be written around source and sink nodes. For a given network, its conservation graph is constructed by merging all the source and sink nodes into a single environment node. This ensures that conservation holds at the environment node. The network topology can be easily obtained from its conservation graph by removing the environment node, and connecting the outgoing edges to source nodes and incoming edges to sink nodes.

For illustration, consider a simple flow network with a source $S$ and 5 sinks $(4, 5, 6, 7, 8)$, and its conservation graph $G_c$ shown in Fig. 1. Conservation equations can be written around every node of $G_c$ in terms of the flow variables $x_1, \ldots, x_8$. It is to be noted that, excluding the source node, there exists a one-to-one correspondence between the nodes and incoming edges in $G$ and this correspondence extends to nodes in $G$ with edges in $G_c$. This observation is used later in the paper for convenient presentation of certain facts.

![Figure 1. Topology of a network and its conservation graph](image)

Now, let there be $n$ edges in the network topology i.e., $|E| = n$ and $x_j \in \mathbb{R}^n$ be the vector corresponding to the actual flows along these edges, at steady state $j$. There exists a set of independent linear equations representing the conservation of flows in the network. Let there be $m$ such equations and they can be written as:

$$Ax_j = 0_{m \times 1},$$

where $A$ is a matrix of dimension $(m \times n)$ giving a model of the network and $0$ is a vector of zeros. There can be infinite such linear models which describe the conservation of flows in the network and these matrices

S.N. and N.B. designed research; S.J.P., S.N. and N.B. performed research; contributed new methods/analytic tools; wrote the paper.

The authors declare no conflict of interest.

2To whom correspondence should be addressed. E-mail: naras@iitdl.ac.in, niravbhatt@iitdl.ac.in
share the same row space. Every such model can be obtained by rotating \( A \) with some non-singular rotation matrix \( Q \) as follows:

\[
Ax = QAx = 0_{m \times 1}.
\]

One set of model equations can be obtained by applying conservation at all the nodes in the conservation graph. The resultant matrix will be the incidence matrix of the graph. Since incidence matrix is not of full rank, we consider a reduced incidence matrix. Another set of model equations is given by the f-cutset matrix since the flows are conserved along the edges in a cutset. For the conservation graph in Fig. 1, a reduced incidence matrix (say \( A_1 \)) and an f-cutset matrix (say \( A_2 \)) are given by:

\[
A_1 = \begin{bmatrix}
1 & -1 & -1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
A_2 = \begin{bmatrix}
1 & 0 & 0 & -1 & 0 & -1 & -1
0 & 1 & 0 & -1 & 1 & 0 & 0
0 & 0 & 1 & 0 & 0 & -1 & -1
\end{bmatrix}
\]

It can be mathematically verified that both the models \( A_1 \) and \( A_2 \) share the same row space (16). Now, to deal with the ambiguity in models, we transform the model into an explicit form by partitioning the flow variables into a set of \( m \) dependent \((x_d)\) and \((n - m)\) independent variables \((x_i)\). The sub-matrices of \( A \) corresponding to \( x_d \) and \( x_i \) are denoted as \( A_d \) and \( A_i \) respectively. A partition is considered ‘valid’ only when \( A_d \) is non-singular. This explicit form has a special feature that it is unique with respect to a valid partition as shown in Corollary 1.

**Corollary 1.** Given a linear model \( Ax = 0 \), there exists a unique model \( R = A_d^{-1}A_i \) with respect to any valid partition of the variables \( x = [x_d \mid x_i] \) such that \( x_d = -Rx_i \).

**Proof.** Let us consider a valid partition of the variables \( x = [x_d \mid x_i] \) which implies that \( A_d \) is non-singular. Therefore,

\[
Ax = A_dx_d + A_ix_i = 0
\]

\[
\implies x_d = -A_d^{-1}A_ix_i = -Rx_i
\]

Now, we rotate \( A \) with \( Q \) and repeat the partitioning as follows:

\[
QAx = QA_dx_d + QA_ix_i = 0
\]

\[
\implies x_d = -(QA_d)^{-1}(QA_i)x_i
\]

\[
\implies x_d = -A_d^{-1}(Q^{-1}Q)A_ix_i = -Rx_i
\]

Therefore, \( R \) is unique for any valid partition of \( x \) irrespective of \( A \) used to obtain it.

Any explicit model can be converted back into an implicit form as:

\[
\begin{bmatrix} I_m & R \end{bmatrix} \begin{bmatrix} x_d \\ x_i \end{bmatrix} = Ax = 0. \tag{2}
\]

The implicit model of the form as in Eq. (2) has a graphical interpretation which can be exploited to recover the graph of the network. This implicit model is indeed an f-cutset matrix \( C_f \) of the conservation graph as shown in the Theorem 1.
Theorem 1. If \([x_d \mid x_i]\) is a valid partitioning of \(x\), then the corresponding model matrix \([I_m \ R]\) is the f-cutset matrix \(C_f\) of \(G_c\) with respect to the spanning tree formed by edges corresponding to \(x_d\) and edges corresponding to \(x_i\) are the chords. (For proof refer to Supporting Information)

Proof. Firstly, we verify if the dimension of \(x_d\) equals the number of branches that a spanning tree can have, which is equal to \(|N| - 1\). Since the reduced incidence matrix is a possible model matrix, its rank \(m\) will be equal to the dimension of \(x_d\). It is known that the rank of a reduced incidence matrix is \(|N| - 1\) which implies that \(|N| - 1 = m\). Therefore, \(x_d\) has sufficient branches to form a spanning tree. Now, we have to prove that the selected \(m\) branches indeed form a spanning tree.

By definition, a graph contains \(|N| - 1\) f-cutsets with respect to a spanning tree with each f-cutset containing only one unique branch and multiple chords. Since the removal of edges in an f-cut separates the graph into two parts, by conservation, the net flow in these edges should be zero. Thus, there exists a set of independent linear relations between the flow variables corresponding to each of the f-cutsets.

Now, as per Corollary 1, \(R\) captures a unique set of independent relations between each of the dependent variables \(x_d\) and the independent variables \(x_i\). This implies that \(R\) actually captures the relations given by f-cutsets with \(x_d\) edges as branches and \(x_i\) edges as chords. Therefore, the \(m\) edges corresponding to \(x_d\) form a spanning tree of \(G\) and \([I_m \ R]\) is the associated f-cutset matrix.

Now, we have to realize a graph which has \([I_m \ R]\) as the f-cutset matrix. This is known as the graph realization problem which has been well studied in the literature (17–20). In general, the graph realized cannot be guaranteed to exactly the original because multiple graphs can have the same \(C_f\) and such graphs are called 2-isomorphic graphs (21). However, topology identification for radial networks can be posed as an arborescence realization problem, which is a special case of graph realization problem (22). This is because the conservation graph \(G_c\) of such networks has an arborescence as one of its spanning trees. If the f-cutset matrix corresponding to this spanning arborescence is identified, then graph realization results in 2-isomorphism which is only upto identifying the direction of the edges (22). As we know that, for a radial network, all the edges are directed away from the source node (root node), 2-isomorphism is simply resolved after graph realization. It can also be observed that with respect to this spanning arborescence, the edges incidence on the sink nodes will become the chords.

For example, consider the network shown in Fig. 1 which has a single source and no loops. Its \(G_c\) has a spanning arborescence spanned by \(x_1, x_2, x_3\) with edges corresponding to sink flows \(x_4, x_5, x_6, x_7, x_8\) as the chords. If the f-cutset matrix corresponding to this spanning arborescence is obtained, then the original topology can be identified by applying the existing algorithm for arborescence realization (22). In Theorem 2, we show that if the f-cutset matrix \(C_f\) of \(G_c\) is transformed to a specific structure, then the corresponding spanning tree will be the desired arborescence.

Theorem 2. The spanning tree \(S_G\) of conservation graph \(G_c\) of a radial network \(G\) is an arborescence if and only if the corresponding \(C_f\) is of the form \([I_m \ C_c]\) and every element of \(C_c\) is either 0 or -1.

Proof. If part: Given \(S_G\) is an arborescence means that addition of a chord to \(S_G\) will result in a fundamental dicycle (22). Hence by definition, every element of non-identity part of f-circuit matrix, \(B_t\) is either 1 or 0 because all the branches and the chord forming a dicycle should be oriented in the same direction. It is known that \(C_c = -B_c^T\) (23). Therefore, every element of \(C_c\) is either 0 or -1.

Only if part: Given every element of \(C_c\) is either 0 or -1, it follows that every element of \(B_t\) is either 1 or 0. This indicates all the edges in each fundamental cycle are oriented in the same direction making them dicycles. If every fundamental cycle in a graph is a dicycle, then the corresponding spanning tree will be an arborescence (22).
Therefore, given that the network is radial, if \( C_f \) is obtained in the form mentioned in Theorem 2, then network topology can be identified exactly. Since \( C_f \) is obtained in the form \( \begin{bmatrix} I_m & R \end{bmatrix} \), we need a methodology to transform \( R \) into \( C_c \) containing only elements 0 and -1.

**Remark 1.** Since the chords correspond to sink flows, \( C_f \) in desired form gives a set of linear equations in which each of the non-sink flows is written in terms of the sink flows i.e., non-sink flows are the desired dependent variables \( x_d \) and sink-flows are the desired independent variables \( x_i \).

In the following section, we describe the steps to topology identification of radial network from network flow measurements.

## 2. Network Reconstruction from Flows

We assume that the flows in all the lines in a radial network are measured at \( N_s \) different steady states. In a power distribution network, the energy meters can be used to measure the incoming energy flow in the lines. In a water distribution network, flow meters can be used to measure the water flow rate in the lines.

Using these measurements, we first find a linear model which satisfies the data. Then, using Theorem 1, we find a f-cutset matrix of the network conservation graph \( G_c \). Finally, we identify the topology of the radial network following Theorem 2.

**A. Learning Linear Model.** Let the measurements at \( N_s \) different steady states be stacked into a matrix \( X_{n \times N} \) which is the only input to the problem. The objective is to identify a linear model \( A_{m \times n} \) which satisfies this data i.e., \( AX = 0 \). Note that \( X \) lies in the \( m \)-dimensional null space of \( A \) which is orthogonal to the \((n - m)\)-dimensional row space of \( A \). A set of vectors which form a basis for this row space will constitute the rows of \( A \). We propose to identify this basis in a novel manner by applying SVD on \( X \). In Theorem 3, we show that applying SVD of \( X \) gives us the subspace \( S \) in which the data lies (null space of \( A \)) and also its orthogonal complement \( S^\perp \) (row space of \( A \)).

**Theorem 3.** If data \( X \) with \( N_s \) samples lies in an \( m \)-dimensional subspace \( S \) of \( \mathcal{R}^n \), then SVD of \( X \) identifies the dimension of \( S \), an orthogonal basis for \( S \) in which \( X \) lies and also an orthogonal basis for \( S^\perp \).

**Proof.** Since the data lies in an \( m \)-dimensional subspace of \( \mathcal{R}^n \), \( X \) will be of rank \( m \) and will have \( m \) non-zero singular values and \((n - m)\) zero singular values. Therefore on finding the SVD of \( X \), the dimension of \( S \) is given by the number of non-zero singular values. This implies, given that

\[
\text{SVD}(X) = USV^T, \tag{3}
\]

\( m \) is equal to the number of non-zero elements of \( S \). It is to be noted that \( U \) is an orthogonal matrix of eigen vectors of \( XX^T \) and \( V \) is an orthogonal matrix of eigen vectors of \( X^T X \). Now based on SVD, \( X \) can be written as:

\[
X = U_1 S_1 V_1^T + U_2 S_2 V_2^T \tag{4}
\]

where \( U_1 \) is the matrix of eigen vectors corresponding to the \( m \) largest eigen values of \( XX^T \) and \( U_2 \) is the matrix of eigen vectors corresponding to the remaining \((n - m)\) smallest eigen values of \( XX^T \). Similarly \( V_1 \) and \( V_2 \) correspond to \( X^T X \). It is to be noted that these eigen values are squares of the singular values of \( X \). Since the smallest \((n - m)\) singular values of \( X \) are equal to zero, we get:

\[
U_2^T X = S_2 V_2^T = 0 \tag{5}
\]

Eq. (5) implies that \( X \) lies in the null space of \( U_2^T \) and since \( U_1^T \) is orthogonal to \( U_2^T \), \( U_1^T \) is a basis for null space in which \( X \) lies i.e., \( U_1^T \) is a basis for \( S \). Consequently \( U_1^T \) is a basis for \( S^\perp \).

Therefore, a linear model satisfying the data is obtained by determining the SVD of \( X \).
B. Obtaining f-cutset matrix through RREF. Having obtained a linear model, we now need to transform it into an appropriate form as discussed in Section 1. In Theorem 1, we showed that an f-cutset matrix can be obtained by finding a valid partition of \( x \). We can find a valid partition through trial and error by verifying if \( A_d \) is non-singular in each case. However, we propose a more elegant way of partitioning the variables. The idea is to find the Reduced Row Echelon Form (RREF) of the model matrix given by PCA. Since it is full rank matrix, the columns can be permuted to get an identity matrix in the first \( m \) columns. The first \( m \) columns then correspond to \( x_d \) and rest to \( x_i \). The RREF of a matrix is unique with respect to the row space and hence is constant even if the matrix is rotated and scaled by an invertible matrix (24). Therefore,

\[
\text{rref}(\hat{A}) = \hat{A}_r = \begin{bmatrix} I_m & R \end{bmatrix} = C_f. \tag{6}
\]

In practice, we observed that the RREF computation suffers from numerical instabilities. This can be mitigated by using both row and column pivoting.

Further, to ensure uniqueness in identification of a radial network, we have to ensure that \( C_c \) has only 0 and -1 elements as a consequence of Theorem 2. In case, this criteria is not met in the RREF step, we propose Algorithm 1 to transform \( R \) to get \( C_c \) in desired form. Following Remark 1, the objective of Algorithm 1 is to identify the non-sink flows and sink flows to get desired \( x_d \) and \( x_i \). It relies on the following implications of Theorem 2 and Remark 1:

1. Let the \( k^{th} \) equation of the obtained model be written as \( x_k + R_k x_i = 0 \), where \( R_k \) is the \( k^{th} \) row of \( R \). If all elements of \( R_k \) are 0 or -1, then \( x_k \) is a non-sink flow.

2. If there are positive elements in \( R_k \), then there exists a non-sink flow in \( x_i \) and a sink flow in \( x_d \) which need to be interchanged. Also, the non-sink flow in \( x_i \) will have larger magnitude compared to other flows in \( k^{th} \) equation.

Algorithm 1 Algorithm to get desired \( C_f \) from \( \hat{A}_r \)

| Let the non-identity part of \( \hat{A}_r \) be \( R = [R_{kj}]_{m \times (n-m)} \) |
| Let the \( k^{th} \) equation from \( \hat{A}_r \) be \( x_k + R_k x_i = 0 \) |
| for \( k \) in 1 to \( m \) do |
| if \( R_{kj} \leq 0 \quad \forall \quad j \) then |
| continue |
| else |
| Set \( L = \{ j : R_{kj} \neq 0 \quad \forall \quad j \} \) |
| \( l = \arg \max_{j \in L} x_j \) |
| Interchange \( x_k \) in \( x_d \) with \( x_l \) in \( x_i \) |
| Find \( \hat{A}_r \) with new set of \( x_d \) and \( x_i \) |

After completing the loop in Algorithm 1, we are guaranteed that \( x_d \) will contain only non-sink flows and \( x_i \) will contain only sink flows. Therefore, we have \( C_c \) in the desired form.

C. Topology identification. With \( C_c \) in desired form, we can apply a graph realization algorithm (17–19) with \( \bar{C}_c \) as input to get the undirected \( G_c \) of the network. The environment node can be separated into source and sink nodes and all the edges be directed away from the source node to get the exact original network. The directions can then be imposed to be directed away from the source node. However, the existing graph realization algorithms are complex to apply as they are general in nature while the graph, in this case, has distinct features. Therefore, we propose a simpler algorithm which directly identifies the connectivity of a radial network, given the f-cutset matrix of its conservation graph \( G_c \) corresponding to its spanning arborescence.
Algorithm 2 takes $C_c$ in desired form as input and gives the exact graph corresponding to a radial network. This algorithm works on the basis that each row of $C_c$ corresponds to a non-sink flow written as a linear combination of sink flows. Therefore, the source flow is a combination of all sink flows and linear combinations of other non-sink flows have common sink flows if they exist on the same directed path. The sink flows are connected to the directed path formed by edges indicated by respective columns in $C_c$. Algorithm 2 first identifies how the non-sink flow edges are connected followed by that of sink flow edges.

Since the dataset only labels the $n$ flows (edges), we follow a specific convention in labelling the $n + 1$ nodes: (i) as all the nodes except the source have exactly one incoming edge, each node is given the same label as the incoming edge (ii) the source node is given the label $n + 1$. This convention is followed in Algorithm 2 while reconstructing the graph. Arrays $s$ and $t$ will constitute the source and target nodes between which directed edges are connected.

**Algorithm 2** Realization of graph having spanning arborescence, from $C_c$

Initialize $s$, $t$ to be empty arrays
Sort rows of $C_c$ based on number of non-zero elements, in descending order and corresponding the labels in $x_{vars}$

$$s[1] \leftarrow (n + 1); \quad t[1] \leftarrow x_{vars}[1]$$

for $k$ in 2 to $m$ do

Set $r_1 = \{ j : C_c^{kj} \neq 0 \}$
for $p$ in $k$ to 1 do

Set $r_2 = \{ j : C_c^{pq} \neq 0 \}$
if $r_1 \cap r_2 \neq \emptyset$ then

$s[k] \leftarrow p; \quad t[k] \leftarrow x_{vars}[k]$

break

for $j$ in 1 to $(n - m)$ do

Set $L = \{ k : R_{kj} == -1 \forall j \}$

$s[m + j] \leftarrow \min L; \quad t[m + j] \leftarrow x_{vars}[m + j]$

3. Noisy Measurements

In general, the measurements are error-prone due to noise and we can model them as follows:

$$y_j = x_j + \epsilon_j \quad [7]$$
$$Y = X + E \quad [8]$$

where $\epsilon_j$ is the random error vector at instance $j$ and $E$ is the matrix of errors. It is assumed that the PCA relates assumptions are satisfied by these measurements (25). In this case, the dimension ($m$) of the desired subspace $S$ is not given by the number of zero singular values because the noisy measurements lie in a subspace that is higher in dimension to the true subspace. To be able to identify $S$ from $Y$, we propose to use PCA and its variants which work under different assumptions of $\mu$ and $\Sigma_e$. While applying PCA, we use the fact that the sample covariance matrix $S_y = \frac{1}{N}YY^T$ will have $(n - m)$ dominant eigen values and $m$ small eigen values, under the assumption that SNR is high. The following are different cases of noise distribution under which PCA can identify the $S$ in which the noise-free data lies:

**A. Homoscedastic Noise.** In this case, it is assumed that noise is i.i.d. i.e., $\mu = 0; \Sigma_e = \sigma^2 I$. In this case, the smallest $m$ eigen values of the expected covariance matrix will be equal to $\sigma^2$. This fact can be used to identify the dimension of $S$. Since we only know the sample covariance matrix $S_y$, we have to perform a hypothesis test to verify how many of the smallest eigen values of the covariance matrix are actually equal,
to ascertain the true value of $m$ (26). Further, the values in non-identity part of $\text{rref}(\hat{A})$ will not be exactly $+1, -1$ or $0$ and cannot be interpreted graphically. Hence, each of these values are rounded to their nearest neighbour among $+1, -1$ and $0$.

**B. Heteroscedastic Noise.** In this case, it is assumed that the noise variables are independent but not identical and $\Sigma_e$ is known. For this case, in (16), it has been shown that by pre-multiplying $Y$ with the inverse of Cholesky factor of $\Sigma_e$, the transformed data matrix ($Y_s$) will have i.i.d. errors. Further, the smallest $m$ eigen values of data covariance matrix will be equal to one. Thus, $Y_s$ can be used to determine the value of $m$ and a basis for $S$ after performing hypothesis testing as in previous case. It is to be noted that even in case of homoscedastic noise, this transformation can be applied to obtain $m$ smallest eigen values equal to one. This enables us to apply the same methodology in both the cases.

**C. Unknown $\Sigma_e$.** In this case, it is assumed that the noise variables are independent but $\Sigma_e$ is not known. Here, we can apply a variant of PCA called Iterative PCA (IPCA) (16). IPCA can estimate $\Sigma_e$ simultaneously with the model corresponding to the data. An important condition to be satisfied for the application of IPCA is $m(m + 1) \geq 2n$.

### 4. Methodology

In general, the following are the steps to identify the topology of a radial network from noisy measurement matrix $Y$:

1. **Case 1:** Given $Y$ and $\Sigma_e$:
   
   (a) Find the Cholesky decomposition of $\Sigma_e$ given by: $\Sigma_e = LL^T$
   
   (b) Transform the data matrix to get: $Y_s = L^{-1}Y$
   
   (c) Apply PCA on $Y_s$: $\text{SVD}(\frac{Y_s}{\sqrt{N}}) = U_sS_sV_s^T$
   
   (d) For different values $\geq 2$, perform hypothesis testing to establish the true value of $m$.
   
   (e) After establishing the value of $m$, choose the last $m$ eigen vectors of sample covariance matrix of $Y_s$ and transform them to get: $\hat{A} = U_{2s}^T L^{-1}$

2. **Case 2:** Given only $Y$:
   
   (a) Apply IPCA on $Y$ to get an estimate for both $\Sigma_e$ and $A$.

3. Find the RREF of $\hat{A}$ obtained in either case 1 or case 2.

4. Round each of the elements of non-identity part of $\text{rref}(\hat{A})$ to the nearest among $-1, 0$ and $+1$.

5. Apply Algorithm 1 to obtain f-cutset matrix in desired form.

6. Finally, apply the graph realization algorithm to obtain the topology.

In the following section, we apply this methodology to identify the topology of a simple radial network from noisy measurements.

### 5. Illustrative Example

Consider the flow network shown in Fig. 1. We generate multiple samples with noise, of all the flows in the network. We wish to recover the true connectivity from this data alone under the assumption that $\Sigma_e$ is known. The simulations are conducted as follows:
A. Data Generation. Data is generated as follows:

1. $x_4, x_5, x_6, x_7$ and $x_8$ are chosen to be independent variables and hence their measurements are generated from independent normal distributions with a mean of $\mu_d = 10$ and a variance of $\sigma_d^2 = 4$.

2. The rest of the flows ($x_1, x_2, x_3$) are obtained using linear relations.

3. A Gaussian noise is added independently to each of the flows with a variance of $\sigma_e^2 = 0.01$ resulting in an SNR of 400. Consequently, the error covariance matrix will be $\Sigma_e = \sigma_e^2 I_n$.

Accordingly, sample dataset with $N = 20$ measurements is generated. For reference, the dataset is provided in the Supplementary Information.

B. Obtaining a Linear Model using PCA. The data matrix is transformed by multiplying it with inverse of Cholesky factor of $\Sigma_e$ to get $Y_s$. PCA is applied on $Y_s$ and the following are the singular values of $\frac{Y_s}{\sqrt{N}}$:

\[
\begin{bmatrix}
655.6 & 42.61 & 22.22 & 18.23 & 16.52 & 1.14 & 0.95 & 0.77
\end{bmatrix}
\]

Now, hypothesis test is performed to determine the value of $m$, the order of the linear model. The test is performed for values $\geq 2$ in ascending order and the test fails at a value of 4 indicating that there are $m = 3$ is the true model order. A linear model corresponding to the data is then given by $\hat{A} = U^T_{2s} L^{-1}$.

C. Obtaining f-cutset matrix through RREF. Clearly, $\hat{A}$ cannot be interpreted graphically. To obtain an f-cutset matrix, we find the RREF of $A$. In this process, $x$ is partitioned into, say, $(x_1, x_6, x_8)$ as the dependent variables and $(x_3, x_7, x_5, x_4, x_2)$ as the independent variables, in that order. The corresponding $\text{rref}(\hat{A})$ is given by:

\[
\begin{bmatrix}
1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & -1 \\
0 & 0 & 1 & -1 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

This is a f-cutset matrix with edges $x_1, x_6, x_8$ forming a spanning tree but it is not in the desired form. Consequently, it can be observed that the spanning tree is not an arborescence. To get the desired $C_f$, we apply the steps in Algorithm 1 and get:

\[
C_f = \begin{bmatrix}
1 & 0 & 0 & -1 & -1 & -1 & -1 & -1 \\
0 & 1 & 0 & -1 & -1 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -1 & -1
\end{bmatrix}
\]

with $(x_1, x_2, x_3)$ as the dependent variables and $(x_4, x_5, x_6, x_7, x_8)$ as the independent variables. As shown in Theorem 2, we have identified the spanning arborescence with edges $(x_1, x_2, x_3)$.

D. Topology Identification. Finally, the network is realised by applying Algorithm 2 on $C_c$. It can be observed that $x_1$ is the source flow since it is written as a linear combination of all sink flows $(x_4, x_5, x_6, x_7, x_8)$. $x_2$ and $x_3$ are connected to the terminal node of $x_1$ since they have common edges with $x_1$ in their linear combinations. $x_4$ is connected to the directed path $x_1, x_2$ as indicated by the corresponding column in $C_f$ and similarly other sink flow edges are connected. Finally, we get the original topology shown in Fig. 1.

6. Time Complexity

In this section, we analyse the time complexity of the proposed methodology for radial networks. Since the objective is to show that the overall algorithm runs in polynomial time complexity, we derive loose bounds and restrict the analysis is to the ideal case of noise-free data. The first step is applying SVD on $X_{n \times N}$ which is shown to be of complexity $O(n^2 N + N^3)$. (27). The next step involved computing RREF of a
matrix of dimension \((m \times n)\) which is of complexity \(O(m^2n)\) \((28)\). Coming to Algorithm 1, it involves a loop with \(m\) iteration and at each iteration, atmost there are maximum and RREF operations involved. This step involves a complexity of \(O(m \times (n + m^2n))\). The final step of topology identification involves graph realization for which we proposed Algorithm 2 which is of complexity \(O(m^2)\). Since all the steps involved can be performed in polynomial time, the overall algorithm is of polynomial time complexity.

7. Simulation Results and Discussions

To corroborate the proposed method, we apply it on randomly simulated radial networks and data. Since the topology graph of any radial network is a directed tree with a root node (source), we categorise the networks into three types to study the effect of tree structure on topology identification. The types include \(\text{Binary networks}\) - networks with tree topology in which every parent node has exactly two child nodes, \(\text{Thin-Long trees}\) - networks with many tree layers but fewer nodes at each layer, and \(\text{Fat-Short trees}\) - networks with fewer tree layers but many nodes at each layer. The simulations are conducted on MATLAB 2018a as follows:

1. For each network, a dataset of \(N_s\) samples per flow variable are randomly generated as follows:

   (a) Firstly, the samples for sink flows are samples randomly from one of the three Gaussian distributions with means \((100, 200, 300)\) and standard deviations \((10, 20, 30)\), respectively.

   (b) For each of the non-sink flows, the sample values are determined by summation of sink flows incident on nodes which are descendants to the nodes on which the non-sink flows are incident.

2. Then, i.i.d. noise is added to each of the samples based on a chosen SNR value. Since, the data is drawn from multiple distributions, we calculate the SNR with respect to the least variance i.e., 10.

This simulation is repeated 100 times for each network and with different SNR values of \((100, 50, 30, 10, 5)\) and the numbers of samples \(N_s\) required (in multiples of \(n\)) for 100% accurate identification are obtained. These results are plotted in a heatmap in the Fig. 2.

The heatmap indicates that the binary networks and long-thin networks require more samples than short-fat networks for networks with number of nodes in the same range and same SNR. This can be attributed to two causes. For the similar number of nodes (i) binary networks and long-thin networks have
more number of linearly independent equations of conservation compared to short-fat networks and (ii) the cutset matrix $C_f$ is more sparse for binary networks and long-thin networks.

8. Conclusions

Reconstruction of network topology is an important problem in network science. This work deals a network reconstruction problem related to conserved networks arising in engineering, biology, and financial fields. A novel concept of conservation graphs is introduced to analyse conserved networks. Further, properties of conservation graphs are discussed from a network reconstruction viewpoint. Then, these properties have been exploited to reconstruct network using principal component analysis. A novel algorithm to reconstruct conserved networks from steady-state flow data has been proposed. The reconstruction algorithm has been corroborated with three different kinds of conserved networks arising in different fields. A comparison study in terms of signal to noise ratio in data, number of nodes, measurement required for reconstructing 100% accurate network has been performed for all three types of conserved networks.

1. Barabási AL., et al. (2016) Network science. (Cambridge university press).
2. Boccaletti S, Latora V, Moreno Y, Chavez M, Hwang DU (2006) Complex networks: Structure and dynamics. Physics reports 424(4–5):175–308.
3. Barabási AL (2009) Scale-free networks: a decade and beyond. science 325(5939):412–413.
4. Watts DJ, Strogatz SH (1998) Collective dynamics of “small-world”networks. nature 393(6684):440.
5. Amaral LAN, Scala A, Barthelemy M, Stanley HE (2000) Classes of small-world networks. Proceedings of the national academy of sciences 97(21):11149–11152.
6. Nepusz T, Vicsek T (2012) Controlling edge dynamics in complex networks. Nature Physics 8(7):568.
7. Han X, Shen Z, Wang WX, Di Z (2015) Robust reconstruction of complex networks from sparse data. Physical review letters 114(2):028701.
8. Angulo MT, Moreno JA, Lippner G, Barabási AL, Liu YY (2017) Fundamental limitations of network reconstruction from temporal data. Journal of the Royal Society Interface 14(127):20160966.
9. Li L, Xu D, Peng H, Kurths J, Yang Y (2017) Reconstruction of complex network based on the noise via qr decomposition and compressed sensing. Scientific reports 7(1):15036.
10. Sontag ED (2008) Network reconstruction based on steady-state data. Essays in Biochemistry 45:161–176.
11. Newman M (2018) Network structure from rich but noisy data. Nature Physics 14(6):542.
12. Zavlanos MM, Julius AA, Boyd SP, Pappas GJ (2011) Inferring stable genetic networks from steady-state data. Automatica 47(6):1113–1122.
13. Pappu SJ, Bhatt N, Pasumarthy R, Rajeswaran A (2018) Identifying topology of low voltage distribution networks based on smart meter data. IEEE Transactions on Smart Grid 9(5):5113–5122.
14. Materassi D, Innocenti G (2010) Topological identification in networks of dynamical systems. IEEE Transactions on Automatic Control 55(8):1860–1871.
15. Srividhya J, Crampin EJ, McSharry PE, Schnell S (2007) Reconstructing biochemical pathways from time course data. Proteomics 7(6):828–838.
16. Narasimhan S, Shah SL (2009) Model identification and error covariance matrix estimation from noisy data using pca. Control Engineering Practice 16(1):146–155.
17. Fujishige S (1980) An efficient pq-graph algorithm for solving the graph-realization problem. Journal of Computer and System Sciences 21(1):63–86.
18. Bixby RE, Wagner DK (1998) An almost linear-time algorithm for graph realization. Mathematics of Operations Research 13(1):99–123.
19. Parker S, Lohse H (1969) A direct procedure for the synthesis of network graphs from a given fundamental loop or cutset matrix. IEEE Transactions on Circuit Theory 16(2):221–223.
20. Whitney H (1933) 2-isomorphic graphs. American Journal of Mathematics 55(1):245–254.
21. Swaminathan RP, Wagner DK (1995) The arborescence-realization problem. Discrete Applied Mathematics 59(3):267–283.
22. Deo N (2017) Graph theory with applications to engineering and computer science. (Courier Dover Publications).
23. Yuster T (1984) The reduced row echelon form of a matrix is unique: A simple proof. Mathematics Magazine 57(2):93–94.
24. Narasimhan S, Bhatt N (2015) Deconstructing principal component analysis using a data reconciliation perspective. Computers & Chemical Engineering 77:74–84.
25. Jolliffe IT (1986) Principal component analysis and factor analysis in Principal component analysis. (Springer), pp. 115–128.
26. Golub GH, Van Loan CF (2012) Matrix computations. (JHU Press) Vol. 3.
27. Fraleigh J, Beauregard R (1995) Linear Algebra. (Addison Wesley).