Topology identification of networks with rank-reduced process noise

J.B.T. Meijer

***Abstract*— Multiple identification techniques are available for modeling networks consisting of nodes, i.e. topology detection where there is either a connection or not. During topology identification it is normally assumed that every node in the network has its own noise source. However it is possible that there are nodes sharing their noise, also known as rank-reduced noise. In this work is determined what the performance influence of rank-reduced noise is on Granger and Bayesian topology identification algorithms. Next the SLS algorithm is tested that is designed with rank-reduced noise in mind. Furthermore an algorithm is designed for doing topology identification in rank-reduced noise networks.**

I. INTRODUCTION

Determining the interconnection structure of brain networks using data is gaining popularity as it gives insight into how the brain processes information, useful for treatment of brain diseases i.e. epilepsy. The brain is complex since many areas are interconnected and additionally, the low measurement quality of the fMRI scans and the absence of external excitations all increase the difficulty for estimating how different parts of the brain interact.

During earlier research the Mozart effect is studied to determine the effect of listening to music on the brain network topology [2]. In this study fMRI scans are made where the activity in different brain regions is determined as time series. Each brain region is then modeled as a node in a dynamical network.

The topology of this network was identified using a Granger algorithm as described in [5], however no relation was found between listening to Mozart music and changes in brain network topology. Next the Bayesian algorithm was used with the same dataset resulting in topology changes as described in [2]. The variance of the estimated network is still considerable, resulting in the objective to reduce the variance of the estimation. The goal of this research is to determine if topology identification of rank-reduced process noise networks is beneficial for this situation. For identification of networks with rank-reduced process noise the Sequential Least Squares (SLS) algorithm as described in [1] is used.

First the definition and notation of a dynamical network is presented. Then the three algorithms (Granger, Bayesian and SLS) used for this research are introduced, which are benchmarked in MATLAB simulations afterwards using a network with rank-reduced process noise. Furthermore an algorithm is designed for topology identification of rank-reduced networks, since SLS does not do topology identification.

II. DYNAMIC NETWORK DEFINITION

Following the network definition as stated in [1], a dynamic network is defined by L scalar *nodes* $w_j(t)$, $j = 1, ..., L$, and K *external variables* r_k, $k = 1, ..., K$. Each node is described as in equation (1).

$$w_j(t) = \sum_{\substack{t=1 \\ l \neq j}}^{L} G^0_{jl}(q) w_l(t) + \sum_{k=1}^{K} R^0_{jk}(q) r_k(t) + v_j(t) \quad (1)$$

where

- q^{-1} is the delay operator, i.e. $q^{-1} w_j(t) = w_j(t-1)$;
- $G^0_{jl}(q)$ is a strictly proper rational transfer function referred to as a *module* in the network;
- R^0_{jk} are proper rational transfer functions, being the (j,k) elements of R^0;
- r_k are *external variables* that can directly be manipulated by the user.

The full network is constructed by combining all nodes:

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} = \begin{bmatrix} 0 & G^0_{12} & \cdots & G^0_{1L} \\ G^0_{21} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & G^0_{L-1} \\ G^0_{L1} & \cdots & G^0_{L\,L-1} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix} + R^0 \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_K \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_L \end{bmatrix} \quad (2)$$

denoted by the matrix equation $w = G^0 w + R^0 r + v$.

The *process noise* is denoted v_j, where the vector process $v = [v_1 \cdots v_L]^T$ is modeled as a stationary stochastic process with rational spectral density, such that there exists a white noise process $e := [e_1 \cdots e_L]^T$ such that $v(t) = H^0(q)e(t)$ with $H^0 \in R^{L \times L}(z)$ as a proper rational transfer function matrix which is monic, stable and stably invertible. The process noise is called *rank-reduced* or *singular* if $p < L$, and a node j is called *noise-free* if $v_j(t) = 0$ for all t. Furthermore v results in the power spectrum matrix Φ_v which is singular if $p < L$ so it can be modelled as rank-reduced white noise.

III. IDENTIFICATION ALGORITHMS

The used identification algorithms used for this project are:

- Granger algorithm [5]
- Bayesian algorithm [4]
- Sparse plus low rank (SLS) algorithm [1]

Below a short introduction for each algorithm is given.

A. Granger algorithm

The Granger algorithm is based on Granger causality and is intended for topology identification, resulting in a connection between a node (1) or no connection (0). The dynamics between nodes are not identified. Also this algorithm does not consider rank-reduced noise. The MVGC Multivariate Granger Causality Matlab Toolbox v1.0 was used during the algorithm benchmark. Below is briefly explained how it works.

The MVGC algorithm calculates G-causalities from empirical time series data using the following steps:

1) Model order estimation through AIC or BIC by fitting a VAR model to time series data
2) Estimating Vector Autoregressive (VAR) model parameters (A_k, Σ) for the selected model order
3) Autocovariance calculation Γ_k from estimated VAR model using reverse Yule-Walker equation with VAR model parameters (A_k, Σ) as inputs

The pth order VAR model has the form of equation (3) where A_k consists of the regression coefficients and ϵ_t the residuals. The residuals covariance matrix is denoted as $\Sigma \equiv \operatorname{cov}(\boldsymbol{\varepsilon}_t)$. Furthermore U_t is a matrix of time series.

$$U_t = \sum_{k=1}^{p} A_k \cdot U_{t-k} + \varepsilon_t \tag{3}$$

Next the autocovariance sequence Γ_k for a covariance-stationary (not necessarily VAR) stochastic process U_t is defined as in equation (4).

$$\Gamma_k \equiv \operatorname{cov}(U_t, U_{t-k}) \quad k = \ldots, -2, -1, 0, 1, 2, \ldots \tag{4}$$

The autocovariance sequence is computed using the Yule-Walker equation, shown in equation (5).

$$\Gamma_k = \sum_{\ell=1}^{p} A_\ell \Gamma_{k-\ell} + \delta_{k0}\Sigma \tag{5}$$

Since for the reverse Yule-Walker equation Γ_k must be invertible it is important that it is nonsingular.

B. Bayesian algorithm

The Bayesian algorithm is also intended for topology identification for full rank noise cases, determined by using a forward-backward search algorithm as described in [4]. The MATLAB implementation from this paper is used during the algorithm benchmark.

C. SLS algorithm

The SLS algorithm estimates dynamics of a MIMO network with rank-reduced noise, as described in [3]. Since this algorithm is important for this research, first is briefly explained how it works.

When Φ_v is singular it can be modelled as in equation (6). Here the noise sources v are split in two groups: v_a and v_b. Here v_a consists of all independent noise sources e_a and v_b consists of copies from noise sources e_b. Then the transfer matrix H can be modelled in three different ways, each with their own properties.

1) Noise models: Three different noise models are used for H: H^0, H_s^0 and H_d^0. The structure of these noise models is shown in equation (6).

$$\begin{bmatrix} v_a \\ v_b \end{bmatrix} = \underbrace{\begin{bmatrix} H_a^0 \\ H_b^0 \end{bmatrix}}_{H^0} e_a = \underbrace{\begin{bmatrix} H_a^0 & 0 \\ \tilde{H}_b^0 & I \end{bmatrix}}_{H_s^0} \begin{bmatrix} e_a \\ e_b \end{bmatrix} = \underbrace{\begin{bmatrix} H_a^0 & 0 \\ 0 & I + \tilde{H}_b^0 \Gamma^{0\dagger} \end{bmatrix}}_{H_d^0} \begin{bmatrix} e_a \\ e_b \end{bmatrix} \tag{6}$$

where H^0 is split in H_a^0 and H_b^0 and multiplied with all independent noise sources in e_a to generate v_a and v_b. H_s^0 is comparable to H^0 however now $\tilde{H}_b^0 := H_b^0 - \Gamma^0$, where $\Gamma^0 =: \lim_{z\to\infty} H_b^0$, is implemented in order to make H^0 monic, stable and stably invertible. Also the copies of noise sources e_b are now added. Thirdly there is H_d^0 that allows for a (block) diagonal model which can be useful for implementation. This results in the two matrices H_s^0 and H_d^0 that are invertible, which is needed for the one-step-ahead predictor discussed later.

Converting H^0 to H_s^0 can be proven using the relationships $\tilde{H}_b^0 = H_b^0 - \Gamma^0$ and $\Gamma^0 e_a = e_b$, since $v_b = \tilde{H}_b^0 e_a + e_b$ can be rewritten to $v_b = (H_b^0 - \Gamma^0)e_a + e_b = H_b^0 e_a - \Gamma^0 e_a + e_b$ and finally $v_b = H_b^0 e_a$. The term for v_a remains the same for both notations.

Converting H^0 to H_d^0 is done using relationship $\Gamma^{0\dagger} e_b = e_a$, derived from $\Gamma^0 e_a = e_b$. This conversion is only possible if the left inverse $\Gamma^{0\dagger}$ of Γ^0 exists, which is true if $\Gamma^0 \in R^{m\times n}$ has rank n. Then $v_b = (I + \tilde{H}_b^0 \Gamma^{0\dagger})e_b$ can be rewritten to $v_b = Ie_b + H_b^0 \Gamma^{0\dagger} e_b - \Gamma^0 \Gamma^{0\dagger} e_b$, where $\Gamma^0 \Gamma^{0\dagger}$ becomes I resulting in $v_b = H_b^0 \Gamma^{0\dagger} e_b$. Since $\Gamma^{0\dagger} e_b = e_a$, this results in $v_b = H_b^0 e_a$ proving that H^0 and H_d^0 are equal.

2) Identifcation methods: For identification purposes the one-step-ahead predictor of equation (7) is used. As shown a invertible H^0 is needed, thus H_s^0 or H_d^0 are needed. In this case H_s^0 is used.

$$\hat{w}(t|t-1) = w - \left(H_s^0(q)\right)^{-1} \left(\left(I - G^0(q)\right) w - R^0(q) r\right) \tag{7}$$

This one-step-ahead predictor is then used in the prediction error shown in equation (8), where a division between nodes with an independent noise source and with a copied noise source is made.

$$\varepsilon(t,\theta) = \begin{bmatrix} \varepsilon_a(t,\theta) \\ \varepsilon_b(t,\theta) \end{bmatrix} := \begin{bmatrix} w_a(t) \\ w_b(t) \end{bmatrix} - \begin{bmatrix} \hat{w}_a(t|t-1,\theta) \\ \hat{w}_b(t|t-1,\theta) \end{bmatrix} \tag{8}$$

Finally the prediction error is used in the optimization problem as shown in (9).

$$\theta^\star = \arg\min_\theta E\left\{\varepsilon_a^T(\theta) Q_a \varepsilon_a(\theta) + \lambda Z(\theta)\right\} \tag{9}$$

where $Z(\theta) = (\Gamma(\theta)\varepsilon_a(t,\theta) - \varepsilon_b(t,\theta))^T (\Gamma(\theta)\varepsilon_a(t,\theta) - \varepsilon_b(t,\theta))$. Here for $\lambda \to \infty$ the unconstrained criterion is equivalent to the constrained criterion. For now $\lambda = \infty$ which might not be the optimal value.

IV. ALGORITHM BENCHMARK

To test the performance of the different dynamic network identification algorithms in a situation with rank-reduced noise, a random network with 5 nodes and a variable number (2 to 5) of independent noise sources is generated in MATLAB. The transfer functions between the nodes are of 2^{nd} order and no external excitations r are present.

Using this network and white noise sources (variance $\sigma = 1$), time series data is generated for all nodes with a length of 1000 samples. These time series data signals are then inserted into the three algorithms to estimate the topology of the generated network. Next the estimated network is compared to the generated network, resulting in a True Positive Rate (TPR) and False Positive Rate (FPR) which are used to evaluate performance.

The TPR expresses the ratio of correctly estimated connections between nodes. The FPR indicates the ratio of connections between nodes where a link was estimated but where in the generated network is no link.

A. Comparing SLS to Granger and Bayesian algorithms

Both Granger and Bayesian algorithms identify the topology of a network, resulting in a connection (1) or no connection (0) between nodes. The SLS algorithm identifies the dynamics resulting in transfer functions between nodes. In order to compare these, the 2-norm of the magnitude of each transfer function is determined and compared to a predefined threshold. If the 2-norm is above the threshold, a connection is detected. This threshold is set to 10 during experiments in the full-rank noise case, based on comparable TPR ratios of the SLS algorithm compared to the Granger and Bayesian algorithms. The effect of different threshold values on the TPR and FPR is shown in Figure 1.

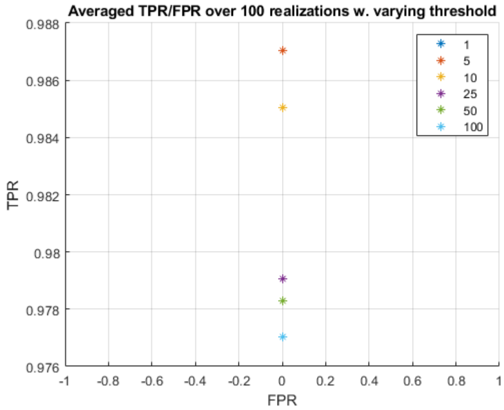

Fig. 1. SLS threshold tuning

B. Simulation results

First a baseline is created using a full rank noise process, every node has its own white noise process. The results of this comparison are shown in Figure 2 where the low FPR value of the SLS algorithm is visible and the TPR ratios are comparable. Next the noise rank is reduced from 5 to 4, resulting in a rank-reduced noise process. The results are shown in Figure 3. Reducing the order of the noise process even further to 3 resulted in Figure 4 and to 2 in Figure 5.

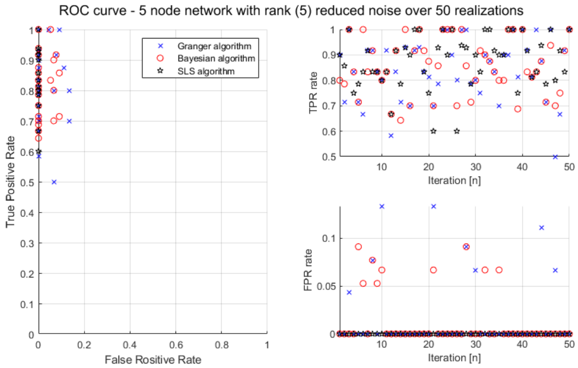

Fig. 2. Comparison of different algorithms using a 5 node network with full rank noise over 50 realizations

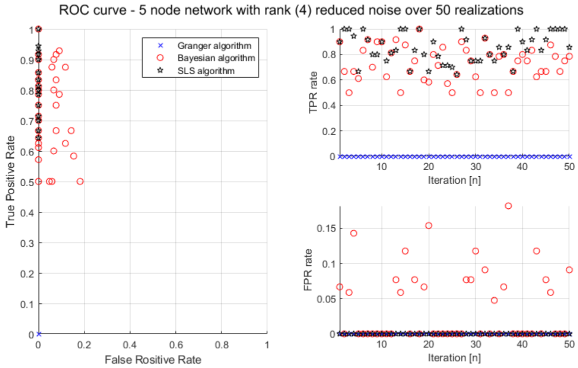

Fig. 3. Comparison of different algorithms using a 5 node network with rank (4) reduced noise process over 50 realizations

Furthermore the performance is evaluated using the distance between the TPR and FPR for each realization, which is computed using $distance = \sqrt{FPR^2 + (1 - TPR)^2}$ where a smaller value implies better performance. The averages of these values for networks with different noise ranks are shown in Table I. The Granger algorithm is not able to identify the rank-reduced systems because the autocovariance matrix Γ_k is singular, thus not invertible. This is solved by enabling an input r, which is done in the next step.

C. Enabling input r

An input is added to determine if the Granger algorithm is able to identify rank-reduced process noise networks if there is an input. A stochastic white noise signal with variance $\sigma = 1$ is added to each individual node. Results of these

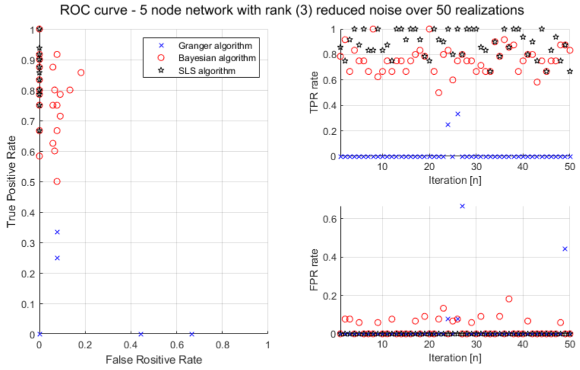

Fig. 4. Comparison of different algorithms using a 5 node network with rank reduced (3) noise process over 50 realizations

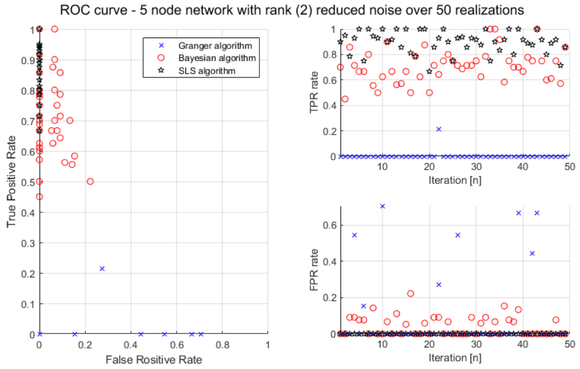

Fig. 5. Comparison of different algorithms using a 5 node network with rank reduced (2) noise process over 50 realizations

simulations are shown in Figure 6, 7, 8 and 9. Also Table II shows the averaged distance results.

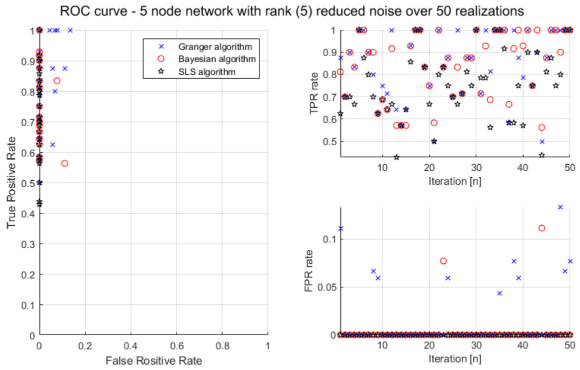

Fig. 6. Comparison of different algorithms using a 5 node network with full rank noise over 50 realizations with input enabled

TABLE I
DISTANCE AVERAGED OVER 50 REALIZATIONS WITH VARYING PROCESS NOISE RANK

Algorithm	rank 2	rank 3	rank 4	rank 5
Granger	0.981	1	1.005	0.143
Bayesian	0.335	0.268	0.274	0.128
SLS	0.197	0.254	0.239	0.242

TABLE II
DISTANCE AVERAGED OVER 50 REALIZATIONS WITH VARYING PROCESS NOISE RANK - INPUT R ENABLED FOR ALL NODES

Algorithm	rank 2	rank 3	rank 4	rank 5
Granger	0.204	0.182	0.181	0.157
Bayesian	0.179	0.175	0.169	0.153
SLS	0.246	0.210	0.241	0.240

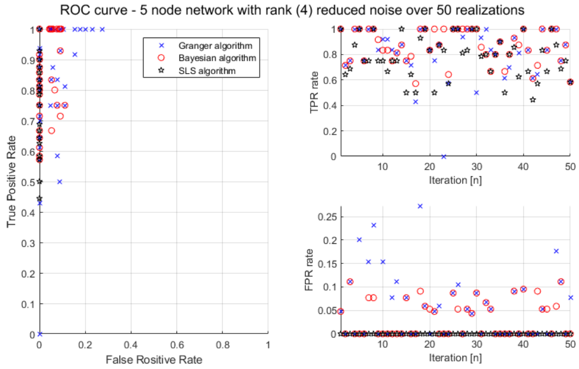

Fig. 7. Comparison of different algorithms using a 5 node network with rank (4) reduced noise process over 50 realizations with input enabled

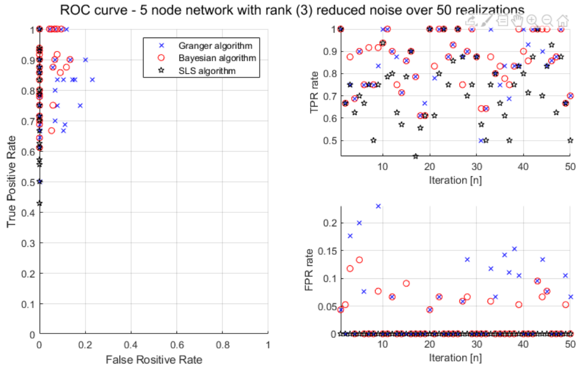

Fig. 8. Comparison of different algorithms using a 5 node network with rank reduced (3) noise process over 50 realizations with input enabled

D. Conclusion

Performance of the algorithms is based on the defined distance. The SLS algorithm outperforms the other two algorithms in every rank-reduced noise situation. In the full rank noise case the Granger and Bayesian algorithm perform better, however this can also be influenced by selecting a different threshold. Lowering the set threshold results in a lower TPR while the FPR remains the same, resulting in a lower distance.

Next to that, the Granger algorithm is almost never able to estimate a network if the noise rank is reduced. This is due to the autocovariance matrix being singular, thus not invertable. For the full rank noise situation the Granger and Bayesian algorithms perform comparable.

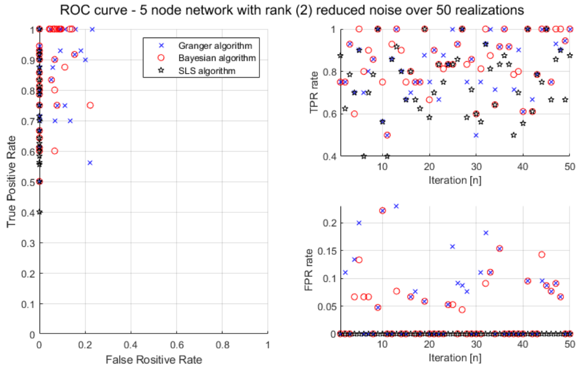

Fig. 9. Comparison of different algorithms using a 5 node network with rank reduced (2) noise process over 50 realizations with input enabled

Furthermore must be noted that the SLS algorithm is created with a rank-reduced case in mind, while the Granger and Bayesian algorithms are designed for full rank noise processes. No research is done to find one of these algorithms adjusted for a rank-reduced case.

V. Rank-reduced topology identification

An algorithm for rank-reduced topology identification is defined and validated on a defined test network using MATLAB simulations.

Assumed is that the structure of the noise model H is known beforehand and G is unknown. Now H is splitted in nodes that have their own independent noise source (H_a) and nodes that have multiple noise sources (H_b). This structure is comparable to H^0 as in equation (6). Also is assumed that all noise sources are connected to at least one node where it is the only noise source.

With the nodes from H_a MISO identification is now possible. After identifying the transfer functions of G towards this node, it is possible to estimate the noise source e.

A. Test network

To validate this approach, a simple test network is defined using three nodes, two independent error sources and one reference signal. This network is shown in Figure 10 and equation (10).

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 & G^0_{12} & G^0_{13} \\ G^0_{21} & 0 & G^0_{23} \\ G^0_{31} & G^0_{32} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} + \begin{bmatrix} r_1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} H_{11} & H_{21} \\ 0 & H_{22} \\ H_{13} & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{10}$$

where e_1 and e_2 are white noise signals with variance $\sigma = 0.1$ and r_1 is white noise with variance $\sigma = 1$ and data length N is set to 1000 samples.

The transfer functions of the network are:

- $G^0_{21}(q) = G^0_{31}(q) = \dfrac{-0.226q^{-1} - 0.161}{q^{-2} - 0.969q^{-1} + 0.367}$,

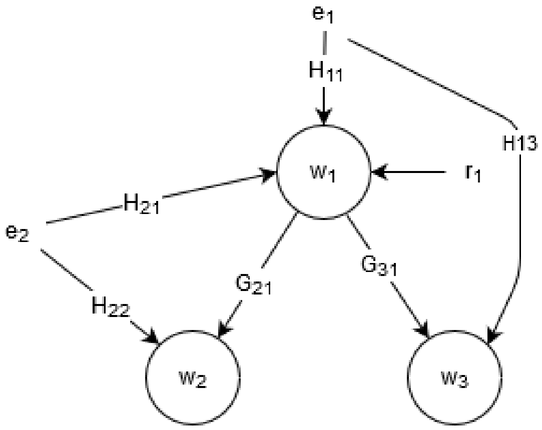

Fig. 10. Three node rank-reduced noise test network

- $G^0_{12}(q) = G^0_{13}(q) = G^0_{23}(q) = G^0_{32}(q) = 0$,
- $H^0_{13}(q) = H^0_{22}(q) = 1$,
- $H^0_{11}(q) = H^0_{21}(q) = \dfrac{1}{q^{-2} - 0.969q^{-1} + 0.367}$

Using this network measurement data is generated after which the network is identified.

B. Identification

Identification is done using a MISO approach with the assumption that the structure of the noise model H is known beforehand while G is unknown. From the diagonal elements in H the error sources are then estimated, for the example network node w_2 and w_3. This results in noise source estimates $\hat{e_1}$ and $\hat{e_2}$. G is expected to be full and the identification problem becomes:

1) $w_2 = G_{21}w_1 + G_{23}w_3 + H_{22}e_2$
2) $w_3 = G_{31}w_1 + G_{32}w_2 + H_{13}e_1$
3) $w_1 = G_{12}w_2 + G_{13}w_3 + H_{11}e_1 + H_{21}e_2$

Starting with node w_2 the transfers $\hat{G}_{21}$ and $\hat{G}_{23}$ are identified using a 4th order ARX model. Here the output is node w_2 and the inputs are w_1 and w_3, resulting in the estimated transfer functions $\hat{G}_{21}$ and $\hat{G}_{23}$. Using these estimations noise signal e_2 is estimated:
$\hat{e}_2 = w_2 - \hat{G}_{21}w_1 + \hat{G}_{23}w_3$. The same approach is used for node w_3 resulting in $\hat{G}_{31}$, $\hat{G}_{32}$ and $\hat{e}_1$. Now all error sources are estimated the nodes with multiple error sources, w_1 in this example, can be estimated.

First we determine $\bar{w}_1 = w_1 - r_1 - H_{11}\hat{e}_1 - H_{21}\hat{e}_2$, which is then used as output for the MISO identification problem where w_2 and w_3 are the inputs. This results in the estimates $\hat{G}_{12}$ and $\hat{G}_{13}$. Now all entries of $\hat{G}$ are determined, resulting in an estimation of the dynamics between the nodes. These dynamics are then converted using the 2-norm approach as with the SLS algorithm, during experiments the threshold is set to 5. The TPR/FPR values of different threshold values are shown in Figure 11, showing that a threshold of 5 is resulting in the lowest distance.

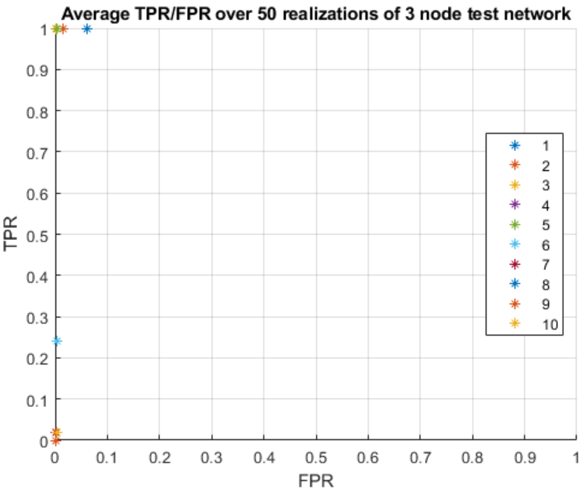

Fig. 11. Rank-reduced topology identification threshold tuning

C. *Simulations*

The performance of this identification approach is validated in MATLAB by creating multiple realizations of the test network and identifying it. Performance is then evaluated by determining the TPR and FPR distance.

After simulating 50 realizations this resulted in an average TPR of 1, average FPR of 0.0014 thus a distance of 0.0014. Lowering the sample length to $N = 50$ resulted in an TPR of 0.74, FPR of 0.1371 and distance of 0.2940.

VI. CONCLUSIONS

Comparing the Granger, Bayesian and SLS algorithms for topology identification of rank-reduced process noise networks resulted in the following conclusion. If there are no inputs present to manipulate the dynamic rank-reduced noise network, SLS outperforms the Granger and Bayesian algorithms. In future work the rank-reduced topology identification can be applied to the fMRI data for a more accurate topology estimation. If the noise in this network is rank-reduced, this will lead to a better estimated topology.

Furthermore an algorithm for rank-reduced process noise topology identification is proposed and validated using simulations. For the given test network this algorithm shows good performance.

REFERENCES

[1] H.M. Weerts, P.M.J. van den Hof, A.G. Dankers, ”Identification of dynamic networks with rank-reduced process noise”, 2017.

[2] R.J.C. van Esch, ”Toplology Detection in Brain Networks”, August 2019.

[3] M. Zorzi, A. Chiuso, ”Sparse plus low rank network identification: A nonparametric approach”, August 2016.

[4] S. Shi, P.M.J. van den Hof, G. Bottegal, ”Bayesian topology identification of linear dynamic networks”, November 2019.

[5] A. L. Dejong, ”Priming the epileptic brain; does Mozart affect cerebral networks?”, 2018.