

Article

Undirected Graphs: Is the Shift-Enabled Condition Trivial or Necessary?

Liyan Chen ^{1,†} , Samuel Cheng ^{1,3,†,*}, Kanghang He ⁴, Lina Stankovic ⁴ and Vladimir Stankovic ⁴¹ College of Electronic and Information Engineering, Tongji University; chenliyan@tongji.edu.cn² Key Laboratory of Oceanographic Big Data Mining & Application of Zhejiang Province, Zhejiang Ocean University³ the School of Electrical and Computer Engineering, University of Oklahoma⁴ the Department of Electronic and Electrical Engineering, University of Strathclyde

* Correspondence: samuel.cheng@ou.edu

† 4800 Caoan Road, Shanghai, P.R. China

Abstract: It has recently been shown that, contrary to the wide belief that a shift-enabled condition (necessary for any shift-invariant filter to be representable by a graph shift matrix) can be ignored because any non-shift-enabled matrix can be converted to a shift-enabled matrix, such a conversion in general may not hold for a directed graph with non-symmetric shift matrix. This paper extends this prior work, focusing on undirected graphs where the shift matrix is generally symmetric. We show that while, in this case, the shift matrix can be converted to satisfy the original shift-enabled condition, the converted matrix is not associated with the original graph, that is, it does not capture anymore the structure of the graph signal. We show via a counterexample, that a non-shift-enabled matrix cannot be converted to a shift-enabled one and still maintain the topological structure of the underlying graph, which is necessary to facilitate localized signal processing.

Keywords: graph signal processing; shift-enabled graphs; shift-invariant filter; undirected graph

1. Introduction

Graph signal processing (GSP) extends classical digital signal processing (DSP) to signals on graphs, and provides a prospective solution to numerous real-world problems that involve signals defined on topologically complicated domains, such as social networks, point clouds, biological networks, environmental and condition monitoring sensor networks [1]. However, there are several challenges in extending classical DSP to signals on graphs, particularly related to the design and application of graph filters.

In classical, one-dimensional DSP, any linear, time-invariant, or shift-invariant, filter that commutes with time shift operator z^{-1} can be represented as a polynomial of z^{-1} leading to Z-transform of the filter. Conversely, if a linear filter can be represented as a polynomial of z^{-1} , the filter is linear and shift-invariant. Unfortunately, this concept does not simply generalize to GSP, partly because the definition of a “shift” for a graph is not obvious [2]. Commonly, in the GSP literature, a graph is uniquely described by a “shift” matrix or a “shift” operator¹, S [3–5], which has been extensively used for time/vertex-domain filter design (see [1], [2] and references therein for frequency-domain and time/vertex-domain filtering). For example, adjacency matrix, for general graphs, and Laplacian matrix, for undirected graphs, are some popular choices for the shift matrix.

In order to make graph filtering feasible, even for very large graphs, it is necessary to perform the filtering operation locally. For example, consider a sensor network represented by a graph, where the edges and edge weights of the graph depend on the distance between the sensors. In this case, efficient

¹ The term “shift” comes from the analogy with z^{-1} operator in Z-transform of classical DSP.

31 filtering boils down to merely mixing the signals acquired by a sensor with those of the nearest sensors.
32 Otherwise, if the filter output at any graph vertex is a linear combination of inputs at *all* vertices,
33 filtering will be practically infeasible for “big data” graphs [5]. Therefore, in practice, we expect that a
34 node can only impose direct influence to an adjacent node through the shift operator. For practical
35 design purposes, it is advantageous to be able to decompose filters in a form of polynomial of such a
36 shift matrix. The importance of this polynomial representation has been reiterated in a recent survey
37 paper (see Section II.F of [1]).

38 Although a nice, but loose, analogy between S and z^{-1} can be established [1], unlike classical
39 DSP, if a graph filter is shift-invariant (the shift matrix commutes with the target filter), this does not
40 automatically imply that a polynomial representation of the filter exists [6]. Ref. [3] argues that, for
41 any shift matrix S , there exists a converted shift matrix \tilde{S} such that graph filter H is a polynomial in
42 \tilde{S} . However, it is not sufficient just to have H to be represented as a polynomial of any arbitrary \tilde{S} .
43 *One should also ensure that \tilde{S} indeed describes the same graph as S* (see details in Definition 2), that is, the
44 converted graph shift should keep the same topological structure as the original one.

45 1.1. Contribution

46 It was shown in [3] that any filter commuting with shift matrix S can be represented as a
47 polynomial in S provided that the characteristic and minimal polynomial of the shift matrix are
48 equal (in the rest of this paper, as in [7], we will refer to this condition as *shift-enabled condition*, see also
49 Definition 1). However, in [3], this condition was immediately disregarded, surmising that one may
50 convert any shift matrix that does not satisfy the shift-enabled condition into one that does. Based on
51 this conclusion, most researchers currently assume that the shift-enabled condition simply holds or
52 ignore the condition completely. However, it was proved in [7], through a counterexample, that such a
53 conversion may not hold for a directed graph with asymmetric shift matrix.

54 In this paper, we focus on undirected graphs, which have wider applications [2], and illustrate
55 with examples that when the symmetric shift matrix of an undirected graph is non-shift-enabled, the
56 conversion suggested in [3] could lead to a very different graph that does not necessarily capture the
57 structure of the original graph signal. Namely, though the conversion would provide a shift-enabled
58 graph that facilitates polynomial representation of the shift-invariant filters, the newly designed graph
59 might no longer capture the structure of the graph signal it was originally designed to model², and
60 does not facilitate performing filtering locally.

61 Referring to our wireless sensor network example in the introduction, in the original graph the
62 output of the filtering at each vertex only involves inputs of the vertex’s immediate neighborhoods.
63 However, in the converted graph, sensors that are far apart might be strongly connected, that is, each
64 output at a vertex could be a linear combination of inputs at almost all vertices, thus filtering in such
65 converted graph will be computationally unaffordable for “big data” graphs in practice which further
66 emphasizes the importance of the shift-enabled condition [7].

67 The outline of the paper is as follows. Section 2 describes the basic concepts and key properties of
68 a shift-enabled graph. Section 3 provides counterexamples to prove that the shift-enabled condition is
69 essential for the symmetric graph. Section 4 concludes the paper.

70 2. Basic Concepts and Properties of Shift-enabled Graphs

71 In this section, we briefly review the concepts of shift-enabled graphs and their properties relevant
72 to this paper. For more details, see [2–5].

² Note that [6] also reiterated the relationship among polynomial representation, shift-invariant, and alias-free filter. However, [6] did not explicitly investigate the implication of the shift matrix conversion as proposed in [3].

Let $\mathcal{G} = (V, A)$ be a graph, where $V = \{v_0, v_1, \dots, v_{n-1}\}$ is a set of vertices and $A \in \mathbb{C}^{n \times n}$ is the adjacency matrix of the graph. Let $\mathbf{x} = (x_0, x_1, \dots, x_{n-1})^T$ be a *graph signal*, where each sample $x_i \in \mathbb{C}$ corresponds to a vertex $v_i \in V$.

In particular, if \mathcal{G} is a directed circular graph, then the corresponding adjacency matrix is given by:

$$A = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}. \quad (77)$$

Then $A\mathbf{x} = (x_{n-1}, x_0, \dots, x_{n-2})^T$, that is, multiplication by A shifts each signal sample to the next vertex. Thus, A is often called shift operator or shift matrix, which is similar to time shift operator z^{-1} in DSP. In practice, adjacency matrix can be replaced by other matrices which reflect the structure of the graph, such as the Laplacian matrix and the normalized Laplacian matrix for undirected graphs, and the probability transition matrix. Here, we use S to denote the general shift matrix, whether it is A , (normalized) Laplacian matrix, or the probability transition matrix.

In classical 1-D DSP, a shift-invariant filter F has a Z-transform (polynomial representation in z^{-1}), that is

$$F(z^{-1}) = \sum_{k=-\infty}^{+\infty} f_k z^{-k},$$

where f_k is polynomial coefficient. Moreover, from the shift-invariance property, it follows that the filtered output of a shifted input is equal to the shifted filtered output of the original input. In other words, the shift operation and the filter commute. That is, $Fz^{-1} = z^{-1}F$, which directly follows from the above polynomial representation (see, e.g., [6]).

Extending this concept to GSP, we also define a shift-invariant filter H as the one that commutes with the shift matrix, i.e., $HS = SH$. However, unlike in the classical DSP case, a shift-invariant filter does not necessarily have a polynomial representation in terms of the shift operator S . Yet, H can be represented as a polynomial in S if the shift matrix S satisfies the following condition.

Definition 1 (Shift-enabled graph [7]). *A graph \mathcal{G} is shift-enabled if its corresponding shift matrix S satisfies $p_S(\lambda) = m_S(\lambda)$, where $p_S(\lambda)$ and $m_S(\lambda)$ are the minimum polynomial and the characteristic polynomials of S , respectively. We also say that S is shift-enabled when the above condition is satisfied. Otherwise, S and the corresponding graph, are non-shift-enabled.*

For shift-enabled graphs, the following theorem is the basis of linear, shift-invariant filter design.

Theorem 1. *The shift matrix S is shift-enabled if and only if every matrix H commuting with S is a polynomial in S [3].*

Note that this theorem implies that as long as the shift matrix S does not satisfy the shift-enabled condition (i.e., $m_S(\lambda) \neq p_S(\lambda)$), there will always be some shift-invariant filters (and thus some filters) that cannot be represented as a polynomial of S . Ref [3] de-emphasized the shift-enabled condition by suggesting that we may work around it with the following theorem.

Theorem 2 (Theorem 2 in [3]). *For any shift matrix S , there exists a converted matrix \tilde{S} and matrix polynomial $r(\cdot)$, such that $S = r(\tilde{S})$ and $m_{\tilde{S}}(\lambda) = p_{\tilde{S}}(\lambda)$.*

While the above theorem is correct, it does not take into account that the target filter H may not be shift-invariant with respect to the converted shift matrix. In particular, for a directed graph, in general, S is not symmetric, and thus not jointly diagonalized with H . Consequently, one can show that generally there exists no converted shift-enabled \tilde{S} that can maintain shift-invariance with the target filter when the graph is directed and S is asymmetric [7].

However, the conversion method suggested in [3] does hold for undirected graphs when H can be jointly diagonalized with S . Yet, as we will show in the following, the converted \tilde{S} may not describe the same graph as the original S . This makes the whole conversion process moot. Hence, the

112 shift-enabled condition is important regardless of whether the graph is directed or not (i.e., the shift
113 matrix is asymmetric or not).

114 3. The Necessity of Shift-enabled Condition for Undirected Graphs

115 Before giving a concrete example, let us first review the conversion process described in [3]. As
116 mentioned earlier, even though the conversion process does not hold for arbitrary shift matrices, it can
117 be applied to symmetric shift matrices.

118 According to Lemma A2 in Appendix A, two symmetric and commuting matrices S and H are
119 simultaneously diagonalizable. Thus, there exists an invertible matrix T such that $S = T\Lambda_S T^{-1}$ and
120 $H = T\Lambda_H T^{-1}$, where Λ_S and Λ_H are composed of the eigenvalues of S and H , respectively. Then,
121 a new matrix $\Lambda_{perturb}$ with distinct diagonal elements can be generated by slightly perturbing the
122 values of Λ_S . The new shift matrix is calculated as $\tilde{S} = T\Lambda_{perturb} T^{-1}$. According to Lemma A1 and
123 Lemma A2, the restructured shift matrix \tilde{S} satisfies $p_{\tilde{S}}(\lambda) = m_{\tilde{S}}(\lambda)$ and $H\tilde{S} = \tilde{S}H$. Hence, from
124 Theorem 1, H is a polynomial in \tilde{S} .

125 However, it is not sufficient to have H represented as a polynomial of any arbitrary \tilde{S} . A natural
126 and basic constraint is that the converted \tilde{S} should facilitate “local processing”, that is, it should
127 describe topologically the same graph, which is essential in virtually all GSP applications, such as filter
128 design [8], sampling [9], denoising [10], and classification [11], otherwise, the conversion is meaningless.
129 To ensure that the converted graph facilitates “local processing”, that is, an implementation of an
130 L -th order polynomial filter requires L data exchanges between neighbouring nodes [6], we introduce
131 Definition 2. In fact, the definition of a matrix describing a graph (see details in Definition 2) is not new
132 in spectral graph theory. In particular, the matrix of “loose description” is widely used in the context of
133 inverse eigenvalue problem and zero-forcing problem [12,13]. We introduce “strict description” since
134 we would like to accommodate graphs with self-loops. In a nutshell, two shift matrices describe the
135 same graph if the conversion from one to another preserves the graph topological structure, implying
136 that filtering under the converted graph can be performed locally. The precise definition is specified as
137 follows.

138 **Definition 2** ([12–14]). *Shift matrices S and \tilde{S} strictly describe the same graph if 1) $S_{i,j} \neq 0$ if and only if*
139 *$\tilde{S}_{i,j} \neq 0$ for any i and j , and 2) \tilde{S} is symmetric if and only if S is symmetric. And we will say S and \tilde{S} loosely*
140 *describe the same graph if the first condition is relaxed to 1') $S_{i,j} \neq 0$ if and only if $\tilde{S}_{i,j} \neq 0$ only for $i \neq j$. That*
141 *is, we allow some i where only $S_{i,i}$ or $\tilde{S}_{i,i}$ equal to 0.*

142 Given this additional constraint that \tilde{S} and S should describe the same graph structure, we can
143 show that it is impossible to guarantee the following three conditions to be satisfied simultaneously:

- 144 • \tilde{S} is shift-enabled (i.e., $p_{\tilde{S}}(\lambda) = m_{\tilde{S}}(\lambda)$).
- 145 • H is shift-invariant on \tilde{S} (i.e., $H\tilde{S} = \tilde{S}H$).
- 146 • \tilde{S} and S strictly or loosely describe the same graph.

147 3.1. A counter-example that \tilde{S} can loosely but not strictly describe the original graph

148 Let us start with a non-shift-enabled graph as shown in Figure. 1(a). The shift matrix³ of the
149 undirected graph is $S = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$. It is clear that $p_S(\lambda) = \lambda^3(\lambda - 2)(\lambda + 2) \neq \lambda(\lambda - 2)(\lambda + 2) =$
150 $m_S(\lambda)$ and hence S is non-shift-enabled. Since shift-enabled condition is not just sufficient but also
151 necessary [7], there must exist a shift-invariant filter not representable as a polynomial of S . Indeed,

³ Without loss of generality, we choose adjacency matrix as the shift matrix in the following examples.

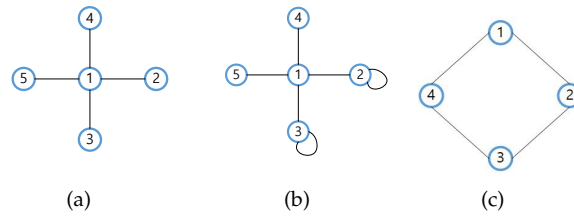


Figure 1. Graph topology used in the examples. (a) Original graph with shift matrix S . (b) Converted shift matrix \tilde{S} which loosely describes the same graph as S . (c) Cycle graph with shift matrix S' .

152 one example for such a filter is $H = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$. It can be readily verified that $HS = 0 = SH$ and

153 thus the filter is shift-invariant, and it is impossible to find polynomial representation of H in terms of
 154 S . Note that $S_{2,3}^n = S_{2,4}^n$ ⁴ for all $n \in \mathbb{N}$. Thus for any polynomial $h(S)$, we must have $h(S)_{2,3} = h(S)_{2,4}$.
 155 But since $H_{2,3} = -1 \neq 0 = H_{2,4}$, $H \neq h(S)$ for any polynomial function $h(\cdot)$.

156 3.1.1. Extension of H to a Class of Filters

Note that we can extend H to the following class of filters that all cannot be represented as polynomials of S :

$$\mathbb{H} = \{\alpha H + q(S) | \alpha \in \mathbb{R}, q(S) \text{ is a polynomial of } S\}. \quad (1)$$

157 Since apparently $q(S)S = Sq(S)$ for any polynomial $q(S)$ and $HS = SH$ as discussed above, any
 158 filter $\alpha H + q(S) \in \mathbb{H}$ commutes with S as well. Thus any filter in \mathbb{H} is shift-invariant. However, since
 159 H is not representable as a polynomial of S , as discussed above, so does $\alpha H + q(S)$.

160 From the examples presented above, we note that when the shift-enabled condition is violated,
 161 we may find an infinite number of shift-invariant filters that are not representable as polynomials of S .

162 3.1.2. Shift-enabled \tilde{S} That Strictly Describes the Original Graph Does Not Exist

First, let us restrict the converted shift matrix \tilde{S} to strictly describe the same graph as S . Thus \tilde{S} could be written as

$$\tilde{S} = \begin{pmatrix} 0 & \tilde{S}_{1,2} & \tilde{S}_{1,3} & \tilde{S}_{1,4} & \tilde{S}_{1,5} \\ \tilde{S}_{1,2} & 0 & 0 & 0 & 0 \\ \tilde{S}_{1,3} & 0 & 0 & 0 & 0 \\ \tilde{S}_{1,4} & 0 & 0 & 0 & 0 \\ \tilde{S}_{1,5} & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2)$$

163 with non-zeros $\tilde{S}_{1,2}$, $\tilde{S}_{1,3}$, $\tilde{S}_{1,4}$, and $\tilde{S}_{1,5}$. We can readily verify that the characteristic polynomial is
 164 $p_{\tilde{S}}(\lambda) = \lambda^3(\lambda^2 - \tilde{S}_{1,2}^2 - \tilde{S}_{1,3}^2 - \tilde{S}_{1,4}^2 - \tilde{S}_{1,5}^2)$ and 0 is the triple eigenvalue of \tilde{S} . According to Lemma A1, a
 165 shift-enabled real symmetric shift matrix has to have unique eigenvalues and thus \tilde{S} is not shift-enabled.
 166 Therefore, all graphs which have the same structure as Figure 1(a) are non-shift-enabled.

167 3.1.3. Shift-enabled \tilde{S} That Loosely Describes the Original Graph Exists

Next, let us relax \tilde{S} so that it may just loosely describe the original graph. In other words, we allow the diagonal elements to be non-zero which maintains most of the topological structure of the original graph. In applications where diffusion or state transition matrices are treated as shift matrices,

⁴ Note that $S_{i,j}^k$ denotes the (i, j) -element of matrix S^k .

the diagonal elements can be interpreted as the returning probabilities of the current state to itself. Thus, the converted shift matrix \tilde{S} can be written as

$$\tilde{S} = \begin{pmatrix} \tilde{S}_{1,1} & \tilde{S}_{1,2} & \tilde{S}_{1,3} & \tilde{S}_{1,4} & \tilde{S}_{1,5} \\ \tilde{S}_{1,2} & \tilde{S}_{2,2} & 0 & 0 & 0 \\ \tilde{S}_{1,3} & 0 & \tilde{S}_{3,3} & 0 & 0 \\ \tilde{S}_{1,4} & 0 & 0 & \tilde{S}_{4,4} & 0 \\ \tilde{S}_{1,5} & 0 & 0 & 0 & \tilde{S}_{5,5} \end{pmatrix}. \quad (3)$$

168 Many solutions that satisfy shift-enabled and shift-invariant conditions can be found. For instance,
 169 $\tilde{S} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$ is one such solution, where the original graph structure is only slightly modified as
 170 shown in Figure 1(b). One can verify that the eigenvalues $(-1.8136, 0, 0.4707, 1, 2.3429)$ of \tilde{S} are distinct
 171 and thus \tilde{S} is shift-enabled. Moreover, one can also readily verify that $H\tilde{S} = \tilde{S}H$. By Theorem 1, the
 172 above two conditions ensure that H is a polynomial in \tilde{S} .

Remark 1. Note that the example from the previous section can be extended to a star graph with more than five vertices. In this case, the shift matrix of a star graph with N vertices is $S_N = \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{pmatrix}$. Consider a

filter $H_N = \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$, and based on Equation 2 shift operator that strictly describing the original

graph satisfies $\tilde{S}_{N(\text{strict})} = \begin{pmatrix} 0 & \tilde{S}_{1,2} & \tilde{S}_{1,3} & \dots & \tilde{S}_{1,N} \\ \tilde{S}_{1,2} & 0 & 0 & \dots & 0 \\ \tilde{S}_{1,3} & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{S}_{1,N} & 0 & 0 & \dots & 0 \end{pmatrix}$. Repeating the similar argument as before, we can readily

verify that the following five conclusions still hold simultaneously: (i) S_N is not shift-enabled for $p_{S_N}(\lambda) = \lambda^{N-2}(\lambda^2 - (N-1))$. (ii) H_N is shift-invariant, i.e., $H_N S_N = 0 = S_N H_N$. (iii) $H_N \neq h(S_N)$ for any polynomial function $h_N(\cdot)$. (iv) H_N can be extended to the following class of filters that none can be represented as polynomials of S_N :

$$\mathbb{H}_N = \{\alpha H_N + q(S_N) | \alpha \in \mathbb{R}, q(S_N) \text{ is a polynomial of } S_N\}.$$

173 (v) Shift-enabled $\tilde{S}_{N(\text{strict})}$ that strictly describes the original graph does not exist.

174 3.2. A Counter Example When the Converted Shift Matrix Can Neither Strictly Nor Loosely Describe the 175 Original Graph

176 Note that there are situations where no shift-enabled \tilde{S} exists even after we relax the graph
 177 structure constraint as in the earlier example. Consider shift matrix $S' = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$ as shown in
 178 Figure 1(c).

179 It can easily be seen that the eigenvalues of S' , $(0, 0, 2, -2)$, are not unique. Thus S' is
 180 non-shift-enabled according to Lemma A1. So we do expect that there exists shift-invariant filter
 181 not representable by S' . Indeed, we can easily show that filter $H' = \begin{pmatrix} 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 \end{pmatrix}$ is such a filter.

182 First, note that $H'S' = S'H'$ and thus H' is shift-invariant under S' . Furthermore, note that
 183 $(S')_{1,2}^n = (S')_{1,4}^n$ for all $n \in \mathbb{N}$, and so $h(S')_{1,2} = h(S')_{1,4}$ for any polynomial $h(S')$. But since $H'_{1,2} =$
 184 $0 \neq 1 = H'_{1,4}$, $H' \neq h(S')$ for any polynomial function $h(\cdot)$.

185 Let us prove that it is impossible to find a converted shift matrix \tilde{S}' which is shift-enabled and
 186 commutes with H' by only changing the weights of nonzero and diagonal elements.

Consider a general symmetric matrix

$$\tilde{S}' = \begin{pmatrix} \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 & \tilde{S}'_{1,4} \\ \tilde{S}'_{1,2} & \tilde{S}'_{2,2} & \tilde{S}'_{2,3} & 0 \\ 0 & \tilde{S}'_{2,3} & \tilde{S}'_{3,3} & \tilde{S}'_{3,4} \\ \tilde{S}'_{1,4} & 0 & \tilde{S}'_{3,4} & \tilde{S}'_{4,4} \end{pmatrix} \quad (4)$$

187 which has arbitrary weights on nonzero and diagonal elements. That is, \tilde{S}' loosely describes the same
188 graph as S' .

$H' = h(\tilde{S}')$ clearly implies that H' commutes with \tilde{S}' , namely, $H'\tilde{S}' = \tilde{S}'H'$ is a necessary condition for $H' = h(\tilde{S}')$. It follows from $H'\tilde{S}' = \tilde{S}'H'$ that $\tilde{S}'_{1,1} = \tilde{S}'_{2,2} = \tilde{S}'_{3,3} = \tilde{S}'_{4,4}$ and $\tilde{S}'_{1,2} = \tilde{S}'_{1,4} = \tilde{S}'_{2,3} = \tilde{S}'_{3,4}$, i.e.,

$$\tilde{S}' = \begin{pmatrix} \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 & \tilde{S}'_{1,2} \\ \tilde{S}'_{1,2} & \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 \\ 0 & \tilde{S}'_{1,2} & \tilde{S}'_{1,1} & \tilde{S}'_{1,2} \\ \tilde{S}'_{1,2} & 0 & \tilde{S}'_{1,2} & \tilde{S}'_{1,1} \end{pmatrix}. \quad (5)$$

189 Following Cayley-Hamilton Theorem [15], if H' is a polynomial in \tilde{S}' , then $H' = h(\tilde{S}') =$
190 $h_0I + h_1\tilde{S}' + h_2\tilde{S}'^2 + h_3\tilde{S}'^3$, where I as the identity matrix. In fact, it is easy to prove that $(\tilde{S}')^k_{1,2} = (\tilde{S}')^k_{1,4}$,
191 for $k = 0, 1, 2, 3$. Hence, $h(\tilde{S}')_{1,2} = h(\tilde{S}')_{1,4}$ which contradicts with $H'_{1,2} \neq H'_{1,4}$. Thus, for this example,
192 the filter H' cannot be represented as a polynomial in the converted shift matrix \tilde{S}' which even just
193 loosely describes the original graph.

194 **Remark 2.** Just as in Remark 1, the above example can be extended to cycle graphs with more vertices.

195 In this case, the shift matrix of cycle graph with N vertices is $S'_N = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & -1 \end{pmatrix}$. Consider the filter

196 $H'_N = \begin{pmatrix} 0 & 0 & \cdots & 0 & -1 & 1 \\ 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ -1 & 1 & \cdots & 0 & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & -1 \end{pmatrix}$, and similar to Equation 4, the shift operators loosely describing the original graph

197 should satisfy $\tilde{S}'_{N(\text{loose})} = \begin{pmatrix} \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 & \cdots & \tilde{S}'_{1,N} \\ \tilde{S}'_{1,2} & \tilde{S}'_{2,2} & \tilde{S}'_{2,3} & \cdots & 0 \\ 0 & \tilde{S}'_{2,3} & \tilde{S}'_{3,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{S}'_{1,N} & 0 & 0 & \cdots & \tilde{S}'_{N,N} \end{pmatrix}$. With the similar argument as before, one can show

198 that $\tilde{S}'_{N(\text{shift-invariant})} = \begin{pmatrix} \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & 0 & \cdots & \tilde{S}'_{1,2} \\ \tilde{S}'_{1,2} & \tilde{S}'_{1,1} & \tilde{S}'_{1,2} & \cdots & 0 \\ 0 & \tilde{S}'_{1,2} & \tilde{S}'_{1,1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tilde{S}'_{1,2} & 0 & 0 & \cdots & \tilde{S}'_{1,1} \end{pmatrix}$. Consequently, one can readily verify the following four

199 conclusions in Section 3.2 are still valid: (i) S'_N is not shift-enabled, since it has repeated eigenvalues, that
200 is, $\lambda_k = 2\cos(2\pi k/N)$ for $k = 0, 1, 2, \dots, N-1$. (ii) H'_N is shift-invariant, i.e., $H'_N S'_N = S'_N H'_N$. (iii)
201 $H'_N \neq h(S'_N)$ for any polynomial function $h_N(\cdot)$. (iv) The shift matrix that loosely describes the original graph
202 and commutes with H'_N must have the form of $\tilde{S}'_{N(\text{shift-invariant})}$, and such a shift-enabled matrix does not
203 exist.

204 4. Conclusions

205 For a non-shift-enabled graph, even if we can easily "transform" the symmetric shift matrix S
206 into one that satisfies the shift-enabled condition, the new \tilde{S} may be irrelevant since it describes a
207 very different graph from S . That is, the operator \tilde{S} on a graph signal may involve mixing inputs far
208 beyond its neighborhood and become impractical for huge graphs. Combined with the necessity of
209 the shift-enabled condition for directed graph [7], we demonstrated in this paper that the shift-enabled
210 condition is essential for any graph structure. A good future direction is to explore a shift that

211 “approximately” describes the original graph as the conversion is quite different. In particular, if it is
 212 already known that one such shift does not exist, one possible direction to explore shift “approximately”
 213 describe the original graph instead (some non-zero off diagonal element may not correspond to an
 214 actual edge). But we will leave this to future study.

215 Note that even though we consider the adjacency matrix as the shift matrix in our examples, the
 216 conclusion applies to other shift matrices. In particular, one can readily verify that the conclusion still
 217 holds if we use the Laplacian matrix as the shift matrix in the example in Section 3.2.

218 **Author Contributions:** writing—original draft preparation, Liyan Chen and Samuel Cheng; writing—review and
 219 editing, Lina Stankovic and Vladimir Stankovic; methodology, Kanghang He.

220 **Funding:** This research was funded by the European Union’s Horizon 2020 research and innovation programme
 221 under the Marie Skłodowska-Curie grant agreement number 734331, the Fundamental Research Funds for
 222 the Central Universities number 0800219369 and National Key Research and Development Project number
 223 2017YFE0119300.

224 **Acknowledgments:** The authors thank B. Zhao for helpful discussions.

225 Appendix A

226 It is easily determined whether a graph is shift-enabled by the following lemmas.

227 **Lemma A1.** *If shift matrix S is a real symmetric matrix, then S is shift-enabled, if and only if all eigenvalues of*
 228 *S are distinct [1].*

229 Lemma A1 indicates that an undirected graph is shift-enabled if and only if its eigenvalues are all
 230 distinct.

231 As both shift matrix S and filter matrix H are symmetric, we can obtain the following lemma.

232 **Lemma A2.** *If shift matrix S and filter matrix H are diagonalizable (this condition always holds for symmetric*
 233 *matrix) then S and H are simultaneously diagonalizable (by an invertible matrix) if and only if $HS = SH$ (see*
 234 *Theorem 1.3.12 in [16]).*

235 References

- 236 1. Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J.M.; Vandergheynst, P. Graph signal processing: Overview,
 237 challenges, and applications. *Proceedings of the IEEE* **2018**, *106*, 808–828.
- 238 2. Shuman, D.I.; Narang, S.K.; Frossard, P.; Ortega, A.; Vandergheynst, P. The emerging field of signal
 239 processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains.
 240 *IEEE Signal Proc. Magazine* **2013**, *30*, 83–98.
- 241 3. Sandryhaila, A.; Moura, J.M. Discrete signal processing on graphs. *IEEE Trans. Signal Processing* **2013**,
 242 *61*, 1644–1656.
- 243 4. Sandryhaila, A.; Moura, J.M. Discrete Signal Processing on Graphs: Frequency Analysis. *IEEE Trans. Signal*
 244 *Processing* **2014**, *62*, 3042–3054.
- 245 5. Sandryhaila, A.; Moura, J.M. Big data analysis with signal processing on graphs: Representation and
 246 processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine* **2014**, *31*, 80–90.
- 247 6. Teke, O.; Vaidyanathan, P. Linear systems on graphs. 2016 IEEE Global Conference on Signal and
 248 Information Processing (GlobalSIP). IEEE, 2016, pp. 385–389.
- 249 7. Chen, L.; Cheng, S.; Stankovic, V.; Stankovic, L. Shift-enabled graphs: Graphs where shift-invariant filters
 250 are representable as polynomials of shift operations. *IEEE Signal Processing Letters* **2018**, *25*, 1305–1309.
- 251 8. Hammond, D.K.; Vandergheynst, P.; Gribonval, R. Wavelets on graphs via spectral graph theory. *Applied*
 252 *and Computational Harmonic Analysis* **2011**, *30*, 129–150.
- 253 9. Gadde, A.; Anis, A.; Ortega, A. Active semi-supervised learning using sampling theory for graph signals.
 254 Proc. 20th ACM SIGKDD Intl. Conf. Knowledge Discovery & Data Mining, 2014, pp. 492–501.
- 255 10. Yang, C.; Cheung, G.; Stankovic, V. Estimating heart rate and rhythm via 3D motion tracking in depth
 256 video. *IEEE Transactions on Multimedia* **2017**, *19*, 1625–1636.

- 257 11. He, K.; Stankovic, L.; Liao, J.; Stankovic, V. Non-Intrusive Load Disaggregation Using Graph Signal
258 Processing. *IEEE Transactions on Smart Grid* **2018**, *9*, 1739–1747.
- 259 12. Hogben, L. Spectral graph theory and the inverse eigenvalue problem of a graph. *Electronic Journal of*
260 *Linear Algebra* **2005**, *14*, 3.
- 261 13. Trefois, M.; Delvenne, J.C. Zero forcing number, constrained matchings and strong structural controllability.
262 *Linear Algebra and its Applications* **2015**, *484*, 199–218.
- 263 14. Fallat, S.M.; Hogben, L. The minimum rank of symmetric matrices described by a graph: a survey. *Linear*
264 *Algebra and its Applications* **2007**, *426*, 558–582.
- 265 15. Lancaster, P.; Tismenetsky, M. *The theory of matrices: with applications*; Elsevier, 1985.
- 266 16. Horn, R.A.; Johnson, C.R. *Matrix analysis*; Cambridge University Press, 2012.