Adaptive filtering for hidden node detection and tracking in networks
Franz Hamilton, Beverly Setzer, Sergio Chavez, Hien Tran, and Alun L. Lloyd

Citation: Chaos 27, 073106 (2017); doi: 10.1063/1.4990985
View online: http://dx.doi.org/10.1063/1.4990985
View Table of Contents: http://aip.scitation.org/toc/cha/27/7
Published by the American Institute of Physics
Adaptive filtering for hidden node detection and tracking in networks

Franz Hamilton,1,2,a) Beverly Setzer,1 Sergio Chavez,1 Hien Tran,1,2 and Alun L. Lloyd1,2,3
1Department of Mathematics, North Carolina State University, Raleigh, North Carolina 27695, USA
2Center for Quantitative Sciences in Biomedicine, North Carolina State University, Raleigh, North Carolina 27695, USA
3Biomathematics Graduate Program, North Carolina State University, Raleigh, North Carolina 27695, USA

(Received 12 March 2017; accepted 19 June 2017; published online 12 July 2017)

The identification of network connectivity from noisy time series is of great interest in the study of network dynamics. This connectivity estimation problem becomes more complicated when we consider the possibility of hidden nodes within the network. These hidden nodes act as unknown drivers on our network and their presence can lead to the identification of false connections, resulting in incorrect network inference. Detecting the parts of the network they are acting on is thus critical. Here, we propose a novel method for hidden node detection based on an adaptive filtering framework with specific application to neuronal networks. We consider the hidden node as a problem of missing variables when model fitting and show that the estimated system noise covariance provided by the adaptive filter can be used to localize the influence of the hidden nodes and distinguish the effects of different hidden nodes. Additionally, we show that the sequential nature of our algorithm allows for tracking changes in the hidden node influence over time.

Published by AIP Publishing. [http://dx.doi.org/10.1063/1.4990985]

In the study of networks, we are often interested in discovering the interactions between measured nodes, referred to as the network’s connectivity. Our ability to obtain an accurate reconstruction of connectivity is often impacted by the presence of hidden nodes. Hidden nodes are parts of the network that we have no measurements, and thus no knowledge, of. They act as unknown drivers on the observed portions of our network, resulting in incorrect connection estimates and consequently inaccurate network inference. Detecting the presence of these hidden nodes, or more specifically the parts of the network they are influencing, is thus critical. We propose an adaptive filtering-based method for the detection and tracking of hidden nodes with application to networks of neurons. By viewing hidden nodes as a problem of missing variables in the model fitting process, we show that the estimate of model error provided by the adaptive filter allows us to localize the effects of the hidden node and also track changes in its influence over time.

I. INTRODUCTION

When analyzing networks, the key to understanding the system dynamics is the accurate identification of the nodal interactions, or connectivity. Methods for reconstructing network connectivity from time series measurements of observed nodes have received considerable attention in the literature (see, for example, Refs. 1–4 and the references within). As an example, the identification of connectivity in networks of neurons is of particular interest since the network structure is known to change as a function of development and in response to various stimuli. Measurements from in vitro or in vivo neuronal networks in the laboratory are often limited to the recorded potential from a subset of active neurons in the network. The challenge is to use these limited, noisy measurements to reconstruct the network structure, several methods for which have recently developed.5–9 Within the neuroscience community, network connections can be classified as structural, functional, or effective.9–14 Structural connectivity pertains to the actual physical links within a network. Functional connectivity is defined by the presence or absence of correlation between the dynamics of any two nodes in a network (non-directional), while effective connectivity describes the directional relationship to the causation between nodes. In our neuronal network examples to follow, when we refer to connectivity, we imply structural connectivity.

The connectivity estimation problem though can be complicated by the presence of unobserved nodes within the network, so-called hidden nodes. By definition, a hidden node is a part of the network for which we have no measurements and thus its very existence is unknown to us. For example, Fig. 1 shows a schematic of a hidden node problem in a network of neurons. In this example, we assume that we have time series measurements from seven nodes in the network (black solid circles), and our goal is to identify the directed connections between nodes (black solid arrows, direction denoted by arrowhead). Complicating matters, there is an eighth node in our network which is hidden from observation (gray dashed circle) and driving an observed node (gray dashed arrow) in the network.

The presence of this hidden node introduces dynamics which are unaccounted for when conducting the analysis of this system. If we were to attempt the identification of connectivity for the network in Fig. 1, the direct effect of the hidden node influence would be errors in the estimated connectivity matrix, particularly the inference of false structural connections between the driven node and the other observed nodes.15 The removal of such false connections

a)fwhamilt@ncsu.edu

1054-1500/2017/27(7)/073106/9/$30.00 27, 073106-1 Published by AIP Publishing.
for detecting hidden nodes in neuronal networks, extending the work of Ref. 8 to include the adaptive filtering algorithm of Ref. 22. The goal of adaptive filtering is to obtain empirical estimates of the system noise (i.e., model error) and observation noise (i.e., measurement error) within the filtering framework. In Ref. 22, these estimates are iteratively updated using the filter calculated innovation at each observation.

The idea behind our method is to use the estimated system noise to identify locations in the network receiving input from the hidden nodes. While we simultaneously obtain an estimate of the observation noise as part of the algorithm, we restrict our focus to the system noise estimate as it will give us a quantification of the model error across the network model. Naturally, there will be a general degree of model error for the proposed mechanistic equations, regardless of hidden node influence, since our model only approximates the system of interest. However, we expect nodes driven by a hidden node to have a higher estimated system noise relative to locations in the network that are not affected. Equally important to the detection of a hidden node is tracking changes in its influence on the network over time. Since the adaptive filter iteratively updates its estimate of the system noise, we show that this allows us to track network nonstationarities introduced by changes in the driving strength of the hidden node. The adaptive nature of our method provides a natural way to track these changes, which can be difficult to detect using the method of Refs. 19–21 due to the required segmentation of the data (it may not be clear a priori how to segment the data to capture these nonstationarities).

We demonstrate the application of our method for hidden node detection and tracking in networks of neurons in Ref. 23. Motivated by experimental conditions in the laboratory, we work under the assumption that only noisy measurements of the neuronal potential are available from each observed node in the network. All other relevant system variables and parameters, including the network connectivity structure, are unknown quantities. While we would like to obtain an accurate reconstruction of the structural connectivity given these noisy observations, there are several hidden nodes acting on the network that we are unaware of resulting in local errors within our connectivity estimate. The goal is to detect the location and number of these hidden nodes, which will allow us to identify the errors in our estimated connectivity, as well as to track any changes in their effects on the network.

**II. NONLINEAR ADAPTIVE FILTERING**

We consider the following nonlinear system with \( n \)-dimensional state vector \( \mathbf{x} \), \( l \)-dimensional parameter vector \( \mathbf{p} \) and \( m \)-dimensional observation vector \( \mathbf{y} \)

\[
\begin{align*}
\mathbf{x}(k+1) &= \mathbf{f}(r(k), \mathbf{x}(k), \mathbf{p}) + \mathbf{w}(k) \\
\mathbf{y}(k) &= \mathbf{h}(r(k), \mathbf{x}(k), \mathbf{p}) + \mathbf{v}(k),
\end{align*}
\]

where \( \mathbf{f} \) describes the evolution of \( \mathbf{x} \) over time and \( \mathbf{h} \) is an observation operator which maps \( \mathbf{x} \) to output \( \mathbf{y} \). \( \mathbf{w} \) and \( \mathbf{v} \) are white noise processes with covariance matrices \( \mathbf{Q} \) and \( \mathbf{R} \), respectively. \( \mathbf{R} \) is the covariance of the observation noise or
measurement error, while \( Q \) is the covariance of the system noise or model error. The model parameters \( p \) may be partially known or completely unknown quantities that need to be estimated with \( x \). Note that we use discrete notation in (1) purely for notational convenience in describing the filter equations below.

Nonlinear Kalman filtering is used in a variety of applications for the estimation of system state and parameters from noisy observations; see, for example, Refs. 8, 22, and 24–35. Here, we consider the unscented Kalman filter (UKF), which approximates a nonlinear function using a deterministic ensemble of \( 2n+1 \) state vectors selected through the unscented transformation.36–38 The UKF is initialized with state \( x^0 \) and covariance \( P^0 \). At the \( k \)th step of the filter, there is an estimate of the state \( x^k \) and the covariance matrix \( P^k \). The square root of \( P^k \) is calculated and used to form the ensemble of state vectors. We denote the \( i \)th ensemble member at step \( k \) as \( x_i^k \). The ensemble can also be rescaled by weighting the members as discussed in Refs. 37 and 38, where \( W_i \) denotes the weight for the \( i \)th ensemble member.

Once formed, the ensemble is propagated forward one time step using model \( f \), and then observed with function \( h \). The weighted average of the resulting state ensemble gives the prior state estimate \( \hat{x}^k \), and the weighted average of the observed ensemble is the predicted observation \( \hat{y}^k \). Denoting the covariance matrices \( P^k \) and \( P^0 \) of the resulting state and observed ensemble, and the cross-covariance matrix \( P^y(k) \) between the state and observed ensembles, we calculate

\[
P^k = \sum_{i=1}^{2n+1} W_i (x_i^k - x^k)(x_i^k - x^k)^T + Q
\]
\[
P^0 = \sum_{i=1}^{2n+1} W_i (y_i^k - y^k)(y_i^k - y^k)^T + R
\]
\[
P^y(k) = \sum_{i=1}^{2n+1} W_i (x_i^k - x^k)(y_i^k - y^k)^T,
\]

and use the equations

\[
K(k) = P^y(k)(P^y(k))^{-1}
\]
\[
P^+(k) = P^k - P^y(k)(P^y(k))^{-1}P^y(k)
\]
\[
x^+(k) = x^k + K(k)(y^k - \hat{y}^k),
\]

to update the state and covariance estimates with the observation \( y^k \). The UKF algorithm as described earlier can be implemented for parameter estimation by considering the parameters \( p \) as auxiliary state variables with trivial dynamics, namely, \( p(k+1) = p(k) \). An augmented \( n + l \) dimensional state vector is formed consisting of the original \( n \) state variables and \( l \) model parameters allowing for simultaneous state and parameter estimation.24,35

In many situations, noise covariance matrices \( Q \) and \( R \) are unknown and have to be estimated from the data, a process known as adaptive filtering.22,39–44 Critical to our detection of hidden nodes in this filtering framework is the adaptive estimation of these noise matrices, with a particular emphasis on the estimation of \( Q \). \( Q \), the covariance of the system noise, gives us a quantification of the error in our proposed model \( f \). The closer the dynamics of \( f \) are to those of the observed system, the smaller the entries of \( Q \) should be. However, if there is a large degree of error in our proposed model, then the entries of \( Q \) should correspondingly be large to reflect this.

As such, we assume that the noise covariance matrices \( Q \) and \( R \) in (1) are unknown, and use a recently developed method for the adaptive fitting of these matrices as part of the filtering algorithm. The method uses the innovations sequence \( e(k) = y(k) - \hat{y}^k(k) \) from (3) to update the estimates \( Q(k) \) and \( R(k) \) of the covariances \( Q \) and \( R \), respectively, at step \( k \) of the filter. For further details of the estimation of these noise matrices and implementation of the algorithm, see Refs. 22 and 45.

III. RECONSTRUCTING NETWORK CONNECTIVITY IN THE PRESENCE OF HIDDEN NODES

As an example, we consider the detection of hidden nodes in networks of \( N \) Hindmarsh-Rose neurons23

\[
\begin{align*}
\dot{V}_i &= y_i - V_i^3 + b_i V_i^2 - z_i + \xi_i + \sum_{j \neq i} N \Gamma(V_j) V_j \\
\dot{y}_i &= 1 - d_i V_i^2 \\
\dot{z}_i &= 5 \times 10^{-4} \left[ s_i \left( V_i - \left( \frac{8}{5} \right) \right) - z_i \right],
\end{align*}
\]

where \( i = 1, 2, \ldots, N \). \( V_i \) corresponds to the membrane potential while \( y_i \) and \( z_i \) describe the fast and slow-scale dynamics, respectively, of neuron \( i \). \( \xi_i \) is a Gaussian random input to each neuron with mean 0 and variance of 3. Each neuron in the network has individual parameters \( b_i \sim N(3, 0.003^2) \), \( d_i \sim N(5, 0.05^2) \), and \( s_i \sim N(4, 0.04^2) \). \( \Gamma(x_j) = \beta_{ij} / (1 + 9e^{-10|x_j|}) \) is a gating function which regulates the communication between neurons in the network, where \( \beta_{ij} \) represents the connectivity coefficient from neuron \( i \) to neuron \( j \). For a network of size \( N \), we have \( N^2 - N \) possible \( \beta_{ij} \) connection parameters since we assume that a node in the network cannot be structurally connected to itself (i.e., \( \beta_{ii} = 0 \)), which is an assumption found in the literature.5–8 We note that the form of \( \Gamma(V_j) \) forces the transfer of information throughout the network only when the neurons are spiking.

Data from these networks are generated using a fourth-order Runge-Kutta method with step size \( h = 0.05 \). Connections in the network are randomly assigned with strengths drawn from a uniform distribution with mean \( \tilde{\beta} \) and variance \( \tilde{\beta} \). Similarly to experimental conditions where only the noisy potential from neurons in the network are available, we assume that our observation of the system is restricted to the \( V_i \) variables only, each of which are sampled at rate \( dt = 0.1 \) and corrupted by Gaussian observational noise with mean 0 and variance of 0.1. All other system variables and parameters are unknown quantities. Figure 2 shows example dynamics from three nodes in a network of Hindmarsh-Rose neurons.
Our observation of the network though is incomplete, in that there are an unknown number of nodes hidden from observation that are driving the network at several unknown locations. Thus, our observations consist of the variables $V_1, V_2, \ldots, V_{N^*}$, for $N^* < N$, where $N$ is the true dimension of the network. Our goal is to identify the affected locations in our network and infer the number of hidden nodes acting on our system.

To do this, we combine the data assimilation framework for estimating network connectivity as proposed in Ref. 8 with the adaptive filtering methodology presented in Ref. 22. In Ref. 8, the authors implemented a nonlinear Kalman filter for connectivity estimation in neuronal networks. They demonstrated the robustness of the filter in handling common sources of error such as noise, poorly known parameters, and model mismatch, as well as the ability to track time-varying connection parameters. We consider a similar methodology here, but incorporate the adaptive filtering framework of Ref. 22 to empirically estimate the $Q$ and $R$ noise matrices.

We assume the following assimilation model when fitting the data:

$$
\begin{align*}
\dot{V}_i &= y_i - V_i^3 + b_i V_i^2 - z_i + 2 + \sum_{j \neq i}^{N^*} \Gamma(V_j) V_j \\
\dot{y}_i &= 1 - d_i V_i^2 \\
\dot{z}_i &= 5 \times 10^{-4} \left[ s_i \left( V_i - \left( \frac{8}{5} \right) \right) - z_i \right],
\end{align*}
$$

where $\Gamma(V_j) = \beta_{ij} / (1 + 9e^{-10V_j})$. There are several important things to note about (5). Notice that in addition to estimating the state of the system, we must estimate neuron parameters $b_i, d_i$, and $s_i$ and connection parameters $\beta_{ij}$. Additionally, we do not account for the random $\xi_i$ input present in (4) in our assimilation model and its absence reflects a form of model mismatch or model error. Finally, the size of the network in (5) is determined by the number of observable nodes $N^*$, meaning that hidden nodes $N^* + 1, N^* + 2, \ldots, N$ are unaccounted for and represent missing, unmodeled variables.

In our initial example, we consider a neuronal network with 20% connectivity (meaning that 20% of possible connections are present) and mean connection strength $\beta = 2.5$. We have noisy observations of the potential from 10 nodes; however, a hidden node is driving nodes 1 and 4. Figure 3 shows the true connectivity matrix of the observed network [Fig. 3(a)] and our resulting estimation of the network structure [Fig. 3(b)], where the $i$th row corresponds to incoming connections into neuron $i$. The grayscale indicates the strength of each connection in the network. When compared to the true network connectivity, our estimation is accurate across the network except at the locations influenced by the hidden node. If we restrict our attention to the first and fourth rows of Fig. 3(b), corresponding to the estimated incoming connections to nodes 1 and 4, respectively, we notice that several false connections are identified. This is expected since the dynamics at these locations cannot be fully explained due to the hidden node influence, and as a result,
the filter tries to reconcile the model mismatch through the formation of these additional connections. This interpretation of the estimated connectivity matrix was only possible because, in this synthetic example, we had knowledge of the true connectivity structure. In experimental studies, only the estimated connectivity matrix is available and thus additional analysis as described below is required to determine the locations affected by the hidden node.

IV. DETECTING HIDDEN NODES

A natural way to detect hidden nodes in our data assimilation framework is to examine the estimate of the system noise, $Q$, provided by the adaptive filter. We can think of the hidden node as a problem of missing variables within our assimilation model. As such, portions of the network that are affected by these unknown drivers should have a higher amount of estimated noise compared to parts of the network that are unaffected. Figure 4(a) shows the total estimated system noise for each node in the network presented in Fig. 3. While the filter updates the noise estimate after each observation, we only show the final converged estimate after all the data are processed. We note that $Q$ is symmetric by definition where the diagonal entries of the matrix correspond to the estimated noise variance at each node in the network, and the off-diagonal entries quantify the covariance between the noise at each node.

Focusing on the diagonal of this matrix, we see that the estimated variances at nodes 1 and 4 are higher relative to the variances elsewhere in the network. This tells us that the dynamics at these two locations in the network are not properly accounted for, which matches our intuition given that they are being driven by the hidden node. While there is a small estimated variance at the unaffected nodes in the network, we expect this since our model does not perfectly match the dynamics of the system that generated the data. What is encouraging is that our method can distinguish between these typical sources of error and the error introduced by the unmodeled effects of the hidden node.

Figure 4(b) shows the average estimated variance for nodes in a 10 neuron network driven by a hidden node (solid black line) and unaffected by the hidden node (dashed black line) as a function of connection strength. Error bars denote standard error over 10 random network realizations. As connection strength increases, the influence of the hidden node becomes stronger and hence we see an increase in the estimated system noise variance at the affected nodes. The variance for nodes not influenced by the hidden node provides a baseline quantification of the model error present throughout the rest of the network. We see a clear distinction between affected and unaffected nodes at all connection strengths except at the lowest strength when the network is weakly connected and the effects of the hidden node are not as strong.

V. INFERRING THE HIDDEN NODE NETWORK STRUCTURE

While the estimated variance found on the diagonal of $Q$ gives us an indication of the parts of the network driven by the hidden node, the off-diagonal entries also reveal important information about our network structure. As previously mentioned, the off-diagonal entries of $Q$ tell us whether the system noise between nodes is correlated or not. Typically, we would expect the noise to be uncorrelated across the network. However, the error introduced by the hidden node is unique in that several locations in our network may be driven by the same hidden node resulting in some similarity in the error statistics. If we examine the off-diagonal entries between nodes 1 and 4 in Fig. 4(a), we see exactly this as there is a prominent estimated covariance between the noise at these two locations. Elsewhere in the network there do not appear to be any substantial covariances between the estimated noises at different nodes.

This surprising result highlights a useful capability of our proposed method. By examining the estimated covariances between parts of our network that we have identified as being driven by a hidden node, we can attempt to disentangle the number of hidden nodes acting on our network as well as distinguish the parts of the network affected by each individual

FIG. 4. (a) Estimated total system noise at each node in the network shown in Fig. 3. Grayscale indicates the level of the noise. Diagonal entries are the variance of the estimated system noise at each neuron in the network and off-diagonal entries are the covariance between the estimated noise, which tells us if the noise between any two nodes in the network is correlated or not. We observe a considerably higher estimated variance at nodes 1 and 4 (rows marked by arrows), locations receiving input from the hidden node, relative to the other nodes in the network. Additionally, we observe a covariance between the estimated process noise for neuron 1 and 4 indicating that they share a common driver. (b) Average estimated system noise variance for neurons driven by the hidden node (solid black line) and unaffected by the hidden node (dashed-dotted black line) plotted as a function of connection strength. Error bars reflect standard error calculated over 10 random network realizations, where each network has 10 observed neurons and 1 hidden node. We see that as connection strength increases and the hidden node has a stronger effect on the network, the estimated noise for the nodes it drives increases. There is a clear separation between affected nodes and those not receiving any hidden node input.
hidden node. To demonstrate this, we consider a network of 10 neurons with 2 hidden nodes. Figure 5(a) shows the true connectivity of this network and Fig. 5(b) our resulting reconstruction of the network connectivity. Here, nodes 3 and 5 are driven by the first hidden node and node 6 is driven by the second hidden node. Examining our estimated connectivity matrix, we once again see a number of false connections found in an attempt to explain the dynamics at these affected nodes. The estimated system noise for this network [Fig. 5(c)] reveals several important details. The variances at nodes 3, 5, and 6 are higher than elsewhere in the network, indicating the influence of a hidden node at these locations. Additionally, we see a strong covariance between the noise at nodes 3 and 5, which share a common link to the first hidden node, while the noise at node 6 shares no covariances with the rest of the network. We can therefore infer that node 6 must be driven by a different hidden node than the one affecting nodes 3 and 5.

The additional details about the network structure that we can infer from the estimated covariances are particularly useful when there are multiple hidden nodes or more complex interactions between the hidden nodes and the observed network. We consider two such example networks. Figure 6(a) shows the estimated system noise for a network with three hidden nodes, where the first hidden node affects nodes 1 and 2, the second hidden node affects nodes 5 and 6, and the third hidden node affects nodes 9 and 10. Again we can identify the affected nodes through the elevated noise covariance levels. Figure 6(b) shows the estimated system noise for a network with 2 hidden nodes, where the first drives nodes 3, 6, and 8 and the second drives nodes 2, 7, and 9. Even in this situation where the hidden node influences the majority of our observed network, we are still able to identify the affected location and distinguish between the different driving forces.

VI. TRACKING HIDDEN NODE INFLUENCE

Equally important to the detection of a hidden node is tracking its influence on the network. In some situations, the strength with which the hidden node drives the network may change over time, such as in the case of neuronal networks where connection strengths are known to change as a function of network development or in response to stimuli. The ability to track any changes in these hidden node effects would allow us further insight into its role on the network dynamics. The difficulty is that we have no a priori knowledge, such as a functional form, of how the hidden node influence changes over time.

Fortunately, our proposed method gives us a direct way to measure these changes with no required ad hoc modifications of its implementation. As previously mentioned, the adaptive filter provides an estimate of the noise covariances after each observation time. The sequential nature of the algorithm allows it the ability to track changes in noise levels. As such, we expect any changes in the strength with
which the hidden node drives the network to appear as a drifting noise estimate. Figure 7 shows examples of our method tracking the time-varying effects of a hidden node in two different networks. The estimated variances for an affected (black solid line) and unaffected (gray solid line) node within each network are shown. We include the noise estimate for an unaffected node to serve as a baseline comparison with which to compare the detected changes in the affected node. The black dotted lines indicate transition points in the network where the driving strength of the hidden node changes.

In the first example [Fig. 7(a)], the influence of the hidden node decreases over time. Initially, over the first 10 s of data, the driving strength of the hidden node is at its strongest, and correspondingly, we obtain a large noise estimate for the affected node. After 10 s, the driving strength decreases by half and so does the estimated noise. After 20 s, the hidden node stops affecting the network, and as such, we see the estimated noise slowly converge to the level of the unaffected node. While there is a small delay in detecting these network nonstationarities, which is to be expected since the filter needs to see enough data first, our method is able to accurately track and capture the changes in the hidden node influence in the form of a drifting noise estimate. In our second example [Fig. 7(b)], we consider a more complex model error scenario. We notice that node 4, when analyzing experimental datasets.

VII. HIDDEN NODE DETECTION UNDER LARGE DYNAMICAL MISMATCH

To test our method under conditions of significant model error caused by a mismatch in dynamics, we consider the detection of hidden nodes in networks of Hodgkin-Huxley neurons

\[
C \dot{V}_i = -g_m m_i^3 h(V - E_i) - g_2 n_i^4 (V - E_2) - g_3 (V_i - E_3)
+ 5 + \xi_i + \sum_{j \neq i}^{N} \Gamma(V_j) V_j
\]

\[
\dot{m}_i = (1 - m_i) x_m(V_i - E_0) - m_i \beta_m(V_i - E_0)
\]

\[
\dot{n}_i = (1 - n_i) x_n(V_i - E_0) - n_i \beta_n(V_i - E_0)
\]

\[
\dot{h}_i = (1 - h_i) x_h(V_i - E_0) - h_i \beta_h(V_i - E_0),
\]

where

\[
\Gamma(V_i) = \frac{\beta_g}{(1 + e^{-10(V_i - 10)})}
\]

\[
x_m(V_i) = \frac{2.5 - 0.1 V_i}{\exp(2.5 - 0.1 V_i) - 1}, \quad \beta_m(V_i) = 4 \exp \left( \frac{-V_i}{18} \right)
\]

\[
x_n(V_i) = \frac{0.1 - 0.01 V_i}{\exp(1 - 0.1 V_i) - 1}, \quad \beta_n(V_i) = \frac{1}{8} \exp \left( \frac{-V_i}{80} \right)
\]

\[
x_h(V_i) = 0.07 \exp \left( \frac{-V_i}{20} \right), \quad \beta_h(V_i) = \frac{1}{\exp(3 - 0.1 V_i) + 1}
\]

\[V_i \text{ describes the membrane potential while } m_i, n_i, \text{ and } h_i \]

describe the gating variables for the \(i\)th neuron. The parameters of the system are set to typical values: \(C = 1, g_1 = 120, g_2 = 36, g_3 = 0.3, E_0 = -65, E_1 = 50, E_2 = -77, E_3 = -54.4. \]

\[\xi_i \text{ is Gaussian random input to each neuron.} \]

In this example, we assume that noisy observations of the membrane potential, sampled at rate \(dt = 0.02\), are available from a network of 5 Hodgkin-Huxley neurons with one unobserved hidden node driving the network. We assimilate these Hodgkin-Huxley data to the Hindmarsh-Rose assimilation model described in (5), scaling the observations prior to assimilation to fit the bounds of the assimilation model. The mismatch in the dynamics between the system that generated the data, (6), and that of the assimilation model, (5), serves as a proxy for the degree of model error we might expect when analyzing experimental datasets.

Figure 8 shows the results of detecting the hidden node in this extreme model error scenario. We notice that node 4, the part of the network being driven by the hidden node, has
If network activity is fully synchronized, then its detection becomes impossible since the dynamics of each node are identical and thus indistinguishable from one other. Furthermore, if a hidden node acts on the entire network, then its detection becomes impossible since the estimated noise at each node would be similar.

We view this work as a first step towards a more robust analysis of networks with hidden observations. While the focus here was solely on detection and tracking, the next step is to reconstruct the hidden node’s driving signal. This will likely involve a two-stage procedure where the first step is the analysis conducted here to detect the number of and distinguish the effects of the hidden nodes. Once these locations are determined, the assimilation model can be expanded to include these missing nodes with proper placement in the network. The data can then be fit to this refined model in an attempt to get an approximation of the unknown dynamics, all the while using our calculated noise statistics to evaluate the error in our model. Additionally, application of our method to experimental data analysis is the subject of future work.

ACKNOWLEDGMENTS

This research was supported by Grant No. RTG/DMS-1246991 from the National Science Foundation. We thank Timothy Sauer and Tyrus Berry for helpful comments and discussion.

FIG. 8. Results from detecting a hidden node in a network of 5 Hodgkin-Huxley neurons. While there is an extreme dynamical mismatch between the dynamics of the Hodgkin-Huxley data and the dynamics of our assimilation model (Hindmarsh-Rose), our method is still able to identify the part of our network influenced by the unknown driver. The affected node (node 4) has an estimated noise twice as large as the noise at the unaffected nodes.

VIII. CONCLUSION

Our proposed methodology for hidden node detection offers several advantageous features when conducting network analysis. The adaptive filtering framework allows us to empirically separate observation error and system error from one another, the latter of which is a natural measure for quantifying the error introduced by the unmodeled dynamics of the hidden node. We demonstrated that the system noise covariance could be used to not only detect which parts of the network are being driven by a hidden node, but it could also be used to distinguish the influence of multiple hidden nodes. Furthermore, our method allows for tracking hidden node influence that may change over time resulting in network nonstationarities. While we demonstrated our method on neuronal networks, its application to networks in general is possible provided that an approximate model of the nodal dynamics is available.

The success of our algorithm for hidden node detection is subject to several factors. In the considered examples, the issue of network observability (i.e., the observations required to reconstruct the network state) was circumvented since the hidden nodes represented a small percentage of the total nodes within the network. However, this question of observability is critical in network analysis and recent work explored the conditions under which the network state can be reconstructed, in particular, deriving the minimum number of observable nodes required for accurate reconstruction. If too few nodes are observed, then reconstruction of the network state and detection of the hidden nodes would suffer. Recent literature examined the role network symmetry plays in synchronizing networks or making the network observable. If network activity is fully synchronized, then detection, let alone connectivity reconstruction, becomes impossible since the dynamics of each node are identical and thus indistinguishable from one other. Furthermore, if a hidden node acts on the entire network, then its detection becomes impossible since the estimated noise at each node would be similar.

We view this work as a first step towards a more robust analysis of networks with hidden observations. While the focus here was solely on detection and tracking, the next step is to reconstruct the hidden node’s driving signal. This will likely involve a two-stage procedure where the first step is the analysis conducted here to detect the number of and distinguish the effects of the hidden nodes. Once these locations are determined, the assimilation model can be expanded to include these missing nodes with proper placement in the network. The data can then be fit to this refined model in an attempt to get an approximation of the unknown dynamics, all the while using our calculated noise statistics to evaluate the error in our model. Additionally, application of our method to experimental data analysis is the subject of future work.

ACKNOWLEDGMENTS

This research was supported by Grant No. RTG/DMS-1246991 from the National Science Foundation. We thank Timothy Sauer and Tyrus Berry for helpful comments and discussion.

30S. Schiff, Neural Control Engineering (MIT Press, 2012).