Dynamic Linepack Depletion Models for Natural Gas Pipeline Networks

Samuel Chevalier, Student Member, IEEE, and Dan Wu, Member, IEEE

Abstract—Given the critical role played by natural gas in providing electricity, heat, and other essential services, better models are needed to understand the dynamics of natural gas networks during extreme events. This paper aims at establishing appropriate simulation models to capture the slow dynamics of linepack depletion for natural gas pipeline networks. Three alternative ordinary differential equation (ODE) simulation techniques are derived and discussed, the first being the conventional one used in the literature. We show that the system of equations associated with this model is degenerate when flux injections are controlled (i.e. specified) at all nodes. The other two present useful frameworks for analyzing how networks respond to system-wide mass flux imbalances. These techniques offer different alternatives for simulating system dynamics based on how sources and loads are chosen to be modeled, and they are all proven to be non-degenerate in a tree-structured network. These proposed techniques are all tested on the 20-node Belgium network. The simulation results show that the conventional model cannot effectively capture linepack depletion under long term system-wide mass flux imbalance, while the proposed models can characterize the network behavior until the linepack is completely depleted.

Index Terms—Dynamic simulation, linepack depletion, natural gas network, reduced order modeling, survival time

I. INTRODUCTION

NATURAL gas continues to be a growing fuel source for the synchronous generators powering the electrical grid. In 2018, natural gas (NG) fired power plants produced 35.5% of the total electricity in the United States [1], up from 13% in 2000 [2]. Since onsite storage of NG fuel is nonexistent for many of these generators [3], the NG pipeline network (NGPN) acts as a critical link in the American energy supply chain. Accordingly, the interdependence between the electrical power grid and the NGPN has increased considerably over the previous decades.

It is increasingly evident that the close degree of coupling between the electrical power grid and NGPNs cannot be ignored when analyzing the propagation of certain types of failures in these systems [4]. A vast literature exists on the modeling of electrical power systems and their various dynamical components [5–7]. Relative to power grids though, the modeling and simulation of NGPNs is challenging due to the characteristic timescale of gas dynamics. While the fast electromagnetic wave transients in power grids may be relaxed to algebraic constraints, NG propagation is a slow phenomena (10 m/s is a characteristic gas flow speed). Despite the potential high fidelity of the results, simulating the nonlinear partial differential equations (PDEs) which model these fluid dynamics can be intractably slow for large NGPNs whose dynamics unfold over the course of many hours. It is therefore necessary to have fast and relatively accurate tools for evaluating the dynamic behaviour of NGPNs under extreme situations, such as severe contingencies.

A variety of platforms have been developed for the purpose of simulating NGPN dynamics. Early reviews of the relevant fluid dynamics, modeling strategies, and numerical solution techniques can be found in [8, 9]. More recent works have focused on approximating network dynamics through discretization in time and space [10–12], so that state space control and optimization techniques can be leveraged for various objectives. By linearizing about an operating point, [13] employs transfer function matrices to describe the dynamics associated with pipelines in a network. For further simplification, [14] identifies the dominant eigenmodes of the linearized NGPN and uses them to build a reduced order model of the system for simplified analysis. In order to simplify the particular gas wave effects while still preserving nonlinearities, [15] analytically integrates across segments of the pipelines in order to build an ODE model. Via adaptive time-stepping, simulation results of the reduced model are compared with those from a full order PDE model and are shown to be comparable in accuracy.

For high fidelity results, authors in [16] employ a second order staggered finite difference discretization method. The approach is computationally efficient, unconditionally stable, and is proven to exactly satisfy mass conservation. For increasing the speed of simulation, [17] first models a NGPN with a set of DAEs. After exploring the so-called tractability index of the DAE set, model order reduction, via proper orthogonal decomposition, is explored and tested. In order to exploit the analytical structure of the equations, [18] applies a finite volume method to convert the PDEs into relevant ODEs. After eliminating the maximum number of algebraic constraints, a preconditioner is developed in order to reduce the computation burden of Jacobian inversion during each Newton iteration. The proposed methods are shown to significantly speed up backward Euler integration of the network dynamics [18].

While there are many sophisticated methods available for simulating intra-pipeline dynamics, the effects of boundary constraints (sources and loads) have been seldom addressed in the literature. Most (if not all) simulation platforms assume constant pressure at several slack nodes, but this further assumes the network has an infinite pool of reserves to pull from at infinitely fast injection rates. While this may be a safe assumption during “normal” system operation, it could certainly fail during certain critical contingencies. In this paper, we investigate alternative methods for applying boundary constraints. Our primary goal is to develop a nonlinear NGPN simulation framework which can better capture the dynamic characteristics of the linepack depletion phenomenon.

S. Chevalier is with Department of Mechanical Engineering, Massachusetts Institute of Technology. E-mail: schev@mit.edu

Dan Wu, is with the Laboratory for Information and Decisions Systems, Massachusetts Institute of Technology. E-mail: danwumit@mit.edu
for extreme contingency situations. Specifically, we require our models to provide an accurate estimation of the survival time during linepack depletion. Thus, rather than the exact transient behavior of the NG, we are primarily concerned with developing methods which can accurately characterize the linepack depletion in the network over time. When these methods are posed properly, simulation results can be collected on a timescale which can enable the statistical analysis of cascading failures in coupled NG and electrical power systems. With a focus on developing fast and well-characterized simulation techniques, the specific contributions of this paper are as follows:

1) We leverage the work presented in [15] by building a full network model on top of the previously proposed framework. This model is proved degenerate if mass flux injections are specified at all boundary points.

2) After presenting the common solution to this degeneracy (introduction of constant pressure slack nodes), we derive two alternative ODE simulation models which each make different load and source modeling assumptions. We prove that each are non-degenerate and explain the particular applications of each.

3) We compare and contrast simulation results associated with these different simulation models for a given set of contingencies. Most notably, we highlight the inability of the common slack node model to capture linepack depletion effects when NG source injections are constrained.

II. BUILDING AN ODE SIMULATION PLATFORM

In this section, we build the relevant set of ODEs which may be used for simulating gas network transients.

A. Model Reduction of the the Euler Equations

We begin by stating the Euler equations [15] which govern unsteady compressible fluid flow in one dimension (x), where all relevant variables and constants are explained in Table I:

\[ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho \nu) = 0 \]  
\[ \frac{\partial}{\partial t} (\nu^2 \rho) + \frac{\partial}{\partial x} \left( \nu^2 \rho \right) + \frac{\partial}{\partial x} \tau = -\lambda \frac{1}{2D} \rho \nu |\nu| - \rho g \sin(\theta). \]  

Equation (1) is a statement of the continuity of mass flow while (2) is a statement of the conservation of momentum.

If we assume temperature is constant in each line segment, then there exists a linear relationship between pressure \( p \) and density \( \rho \):

\[ p = a^2 \rho, \]

where \( a^2 = ZRT/M \) comes from the ideal gas law (plus some nonideality correction factor \( Z \)). Further assuming the gravitation forces caused by elevation changes are negligible (\( \theta \approx 0 \)), and that fluid flows are much slower than the speed of sound (\( \nu \ll a \)), the PDEs simplify to

\[ \frac{\partial}{\partial t} \rho + \frac{\partial}{\partial x} \rho \phi = 0 \]  
\[ \frac{\partial}{\partial t} \phi + a^2 \frac{\partial}{\partial x} \phi = -\frac{\lambda}{2D} \phi |\phi|. \]

We consider some line of length \( L \). This line is spatially discretized into \( N \) segments, each of length \( l = L/N \). Intermediate nodes are thus defined at \( x = \{0, l, 2l,...Nl = L\} \). As in [15], we integrate the mass and momentum equations over the length of a line segment \( x \in [0, l] \), using the trapezoidal rule to approximate the integrals where necessary. The resulting nonlinear differential equations contain derivatives with respect to time only:

\[ \frac{l}{2} (\dot{\rho}_l + \rho_0) + (\phi_l - \phi_0) = 0 \]  
\[ \frac{l}{2} (\dot{\phi}_l + \phi_0) + a^2 (\rho_l - \rho_0) = -\frac{\lambda}{4D} (\phi_l + \phi_0) \left| \phi_l + \phi_0 \right| \rho_l + \rho_0, \]

where subscripts 0 and \( l \) indicate variables corresponding to the beginnings and ends of line segments, respectively. To specify a complete network model with an arbitrary number of lines connecting sources and loads, these equations must be complimented by two others. The first is a statement of the conservation of mass flux at each node; the second states that the intra-nodal pressures on the incoming and outgoing sides of a node are algebraically related according to some compressor amplification factor \( \alpha \). If no compressor is present at a node, \( \alpha = 1 \). This parameter may be independently controlled by the system operators as necessary and is treated as a system input. Via PDE integration, [15] confirms the validity of this reduction technique for a slowly varying input on a small test system.

Remark 1. The physical accuracy associated with the model reduction in (6)-(7) can be increased arbitrarily if we integrate \( l \to 0 \).

B. NGPN Simulation Model

We now set up a system of equations to describe the dynamics of a full NGPN. Our primary goal is to investigate the phenomenon of linepack depletion. Accordingly, in writing the conservation laws, we assume that the load and source mass flux injections are specified inputs (typically constant). We also assume all density states are free variables, i.e., they may evolve freely according to the dynamics of the system.

Definition 1. The NG simulation model where all mass flux injections are specified (i.e. determined) and all density states are free variables is referred to as the flux determined model.
As shall be shown, the differential system associated with the flux determined system is degenerate and cannot be simulated uniquely, so alterations must be made.

We define an arbitrary network with \( n \) physical nodes and \( m \) physical branches. Once the system lines have been properly discretized, we include the intermediate nodes and branches to define a total of \( n \gg \tilde{n} \) nodes and \( m \gg \tilde{m} \) branches. The standard directed incidence matrix related to the system is given by \( E \in \mathbb{R}^{m \times n} \). The graph is directed in the same direction in which line flows are normally positive. We define two vectors: \( \overrightarrow{p} \in \mathbb{R}^{2m \times \tilde{n}} \) and \( \overrightarrow{\phi} \in \mathbb{R}^{2m \times \tilde{n}} \). The vector \( \overrightarrow{p} \) is filled with the density states on either side of each of the \( n \) nodes, where “-” indicates the inflowing side and “+” indicates the outflowing side of each node. The vector \( \overrightarrow{\phi} \) is filled with the mass flux states on either side of each of the \( m \) lines, where “0” indicates the beginning side and “f” indicates the ending side of each line:

\[
\overrightarrow{p} = \begin{bmatrix}
\rho_{(1)}^{-} \\
\vdots \\
\rho_{(n)}^{-} \\
\rho_{(1)}^+ \\
\vdots \\
\rho_{(n)}^+
\end{bmatrix}, \quad (8) \quad \overrightarrow{\phi} = \begin{bmatrix}
\phi_{(1)}^0 \\
\vdots \\
\phi_{(m)}^0 \\
\phi_{(1)}^f \\
\vdots \\
\phi_{(m)}^f
\end{bmatrix}. \quad (9)
\]

Next, we split these vectors in half and define subset vectors:

\[
\rho_- = \begin{bmatrix}
\rho_{(1)}^-
\vdots \\
\rho_{(n)}^-
\end{bmatrix}, \quad \rho_+ = \begin{bmatrix}
\rho_{(1)}^+ \\
\vdots \\
\rho_{(n)}^+
\end{bmatrix}, \quad \phi_0 = \begin{bmatrix}
\phi_{(1)}^0 \\
\vdots \\
\phi_{(m)}^0
\end{bmatrix}, \quad \phi_f = \begin{bmatrix}
\phi_{(1)}^f \\
\vdots \\
\phi_{(m)}^f
\end{bmatrix}.
\]

We define \( \alpha \in \mathbb{R}^{n \times 1} \) as the vector of compressor ratios. The compressors relate the density differentials via

\[
\rho_+ = \text{diag} \{ \alpha \} \rho_-.
\]

Fig. 1 illustrates the relationships between compressor constants and flux injection, flux line flow, and density variables. Since (10) is a linear algebraic equation, we also have \( \dot{\rho}_+ = \text{diag} \{ \alpha \} \dot{\rho}_- \). For notational convenience, we introduce the density variable vector \( \rho \equiv \rho_- \).

Next, we must define a mass-flux conservation law. In this network, just as in power systems, a positive injection is defined to be a source of supply, while a negative injection represents a load. We define the load and source injection vector \( \mathbf{d} \in \mathbb{R}^{n \times 1} \) whose relationship satisfies

\[
K_0 \phi_0 + K_1 \dot{\phi}_0 = \mathbf{d}
\]

where \( K_1 \in \mathbb{R}^{n \times m} \) and \( K_0 \in \mathbb{R}^{n \times m} \) codify, respectively, which lines enter and leave each node (based on the convention of the incidence matrix):

\[
K_0(i,j) = \begin{cases} 
1 & \text{Line } j \text{ leaves node } i \\
0 & \text{Line } j \text{ does not leave node } i 
\end{cases}
\]

(12)

\[
K_1(i,j) = \begin{cases} 
-1 & \text{Line } j \text{ enters node } i \\
0 & \text{Line } j \text{ does not enter node } i 
\end{cases}
\]

(13)

Alternatively, \( K_0 = \frac{1}{2} [E]^T \frac{1}{2} [E] \), \( K_1 = \frac{1}{2} [E]^T - \frac{1}{2} [E] \), and \( E^T = K_1 + K_0 \). Equation (11) is a linear algebraic equation which is entirely analogous to Kirchhoff’s Current Law in an electrical network. In taking its time derivative, we have

\[
K_0 \dot{\phi}_0 + K_1 \dot{\phi}_1 = \mathbf{d}.
\]

Remark 2. While (11) represents the conservation of mass at each node in the network, (12) represents the continuity of differential mass flow on a pipeline. These are physically similar, yet characteristically different, processes.

Across each line, we must satisfy the conservation laws. Using the proposed framework, (6) may be written as

\[
0 = \text{diag} (1/2) \left( K_0 (\dot{\rho}_+ - K_1^T \rho_- ) + (\phi_1 - \phi_0) \right)
\]

(14a)

\[
= \text{diag} (1/2) \left( K_0^T \text{diag} \{ \alpha \} - K_1^T \right) \dot{\rho} + \phi_1 - \phi_0. \quad (14b)
\]

where \( l \in \mathbb{R}^{m \times 1} \) is a vector of line lengths. In considering [7], we note the Hadamard product \( \odot \) which performs element by element multiplication on a set of vectors, and Hadamard division \( \oslash \) which performs element by element division on a set of vectors. We also define function \( f \), which is a function of identically sized vectors \( \mathbf{a}, \mathbf{b}, \) and \( c \):

\[
f(x, y, z) := (x + y) \odot |x + y| \oslash z. \quad (15)
\]

We also define a set of \( \Gamma \) matrices:

\[
\Gamma_2 = \text{diag} (1/2) \quad (16)
\]

\[
\Gamma_3 = \text{diag} \{ \alpha \}^2 (K_1^T + K_0^T \text{diag} \{ \alpha \}) \quad (17)
\]

\[
\Gamma_4 = \text{diag} \{ \lambda \odot D / 4 \} \quad (18)
\]

\[
\Gamma_5 = K_1^T \text{diag} \{ \alpha \} - K_1^T \quad (19)
\]

where \( \alpha \in \mathbb{R}^{m \times 1} \) is the vector of line velocities from \([5]\), \( \lambda \in \mathbb{R}^{m \times 1} \) is the vector of darcy friction factors, and \( D \in \mathbb{R}^{m \times 1} \) is the vector of pipe diameters. Thus, (7) can be written as

\[
\Gamma_2 (\dot{\phi}_1 + \dot{\phi}_0) = \text{diag} \{ \alpha \}^2 (K_1^T \rho_- + K_0^T \rho_+) \quad (20a)
\]

\[
-\Gamma_4 (\phi_1 + \phi_0) \odot |\phi_1 + \phi_0| \odot (K_0^T \rho_+ - K_1^T \rho_-) = \Gamma_3 \rho - \Gamma_4 f (\phi_1, \phi_0, \Gamma_5 \rho). \quad (20b)
\]

We may now assemble the full set of differential equations which are necessary to describe the dynamics of this network:

\[
K_1 \dot{\phi}_1 + K_0 \dot{\phi}_0 = \mathbf{d} \quad (21)
\]

\[
\Gamma_1 \rho = \phi_0 - \phi_1 \quad (22)
\]

\[
\Gamma_2 (\dot{\phi}_1 + \dot{\phi}_0) = \Gamma_3 \rho - \Gamma_4 f (\phi_1, \phi_0, \Gamma_5 \rho). \quad (23)
\]
Next, we define the state variable vector $\mathbf{x} \in \mathbb{R}^{(n+2m) \times 1}$,
\[
\mathbf{x} = \begin{bmatrix} \rho \\ \phi_0 \\ \phi_l \end{bmatrix},
\]
and we define the RHS of the set of differential equations (21)-(23) as $G(\mathbf{x})$, such that
\[
\begin{pmatrix}
0 & K_0 & K_L \\
\Gamma_1 & 0 & 0 \\
0 & \Gamma_2 & \Gamma_2 \\
\end{pmatrix}
\mathbf{x} = G(\mathbf{x}).
\]

The full differential model is compactly stated $M_\mathbf{x} \mathbf{x} = G(\mathbf{x})$.

**Remark 3.** The differential order of (25) may be reduced by eliminating the out-flowing flux variable at all nodes (virtual or real) with a single in-flowing flux variable (or vice versa). In this case, $\phi_0^k = \phi_0^{k+1}$. This is similarly noted in [18].

Before concluding this subsection, we define a NG pipeline system $\Sigma_c$ which has a set of specific properties.

**Definition 2.** Consider a NG pipeline system $\Sigma_c$ whose graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has edge (line) set $\mathcal{E}$, $|\mathcal{E}| = m$, vertex (node) set $\mathcal{V}$, $|\mathcal{V}| = n$, and directed nodal incidence matrix $\mathbf{E} \in \mathbb{R}^{m \times n}$. The graph of $\Sigma_c$ has a connected tree structure, meaning $n = m + 1$, and the dynamics of the network are codified by (25).

**C. Degeneracy of the Flux Determined NG Simulation Model**

Now that a differential model for the flux determined model has been derived, we may investigate its shortcomings.

**Theorem 1.** Consider system $\Sigma_c$. The associated coefficient mass matrix $M_\mathbf{x}$ is a singular matrix.

**Proof.** We note that $\Gamma_1 \in \mathbb{R}^{m \times n}$ in (25) is structurally equivalent to incidence matrix $\mathbf{E}$. Since in a tree network, $m = n - 1$, the maximum rank of $\Gamma_1$ is $m$. Even assuming full row rank of $\Gamma_1$, column rank must be degenerate. Since the columns of $\Gamma_1$ are surrounded by 0 entries on the top and bottom in (25), degenerate column rank of $\Gamma_1$ implies degenerate column rank of $M_\mathbf{x}$, further implying matrix singularity.

The results of (1) are clear: the system in (25) cannot be simulated as formulated because the coefficient matrix cannot be inverted. Intuitively, this may seem to be a perplexing result, because in a physical system, if all injections are controlled (i.e. determined), there will certainly be a physical response. Why can the model in (25) not uniquely predict this response?

To investigate this result, we may consider a two-node system where flux is determined according to $\phi_0 = d_s$ at the source node and $\phi_L = d_l$ at the load node. Assuming constant injections at either end, we have $\phi_{L} = \phi_{0} = 0$. In setting $\lambda/2Da^2 = L/2 = 1$, the system (6)-(7) simplifies to
\[
\dot{\rho}_L + \dot{\rho}_0 = d_s - d_l
\]
and
\[
\dot{\rho}_L^2 - \dot{\rho}_0^2 = -(d_l + d_s)^2.
\]

We now consider the situation where the load increases instantaneously by $\epsilon_l$. Due to algebraic constrain (27), one of the densities must also change instantaneously. Herein lies the degeneracy: either density can change, so long as their squared difference is equal to $-(d_l + d_s)^2$. Since there is only 1 differential equation, we may eliminate either $\rho_L$ or $\rho_0$ with the algebraic constraint and treat the other as a true state variable. In the case where a load perturbation occurs, it would be physically meaningful to eliminate the load density and treat it as an algebraic variable which can instantaneously respond to the injection change; this observation will inform one of our proposed simulation formulations, along with the following theorem.

**Theorem 2.** Consider system $\Sigma_c$. To achieve invertibility of the coefficient mass matrix $M_\mathbf{x}$, the elimination of a density state from the system is a necessary condition.

**Proof.** For full rank of $M_\mathbf{x}$ to be achieved without violation of its structure, full column rank of $\Gamma_1$ is necessary. Matrix $\Gamma_1$ is structurally equivalent to incidence matrix $\mathbf{E}$ and thus has rank $n - 1 = m$ (i.e. full row rank). To achieve full column rank of $\Gamma_1$, one column must be eliminated. Since the deletion of a column from the incidence matrix associated with a tree network is a full rank matrix, the deletion of a column of $\Gamma_1$ will render the submatrix, termed $\Gamma_1'$, full rank. All columns of $\Gamma_1$ in (25) are associated with a density state variable. Therefore, the deletion of at least one density state is necessary.

Regardless of how the system is altered, a density state must be eliminated from the state variable vector in order to invert the mass matrix. The following section will use these results to build a set of realizable simulation models.

**III. SIMULATION TECHNIQUES**

With the observations of Theorems 1 and 2 in mind, this section introduces three alternative simulation techniques based on different assumptions about the nature of the sources and loads in the system.

**A. Technique 1: Infinite Flux Reservoir**

This modeling technique holds at least one pressure state in the system constant, and it has been implicitly employed in other works [10], [15]. Its framework may be a poor one for investigating linepack depletion, though. Because a density state variable must be eliminated, we must delete a corresponding equation in order to ensure that the number of equations continues to match the number of state variables. The following theorem shows how non-singularity of the mass matrix can be achieved.

**Theorem 3.** Consider system $\Sigma_c$. The elimination of a pressure state and a flux conservation equation in (27) from $M_\mathbf{x}$ will yield a square, full rank matrix $M''_\mathbf{x}$.

**Proof.** By Theorem 2 when a pressure state has been eliminated, submatrix $\Gamma_1'$ will be full rank. We define
\[
M'_x = \begin{bmatrix} 0 & K_0 & K_L \\
\Gamma_1' & 0 & 0 \\
0 & \Gamma_2 & \Gamma_2 \end{bmatrix}.
\]
We eliminate one row from the top subsection of $M'_x$ (from matrices $K_0$ and $K_L$) and consider the rank of the submatrix $\tilde{M}'_x$, where

$$\tilde{M}'_x = \begin{bmatrix} K'_0 & K'_L \\ \Gamma_2 & \Gamma_2 \end{bmatrix},$$

and the prime notation on $K'_0$ and $K'_L$ indicates the deletion of a row. We take the determinant of $\tilde{M}'_x$:

$$\det(\tilde{M}'_x) = \det(K'_0 - K'_L \Gamma_2^{-1} \Gamma_2) \det(\Gamma_2) = \det(K'_0 - K'_L) \sigma. \quad (30a)$$

where $\sigma \neq 0$ since $\Gamma_2$ is a diagonal matrix. Since $K'_0 - K'_L = |E|^{-1}$ (meaning one row has been eliminated), $\det(\tilde{M}'_x) \propto \det(|E|^{-1}) \neq 0$ since the reduced incidence matrix of a connected tree graph is full rank. Because $\tilde{M}'_x$ and $\Gamma_1'$ are both full rank, it implies that matrix $M''_x$, which is defined by

$$M''_x = \begin{bmatrix} 0 & K'_0 & K'_L \\ \Gamma_1' & 0 & 0 \\ 0 & \Gamma_2 & \Gamma_2 \end{bmatrix},$$

is also full rank.

Therefore, the elimination of a conservation equation from (21) and the elimination of a density state from (24) will yield a set of equations which can be uniquely simulated. When this happens, the density state becomes a model input, and the injection variable $d$ associated with the eliminated conservation equation is also eliminated. When the density state and conservation equation are eliminated at the same node, this node then has the interpretation of a so-called “slack” node, whose definition follows.

**Definition 3.** A slack node is a node whose density (i.e. pressure) is specified and whose corresponding flux injection can take any unbounded instantaneous value to meet the specified density constraint.

The assumption behind the slack node is that it has the ability to pull from an infinite flux reservoir. This slack node is entirely analogous to the infinite bus in dynamical power system simulations, where complex power injection can take any value such that a specified complex voltage value is met.

**Simulation Model Statement:** To implement this simulation technique, we assume without loss of generality (WLOG) that the first node in $\Sigma$ will be treated as a slack node. We split system $G$ into the to-be eliminated conservation equation $G_1$ and the remaining equations $\tilde{G}$:

$$G = \begin{bmatrix} G_1 \\ \tilde{G} \end{bmatrix}. \quad (32)$$

We also split state variable vector $\mathbf{x}$ into the to-be eliminated density variable $\rho^{(1)}$ and the remaining variables $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} \rho^{(1)} \\ \mathbf{x} \end{bmatrix}. \quad (33)$$

We parse the system according to

$$\begin{bmatrix} M_{x1} & M_{x2} \\ M_{x3} & M_{x4} \end{bmatrix} \begin{bmatrix} \dot{\rho}^{(1)} \\ \dot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} G_1(\rho^{(1)}, \mathbf{x}) \\ G(\rho^{(1)}, \mathbf{x}) \end{bmatrix}, \quad (34)$$

and we eliminate the top equation in (34). Finally, we parse the second equation, such that

$$M_{x4} \dot{\mathbf{x}} = \tilde{G}(\rho^{(1)}, \mathbf{x}) - M_{x3}\dot{\rho}^{(1)}. \quad (35)$$

The final simulation model is given by

$$\dot{\mathbf{x}} = M_{x4}^{-1}(\tilde{G}(\rho^{(1)}, \mathbf{x}) - M_{x3}\dot{\rho}^{(1)}) \quad (36)$$

$$\dot{\rho}^{(1)} = \frac{d}{dt}\rho^{(1)} \quad (37)$$

where $\rho^{(1)}$ is specified by the user. If constant pressure is assumed, $\dot{\rho}^{(1)} = 0$ and the formulation simplifies. This process may be repeated for as many constant pressure slack node sources as may be desired. The primary drawback of this simulation technique is that source fluxes are treated as unconstrained injections. Thus, linepack depletion cannot be exploited properly in contingency situations, since the system has an infinite flux reservoir at its disposal.

**B. Technique 2: Finite Flux Reservoir via Sigmoid Function**

As a realistic modification to the infinite flux reservoir model, we may hypothesize the existence of a slack node with two nonidealities: (i) its flux output has a finite upper limit, and (ii) its density of injection decreases in value as flux injection saturates. Thus, density is treated as some algebraic function of flux. To create this nonideal slack node, we introduce a state variable which parameterizes the mass flux flowing from a slack source. In particular, we consider a situation where the reservoir at a source has an output flow limit; we may codify this limit with a sigmoid function. Replacing the injection variable $d$ by the injection function $\varphi_m S_1(z)$, the flux injection can be written as

$$\phi_0 = \varphi_m S_1(z), \quad (38)$$

where $S_1(z)$ is the sigmoid function given by

$$S_1(z) = \frac{e^z}{1 + e^z} \quad (39)$$

and the constant $\varphi_m$ represents the maximum upper limit of flux which the source can produce; this limit is approached as $z \to \infty$. As the flux flow saturates, rather than holding density constant, we wish to parameterize the density of the source node as a monotonically decreasing function of flux. To do so, we write

$$\rho = \varphi_m S_2(\phi_0), \quad (40)$$

where $\varphi$ is some nominal density of the node, and $S_2(z)$ is a sigmoid function flipped about the y-axis:

$$S_2(z) = \frac{e^\gamma(\varphi_M - \varphi_0)}{1 + e^\gamma(\varphi_M - \varphi_0)} \quad (41)$$

The constant $\varphi_M$ represents the flux value at which the density reaches half its nominal value, and the constant $\gamma > 0$ controls the speed at which the nodal density decreases as the flux
injection approaches saturation. Using the chain rule \( \frac{df}{dt} = \frac{df}{dx} \cdot \frac{dx}{dt} \), the flux and density function time derivatives are

\[
\dot{\phi}_0 = \phi_m (S_1(z) - S'_2(z)) \frac{\dot{z}}{h_1(z)} \\
\dot{\rho} = \Phi (S'_2(\phi_0) - S_2(\phi_0)) \frac{\dot{\phi}_0}{h_2(\phi_0)}.
\]

(42)

(43)

As a clarification, the slack pressure \( \rho \) is not a state variable, but knowledge of its derivative is still essential for simulating the network. We again assume, WLOG, that the first node in \( \Sigma_c \) is treated as a slack node, with constrained flux state \( \phi_0 \) and algebraic density \( \rho \). We also reorder system \( \mathbf{G} \) such that the conservation law equation associated with the slack node is altered to \( (42) \) and placed at the end of the equation vector. Thus, by borrowing the formulation from \( (34) \), the updated system may be described according to

\[
\begin{bmatrix}
M_{x3} & M_{x4} & 0 \\
0 & -e_k & h_1(z) \\
- \end{bmatrix}
\begin{bmatrix}
\dot{\rho} \\
\dot{x} \\
\dot{z}
\end{bmatrix}
= \begin{bmatrix}
\dot{\mathbf{G}}(\rho, \dot{x}) \\
0
\end{bmatrix}.
\]

(44)

where \( e_k \) is a row vector of zeros with a single 1 at index \( k \), i.e. the index of state variable \( \phi_0 \). Since \( M_{x3} \rho = M_{x3} h_2(\phi_0) \dot{\phi}_0 \), then \( (44) \) may be simplified to

\[
\begin{bmatrix}
M_{x5} \\
0 \\
- \end{bmatrix}
\begin{bmatrix}
\dot{x} \\
\dot{z}
\end{bmatrix}
= \begin{bmatrix}
\dot{\mathbf{G}}(\rho, \dot{x}) \\
0
\end{bmatrix}.
\]

(45)

If slack flux \( \phi_0 \) is the final state variable in \( \dot{x} \), then \( M_{x5} \) is

\[
M_{x5} = \left[ M_{x4}^{(1)} \mid M_{x4}^{(2)} \mid \cdots \mid M_{x4}^{(n)} + h_2(\phi_0) M_{x3} \right],
\]

(46)

where \( M_{x4}^{(i)} \) is the \( i \)th column of matrix \( M_{x4} \). By Theorem 4, matrix \( M_{x4} \) is nonsingular. The vector \( h_2(\phi_0) M_{x3} \) has its only nonzero (and negative definite) entry at the index of the slack node density state variable. When viewed as a perturbation of \( M_{x4} \), for all practical purposes, \( M_{x5} \) will also be a nonsingular matrix. To simulate the system \( (45) \), \( M_{x5} \) must be inverted. By inspection, the inverse of \( M_{x5} \) is given by

\[
M^{-1}_{x5} = \begin{bmatrix}
M^{-1}_{x4} & 0 \\
\frac{M^{-1}_{x4} (k,z)}{h_1(z)} & 1 \\
\frac{M^{-1}_{x4} (k,z)}{h_1(z)} & 1
\end{bmatrix}.
\]

(47)

where \( M^{-1}_{x4} (k,z) \) refers to the \( k \)th row of \( M^{-1}_{x5} \).

**Theorem 4.** Matrix \( M_{x5} \) of \( (45) \) is nonsingular if \( |z| < \infty \).

**Proof.** Since matrix \( M_{x5} \) is full rank, square matrix \( M_{x5} \) will clearly have full row rank as long as \( h_1(z) \neq 0 \). Assuming \( \phi_m \neq 0 \), \( h_1(z) = 0 \) iff \( S_1(z) = 0 \) or \( S_1(z) = 1 \), which only occur when \( z = \pm \infty \).

When a source flux injection is modeled by a sigmoid function, its state variable \( z \) will tend to “blow up” when the upper flow limit \( \phi_m \) is approached. This is problematic numerically, rather than physically, so a helpful workaround is to artificially constrain the growth of \( z \). This may be accomplished through the application of a constraint function \( d(z) = 1 + e^{z^2 - 10^2} \) to \( h_1(z) \). From \( (42) \),

\[
\dot{z} = \frac{\dot{\phi}_0}{h_1(z) \left( 1 + e^{z^2 - 10^2} \right)}.
\]

(48)

where the \( d(z)^{-1} \) term essentially ensures \( z \ll \infty \). The matrix inversion expression of \( (47) \) may be updated by simply replacing \( h_1(z) \) with \( h_1(z) \cdot d(z) \).

**Simulation Model Statement:** To state the model compactly, we augment the state variable vector \( \dot{\mathbf{x}} \) from \( (33) \) by adding \( z: \dot{x}_n = \dot{x} z \). We also augment \( \mathbf{G} \) by adding a zero to the bottom row to form augmented vector \( \mathbf{G}_a: \)

\[
\begin{bmatrix}
\dot{x}_m \\
\dot{x}_n
\end{bmatrix} = \begin{bmatrix}
M^{-1}_s \mathbf{G}_a(\rho, \dot{x}_n) \\
\rho = \mathbf{G}_a(\rho, \dot{x}_n)
\end{bmatrix}.
\]

(49)

(50)

This model may be used to investigate how a system responds when source mass flux flow limits are reached.

**C. Technique 3: Constant Flux Sources**

While Technique 1 assumed infinitely variable mass flux sources and technique 2 assumed constrained mass flux sources, technique 3 assumes constant (or specified) mass flux sources at all nodes. This is ultimately accomplished by converting a density state variable into an algebraic variable. According to Theorem 2, the deletion of a density state is a necessary condition for mass matrix nonsingularity. While Theorem 3 showed that we may delete a mass flux conservation equation to ensure this nonsingularity, the following theorem shows that we may instead delete a momentum conservation equation from \( (23) \) to ensure nonsingularity.

**Theorem 5.** Consider system \( \Sigma_c \). The elimination of a pressure state and a flux momentum equation in \( (23) \) from \( M_x \) will yield a square, full rank matrix \( M''_x \).

**Proof.** We borrow matrix \( M''_x \) from Theorem 3 but we alter its definition by the elimination of one row (row \( i \)) from the bottom subsection of \( M'_x \) instead of the top:

\[
M''_x = \begin{bmatrix}
K_0 & K_L \\
\Gamma_2 & \Gamma_2
\end{bmatrix}.
\]

(51)

We now assume corresponding line \( i \) connects nodes \( j \) and \( k \). Our goal is to show that matrix \( (51) \) has full row rank. To do so, we remove two columns from \( (51) \): column \( i \) and column \( i + m \). Each of these columns will have a single non-zero entry, at indices \( j \) and \( k \) respectively, so the row rank of the matrix necessarily drops by 2. By eliminating these columns, we are effectively eliminating line \( i \) from the incidence matrix and our system \( \Sigma_c \) is no longer fully connected. We now eliminate rows \( j \) and \( k \). This effectively eliminates nodes \( j \) and \( k \). By performing these operations, we are left with matrices which we define to be called \( K_0, K_L \), and \( \Gamma_2 \). Implicitly contained in \( K_0 \) and \( K_L \) are two tree-structured subgraphs which have both been reduced. By defining

\[
\tilde{M''}_x = \begin{bmatrix}
K_0 & K_L \\
\Gamma_2 & \Gamma_2
\end{bmatrix},
\]

(52)
we employ the same tactics used in the Theorem $3$ proof:

\[
\det(M''_x) = \det(K_0 - K_L \hat{\Gamma}_2^{-1}\hat{\Gamma}_2) \det(\hat{\Gamma}_2) = \det(K_0 - K_L) \sigma. \tag{53b}
\]

where $\sigma \neq 0$ since $\hat{\Gamma}_2$ is a diagonal matrix. Matrix $K_0 - K_L = |E|'$ is in fact a nonsingular matrix, because it represents the block diagonal concatenation of two reduced tree incidence matrices which are in themselves both nonsingular. Since the determinant in (53) is nonzero, then the following is implied:

\[
\text{rank}(M''_x) = \text{rank}(\hat{M}''_x) + 2 = n + m - 1. \tag{55}
\]

because full row rank of a square matrix implies full column rank. Thus, (51) is a full rank matrix. By direct extension, matrix $M''_x$, which is defined by

\[
M''_x = \begin{bmatrix}
0 & K_0 & K_L \\
\Gamma'_0 & 0 & 0 \\
0 & \Gamma'_2 & \Gamma'_2
\end{bmatrix}, \tag{56}
\]

is also full rank.

We now leverage the results of Theorem $3$ in order to define a simulation technique which allows for flux injections to be simultaneously specified at all nodes. To do so, we define a balancing node.

**Definition 4.** A balancing node is a node whose flux injection is specified and whose nodal density $\rho$ is transformed from a state variable into an algebraic variable.

Rather than becoming a specified input, as with the other two proposed simulation techniques, the balancing node’s density variable becomes an algebraic variable whose value can be computed analytically. The decision of which node is chosen to be this balancing node is an important point which shall be considered later in this subsection. For now, we consider some balancing node which is connected to the rest of the tree through a single link.\footnote{Any node in the network can be selected as the balancing node, so long as the dynamics of an interconnecting line can be described via (57).} The momentum equation associated with its interconnection line is defined as

\[
\frac{l}{2} (\dot{\phi}_l + \dot{\phi}_0) + a^2 (\rho_l - \rho_0) = -\frac{\lambda}{4D} (\dot{\phi}_l + \dot{\phi}_0) \frac{\dot{\phi}_l + \dot{\phi}_0}{\rho_l + \rho_0}, \tag{57}
\]

where we have neglected nodal indices for notational simplicity. At this balancing node, we assume flux is exogenously specified, WLOG, we further assume constant flux such that $\dot{\phi}_0 = 0$. Thus, we may form a quadratic equation in $\rho_0$:

\[
0 = \left[ -\frac{2a^2}{l} \rho_0^2 + \left( \dot{\phi}_l \right) \rho_0 + \left[ \frac{\lambda}{2D} (\dot{\phi}_l + \dot{\phi}_0) |\dot{\phi}_l + \dot{\phi}_0| \rho_l \rho_0 + \frac{2a^2}{l} \rho_0^2 \right] \right]. \tag{58}
\]

The value of $\rho_0$ is a function of state variables $\phi_0$, $\dot{\phi}_l$ and $\rho_l$ along with state variable derivative $\dot{\phi}_l$. Assuming all of these values are known numerically, $\rho_0$ can be computed analytically via the quadratic formula. We thus write $\rho_0$ as a function $g$ of state variable vector $x$ and derivative $\dot{\phi}_l$:

\[
\rho_0 = g(x, \dot{\phi}_l). \tag{59}
\]

To compute $\rho_0$, we must know $x$ and $\dot{\phi}_l$ numerically. The state variable values of $\phi_0$, $\dot{\phi}_l$ and $\rho_l$ are certain to be known (at each numerical time step). In the following theorem, we show that $\dot{\phi}_l$ can be numerically computed without numerical knowledge of $\rho_0$ or its time derivative.

**Theorem 6.** The numerical value of the flux derivative $\dot{\phi}_l$ at the far end of the line attached to a balancing node may be computed without knowledge of the density at the balancing node $\rho_0$ or its derivative $\dot{\rho}_0$.

**Proof.** Consider some system of equations given by

\[
K_1 \dot{\phi}_l + K_0 \dot{\phi}_0 = d \tag{60}
\]

\[
[\Gamma_2 (\dot{\phi}_l + \dot{\phi}_0)]' = [\Gamma_3 \rho - \Gamma_4 f (\phi_l, \phi_0, \Gamma_5 \rho)]', \tag{61}
\]

where the prime notation indicates the deletion of one momentum equation, i.e. (57). The mass matrix associated with this system exactly corresponds to (51) which has been shown to be nonsingular. If the momentum equation associated with the balancing node’s line is the equation which was deleted, then (61) will not contain the density of the balancing node or its derivative. Therefore, all flux flow derivatives, including $\dot{\phi}_l$, may be solved for through matrix inversion of (51), without knowledge of $\rho_0$ or $\dot{\rho}_0$. \hfill $\Box$

With the results of Theorem 6, we may assume there is some function $y$ which computes the necessary flux flow derivative value: $\dot{\phi}_l = y(x)$. Thus, we may solve (58) via the quadratic formula if we first substitute in $\dot{\phi}_l = y(x)$.

While the solution to (58) is necessary, we shall show that the model associated with this simulation technique additionally depends on the density derivative $\dot{\rho}_l$ at the balancing node. The value of $\dot{\rho}_0$ may be computed via the chain rule:

\[
\dot{\rho}_0 = \frac{d}{dt} g(x, \dot{\phi}_l) = \sum_i \frac{d}{dx_i} g(x, y(x)') x_i. \tag{62b}
\]

The vector $x_i$ will include flux flow derivatives as well as density derivatives. As proved in Theorem 6, the flux flow derivatives can be computed without $\rho_0$ or $\dot{\rho}_0$. The density derivatives are considered in the following theorem.

**Theorem 7.** Once a balancing node has been selected, the density state derivatives $\dot{\rho}_l$ may be uniquely computed.

**Proof.** When the expression $\Gamma_1 \dot{\rho} = \phi_0 - \phi_1$ from (22) has been altered such that a density state derivative is removed, $\Gamma_1$ loses a column and is transformed into $\Gamma'_1$ from (28). Because $\Gamma'_1$ has been shown to be a nonsingular matrix, the system in (22) may be solved by specifying a single density derivative (i.e. the one that was removed) if both $\phi_0$ and $\phi_1$ are known.

Alternatively, instead of specifying a density derivative state to solve (22), a new linear equation may be introduced which specifies the relationship of the “eliminated” state with the remaining states. Since (22) represents such a linear relationship, and since this relationship is physically independent from the processes used to model the relationships codified by (22), then a unique solution for the density derivative states may be solved for. \hfill $\Box$
We thus seek to use (62b) in order to add a new linearly independent row to \( \Gamma_1 \). To do so, we restate (62b) as

\[
\begin{bmatrix}
1 & -a_1 & \cdots & -a_n
\end{bmatrix} \hat{\rho} = b,
\]

where \( \hat{\rho} \) is the unknown vector, and known \( a_i, b \) are given by

\[
a_i = \frac{d}{dx_i} \Phi(x, y(x)), \quad i \in \{1 \ldots n\}
\]

\[
b = \sum_{i \notin \{1 \ldots n\}} \frac{d}{dx_i} \Phi(x, y(x)) x_i.
\]

Thus, we may build augmented matrix \( \Gamma_{1a} \) by appropriately concatenating matrix \( \Gamma_1 \) and vector \( \nu : \Gamma_{1a} = \Gamma_1 \nu \). Finally,

\[
\hat{\rho} = \Gamma_{1a}^{-1} \begin{bmatrix} \Phi_0 - \Phi_L \\ b \end{bmatrix}.
\]

**Simulation Model Statement:** In order to explicitly state the model associated with this simulation technique, we restate (64), but we reorder the system \( G \) into \( G' \) such that the first equation \( G'_1 \) is the momentum equation associated with the line attached to the balancing node:

\[
\begin{bmatrix}
M_{x1}^r & M_{x2}^r \\
M_{x3}^r & M_{x4}^r
\end{bmatrix}
\begin{bmatrix}
\dot{\rho}^{(1)} \\
\dot{x}
\end{bmatrix} =
\begin{bmatrix}
G'_1(\rho^{(1)}, \tilde{x}) \\
G'^r(\rho^{(1)}, \tilde{x})
\end{bmatrix},
\]

where the first density variable is also associated with the balancing node. This system may be solved according to

\[
\dot{x} = M_{x4}^{-1}(G'^r(\rho^{(1)}, \tilde{x}) - M_{x3}\dot{\rho}^{(1)})
\]

\[
\rho^{(1)} = \Phi(\tilde{x}, y(\tilde{x}))
\]

\[
\dot{\rho}^{(1)} = \frac{d}{dt} g(\tilde{x}, y(\tilde{x})),
\]

where the first equation from (67) is used to build function \( g \), the results of Theorem 2 are used to build function \( y \), and the results of Theorem 7 and (66) may be used for taking the derivative of the density. In a large network, it may be numerically expedient to write (68) as \( 0 = g(\rho, \nu, \phi_1) = q(\rho, x, y(x)) = Q(\rho, x) \). In noting that

\[
0 = \frac{dQ}{d\rho_0} \rho_0 + \sum_i \frac{dQ}{dx_i} \frac{dx_i}{dt},
\]

the time derivative \( \dot{\rho}_0 \) may be solved for by rearranging (71).

**Selecting the Balancing Node:** In a system where all flux injections are specified, it may not be obvious which node should be selected as the balancing node. The physical characteristics of this selection are important to consider, though. Since the balancing node’s density variable will transform into an algebraic variable, it will have the obligation of reacting instantaneously to any local imbalance or change in the system. For this reason, the node selected as the balancing node should generally be the node closest to the location of any system disturbance (e.g. loss of compressor, increase in load, loss of supply, leak in line). For an intuitive explanation of this rationale, we refer to the paragraph preceding Theorem 2. In this two-node example, we showed that there is inherent degeneracy for a load flux perturbation \( \epsilon_t \), but we explained that load density, rather than source density, should be responsible for instantaneous response: immediate propagation to the source wouldn’t be physical. Therefore, the load should be selected as the balancing node. This choice, though, should be thought of as a dynamic choice, because if there is a subsequent perturbation at the source, the source should then be reassigned as the balancing node. In this way, any time the system is perturbed, the balancing node should be reassigned nearest to the location of the perturbation.

**IV. Test Results**

In this section, we compare the previously derived simulation techniques via tests on the 20 node Belgium network; this system was reconstructed based on the model presented in [19], with some alterations. From Fig. 2 this radial network has 19 lines and two active compressors \( c_1 \) and \( c_2 \). The longest line in the network is 98 km long, but all lines are finitely discretized into \( l = 5 \) km sections. All simulation code and network data are posted online for open source access.

A. Test 1: Sharp Load Increase at Node 16

In this test, the load at node 16 was doubled\(^4\) in value over the course of 1000 seconds. The system was further simulated for 36 hours with the three alternative techniques.

1) Technique 1: Infinite Flux Reservoir: In this trial, source node 8 was chosen as the constant density slack node. The flux flow, demand, and injection variables for all 36 hours are seen in panel (a) of Fig. 3. Panel (b) shows the nodal densities in the system. In this trial, the slack node completely compensated for the increase in load and the system converged to a new, stable equilibrium.

2) Technique 2: Finite Flux Reservoir via Sigmoid Function: In this trial, source node 8 was converted from a constant density node to a node whose flux injection was constrained by (58) and whose density of injection was governed via (40). While the flux consumed by the load increased by 125 \( \text{kg/s} \cdot \text{m}^2 \), \( \phi_m \) was chosen such that the source could only provide an additional 75 \( \text{kg/s} \cdot \text{m}^2 \). Additionally, \( \gamma = 0.1 \) was selected. The results are portrayed in Fig. 4. As expected, the source saturated at 75 \( \text{kg/s} \cdot \text{m}^2 \) above its initial steady state.

\(^4\) github.com/SamChevalier/Natural-Gas-Simulations-TCNS

\(^3\) Increasing load by 125 \( \text{kg/s} \cdot \text{m}^2 \) increased total system load by 34%.
value, and $\rho_8$ did not hold constant. Due to a constant inherent flux imbalance, the linepack depletion phenomenon began, and the system experienced a pressure collapse as the density kept declining. This is in significant contrast with the Infinite Flux Reservoir simulation results shown in Fig. 3 in which case the flux was finally re-balanced and the system stabilized to a new equilibrium point. The simulation result further implies the survival time of the system during the collapse event. For example, if the pipeline density has a minimum requirement at, say, 30 kg/m$^3$, then Fig. 4 (b) indicates that the system can survive for 33 hours. However, in Fig. 5 the finite Flux Reservoir model suggests that the system can always survive. This comparison validates that our proposed Finite Flux Reservoir model can capture the slow dynamics of the linepack depletion for severe contingency evaluations, and it can provide an estimation of the system survival time.

3) Technique 3: Constant Flux Source: This model is more restrictive than the second one above since all the source node flux inputs are rigid without any flexibility. Thus, we expected that the system could only survive for a shorter period of time. In this trial, source node 8 was selected as the balancing node. Its density state was thus converted into an algebraic variable and its flux injection was held fixed. The results are shown in Fig. 5. The flux injection at both source nodes are held constant, but nodal densities drop dramatically due to the severe flux imbalance in the network. If we again consider the minimum density requirement to be 30 kg/m$^3$, the survival time was about 18 hours, which is 45.5% shorter than the survival time of the second model. At $t = 36$ hours, the linepack was completely depleted.

B. Test 2: Partial Loss of Compressor at Node 17

In this test, the compressor at node 17 was intentionally compromised. Its original density amplification value of 30.28% (i.e. $\alpha = 1.3028$) was decreased by 10% over the course of 200 seconds using a smooth sigmoid function. The system was further simulated for several hours with the three alternative techniques. Since source saturation wasn’t a significant factor in this experiment, the technique 1 and technique 2 results were sufficiently similar. For this reason, the finite flux reservoir results are not reported.

In applying technique 1, node 8 was again chosen as the constant density slack node, and in applying technique 3, node 8 was selected as the balancing node. The flux flows associated with the discretized lines downstream from the compressor (i.e. lines connecting nodes 18, 19 and 20) are plotted in Fig. 6. Panel (a) corresponds to the constant density simulation, while panel (b) corresponds to the constant flux simulation. After the compressor failure, there were significant flow reversals due to the sudden severe drop in pressure at the compressor node; these flows became increasingly less severe if one moves farther away from the compressor node. Until $t \approx 0.2$ hours, the local dynamics were approximately identical. The flux then rebounded more quickly in panel (b) due to an impending pressure spike which occurred at the source node. In order to hold flux constant against the back pressure of the compressor failure, the pressure needed to spike in the constant flux model. This ultimately allowed the system to rebound more quickly.

\[\text{This selection was made for comparison purposes. As previously noted, node 16 is actually the most suitable choice for balancing node, since this is the location of the disturbance.}\]

\[\text{In this work, the local flow reversal is accepted for the compromised compressor node. Future work will discuss restricted models when only mono-directional flow is permitted.}\]
than in the simulation with the constant density node, where density remained constant.

Unlike the previous contingency, the loss of a compressor did not cause a system-wide flux imbalance (there were local flux imbalances during the transients, of course). Eventually, both simulations found a new steady state. Interestingly, the resulting steady state density values found by each of the models were different, despite the fact that all flux injections were identical in both steady state solutions. Fig. 7 shows the gas density of the physical and intermediate nodes downstream of node 17. As can be seen, the density of the first simulation decreased more sharply in the initial two hours than it did in the second simulation. By hour 5, both systems were approaching convergence to a new steady state, but the first system converged to a steady state with lower density values than the second system. This is because the reference density for the first system was given at the source node, while the reference density for the second system was computed through \[(58)\] at the selected node.

V. CONCLUSION

In this paper, three alternative simulation models for studying dynamic linepack depletion in NGPNs are derived and discussed. The conventional model is shown to be degenerate under specified mass flux inputs for every node. Thus, it is not capable of characterizing linepack depletion in the case of a system-wide mass flux imbalance. The other two proposed models are shown to be non-degenerate under specified mass flux inputs. They are novel and present new opportunities for characterizing the survival time of NGPNs in the context of extreme contingencies. Using the proposed methods, we simulated and analyzed two specific potential contingencies: a sudden load increase, which could be caused by the emergency dispatch of gas-fired power plants or an unanticipated large scale leakage; and the partial loss of a compressor, which could be caused by the malfunctioning of a compressor station or loss of power. In terms of evaluating linepack depletion and system survival time, the simulation results validated the anticipations of the proposed models. Other contingencies, such as line leak, system bifurcation, or sudden loss of a source, will be characterized in the future publications. Additionally, this framework will be useful in characterizing the interdependency of different networks and the propagation of failures between electrical power systems and NGPNs.

REFERENCES

[1] B. Tyra, “Electric power monthly with data for june 2019,” Energy Information Administration, Tech. Rep., 2019.
[2] M. Johnson, “Electric power monthly with data for october 2001,” Energy Information Administration, Tech. Rep., 2002.
[3] V. Ekstrom, “Grid reliability and the role of natural gas,” Energy Futures, May 2014.
[4] M. Chertkov, M. W. Fisher, S. Backhaus, R. Bent, and S. Misra, “Pressure fluctuations in natural gas networks caused by gas-electric coupling,” 2015 48th Hawaii International Conference on System Sciences, pp. 2738–2747, 2015.
[5] P. Kundur et al., Power system stability and control. McGraw-Hill New York, 1994, vol. 7.
[6] R. Sauer and M. Pai, Power System Dynamics and Stability. Stipes Publishing L.L.C., 2006. [Online]. Available: https://books.google.com/books?id=W9PAAAACAAJ
[7] J. Glover, M. Sarma, and T. Overbye, Power Systems Analysis and Design. Cengage Learning, 2007. [Online]. Available: https://books.google.com/books?id=MAPV_vkuc00C
[8] A. J. Osiadacz, “Simulation of transient gas flows in networks.” 1984.
[9] A. Thorley and C. Tiley, “Unsteady and transient flow of compressible fluids in pipelines: A review of theoretical and some experimental studies,” International Journal of Heat and Fluid Flow, vol. 8, no. 1, pp. 3 – 15, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0142727087900430
[10] T. Mak, P. Van Hentenryck, A. Zlotnik, and R. Bent, “Dynamic compressor optimization in natural gas pipeline systems,” INFORMS Journal on Computing, vol. 31, 01 2019.
[11] F. Ulthoorn, “State-space estimation with a bayesian filter in a coupled pde system for transient gas flows,” Applied Mathematical Modelling, vol. 39, 07 2014.
[12] K. Wen, Z. Xia, W. Yu, and J. Gong, “A new lumped parameter model for natural gas pipelines in state space,” Energies, vol. 11, p. 1971, 07 2018.
[13] Y. Zhou, C. Gu, H. Wu, and Y. Song, “An equivalent model of gas networks for dynamic analysis of gas-electric systems,” IEEE Transactions on Power Systems, vol. 32, no. 6, pp. 4255–4264, Nov 2017.
[14] M. Behbahani-Nejad and Y. Shekari, “Reduced order modeling of natural gas transient flow in pipelines,” International Journal of Mechanical and Mechanatronics Engineering, vol. 2, no. 7, pp. 868 – 872, 2008. [Online]. Available: http://waset.org/Publications?p=19
[15] V. Gyrya and A. Zlotnik, “An explicit staggered-grid method for numerical simulation of large-scale natural gas pipeline networks,” Applied Mathematical Modelling, vol. 65, pp. 34 – 51, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0307904X1830606X
[16] S. Gruneld, L. Jansen, L. Hornung, T. Clees, C. Tischendorf, and P. Benner, “Model order reduction of differential algebraic equations arising from the simulation of gas transport networks,” in Progress in Differential-Algebraic Equations, S. Schöps, A. Bartel, M. Günther, E. J. W. ter Maten, and P. C. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 183–205.
[17] Y. Qiu, S. Gruneld, M. Stoll, and P. Benner, “Efficient numerical methods for gas network modeling and simulation.” 2018.