Accepted version

Aaron C. Zecchin, Martin F. Lambert and Angus R. Simpson
Inverse laplace transform for transient-state fluid line network simulation
Journal of Engineering Mechanics-ASCE, 2012; 138(1):101-115

http://doi.org/10.1061/(ASCE)EM.1943-7889.0000311

Inverse Laplace transform for transient-state fluid line network simulation

by

Zecchin, A.C., Simpson, A.R. and Lambert, M.F.

Journal of Engineering Mechanics

Citation:
Zecchin, A.C., Simpson, A.R. and Lambert, M.F. (2012). “Inverse Laplace transform for transient-state fluid line network simulation.” *Journal of Engineering Mechanics*, 138(1), 101–115, Online Publication Date: 30 Jul 2011, ISSN (print): 0733-9399, ISSN (online): 1943-7889.

The Inverse Laplace Transform for Transient-State Fluid Line Network Simulation

Aaron C. Zecchin[a,b,*], Martin F. Lambert[a,b], Angus R. Simpson[a,b]

[a] *School of Civil, Environmental and Mining Engineering*
[b] *The University of Adelaide*

Abstract

As surveyed within Zecchin (2010), inverse Laplace transform methods have a long history in the development of time-domain fluid line models. This paper presents a study combining the new Laplace-domain input/output (I/O) model derived from the network admittance matrix with the Fourier series expansion numerical inverse Laplace transform (NILT) to serve as a time-domain simulation model. A series of theorems are presented demonstrating the stability of the I/O model, which is important for the construction of the NILT method. In Zecchin (2010) the Fourier series expansion algorithm was studied, where qualitative relationships between the parameters and numerical errors were analysed, and reliable parameter heuristics were developed. These heuristics are used for a series of numerical examples dealing with networks of 11, 35, 51 and 94 pipes using the five different pipe models. The examples are used as the basis from which the accuracy and numerical efficiency of the proposed NILT are compared to the standard method of characteristics (MOC) model for transient pipeline networks. Findings show that not only is the proposed NILT is very efficient numerically in comparison to the pipe types involving convolution operations, but it is accurate for networks comprised of both linear and nonlinear pipe types.

Key words: pipeline transients; inverse Laplace transform; fluid line networks

*Corresponding author
Email address: azecchin@civeng.adelaide.edu.au (Aaron C. Zecchin)

1. Introduction

The existence of fluid line networks within many natural and man made systems has meant that stable, accurate and computationally efficient methods to model their transient behaviour is of broad interest. Much of the research literature has focused on the use of discrete partial differential equation (PDE) solvers, however, as outlined within Zecchin (2010) there has been significant interest on the development of time-domain models based on the inverse Laplace transform (ILT) of the Laplace-domain solutions of the fluid line equations. These approaches possess many advantages over their discrete counterparts in that they do not suffer from the stability, accuracy, and computational efficiency issues associated with the grid-based computations of discrete methods.

Many of the ILT-based methods represent elegant and novel approaches to inverting the Laplace transform (LT) of the fluid line equations. A limitation, however, is that most are formulated for only single pipelines, with few being formulated for a limited class of compound lines (Margolis and Yang, 1985; Yang and Tobler, 1991), and even fewer approaches being able to deal with the case of general networks (Suo and Wylie, 1989; Kojima et al., 2002). Despite this, no systematic and detailed study has been performed to compare ILT methods to their discrete counterparts for general fluid line network structures. This paper undertakes such a study. The time-domain numerical inverse Laplace transform (NILT) method adopted here consists of a combination of the Fourier series expansion method, studied in detail in Zecchin (2010), combined with the recently developed Laplace-domain network admittance model (Zecchin et al., 2009b,a). New results concerning the stability of the input/output (I/O) form of this model are presented. The proposed time-domain NILT method is compared to the method of characteristics (MOC) in terms of accuracy and computational efficiency for 20 different case studies (four different network structures and five different pipeline model types).

The paper is structured as follows. The system of fluid line network equations is outlined in Section 2, where the Laplace-domain solution of this system from Zecchin et al. (2009b) is described. Section 3 the outlines the framework for the proposed NILT model based

on the Laplace-domain in three stages: first the I/O Laplace-domain model is derived and its stability is proved; second the I/O time-domain model based on the inverse Laplace transform is outlined; and last the adopted NILT from Crump (1976) is presented. The case study details for the numerical experiments are outlined in Section 4. Section 5 presents the results and provides analysis for a range of different numerical experiments. The numerical experiments take two different forms: a study of the accuracy of the NILT; a study of the computational efficiency of the NILT. The conclusions are given in Section 6.

2. Laplace-Domain Network Model

To facilitate the discussion of the fluid line network equations, it is convenient to describe a network as a connected graph $\mathcal{G}(\mathcal{N}, \Lambda)$ (Diestel, 2000) consisting of the node set $\mathcal{N} = \{1, 2, ..., n_n\}$, and the link set $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_{n_\Lambda}\}$ and where $\lambda_j = (i_{uj}, i_{dj})$ where $i_{uj}, i_{dj} \in \mathcal{N}$ are the upstream and downstream nodes of link j respectively. Each node is associated with a lumped hydraulic component that is connected to a number of links, and each link is associated with a distributed pipe element where the directed nature of the link describes the positive flow direction sign convention of the element. There are two link sets associated with each node, these are Λ_{ui} and Λ_{di} which correspond to the set of links directed from and to node i respectively, that is $\Lambda_{ui} = \{(i,k), k \in \mathcal{N} : (i,k) \in \Lambda\}$ and $\Lambda_{di} = \{(k,i), k \in \mathcal{N} : (k,i) \in \Lambda\}$). Note that the first set corresponds to the links whose upstream node is i and the second set correspond to the links whose downstream node is i. With the given notation, a fluid line network be defined as the pair $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P})$ where $\mathcal{G}(\mathcal{N}, \Lambda)$ is the network graph of nodes $\mathcal{N}$ and links Λ, and $\mathcal{P} = \{\mathcal{P}_j : \lambda_j \in \Lambda\}$ is the set of pipeline properties where $\mathcal{P}_j$ are the properties for pipe j (*i.e.* length l_j, diameter, roughness *etc.*). The state space of the network $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P})$ is given by the distributions of pressure and flow along each line of the network, which are given by

$$\boldsymbol{p}(\boldsymbol{x}, t) = \begin{bmatrix} p_1(x_1, t) \\ \vdots \\ p_{n_\Lambda}(x_{n_\Lambda}, t) \end{bmatrix}, \quad \boldsymbol{q}(\boldsymbol{x}, t) = \begin{bmatrix} q_1(x_1, t) \\ \vdots \\ q_{n_\Lambda}(x_{n_\Lambda}, t) \end{bmatrix}, \tag{1}$$

respectively, where $\boldsymbol{x} = [x_1 \cdots x_{n_\Lambda}]^T$ is the vector of spatial coordinates, (*i.e.* $\boldsymbol{x} \in \mathcal{X} = \mathcal{X}_1 \times \cdots \mathcal{X}_{n_\Lambda}$ where $\mathcal{X}_j = [0, l_j]$), $t \in \mathbb{R}$ is time, and n_Λ is the number of links.

2.1. Network equations

For a given network $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P})$, a time-domain simulation involves the computation of the transient response of the states (1) for a specific set of specified initial and boundary conditions. Each node either has a controlled nodal pressure (as in the case of a reservoir) or a controlled nodal flow (as in the case of a controlled flow node or a junction). Therefore, partitioning the set of nodes as $\mathcal{N} = \mathcal{N}_J \cup \mathcal{N}_d \cup \mathcal{N}_r$ where $\mathcal{N}_J$ is the set of junctions, $\mathcal{N}_d$ is the set of demand nodes, and $\mathcal{N}_r$ is the set of reservoir nodes, the system of dynamic equations governing the network states (1) is

$$\frac{\partial p_j}{\partial x_j} + \frac{\rho}{A_j^2}\left(\frac{\partial q_j}{\partial t} + \mathcal{R}_j\left[q_j\right]\right) = 0, \quad x_j \in \mathcal{X}_j,\, \lambda_j \in \Lambda, \tag{2}$$

$$\frac{\partial q_j}{\partial x_j} + \frac{A_j^2}{\rho c_j^2}\left(\frac{\partial p_j}{\partial t} + \mathcal{C}_j\left[p_j\right]\right) = 0, \quad x_j \in \mathcal{X}_j,\, \lambda_j \in \Lambda, \tag{3}$$

$$p_j(\varphi_{ji}, t) - p_k(\varphi_{ki}, t) = 0, \quad \lambda_j, \lambda_k \in \Lambda_i,\, i \in \mathcal{N}_J \cup \mathcal{N}_d \tag{4}$$

$$p_j(\varphi_{ji}, t) - \psi_{ri}(t) = 0, \quad \lambda_j \in \Lambda_i,\, i \in \mathcal{N}_r, \tag{5}$$

$$\sum_{\lambda_j \in \Lambda_{di}} q_j(l_j, t) - \sum_{\lambda_j \in \Lambda_{ui}} q_j(0, t) = 0, \quad i \in \mathcal{N}_J \tag{6}$$

$$\theta_{di}(t) + \sum_{\lambda_j \in \Lambda_{di}} q_j(l_j, t) - \sum_{\lambda_j \in \Lambda_{ui}} q_j(0, t) = 0, \quad i \in \mathcal{N}_d \tag{7}$$

$$p_j(x, 0) = p_j^0(x),\, q_j(x, 0) = q_j^0(x), \quad x \in \mathcal{X}_j, \lambda_j \in \Lambda \tag{8}$$

where: for the fluid lines ρ is the fluid density, c_j, A_j, x_j are the fluid line wavespeed, the cross-sectional area, and the axial coordinate, and $\mathcal{R}_j$ and $\mathcal{C}_j$ are the integrodifferential line resistance and compliance operators (Zecchin, 2010); for the nodes ψ_{ri} is the controlled temporally varying reservoir pressure for the reservoir nodes in the reservoir node set $\mathcal{N}_r$, θ_{di} is the controlled temporally varying nodal demand for the demand nodes in the demand node set $\mathcal{N}_d$; p_j^0 and q_j^0 are the initial distribution of pressure and flow in each pipe $\lambda_j \in \Lambda$; and $\varphi_{ji} = l_j$ if $\lambda_j \in \Lambda_{di}$ and 0 otherwise.

The network equations (2)-(8) can be divided into five groups: (2) and (3) are the unsteady equations of motion and mass continuity for each fluid line; (4) and (5) are the nodal equations of equal pressures in pipe ends connected to the same node for junctions (nodes for which the inline pressure is the free variable) and reservoirs (nodes for which the nodal flow is the free variable) respectively; (6) and (7) are the nodal equations of mass conservation for junctions and demand nodes; and (8) are the initial conditions for the link states and node states.

2.2. Laplace-domain network admittance matrix

Based on (2)-(8) with a homogeneous initial condition (8) and linearised $\mathcal{R}_j$ and $\mathcal{C}_j$, Zecchin et al. (2009b) derived the form of the network admittance matrix mapping from the nodal pressures $\mathbf{\Psi}$ to the nodal flows $\mathbf{\Theta}$, where the nodal states are defined as

$$\mathbf{\Psi} = [\Psi_1 \cdots \Psi_{n_n}]^T, \quad \mathbf{\Theta} = [\Theta_1 \cdots \Theta_{n_n}]^T.$$

The main results of this work are briefly reviewed below.

The Laplace-domain solution of (2)-(3) can be organised into the end-to-end transfer matrix admittance form (Goodson and Leonard, 1972)

$$\begin{bmatrix} Q_j(s,0) \\ -Q_j(s,l_j) \end{bmatrix} = \frac{1}{Z_{cj}(s)} \begin{bmatrix} \coth \Gamma_j(s) & -\operatorname{csch} \Gamma_j(s) \\ -\operatorname{csch} \Gamma_j(s) & \coth \Gamma_j(s) \end{bmatrix} \begin{bmatrix} P_j(s,0) \\ P_j(s,l_j) \end{bmatrix}, \tag{9}$$

where Γ_j is the propagation operator, and Z_c is the characteristic impedance which are given by

$$\Gamma(s) = \frac{l_j}{c_j}\sqrt{[s + R_j(s)]\,[s + C_j(s)]}, \quad Z_{cj}(s) = \frac{c_j \rho}{A_j}\sqrt{\frac{s + R_j(s)}{s + C_j(s)}}$$

where R_j and C_j are the Laplace transforms of the linearised approximations of $\mathcal{R}$ and $\mathcal{C}$ respectively (typically the only term requiring linearisation is the steady-state quadratic term in $\mathcal{R}$, as for turbulent flow, $\mathcal{R}[q] = \overline{\mathcal{R}}[q] + O\left\{(q - q_o)^2\right\}$ where q_o is a reference flow rate (Wylie and Streeter, 1993)). The admittance matrix functions for each link $\lambda \in \Lambda$ can

be organised into the matrix form

$$\begin{pmatrix} \boldsymbol{Q}(s,\boldsymbol{0}) \\ -\boldsymbol{Q}(s,\boldsymbol{l}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{Z_c}^{-1}(s)\coth\boldsymbol{\Gamma}(s) & -\boldsymbol{Z_c}^{-1}(s)\operatorname{csch}\boldsymbol{\Gamma}(s) \\ -\boldsymbol{Z_c}^{-1}(s)\operatorname{csch}\boldsymbol{\Gamma}(s) & \boldsymbol{Z_c}^{-1}(s)\coth\boldsymbol{\Gamma}(s) \end{pmatrix} \begin{pmatrix} \boldsymbol{P}(s,\boldsymbol{0}) \\ \boldsymbol{P}(s,\boldsymbol{l}) \end{pmatrix}$$

with link state vectors

$$\boldsymbol{P}(s,\boldsymbol{x}) = \left[P_1(s,x_1),\ldots,P_{n_\Lambda}(s,x_{n_\Lambda})\right]^T, \quad \boldsymbol{Q}(s,\boldsymbol{x}) = \left[Q_1(s,x_1),\ldots,Q_{n_\Lambda}(s,x_{n_\Lambda})\right]^T,$$

and the diagonal link function matrices

$$\boldsymbol{\Gamma}(s) = \operatorname{diag}\left\{\Gamma_1(s),\ldots,\Gamma_{n_\Lambda}(s)\right\}, \boldsymbol{Z_c}(s) = \operatorname{diag}\left\{Z_{c1}(s),\ldots,Z_{cn_\Lambda}(s)\right\},$$

where $\boldsymbol{x} = [x_1,\ldots,x_{n_\Lambda}]^T$ is the vector of spatial coordinates, and $\boldsymbol{x} = \boldsymbol{0}$ ($\boldsymbol{x} = \boldsymbol{l}$) corresponds to all coordinates set at their start (or end) points. Defining the upstream and downstream node incidence matrices as

$$\left\{\boldsymbol{N_u}\right\}_{i,j} = \begin{cases} 1 & \text{if} \quad \lambda_j \in \Lambda_{u,i} \\ 0 & \text{otherwise} \end{cases}, \qquad \left\{\boldsymbol{N_d}\right\}_{i,j} = \begin{cases} 1 & \text{if} \quad \lambda_j \in \Lambda_{d,i} \\ 0 & \text{otherwise} \end{cases},$$

the upstream and downstream pressure and flow link variables can be related to the pressure and flow nodal variables by the matrix equations

$$\begin{pmatrix} \boldsymbol{P}(s,\boldsymbol{0}) \\ \boldsymbol{P}(s,\boldsymbol{l}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{N_u} & \boldsymbol{N_d} \end{pmatrix}^T \boldsymbol{\Psi}(s), \quad \begin{pmatrix} \boldsymbol{N_u} & \boldsymbol{N_d} \end{pmatrix} \begin{pmatrix} \boldsymbol{Q}(s,\boldsymbol{0}) \\ -\boldsymbol{Q}(s,\boldsymbol{l}) \end{pmatrix} = \boldsymbol{\Theta}(s),$$

which are expressions of the pressure preservation and mass conservation for a simple node, respectively (*i.e.* matrix versions of equations (4) and (6) respectively). Combining these link and node relationship expressions with the link functions (10) yields an admittance

matrix expression for the network dynamics

$$\boldsymbol{Y}(s)\boldsymbol{\Psi}(s) = \boldsymbol{\Theta}(s)$$

where $\boldsymbol{Y}(s)$ is the symetric $n_n \times n_n$ admittance matrix given by

$$\boldsymbol{Y}(s) = \begin{pmatrix} \boldsymbol{N_u} & \boldsymbol{N_d} \end{pmatrix} \begin{pmatrix} \boldsymbol{Z_c}^{-1}(s)\coth\boldsymbol{\Gamma}(s) & -\boldsymbol{Z_c}^{-1}(s)\operatorname{csch}\boldsymbol{\Gamma}(s) \\ -\boldsymbol{Z_c}^{-1}(s)\operatorname{csch}\boldsymbol{\Gamma}(s) & \boldsymbol{Z_c}^{-1}(s)\coth\boldsymbol{\Gamma}(s) \end{pmatrix} \begin{pmatrix} \boldsymbol{N_u} & \boldsymbol{N_d} \end{pmatrix}^T \quad (10)$$

which possesses the elementwise representation

$$\{\boldsymbol{Y}(s)\}_{i,k} = \begin{cases} \displaystyle\sum_{\lambda_j \in \Lambda_i} Z_{cj}^{-1}(s)\coth\Gamma_j(s) & \text{if } k = i \\ -Z_{cj}^{-1}(s)\operatorname{csch}\Gamma_j(s) & \text{if } \lambda_j \in \Lambda_i \cap \Lambda_k \\ 0 & \text{otherwise} \end{cases}. \quad (11)$$

An important property of most physical systems is that they are strictly passive. This means that the systems absorb or dissipate energy (Desoer and Vidyasagar, 1975), and is defined by the requirement that, at any point in time, the cumulative energy transfered into a system is greater than the energy transfered out of a system. It turns out that the nodal admittance matrix $\boldsymbol{Y}$ represents a strictly passive system conditional on the strict passivity of the network elements. This is defined within the following theorem.

Theorem 1. *The network admittance matrix $\boldsymbol{Y}$ in (10) for the network $(\mathcal{G}(\mathcal{N}, \Lambda), \mathcal{P})$ is strictly passive if the link admittance matrices for each $\lambda \in \Lambda$ are strictly passive.*

Proof. The matrix $\boldsymbol{Y}$ is strictly passive if (i) it is analytic in the open right hand complex plane $\mathbb{C}_+$, (ii) represents the transform of a real valued function, and (iii) has the property that $\mathbb{R}\mathrm{e}\{Y(s)\}$ is strictly positive definite for $s \in \mathbb{C}_+$ (see Desoer and Vidyasagar (1975) for details). As the terms in $\boldsymbol{Y}$ are simply additions of the terms in the link admittance functions, properties (i) and (ii) are clearly satisfied if all link admittance matrices are

strictly passive. For (iii), it is required that

$$\boldsymbol{x}^H \mathbb{R}\mathrm{e}\left\{\boldsymbol{Y}(s)\right\}\boldsymbol{x} > 0 \quad \text{for all } \boldsymbol{x} \in \mathbb{C}^{n_n} \text{ such that } ||\boldsymbol{x}|| > 0. \tag{12}$$

on $s \in \mathbb{C}_+$. It can be demonstrated that the quadratic form in (12) can be expressed as

$$\boldsymbol{x}^H \mathbb{R}\mathrm{e}\left\{\boldsymbol{Y}(s)\right\}\boldsymbol{x} = \sum_{(i,k)=\lambda_j \in \Lambda} \begin{bmatrix} x_i \\ x_k \end{bmatrix}^H \mathbb{R}\mathrm{e}\left\{ \frac{1}{Z_{cj}(s)} \begin{bmatrix} \coth\Gamma_j(s) & -\operatorname{csch}\Gamma_j(s) \\ -\operatorname{csch}\Gamma_j(s) & \coth\Gamma_j(s) \end{bmatrix} \right\} \begin{bmatrix} x_i \\ x_k \end{bmatrix}.$$

Each of the 2×2 quadratic terms within the summation involve the link admittance matrices as in (9). Each of these terms is clearly nonnegative provided that the link admittance matrices are strictly passive (*i.e.* they have a positive definite real part). Hence, the summation is positive provided that $||\boldsymbol{x}||$ is nonzero. □

Theorem 1 is conditional on the strict passivity of the link admittance functions. This was demonstrated in Zecchin (2010) to be conditional on the strict passivity of the resistive and compliance functions $\mathcal{R}$ and $\mathcal{C}$. For all physically realisable models, $\mathcal{R}$ and $\mathcal{C}$ are strictly passive as they do not create energy.

3. Framework for Network Time-Domain Simulations

The time-domain simulation of a $(\mathcal{G}(\mathcal{N},\Lambda),\mathcal{P})$ network involves computing the time varying unknown states of a network for a given hydraulic scenario, where a hydraulic scenario is defined as a well posed specification of the boundary conditions for the network (taking the initial conditions as homogeneous). Given a system with n_r reservoirs, and n_d demand nodes ($n_n = n_r + n_d$), in the context of the network equations (2)-(7), the boundary conditions are the nodal demands $\boldsymbol{\theta}_d$ (organised as a $n_d \times 1$ vector), and the reservoir pressures $\boldsymbol{\psi}_r$ (organised as a $n_r \times 1$ vector). Within the context of an I/O model, these boundary conditions serve as inputs, where the outputs can be taken as the unknown nodal states, that is the nodal pressures at the demand nodes $\boldsymbol{\psi}_d$ ($n_d \times 1$) and the nodal flows at the reservoir nodes $\boldsymbol{\theta}_r$ ($n_r \times 1$). Given that the Laplace transforms of $\boldsymbol{\theta}_d$ and $\boldsymbol{\psi}_r$ exist, the

Laplace-domain expression of the I/O transfer function is given by

$$\begin{pmatrix} \boldsymbol{\Psi}_d(s) \\ \boldsymbol{\Theta}_r(s) \end{pmatrix} = \boldsymbol{H}(s) \begin{pmatrix} \boldsymbol{\Theta}_d(s) \\ \boldsymbol{\Psi}_r(s) \end{pmatrix} \tag{13}$$

where $\boldsymbol{H}$ is the $n_n \times n_n$ I/O network transfer matrix. The focus in this section is the development of a time-domain simulation model based on (13).

3.1. Laplace-domain input/output model

The Laplace-domain form of $\boldsymbol{H}$ is now derived, and its stability is demonstrated. Ordering the nodal state vectors of pressure and flow as

$$\boldsymbol{\Psi}(s) = \begin{pmatrix} \boldsymbol{\Psi}_d(s) \\ \boldsymbol{\Psi}_r(s) \end{pmatrix}, \qquad \boldsymbol{\Theta}(s) = \begin{pmatrix} \boldsymbol{\Theta}_d(s) \\ \boldsymbol{\Theta}_r(s) \end{pmatrix}$$

the network matrix equation (10) can be expressed in the following partitioned form

$$\begin{pmatrix} \boldsymbol{Y}_d(s) & \boldsymbol{Y}_{d\text{-}r}(s) \\ \boldsymbol{Y}_{r\text{-}d}(s) & \boldsymbol{Y}_r(s) \end{pmatrix} \begin{pmatrix} \boldsymbol{\Psi}_d(s) \\ \boldsymbol{\Psi}_r(s) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Theta}_d(s) \\ \boldsymbol{\Theta}_r(s) \end{pmatrix} \tag{14}$$

where $\boldsymbol{Y}_d$ is the $n_d \times n_d$ system matrix for the subsystem comprised of the demand nodes, $\boldsymbol{Y}_r$ is the $n_r \times n_r$ system matrix for the subsystem comprised of the reservoir nodes, and $\boldsymbol{Y}_{d\text{-}r}$ ($\boldsymbol{Y}_{r\text{-}d}$) are the $n_d \times n_r$ ($n_r \times n_d$) partitions of the network matrix that corresponding to the nodal flow contribution at the demand (reservoir) nodes admitted from the nodal pressures at the reservoir (demand) nodes. Note that $\boldsymbol{Y}_d$ and $\boldsymbol{Y}_r$ are symmetric and $\boldsymbol{Y}_{d\text{-}r} = \boldsymbol{Y}_{r\text{-}d}^T$. Given this partitioning, Zecchin et al. (2009b) derived the I/O map (13) where

$$\boldsymbol{H}(s) = \begin{pmatrix} \boldsymbol{Y}_d^{-1}(s) & -\boldsymbol{Y}_d^{-1}(s)\,\boldsymbol{Y}_{d\text{-}r}(s) \\ \boldsymbol{Y}_{r\text{-}d}(s)\,\boldsymbol{Y}_d^{-1}(s) & \boldsymbol{Y}_r(s) - \boldsymbol{Y}_{r\text{-}d}(s)\,\boldsymbol{Y}_d^{-1}(s)\,\boldsymbol{Y}_{d\text{-}r}(s) \end{pmatrix}. \tag{15}$$

For computational reasons, it is necessary that $\boldsymbol{H}$ is a stable map. The stability of $\boldsymbol{H}$ is demonstrated to be dependent on the strict passivity of the network elements in the

following theorem.

Theorem 2. *For a given network* $(\mathcal{G}(\mathcal{N},\Lambda),\mathcal{P})$*, the I/O transfer matrix* $\boldsymbol{H}$ *defined by (15) is stable if all the admittance matrices for all links* $\lambda \in \Lambda$ *are strictly passive.*

Proof. The matrix $\boldsymbol{H}$ is stable provided all its elemental functions are analytic on $s \in \mathbb{C}_+$, which holds only if $\boldsymbol{Y}_r$, $\boldsymbol{Y}_{d\text{-}r}$, $\boldsymbol{Y}_{r\text{-}d}$ and $\boldsymbol{Y}_d^{-1}$ all have analytic elemental functions within this domain. As the matrices $\boldsymbol{Y}_r$, $\boldsymbol{Y}_{d\text{-}r}$, $\boldsymbol{Y}_{r\text{-}d}$ are submatrices of the strictly passive $\boldsymbol{Y}$, they are thus analytic on $s \in \mathbb{C}_+$ (strict passivity of $\boldsymbol{Y}$ is ensured by Theorem 1). The matrix $\boldsymbol{Y}_d$ is strictly passive as it is a principal minor of the strictly passive $\boldsymbol{Y}$ (Zecchin, 2010). Hence $\boldsymbol{Y}_d^{-1}$ itself is also strictly passive (Zecchin, 2010), and being strictly passive, it is also stable (Triverio et al., 2007). □

3.2. Time-domain input/output model

By the convolution theorem of the ILT (Franklin et al., 2001), the time-domain representation of (13) is given by

$$\begin{pmatrix} \boldsymbol{\psi}_d(t) \\ \boldsymbol{\theta}_r(t) \end{pmatrix} = \int_0^t \boldsymbol{h}(t-\tau) \begin{pmatrix} \boldsymbol{\theta}_d(\tau) \\ \boldsymbol{\psi}_r(\tau) \end{pmatrix} \mathrm{d}\tau \tag{16}$$

where the lower case symbols are the time-domain counterparts of their Laplace transforms. Since the impulse response matrix $\boldsymbol{h}$ is not analytically available, for any time point t, the computation of (16) first requires (i) the computation of $\boldsymbol{h}(\tau)$ on $\tau \in [0,t]$ via the ILT of $\boldsymbol{H}(s)$, and (ii) the convolution operation of $\boldsymbol{h}$ with the inputs. That is, the outputs are computed by

$$\int_0^t \mathcal{L}^{-1}\left\{\boldsymbol{H}(s)\right\}(t-\tau) \begin{pmatrix} \boldsymbol{\theta}_d(\tau) \\ \boldsymbol{\psi}_r(\tau) \end{pmatrix} \mathrm{d}\tau. \tag{17}$$

In the interest of computational efficiency, (17) can be calculated more efficiently by computing the impulse response $\mathcal{L}^{-1}\left\{\boldsymbol{H}(s)\right\}(\tau), \tau \in [0,t]$ *a priori* so that only the convolution would require computation at each time point. This approach still requires the NILT of each

elemental function of $\boldsymbol{H}$, which is still computationally very expensive. A more computationally efficient strategy of calculating (16) is

$$\begin{pmatrix} \boldsymbol{\psi}_d(t) \\ \boldsymbol{\theta}_r(t) \end{pmatrix} = \mathcal{L}^{-1} \left\{ \boldsymbol{H}(s) \begin{pmatrix} \boldsymbol{\Theta}_d(s) \\ \boldsymbol{\Psi}_r(s) \end{pmatrix} \right\} (t), \tag{18}$$

that is, the convolution is performed in the Laplace-domain, and the resulting function is then inverted. Despite the fact that both approaches involve the calculation of $\boldsymbol{H}(s)$ at discrete points along some contour $s = a + \mathrm{i}\omega_i, i = 0 \ldots, N$, the calculation of (18) is computationally simpler than (17) for two reasons:

1. Equation (18) requires a far reduced number of NILT calculations. As mentioned, (17) involves the NILT of all the elemental functions of $\boldsymbol{H}(s)$, which comprises $(n_d + n_r) \times (n_d + n_r)$ functions. In comparison, for the computation of the matrix multiplication for (17) in the Laplace-domain, only $(n_d + n_r)$ functions require numerical inversion, which is the square root of the NILTs required by (17).
2. Equation (17) requires the calculation of a convolution at each time point, and (18) does not. Despite the fact that, for a hydraulic scenario, the nodal boundary conditions $\boldsymbol{\theta}_d(t)$, and $\boldsymbol{\psi}_r(t)$ are typically specified as functions of time, the Laplace transforms are typically analytically available. This fact reinforces the computational efficiency of the convolution calculation in the Laplace-domain.

Therefore, given these merits (18) is the framework of the NILT model that has been adopted in this research.

3.3. Numerical inverse Laplace transform

Given the Laplace transformable function f with Laplace transform $F = \mathcal{L}\{f\}$, then the function f can be expressed as the Bromwich contour integral over F given by

$$f(t) = \mathcal{L}^{-1}\{F\}(t) = \frac{1}{2\pi\mathrm{i}} \int_{a-\mathrm{i}\infty}^{a+\mathrm{i}\infty} F(s) e^{st} \mathrm{d}s \tag{19}$$

where a is any number such that the location of all singularities of F have a real part less than a. For the application of interest here, the Laplace-domain function for the i-th output is given by

$$F(s) = \boldsymbol{H}_i(s) \begin{pmatrix} \boldsymbol{\Theta}_d(s) \\ \boldsymbol{\Psi}_r(s) \end{pmatrix} \tag{20}$$

where $\boldsymbol{H}_i$ is the i-th row of $\boldsymbol{H}$. From Theorem 2, it is known that all elemental functions of $\boldsymbol{H}$ are stable. Therefore, given that $\boldsymbol{\theta}_d$ and $\boldsymbol{\psi}_r$ are exponentially bounded functions, it holds that appropriate values of the contour location a are given by $a \geq \alpha$ where α is some real number such that

$$|\psi_{ri}(t)|, |\theta_{dj}(t)| \leq Me^{\alpha t}, \quad i = 1, \ldots, n_r, \ j = 1, \ldots, n_d$$

for $t \in \mathbb{R}_+$ where M is a real constant. For most scenarios of interest, the boundary perturbations are of finite energy, or they approach a finite valued upper bound (such as a step input resulting from a valve closure). In such cases, it holds that $a > 0$ is appropriate.

In all but the most simple instances, (19) can only be calculated using NILT methods. The work Zecchin (2010) provided a detailed survey of methods dealing with simple pipeline networks, for which analytic approximations to f can be obtained. As arbitrary networks are under consideration here, numerical methods must be pursued. Within this work, the Fourier series expansion algorithm (Crump, 1976; Abate and Whitt, 1992) is used as the NILT approximation to f, and is given by

$$\begin{aligned} f(t) &\approx \tilde{f}(t|a, N, \Delta\omega) \\ &= \frac{e^{at}\Delta\omega}{\pi} \left[\frac{F_0}{2} + \sum_{k=1}^{N} \mathbb{R}\mathrm{e}\left\{F_k\right\} \cos\left(k\Delta\omega t\right) - \mathbb{I}\mathrm{m}\left\{F_k\right\} \sin\left(k\Delta\omega t\right) \right]. \end{aligned} \tag{21}$$

where $F_k = F(a + \mathrm{i}k\pi\Delta\omega), k = 0, 1, \ldots, N$, and $\Delta\omega$ is the discretisation interval on the line $\mathbb{R}\mathrm{e}\left\{s\right\} = a$ (refer to (Zecchin, 2010) for details).

Given the harmonic nature of $F(s)$ for pipe line systems, Zecchin (2010) proposed the reparameterisation dealing with the number of harmonics used in the inversion N_H, and the

Table 1: Case study details for the four networks. Square braces indicate the interval range of the properties.

Network property	Network case study			
	11-pipe	35-pipe	51-pipe	94-pipe
number of nodes	6	20	35	87
nodal demands (L/s)	126	[0.85, 117]	[28, 142]	[42, 420]
pipe diameters (mm)	240	[254, 1524]	[305, 1524]	[250, 1200]
pipe lengths (m)	[457, 1372]	[883, 3109]	[450, 994]	[10, 4800]

number of discretisations of each harmonic N_Δ, where the following relationships hold

$$N = N_H \cdot N_\Delta, \quad \Delta\omega = \frac{\Delta\Omega}{N_\Delta}.$$

where $\Delta\Omega$ is a nominal frequency bandwidth for the harmonics of the network.

4. Case Study Details

The primary interest within this paper is the suitability of the linear NILT approach for the time-domain simulation of pipeline networks comprised of both linear and nonlinear pipes. The important issues pertaining to the suitability of the NILT method are (i) the accuracy of the method to approximate the true dynamics, and (ii) the relative computational efficiency of the method with respect to alternative simulation approaches. To undertake this analysis, many numerical experiments were undertaken comparing the proposed NILT method combining (20) and (21) with the commonly used MOC approach (Chaudhry, 1987; Wylie and Streeter, 1993). Within the experiments, 20 different case studies were considered comprised of four different networks with five different pipeline models. These, as well as the adopted parameter settings for the NILT are outlined in the following.

4.1. Case study networks

The four different networks used within in the study are depicted in Figures 1 to 4, and their properties are summarised in Table 1. The 11-pipe network is adapted from Pudar and

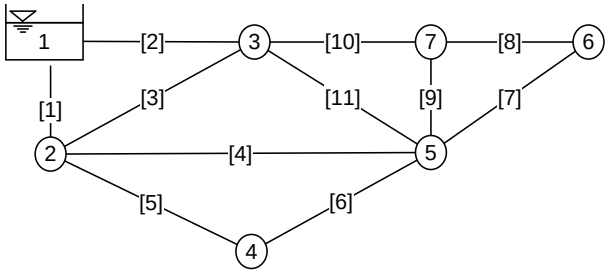

Figure 1: The 11-pipe network.

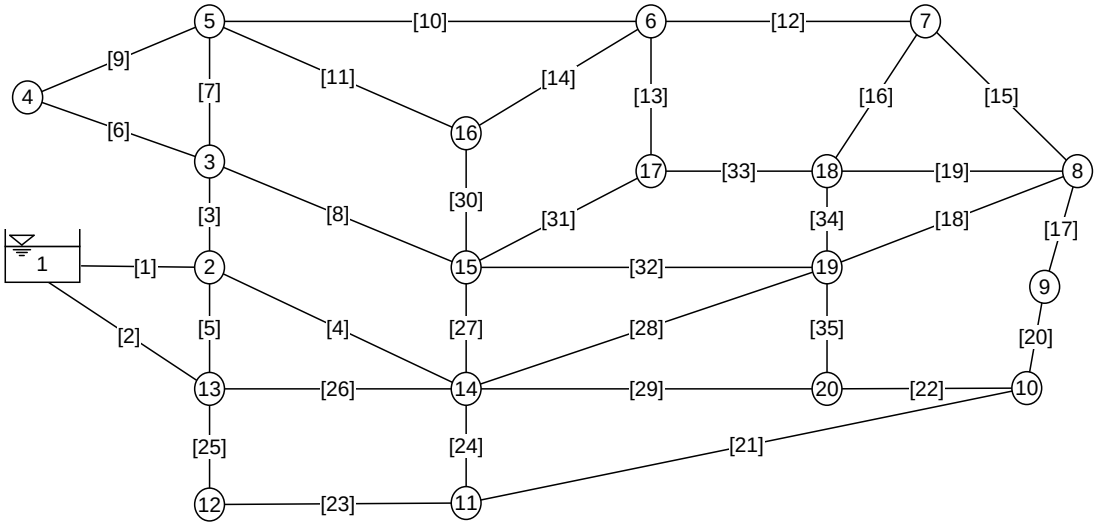

Figure 2: The 35-pipe network.

Liggett (1992), the 35-pipe and 51-pipe networks from Vítkovský (2001), and the 94-pipe network from Datta and Sridharan (1994). Interested readers are referred to Zecchin (2010) for all network details.

4.2. Pipeline models

Each of the networks were studied with five different pipe types, namely the laminar-steady-friction (LSF) model (Wylie and Streeter, 1993), the turbulent-steady-friction (TSF) model (Wylie and Streeter, 1993), the laminar-unsteady-friction (LUF) model (Zielke, 1968), the turbulent-unsteady-friction (TUF) model (Vardy and Brown, 2007), and the viscoelastic

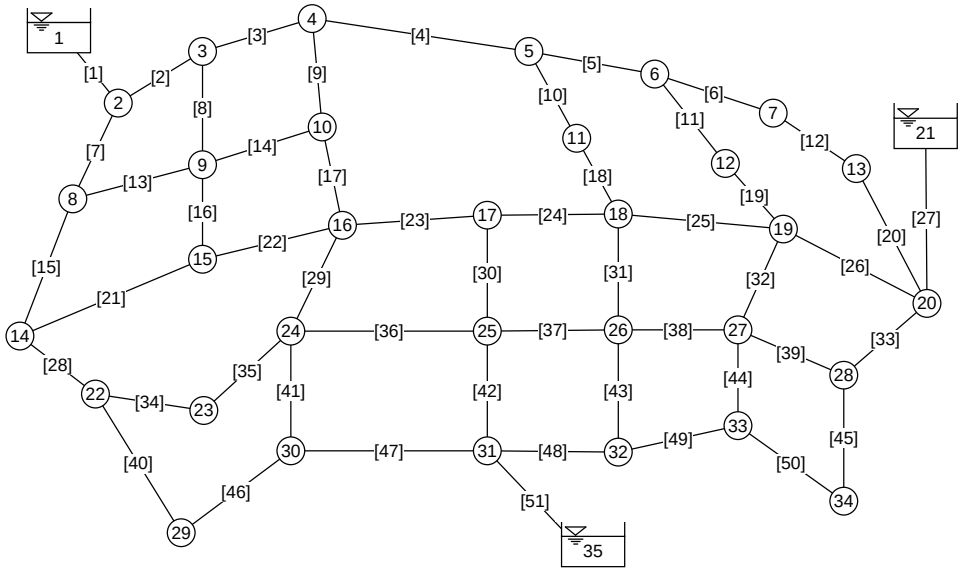

Figure 3: The 51 pipe network.

(VE) model (Rieutord and Blanchard, 1979). The models LSF, TSF, LUF, and the TUF are different forms of the resistive operator $\mathcal{R}$, and VE is a different model for the compliance operator $\mathcal{C}$. The most basic steady-state models, LSF and TSF, are given by the quasi-steady expressions

$$\mathcal{R}_{\text{LSF}}[q](x,t) = 32\frac{\nu}{D^2}q(x,t), \quad \mathcal{R}_{\text{TSF}}[q](x,t) = \frac{f}{2DA}|q(x,t)|q(x,t)$$

where ν is the kinematic viscosity, and f is the Darcy-Weisbach friction factor (Wylie and Streeter, 1993). The only nonlinearity within the all the models is the quadratic term in $\mathcal{R}_{\text{TSF}}$ which is linearised as

$$\overline{\mathcal{R}}_{\text{TSF}}[q](x,t) = \frac{fq_o}{AD}q(x,t)$$

where q_o is the operating flow about which the linearisation is taken (the steady-state flow). The steady-state models do not account for the influence of the time-varying cross-sectional velocity profiles on the resistance function. Working from the basis of 2-D axisymetric flow, Zielke (1968) and Vardy and Brown (2003, 2004, 2007) derived unsteady friction terms

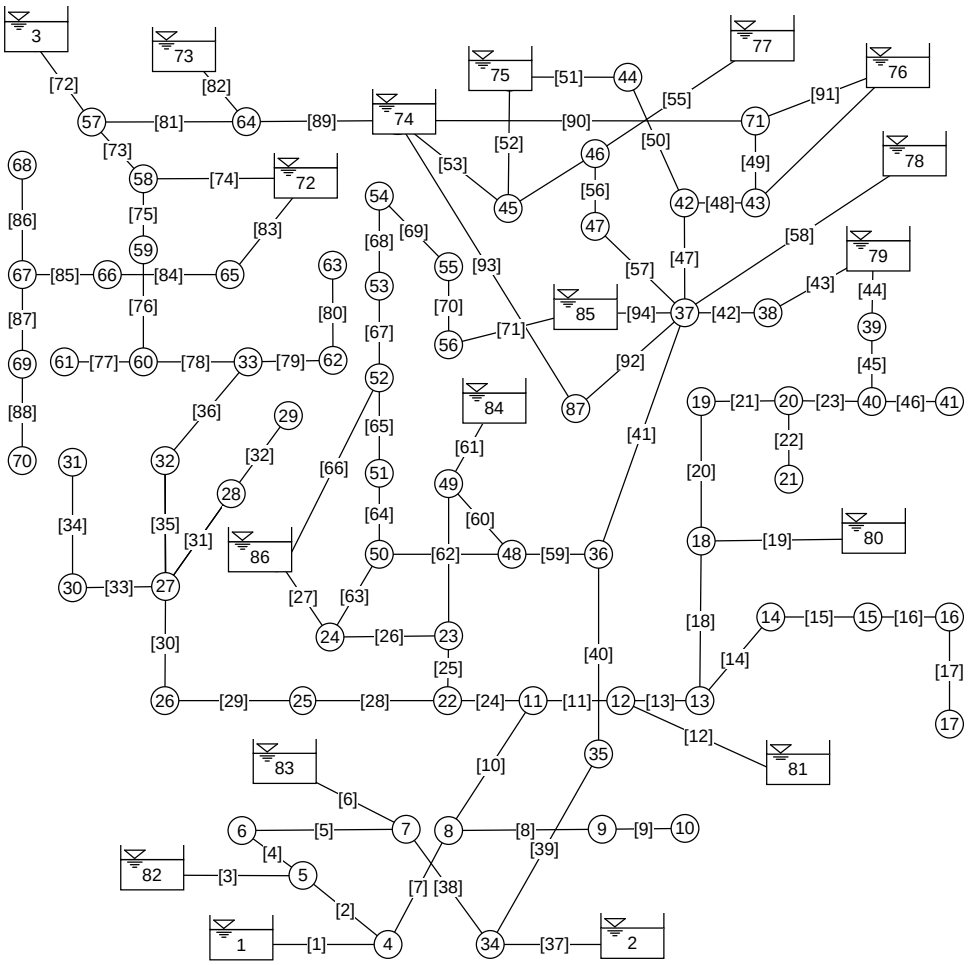

Figure 4: The 94-pipe network.

for laminar and turbulent flow, respectively. The incorporation of the the unsteady term within the resistance function models takes the form of a convolution operation on the flow acceleration, and these models are given by the following modifications on their steady-state counterparts

$$\mathcal{R}_{\mathrm{LUF}}[q](x,t) = \mathcal{R}_{\mathrm{LSF}}[q](x,t) + 16\frac{\nu}{D^2}\int_0^t w_{\mathrm{LUF}}(t-\tau)\frac{\partial q}{\partial t}(x,\tau)\mathrm{d}\tau$$

$$\mathcal{R}_{\mathrm{TUF}}[q](x,t) = \mathcal{R}_{\mathrm{TSF}}[q](x,t) + 16\frac{\nu}{D^2}\int_0^t w_{\mathrm{TUF}}(t-\tau)\frac{\partial q}{\partial t}(x,\tau)\mathrm{d}\tau$$

where w_{LUF} and w_{TUF} are weighting functions.

For all the models LSF, TSF, LUF, and TUF, the pipe is taken as elastic with $\mathcal{C} \equiv 0$, however, for the VE model, the compliance operator is given by Rieutord and Blanchard (1979)

$$\mathcal{C}_{\mathrm{VE}}[p](x,t) = \rho c^2 \frac{\alpha D}{eE} \int_0^t \frac{\partial J}{\partial t}(t-\tau)\frac{\partial p}{\partial t}(\tau)\mathrm{d}\tau$$

where α is the restraint parameter, E is the pipe materials Young's modulus, e is the pipe wall thickness, and J is the material compliance curve.

For all cases, w_{LUF} has been taken as the 10-term exponential approximation from Vítkovský et al. (2004), w_{TUF} has been taken as the 13-term approximation from Vardy and Brown (2007), $\partial J/\partial t$ has been taken as the single term Kelvin-Voigt model for mildly viscoelastic mortar-lined steel pipes (Stephens, 2008), $c = 1000$ m/s (unless otherwise specified), $f = 0.02$ for $\mathcal{R}_{\mathrm{TSF}}$, $\epsilon/D = 0.001$ for $\mathcal{R}_{\mathrm{TUF}}$, and $e/D = 0.1$ for $\mathcal{C}_{\mathrm{VE}}$ where all pipes are only restrained at the ends.

4.3. Parameter settings

Within Zecchin (2010), a detailed parametric sensitivity analysis was undertaken applying (21) to a series of dimensionless single pipeline system. Heuristics were developed for the parameters a, N_H and N_Δ. However, as the Laplace variable s for the dimensionless system is nondimensionalised by a factor of the pipeline period $T_l = l/c$ (that is $\tilde{s} = T_l s$ where the tilde denotes properties pertaining to the dimensionless system), a reinterpretation of the parameters that are directly related to the s domain (a and the nominal harmonic bandwidth $\Delta\Omega$ upon which N_Δ depends) are required for dimensional systems. For single pipe systems, the reinterpretation is simply a multiplication of the dimensionless values by a factor of $1/T_l$. However, for a network $(\mathcal{G}(\mathcal{N},\Lambda),\mathcal{P})$ containing many pipes, there is no single period as each pipeline has a different period. Based on trialling a number of different statistics, a preliminary analysis demonstrated that the most suitable choice for the indicative period of a network period (denoted by T_l^*) is

$$T_l^* = \max_{\lambda\in\Lambda} \{T_{l\lambda}\}, \tag{22}$$

which is the period associated with the largest pipeline period within the network. The reason for this is twofold. Firstly, as a result of the e^{at} term in the integration (21), it is important to keep $a = \tilde{a}/T_l$ small hence motivating the use of larger values of T_l^*. The second reason is to do with the spacing of the system harmonics throughout $\mathbb{C}$. The harmonic spacing for nondimensional pipeline is approximately $\Delta\widetilde{\Omega} = \pi/2$. However, multi-pipe networks do not contain uniformly spaced harmonics, hence the spacing between the harmonics along an integration contour can be highly irregular. From this perspective, it is necessary to adopt a nominal harmonic width that is small enough to adequately discretise the harmonics with the thinnest bandwidth. Such harmonics are those associated with the longest period, and hence $\Delta\Omega = \pi/(2T_l^*)$ has been taken as the nominal harmonic bandwidth.

To demonstrate the robustness of the method, the parameters were not calibrated to each case study, but the heuristics were used with $\tilde{a} = 0.07$ and $N_\Delta = 41$ (as suggested in Zecchin (2010)), and a range of N_H was used ($N_H \in \{250, 500, 1000, 2000\}$) to study the accuracy versus computational efficiency trade-off.

5. Computational Results and Analysis

Examples of specific case studies are used to highlight important issues relating to the accuracy (Section 5.1) and computational efficiency (Section 5.2) of the NILT. Following the examples, general results observed for all case studies are discussed at length.

5.1. Accuracy studies

Five detailed examples are firstly presented and discussed below, where each example is based on the 51-pipe network whose pipes are comprised of one of the five pipe types[1]. These examples are used to explore the issues pertaining to each different pipe type. Following these examples, a detailed analysis of the results for all four networks is given, where general conclusions are drawn.

[1]It is recognised that in the case of the laminar pipe models, the assumption of laminar flow may be violated. This however is not of concern within this work, as the emphasis is on the ability of the NILT to approximate the MOC for a range of different models for $\mathcal{R}$. That is, the comparative dynamical behaviour is of primary interest.

5.1.1. Qualitative analysis of different pipeline models for the 51-pipe network

The results for the experiments with the 51-pipe network with the five different pipeline models is presented within this section. For each case, the network was excited into a transient state by temporarily halting the demand at nodes $\{12, 17, 27, 30\}$ for a period of $\{1.0, 0.5, 0.3, 0.4\}$ seconds. The pressure responses of the network at node 25 for the first 100 s, as computed by the MOC on a temporal grid of $\Delta t = 0.001$ s (required for a Courant number of 1), are given in subfigure (a) for Figures 5-9 respectively for models LSF, TSF, LUF, TUF and VE. Within these figures, subfigures (b)-(d) show the error functions for the NILT approximations for different values of $N_H = 250$, 500, 1000, where $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$.

From the experiments for the LSF model, a consideration of the $E_{1000}(t)$ error function shows that an extremely accurate simulation for this pipe type with the NILT is achievable. Comparing the error functions $E_{1000}(t)$, $E_{500}(t)$, and $E_{250}(t)$ it is observed that there is an order of magnitude increase in the accuracy when doubling the number of harmonics included in the NILT from 250 to 500, and that the increase from 500 to 1000 harmonics yields a near indistinguishable error. The highest error in the NILT occurred in the first stages of the pressure response, where closer analysis shows that the error was associated with a Gibbs-type oscillation in the NILT approximation resulting from the sharpness of the pressure wave. For the larger time scales, the errors remain within reasonably small bounds (*i.e.* $|E_{1000}(t)| < 1$ kPa, $|E_{500}(t)| < 3$ kPa, and $|E_{250}(t)| < 10$ kPa).

Similarly with the TSF experiments, the error functions $E_{250}(t)$, $E_{500}(t)$, and $E_{1000}(t)$ in Figure 6 exhibit their maximum error in the early stages of the pressure response due to the sharpness of the pressure front at this time. However, a qualitativly different behaviour of the error functions is observed for the TSF experiments as opposed to that for the LSF experiments. For the smaller time points the errors appear to follow a trend (as opposed to being approximately uniformly distributed about the 0 level) and at the larger time points, the errors do not consist of high frequency oscillations as in for the LSF, but they have a much lower variability, particularly in the case of the $E_{250}(t)$ function.

These observed differences can be explained by considering the dynamic behaviour of

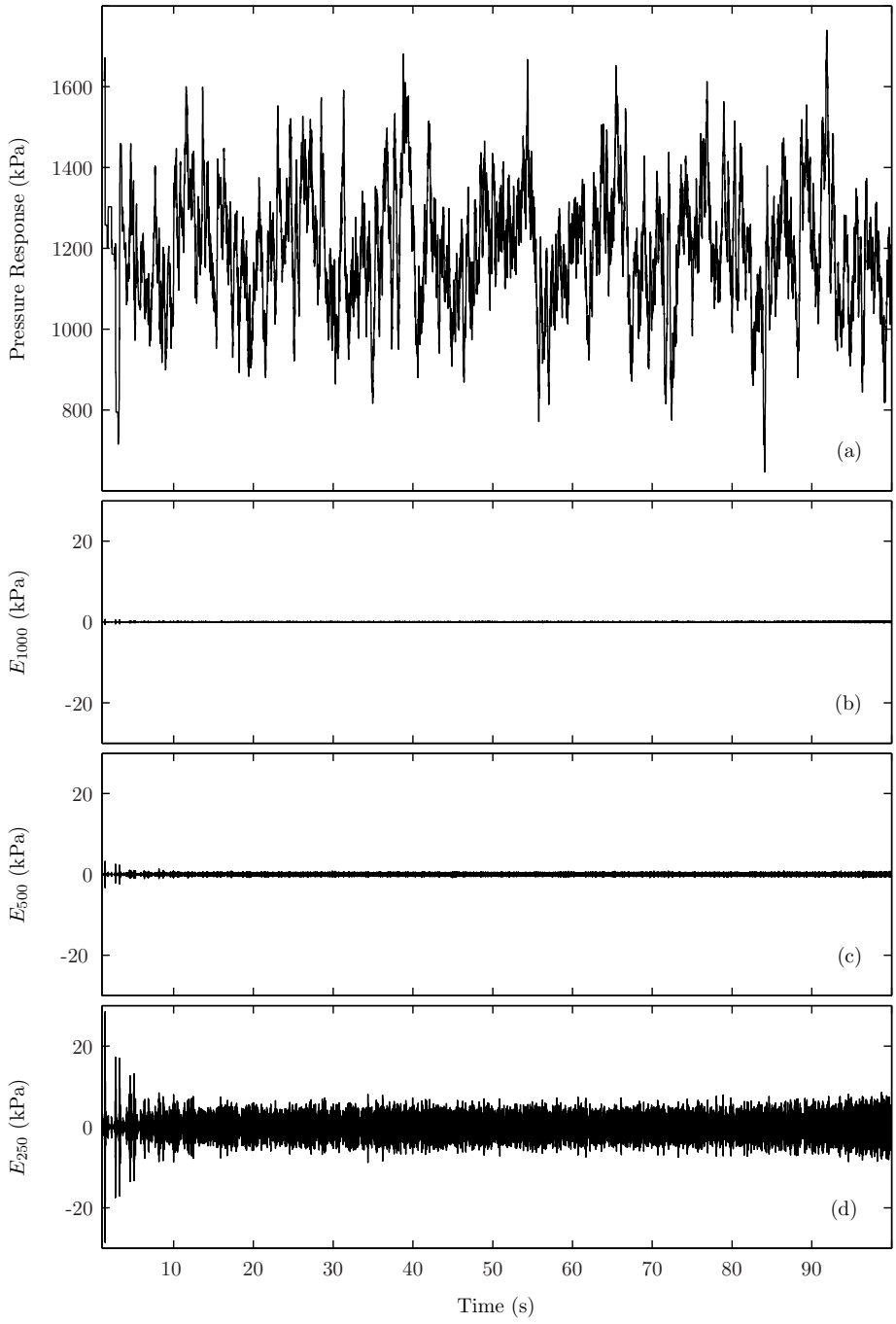

Figure 5: Comparison of MOC and the NILT for the laminar-steady-friction (LSF) 51-pipe network for the pressure response at node 25, where the subfigures show: (a) the pressure response $f(t)$ computed by the MOC; and (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

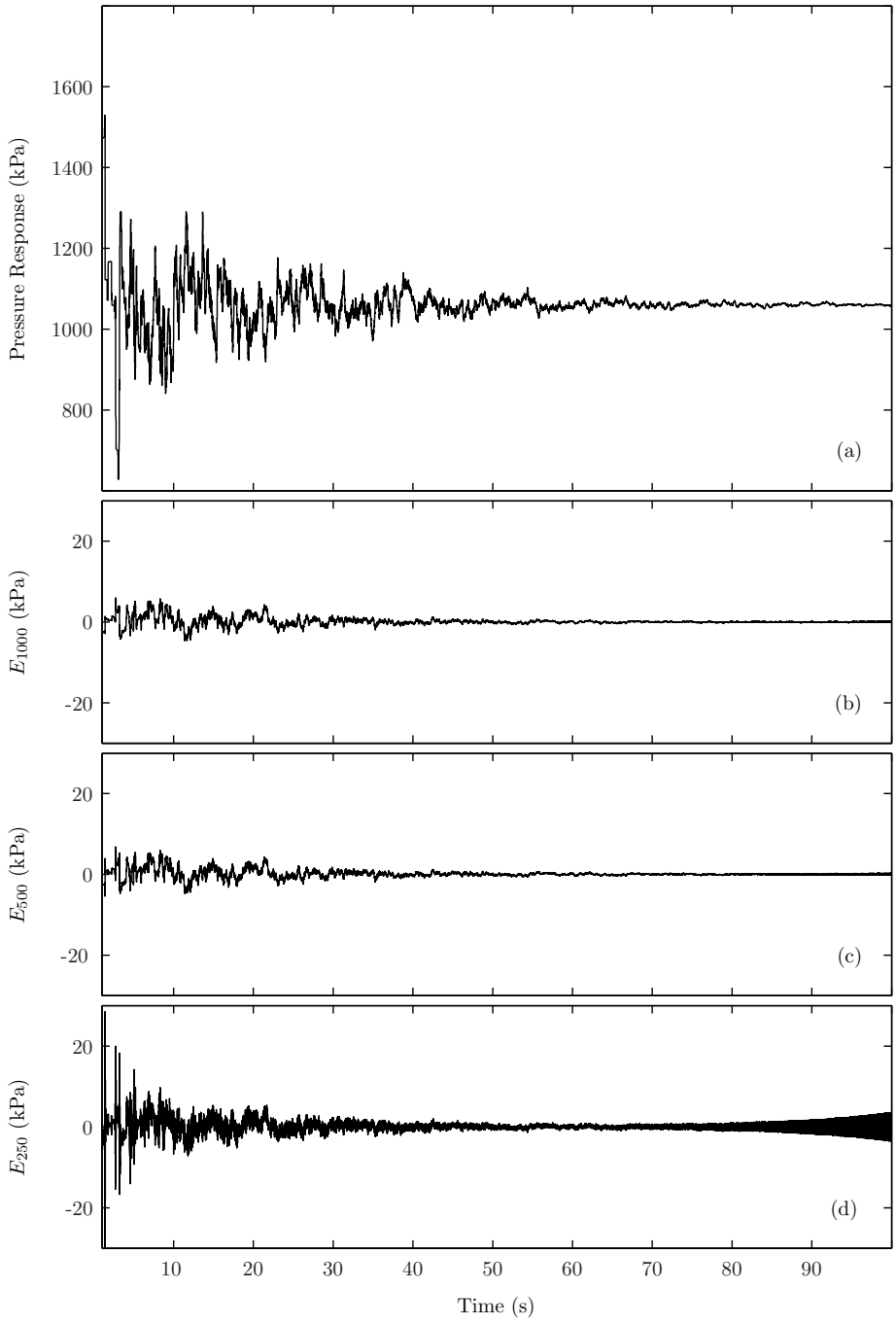

Figure 6: Comparison of MOC and the NILT for the turbulent-steady-friction (TSF) 51-pipe network for the pressure response at node 25, where the subfigures show: (a) the pressure response $f(t)$ computed by the MOC; and (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

the pipe types. Firstly, the lower magnitude in the high frequency components in the error, observed in this example, result from the higher energy dissipation of the TSF pipes. This means that the energy in the higher frequencies did not persist into the longer time-scales, as was observed for the LSF pipes. Secondly, the bias (or trend) observed in the error was due to the fact that the NILT is a linear approximation (in the TSF case but not the LSF case). The nonlinear error associated with the NILT is dependent on the square of the size of the deviation of the flow from the operating point $|q(t) - q_0|^2$. Therefore, when the deviation was large enough, the linear model yielded a biased approximation of the nonlinear model. This phenomena is also observed in the TUF experiments.

The comparative behaviour of the error functions is also different for this example as opposed to the LSF expriments. It is seen that a reduction in the error is achieved by doubling the number of harmonics used in the NILT from 250 to 500. However, the reduction in error for $E_{1000}(t)$ in comparison to $E_{500}(t)$ is small. This observation is consistent with the discussion in the previous paragraph, in that the higher dissipation of this network means that the higher frequency components do not contribute much to the signal reconstruction in the NILT, but that the observed error is mainly associated with the nonlinearities. Finally, the error function $E_{250}(t)$ increases in magnitude for the larger time, resulting from the truncation error associated with $E_{250}(t)$.

As with the LSF experiments, the accuracy that is achievable with the NILT for linear pipe types is demonstrated by the extremely low magnitude of the $E_{1000}(t)$ error function in Figure 7 for the LUF model. As the pipe type for this example is linear, there is no trend in the error functions as with the TSF model. However, in comparison to the LSF model, the errors are generally smaller for the larger time scales. This reduction in the error is attributed to the higher dissipation rate in the LUF pipes as opposed to the LSF pipes, resulting in a lower contribution to the time-domain behaviour from the higher frequencies.

As the TUF pipe type is nonlinear, the behaviour of the error functions in Figure 8 is qualitatively similar to that for the TSF, but with even less high frequency components for the larger time scales resulting from the increased dissipation rate from the unsteady friction contribution. As with this example, the decrease in the error for increasing the number of

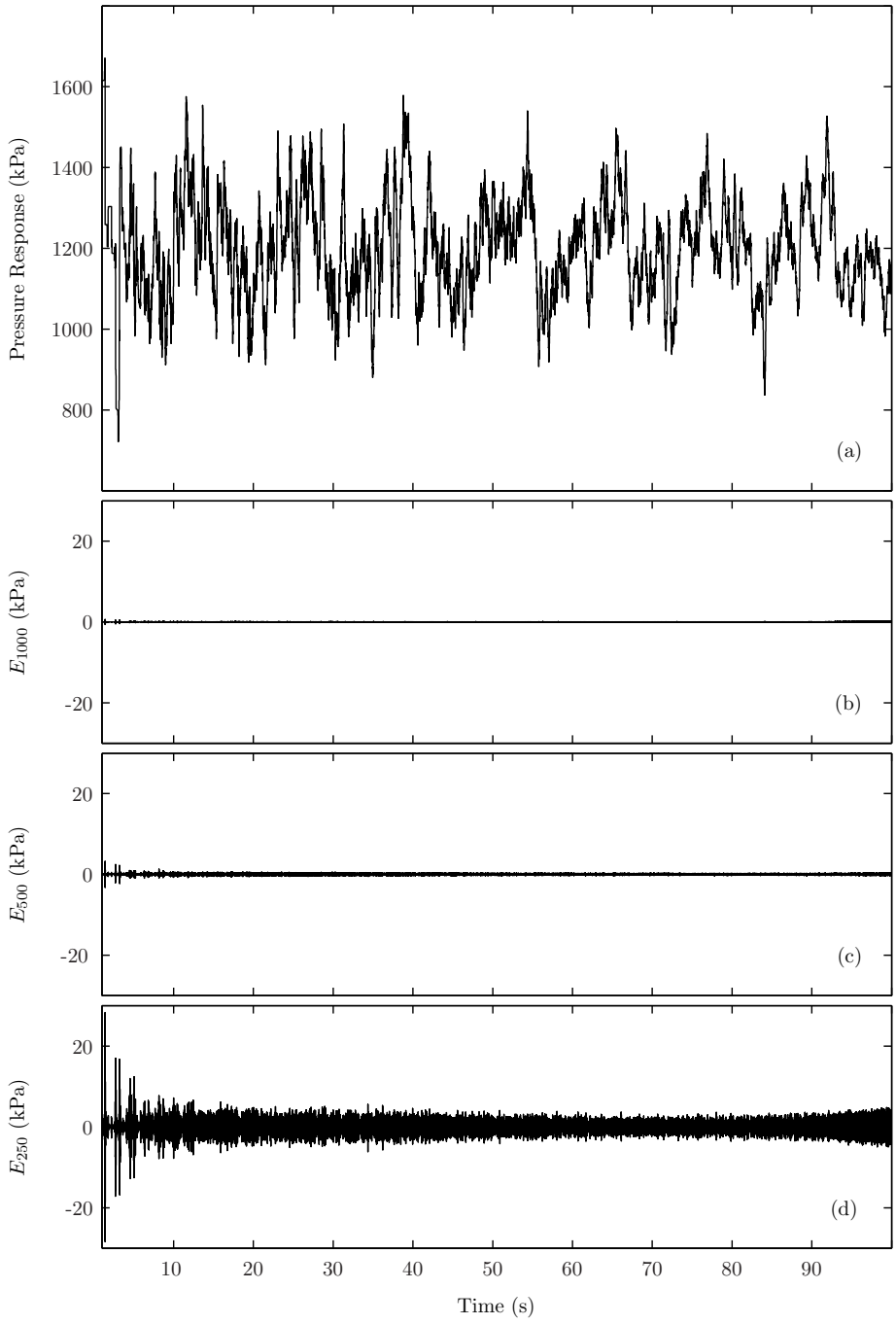

Figure 7: Comparison of MOC and the NILT for the laminar-unsteady-friction (LUF) 51-pipe network for the pressure response at node 25, where the subfigures show: (a) the pressure response $f(t)$ computed by the MOC; and (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

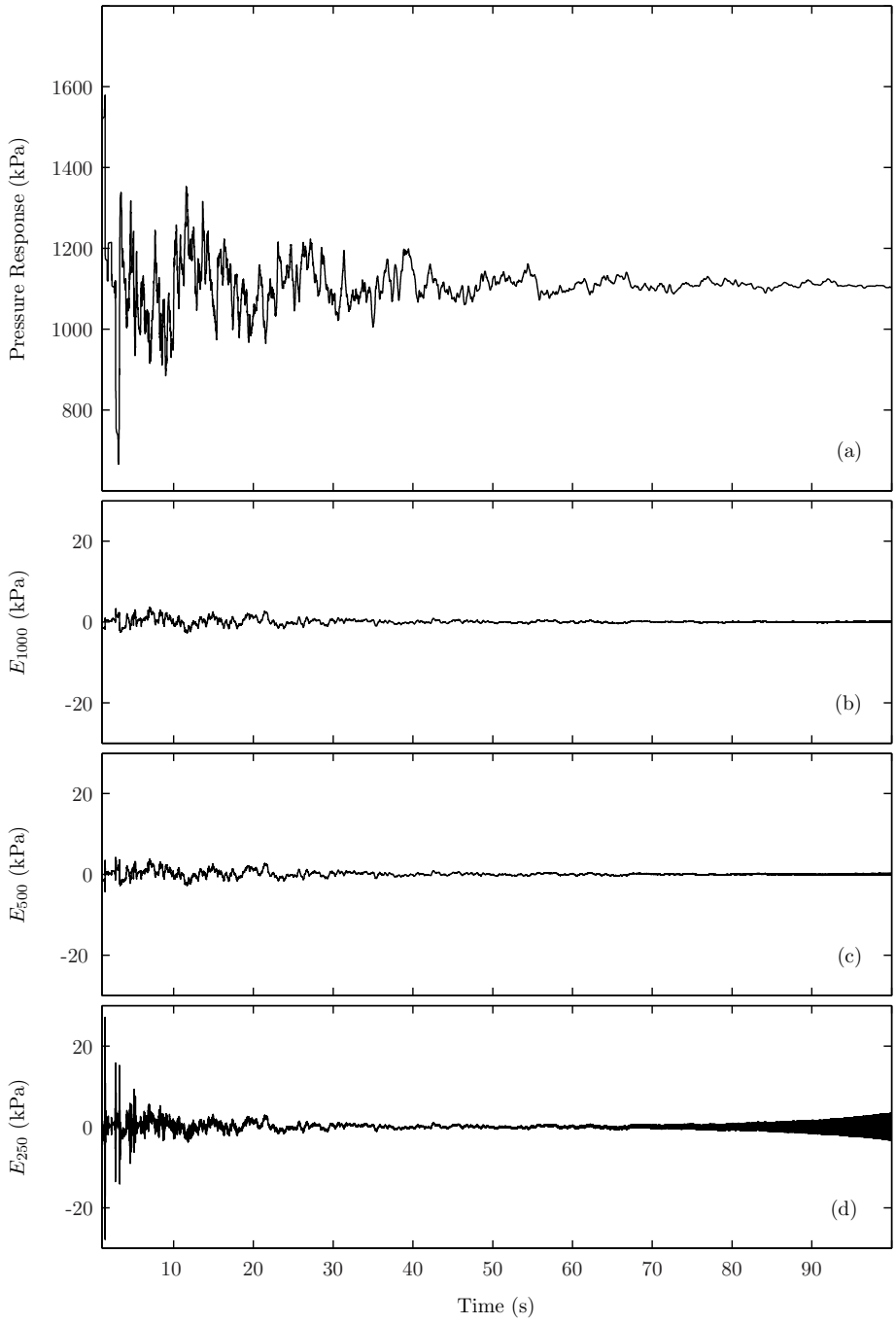

Figure 8: Comparison of MOC and the NILT for the turbulent-unsteady-friction (TUF) 51-pipe network for the pressure response at node 25, where the subfigures show: (a) the pressure response $f(t)$ computed by the MOC; and (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively.

harmonics from 500 to 1000 is small as the error arises from the nonlinear dynamics in the MOC model

As with the LSF and LUF experiments, the ability for the NILT to accurately model VE pipes is observed by the low magnitude of both the E_{1000} and E_{500} error functions in Figure 9. It is observed that the errors for this pipe type are the lowest of all pipe types. This is due to the higher dissipation in the VE model. As observed in Zecchin (2010), a more dissipative model manifests itself as a smoother function in the Laplace-domain that decays faster for larger $|\omega|$. Such functions are more accurately numerically integrated.

5.1.2. Quantitative analysis of different pipeline models

Accuracy comparisons for 65 case studies (three to four different values of N_H for five network types in five different pipe types) are summarised in Tables 2-5. The results are presented in terms of the L_∞ error of the pressure response between the NILT approximation and the MOC (defined as $||E_{N_H}(t)||_\infty$), where the error is taken as the maximum over 5 nodes for the 11-pipe network and 10 nodes for the other networks. To provide a comparative assessment of the magnitude of the errors relative to the excitation of the system, the L_∞ norm of the MOC with respect to the steady-state value[2] is also given. The use of this norm allows for a comparison of the maximum magnitude of the approximations error as a ratio of the maximum magnitude of transient fluctuations about the steady-state point. The errors as a percentage of the norms are given in Tables 2-5 in italics.

From Tables 2-5 it is observed that for the highest number of harmonics N_H in each table, the normalised error was less than 4% for most case studies, with some of the case studies achieving errors of less than 1% for the 11-pipe and 51-pipe networks. This level of accuracy for the nonlinear case studies is greater than expected, particularly given the relatively large transient perturbation from the steady-state operating point, as indicated by the L_∞ norm of the MOC excitation magnitude. The errors for the 35-pipe and 94-pipe networks were consistently greater than the other two networks, and were also more

[2]That is, given the function $f(t)$ as computed by the MOC, the relative L_∞ norm is defined as $||f(t)-f_o||_\infty$ where f_o is the steady state value of $f(t)$.

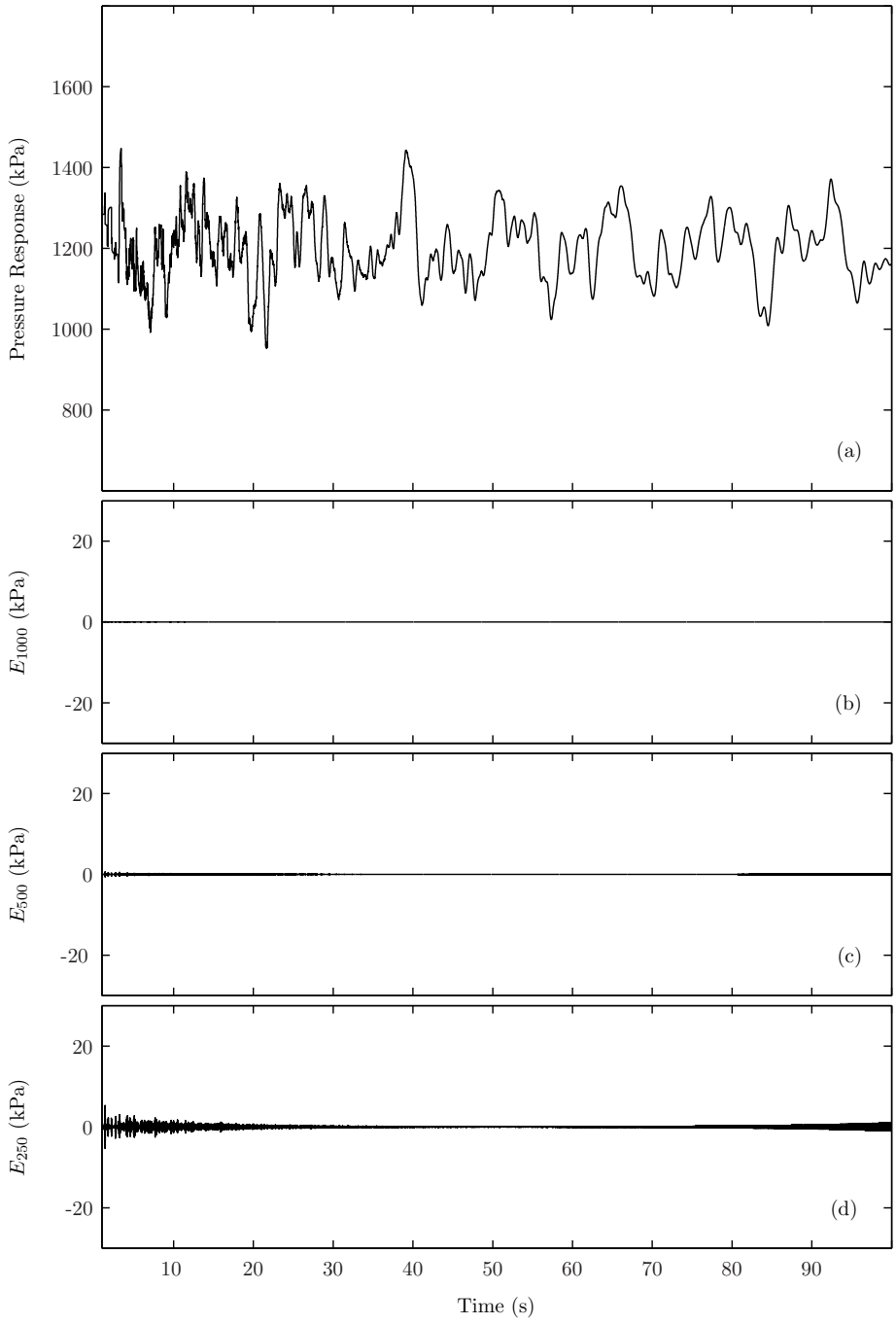

Figure 9: Comparison of MOC and the NILT for the viscoelastic (VE) 51-pipe network for the pressure response at node 25, where the subfigures show: (a) the pressure response $f(t)$ computed by the MOC; and (b)-(d) the errors $E_{N_H}(t) = \tilde{f}(t|a, \Delta\omega, N_H) - f(t)$ for $N_H = 1000$, 500, and 250, respectively..

Table 2: The L_∞ excitation magnitudes for the MOC and the L_∞ errors for the NILT methods, for varying N_H applied to the 11-pipe case studies for the different pipe types. The NILT errors, presented as a percentage of the MOC norms, are given in italics.

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	665.91	395.66	429.47	355.16	379.65
1000	1.07 *(0.2%)*	3.93 *(1%)*	1.07 *(0.3%)*	2.99 *(0.8%)*	1.07 *(0.3%)*
500	6.15 *(0.9%)*	6.93 *(1.8%)*	6.14 *(1.4%)*	6.95 *(2%)*	6.11 *(1.6%)*
250	39.94 *(6%)*	40.47 *(10.2%)*	39.95 *(9.3%)*	40.45 *(11.4%)*	39.95 *(10.5%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the number of harmonics N_H. [b] The norm magnitudes for the MOC are taken relative to the steady state value. [c] The L_∞ errors are based on the maximum of the norms from 5 nodes (*i.e.* $\{2, 3, 4, 5, 6\}$).

Table 3: The L_∞ excitation magnitudes for the MOC and the L_∞ errors for the NILT methods, for varying N_H applied to the 35-pipe case studies for the different pipe types. The NILT errors, presented as a percentage of the MOC norms, are given in italics

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	628.22	508.69	589.28	508.55	507.00
1000	15.06 *(2.4%)*	16.04 *(3.2%)*	14.92 *(2.5%)*	13.98 *(2.7%)*	13.87 *(2.7%)*
500	71.84 *(11.4%)*	71.82 *(14.1%)*	71.84 *(12.2%)*	71.80 *(14.1%)*	71.88 *(14.2%)*
250	134.65 *(21.4%)*	134.81 *(26.5%)*	134.65 *(22.8%)*	134.74 *(26.5%)*	134.00 *(26.4%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the number of harmonics N_H. [b] The norm magnitudes for the MOC are taken relative to the steady state value. [c] The L_∞ errors are based on the maximum of the norms from 10 nodes (*i.e.* $\{3, 5, 7, 8, 10, 11, 12, 13, 15, 19\}$).

sensitive to reductions in N_H, as the errors increased markedly more for the reduction from $N_H = 1000$ to $N_H = 250$. Values of $N_H = 500$ are adequate for high accuracy for the 11-pipe

Table 4: The L_∞ excitation magnitudes for the MOC and the L_∞ errors for the NILT methods, for varying N_H applied to the 51-pipe case studies for the different pipe types. The NILT errors, presented as a percentage of the MOC norms, are given in italics.

Method[a]	L_∞ norms[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	1208.89	948.20	1071.98	986.33	1062.28
1000	0.81 *(0.1%)*	33.60 *(3.5%)*	0.80 *(0.1%)*	16.47 *(1.7%)*	0.77 *(0.1%)*
500	6.81 *(0.6%)*	34.51 *(3.6%)*	6.75 *(0.6%)*	17.36 *(1.8%)*	6.56 *(0.6%)*
250	56.23 *(4.7%)*	60.51 *(6.4%)*	55.71 *(5.2%)*	54.23 *(5.5%)*	54.05 *(5.1%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the number of harmonics N_H. [b] The norm magnitude values for the MOC are taken relative to the steady state value. [c] The L_∞ errors are based on the maximum of the norms from 10 nodes (*i.e.* $\{3, 6, 8, 10, 15, 19, 22, 25, 28, 32\}$).

Table 5: The L_∞ excitation magnitudes for the MOC and the L_∞ errors for the NILT methods, for varying N_H applied to the 94-pipe case studies for the different pipe types. The NILT errors, presented as a percentage of the MOC norms, are given in italics.

Method[a]	L_∞ magnitudes[b] and errors[c] for pipe types (kPa)				
	LSF	TSF	LUF	TUF	VE
MOC	1363.40	937.44	1094.56	1642.57	1072.90
2000	50.32 *(3.7%)*	69.48 *(7.4%)*	44.50 *(4.1%)*	53.72 *(3.3%)*	40.89 *(3.8%)*
1000	154.22 *(11.3%)*	142.48 *(15.2%)*	137.86 *(12.6%)*	71.96 *(4.4%)*	126.70 *(11.8%)*
500	236.14 *(17.3%)*	204.71 *(21.8%)*	206.94 *(18.9%)*	126.22 *(7.7%)*	189.58 *(17.7%)*
250	293.31 *(21.5%)*	245.51 *(26.2%)*	260.39 *(23.8%)*	182.94 *(11.1%)*	240.13 *(22.4%)*

[a] The numbers refer to the NILT simulations where the number specified corresponds to the number of harmonics N_H. [b] The norm magnitude for the MOC are taken relative to the steady state value. [c] The L_∞ errors are based on the maximum of the norms from 10 nodes (*i.e.* $\{9, 16, 36, 39, 48, 56, 57, 62, 69, 71\}$).

and 51-pipe networks, but $N_H = 1000$ was needed for high accuracy in the 35-pipe network, and $N_H = 2000$ for high accuracy in the 94-pipe network.

With respect to the pipe types, as observed in the examples, the greatest errors occurred for the cases of the nonlinear TSF and TUF types. The TUF approximation was generally more accurate than the TSF. This observation is explained by the fact that the unsteady friction operator in the TUF model is in fact linear (Vardy and Brown, 2007), therefore, in comparison to the TSF model, a proportionally greater degree of the dissipation behaviour for the TUF is linear in nature.

As observed in the examples, the LSF generally yielded the highest error for the linear pipe types, in an absolute sense, but in a relative sense the errors as a percentage of the MOC norm were similar to those for the LUF and the VE, as the MOC norm for the LSF cases was typically higher than these other pipe types. The VE cases typically yielded the lowest error, once again being attributed to the higher energy dissipation rate of these pipe types (as is observed by the relatively smoother pressure response in Figure 9).

5.2. Computational efficiency studies

One of the advantages of the NILT as an efficient hydraulic simulator is that it does not require the computation of the complete network state as it deals only with the composition of transfer functions from the input boundary conditions to the output nodal response variables. This is in contrast to all discrete methods, such as the MOC, that require a discretisation of the state and the computation of all lumped state variables at each time step. Despite its efficiency, an implication of these different approaches, however, is that the computational time of the NILT is dependent on the number of measurement points[3] N_m within the network, whereas the computational time of discrete methods, like the MOC, is not.

To explain this further, with respect to (18), the computation of $\psi_i(t)$ ($N_m = 1$), the

[3]That is, spatial points at which the transient response is to be computed.

pressure response at node $i \in \mathcal{N}_d$, is efficiently calculated by

$$\psi_i(t) = \mathcal{L}^{-1}\left\{\Psi_i(s)\right\}(t) = \mathcal{L}^{-1}\left\{\boldsymbol{H}_i(s)\begin{pmatrix}\boldsymbol{\Theta}_d(s)\\ \boldsymbol{\Psi}_r(s)\end{pmatrix}\right\}(t), \tag{23}$$

where $\boldsymbol{H}_i$ is the i-th row of the system matrix $\boldsymbol{H}$. That is, (23) involves only a vector multiplication (at each s) and the ILT of only a single Laplace-domain function $\Psi_i(s)$. This can be performed without evaluating any of the other response nodal states in $\boldsymbol{\psi}_d(t)$ or $\boldsymbol{\theta}_r(t)$. Similarly, to calculate the pressure response at any two nodes $\psi_i(t)$ and $\psi_j(t)$ ($N_m = 2$), the most efficient approach is

$$\begin{bmatrix}\psi_i(t)\\ \psi_j(t)\end{bmatrix} = \mathcal{L}^{-1}\left\{\begin{bmatrix}\Psi_i(s)\\ \Psi_j(s)\end{bmatrix}\right\}(t) = \mathcal{L}^{-1}\left\{\begin{pmatrix}\boldsymbol{H}_i(s)\\ \boldsymbol{H}_j(s)\end{pmatrix}\begin{pmatrix}\boldsymbol{\Theta}_d(s)\\ \boldsymbol{\Psi}_r(s)\end{pmatrix}\right\}(t), \tag{24}$$

which requires approximately twice[4] the number of operations of (23) (two vector multiplications, and the ILT of two functions). Given this dependency on the number of state variable points of interest, it is important to include this as a parameter in the numerical computational studies.

As with Section 5.1, firstly a couple of specific examples are given, followed by a general analysis of the computational timings for all case studies from Section 5.1. Simulations were performed on a 2.13 GHz Linux machine and the CPU timings were evaluated by the `procstat` routine to ensure that the exact processing time of the simulation in the CPU was recorded.

5.2.1. Qualitative analysis for different pipeline models

For the qualitative analysis, results from experiments on the 51-pipe network with both the TSF and TUF pipeline models is given. The experiments consisted of 10^5 time point computations (*i.e.* 100 s simulation time at a temporal discretisation of $\Delta t = 0.001$ s), where the computational times for these experiments for the MOC and the NILT are given

[4]The computational requirements are only approximately double, as there is some computational saving in computing the NILT for two functions at a time (as opposed to two functions independently).

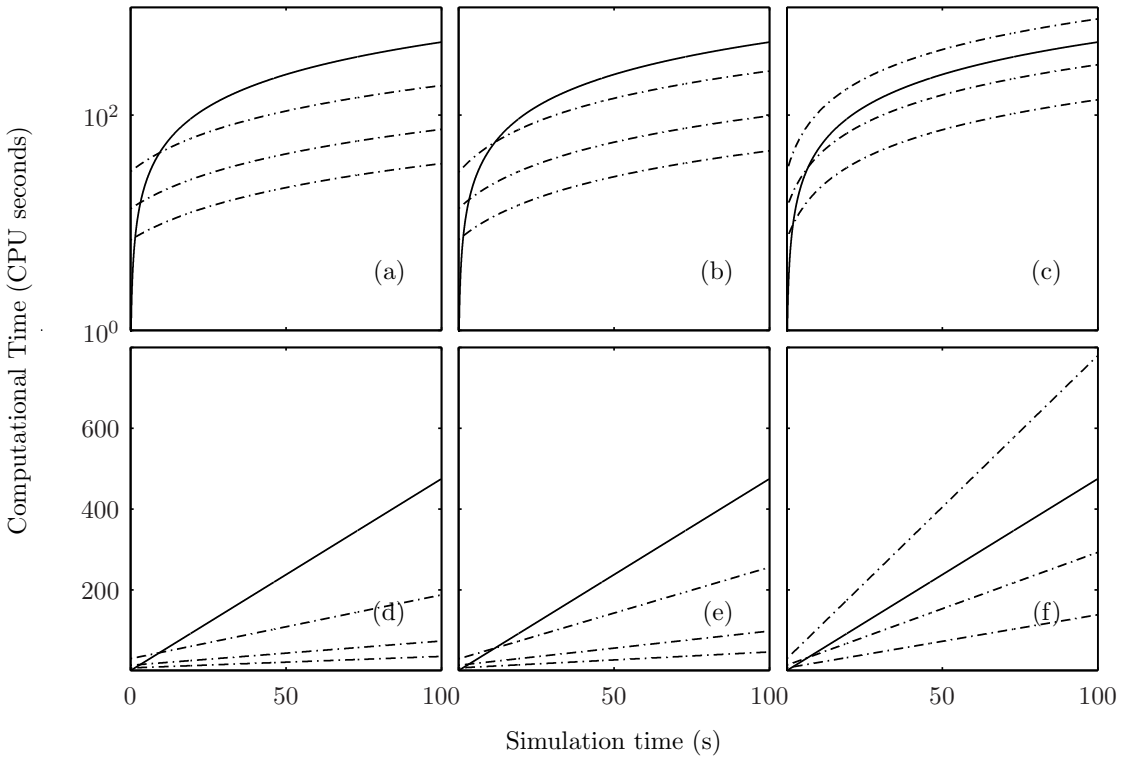

Figure 10: Computational times versus simulation time for the TSF 51-pipe network for the MOC ($-$) and the NILT ($- \cdot -$). The three lines for the NILT correspond to N_H=250 (lowest), N_H=500 (middle), and N_H=1000 (highest). Figures (a)-(c) show the computational time on log scale for the case of the pressure response being computed at 1, 2, and 10 nodes, respectively. Figures (d)-(f) repeat these plots on a linear scale. Computational times are in CPU seconds. Simulations were performed on a 2.13 GHz Linux machine.

in Figures 10 and 11 for $N_m = 1$, 2, and 10.

Considering the computational requirements for a given number of nodal calculations in Figure 10, qualitatively, the computational time of the MOC and the NILT demonstrate an interesting behaviour. The MOC required minimal startup time, where the initialisation procedures simply involve setting the state variables to the initial steady-state values[5]. The increase in computational time for the MOC is linear with the simulation time t. The NILT however has a more computationally expensive overhead in the initialisation procedures, as observed more clearly in the Figure 10(a)-(c) log scale plots. This is attributed to the fact that before any time-points can be computed, the complex coefficients $F(a + \mathrm{i}n\Delta\omega), n =$

[5]The computation of the initial steady-state hydraulic solver is not included in these comparative timing studies as it is was the same for both the MOC and the NILT.

$0, 1, \ldots, N_\Delta \cdot N_H$ must be determined by (18), which for this example involved the inversion of a complex 32×32 matrix. Therefore, as observed in Figure 10, the initial computational overhead is greater as N_H is increased. Similarly to the MOC, the computational time for the NILT increased linearly with t. As is clear on the Figure 10(d)-(f) linear plots, the startup time for the NILT is only a small portion of the overall computational time.

Doubling the number of measurement nodes from 1 to 2 [Figures 10(a) and 10(d) compared to Figures 10(b) and 10(e)] appears to have small impact on the increase in computational time (particularly for the smaller N_H). For these cases, the NILT is more computationally efficient for larger t. However, as the number of measurement nodes is increased to 10, the computational requirements of the NILT for all N_H are dramatically increased, such that, as seen in 10(e), the MOC is more efficient than the NILT with $N_H = 1000$ for all t.

The behaviour of the computational time as a function of the simulation time t for the TUF from Figure 11 is qualitatively similar to that for the TSF in Figure 10, with the significant quantitative difference being the computational times for the MOC. As a time-domain operator, the TUF in the MOC involves the evaluation of a convolution to model the unsteady component to the fluid shear stresses at every spatial point. Under the efficient Vardy and Brown (2007) algorithm, this convolution is transformed into a single step difference equation in a finite number of states. Therefore, in comparison to the TSF, the TUF involves the storage of these additional states and the calculation of the difference equation at each time point for each spatial point. The computational impact of this is observed to be a near quadrupling of the computational cost of the TUF in comparison with the TSF.

In contrast, the Laplace-domain representation of the TUF does not require any additional states, but the Vardy and Brown (2007) algorithm serves to introduce a rational function[6] into the resistance transfer function. Therefore, in comparison with the TSF, the computational difference in the NILT method occurs only in the initialisation time, that is, only when the $\boldsymbol{H}(s)$ matrix in (18) is computed. From Figure 11, it is clear that this cost is

[6]The introduced rational function is of the order of the number of introduced states.

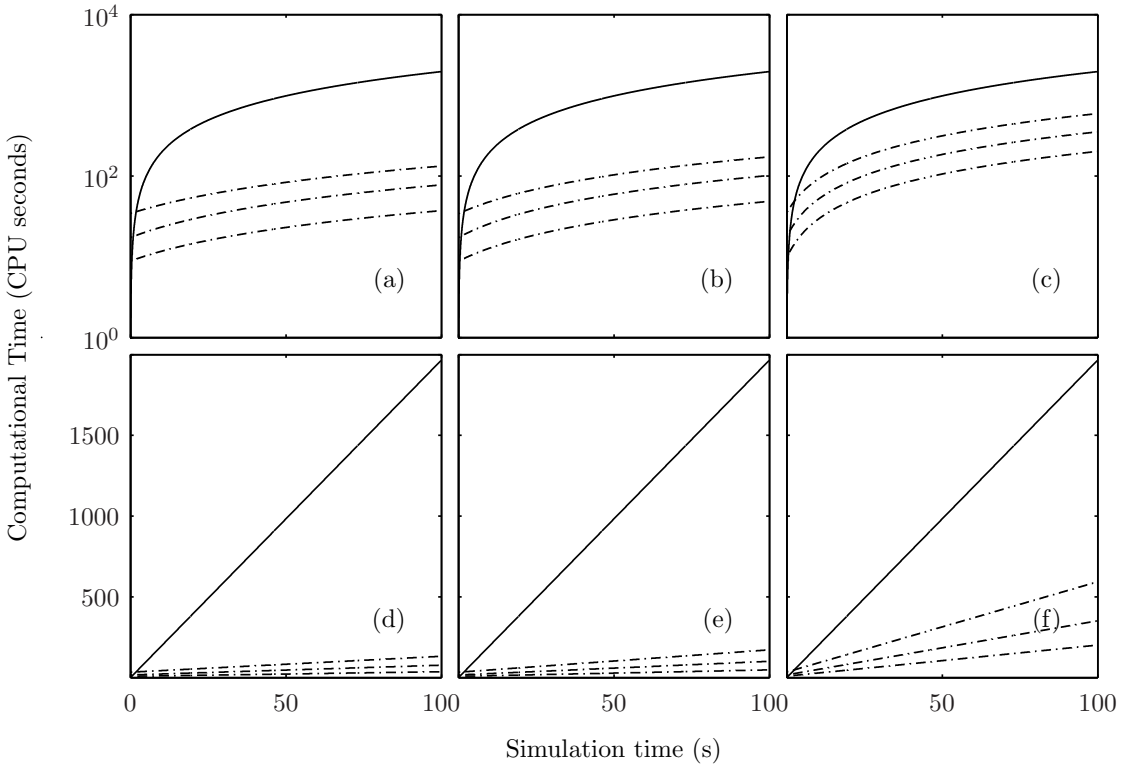

Figure 11: Computational times versus simulation time for the 51-pipe network with TUF pipes for the MOC($-$) and the NILT ($-\cdot-$). The three lines for the NILT correspond to N_H=250 (lowest), 500 (middle), and 1000 (highest). Figures (a)-(c) show the computational time on log scale for the case of the pressure response being computed at 1, 2, and 10 nodes, respectively. Figures (d)-(f) repeat these plots on a linear scale. Computational times are in CPU seconds. Simulations were performed on a 2.13 GHz Linux machine.

small. Consequently, once the complex coefficients $F(a + \mathrm{i}n\Delta\omega), n = 0, 1, \ldots, N_\Delta \cdot N_H$ have been computed, the computational cost of the NILT at each time point for the TUF is the same as for the TSF. This results in the NILT being more computationally efficient than the MOC for all cases as depicted in Figure 11.

5.2.2. Quantitative analysis for different pipeline models

To generalise the study, numerical timing experiments were performed on the four different networks from Section 5.1 in the five different pipe types, creating a total of 20 different network types. For the NILT, the timing experiments were performed for the number of harmonics $N_H \in \{250, 500, 1000\}$ for the 11-pipe, 35-pipe and 51-pipe network and $N_H \in \{250, 500, 1000, 2000\}$ for the 94-pipe network, with the number of computational

Table 6: Computational times[a] of the NILT relative to the MOC for the 11-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_{ms}	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
1000	5	2.770[b]	2.793	0.938	0.700	1.354
	2	1.514	1.535	0.513	0.381	0.746
	1	1.105	1.094	0.370	0.278	0.529
500	5	1.398	1.392	0.465	0.350	0.664
	2	0.767	0.751	0.256	0.190	0.368
	1	0.549	0.549	0.183	0.140	0.269
250	5	0.690	0.700	0.232	0.175	0.338
	2	0.385	0.378	0.129	0.096	0.187
	1	0.279	0.272	0.092	0.069	0.134
(MOC		*74.1*	*75*	*223.5*	*297.4*	*154.9)*

[a]Simulations were performed on a 2.13 GHz Linux machine were the CPU timings were evaluated by the procstat routine. [b] This means that the NILT took 2.770 times the computational time for the MOC, which in this case is $2.77 \times 74.1 = 205.3$ CPU seconds.

measurement points $N_m \in \{1, 2, 5\}$ for the 11-pipe network and $N_m \in \{1, 2, 5, 10\}$ for the other three networks. Consequently, the study comprised 20 MOC simulations, and 245 NILT simulations, each for 10^5 time points.

Tables 6-9 summarise the numerical experiments for the computational timing studies. Presented in italics are the computational times (CPU seconds) of the MOC for the different network types and, for convenience, the computational times of the NILT are presented as a ratio with the corresponding MOC time (*i.e.* relative computational times greater than 1 indicate that the MOC was more efficient than the NILT for the particular case).

At a first observation, Tables 6-9 show the expected result that the computational time of the NILT are approximately proportional to the number of harmonics N_H involved in the inversion process (*i.e.* a doubling of N_H is matched by a doubling of the computational time). In comparison, the computational cost of the NILT is not linear with the number of measurement nodes N_m, but the incorporation of each additional node for $N_m > 1$ costs

Table 7: Computational times[a] of the NILT relative to the MOC for the 35-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_{ms}	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
1000	10	0.775	1.132	0.347	0.206	0.415
	5	0.539	0.587	0.122	0.144	0.242
	2	0.262	0.310	0.068	0.076	0.134
	1	0.192	0.209	0.074	0.049	0.099
500	10	0.522	0.404	0.116	0.123	0.286
	5	0.337	0.230	0.073	0.065	0.161
	2	0.181	0.127	0.041	0.027	0.083
	1	0.129	0.092	0.030	0.020	0.070
250	10	0.141	0.141	0.045	0.032	0.073
	5	0.155	0.081	0.026	0.019	0.042
	2	0.077	0.045	0.014	0.010	0.024
	1	0.066	0.033	0.011	0.008	0.018
(MOC		*649.7*	*655.4*	*2106.7*	*2916.8*	*1280.7)*

[a]Simulations were performed on a 2.13 GHz Linux machine were the CPU timings were evaluated by the `procstat` routine.

approximately an additional 1/3 of the computational time required for the first node (*i.e.* at each time point, there are operations that need to be performed only once for all nodes).

As is clear in Tables 7-9, the NILT compares more favorably with the MOC for the larger networks, with relative computational times reaching as low as 0.008 (*i.e.* two orders of magnitude less time then the MOC). This is attributed to the fact that the MOC has an increasing computational expense for larger networks. In contrast, the only computational overhead associated with large networks for the NILT is in the initialisation phase, which, from the qualitative analysis, was observed to only contribute minimally to the overall computational time. The MOC was only faster than the NILT for the cases of the numerically simple LSF and TSF, and this was only observed for the longer NILT simulations (*i.e.* higher N_H and N_m).

With regard to the more numerically involved LUF, TUF and VE pipe types, the NILT

Table 8: Computational times[a] of the NILT relative to the MOC for the 51-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_{ms}	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
1000	10	1.614	1.430	0.545	0.419	0.890
	5	0.753	0.613	0.260	0.238	0.524
	2	0.428	0.354	0.160	0.103	0.263
	1	0.327	0.334	0.100	0.068	0.177
500	10	0.638	0.562	0.203	0.203	0.313
	5	0.374	0.329	0.120	0.121	0.186
	2	0.216	0.189	0.069	0.059	0.110
	1	0.162	0.141	0.053	0.040	0.109
250	10	0.296	0.260	0.138	0.071	0.140
	5	0.174	0.152	0.073	0.060	0.083
	2	0.101	0.088	0.033	0.032	0.049
	1	0.076	0.066	0.025	0.025	0.038
(MOC		*456.7*	*523.2*	*1458.5*	*1911.9*	*874.7)*

[a]Simulations were performed on a 2.13 GHz Linux machine were the CPU timings were evaluated by the `procstat` routine.

was unconditionally more efficient (except for the 11-pipe $\{N_H, N_{ms}\} = \{1000, 5\}$ case). The more expensive MOC times for these cases are clearly attributed to the increased number of numerical operations involved in evaluating the convolutions for the unsteady friction and viscoelastic operators. Using the N_H values from Tables 2-5, a NILT simulation with a small error[7] for 5 nodes was observed to be on average 50%, 17%, 14% and 40% of the MOC time for the 11-pipe, 35-pipe, 51-pipe and 94-pipe networks, respectively. As the LUF and the TUF pipe types were more numerically involved than the VE pipe type for the MOC, the computational saving of the NILT is greater for these cases.

[7]Here *small* error is taken to mean: less that 1% for the 11-pipe network (*i.e.* the case $N_H = 500$); less that 3% for the 35-pipe network (*i.e.* the case $N_H = 1000$); less that 2% for the 51-pipe network (*i.e.* the case $N_H = 500$); and less than 4.1% for the 94-pipe network (*i.e.* the case $N_H = 2000$)

Table 9: Computational times[a] of the NILT relative to the MOC for the 94-pipe case study. The computational times for the MOC simulations (in CPU seconds) are given in italics. All times are based on the computation of 10^5 simulation time points.

N_H	N_{ms}	Relative computational times for pipe type				
		LSF	TSF	LUF	TUF	VE
2000	10	1.826	2.047	0.527	0.389	0.828
	5	1.201	1.266	0.338	0.256	0.596
	2	0.739	0.803	0.236	0.169	0.429
	1	0.630	0.695	0.193	0.141	0.291
1000	10	0.938	1.059	0.369	0.271	0.521
	5	0.560	0.601	0.212	0.165	0.357
	2	0.363	0.422	0.144	0.106	0.237
	1	0.301	0.337	0.120	0.089	0.198
500	10	0.422	0.479	0.158	0.133	0.244
	5	0.304	0.299	0.101	0.074	0.178
	2	0.192	0.199	0.070	0.050	0.113
	1	0.129	0.164	0.058	0.042	0.092
250	10	0.203	0.216	0.088	0.066	0.139
	5	0.131	0.143	0.053	0.040	0.085
	2	0.085	0.093	0.035	0.026	0.056
	1	0.070	0.078	0.029	0.021	0.045
(MOC		*3235.0*	*2865.4*	*8313.0*	*11335.0*	*5385.1)*

[a]Simulations were performed on a 2.13 GHz Linux machine were the CPU timings were evaluated by the `procstat` routine.

6. Conclusions

The focus of this paper has been on the use of the linear Laplace-domain network model from Zecchin et al. (2009a) as an alternative time-domain hydraulic simulator by way of the numerical inverse Laplace transform (NILT). The use of the inverse Laplace transform (ILT) in the development of time-domain models from their Laplace transforms (LTs) has been extensive, however little attention has been given to full network models, with the exception being the impulse response method (IPREM) (Suo and Wylie, 1989). The approach presented here is entirely novel in that it couples the Laplace-domain input/output model from

(Zecchin et al., 2009a) in a computationally efficient way with the Fourier series expansion NILT from Abate and Whitt (1995). The parameters of the NILT have been studied in detail in (Zecchin, 2010). These heuristics were successfully used in the application of the NILT to 20 different case studies in this paper (four different networks in five different pipe types). The focus of the studies were on the accuracy and computational efficiency of the proposed NILT.

For the cases considered, the NILT was found to provide accurate approximations for all case studies, even networks with nonlinear pipe types. The accuracy was observed to be greater for the more highly dissipative networks. For large networks, NILT was found to be computationally efficient compared with the method of characteristics (MOC). This relative efficiency was observed to be especially true for the case studies with more complex pipe types involving convolution operations, as these operations exert little additional computational time on the NILT. In addition to the computational efficiency, the NILT possesses the desirable property that it correctly captures wave propagation delays without the need for fine computational grids. This property arises from the fact that the NILT does not involve the discretisation of the network state, and it is able to compute the network state at any time point without computing the state at the preceding time points. As such, the NILT represents a worthy alternative approach for modelling networks involving pipes with greatly varying wavespeeds.

7. Acknowledgements

This research has been financially supported by the Australian Research Council

References

Abate, J., Whitt, W., 1992. The Fourier-series method for inverting transforms of probability distributions. Queueing Systems Theory and Applications 10 (1-2), 5–88.

Abate, J., Whitt, W., 1995. Numerical inversion of Laplace transforms of probability distributions. ORSA Journal on Computing 7 (1), 36–43.

Chaudhry, M. H., 1987. Applied Hydraulic Transients, 2nd Edition. Van Nostrand Reinhold Co., New York, USA.

Crump, K. S., 1976. Numerical inversion of Laplace transforms using a Fourier-series approximation. J. ACM 23, 89–96.

Datta, R. S. N., Sridharan, K., 1994. Parameter estimation in water-distribution systems by least squares. Journal of Water Resources Planning and Management, ASCE 120 (4), 405–422.

Desoer, C. A., Vidyasagar, M., 1975. Feedback Systems: Input-Output Properties. Academic Press, New York.

Diestel, R., 2000. Graph Theory, electronic edition Edition. Springer-Verlag, New York, USA.

Franklin, G. F., Powell, J. D., Emami-Naeini, A., 2001. Feedback control of dynamic systems, 4th Edition. Prentice Hall PTR, Upper Saddle River, N. J. London.

Goodson, R. E., Leonard, R. G., 1972. A survey of modeling techniques for fluid line transient. Journal of Basic Engineering, ASME 94, 474–482.

Kojima, E., Shinada, M., Yu, J., 2002. Development of accurate and practical simulation technique based on the modal approximations for fluid transients in compound fluid-line systems (1st report: Establishment of fundamental calculation algorithm and basic considerations for verification of its avaliability). International Journal of Fluid Power 3 (2), 5–15.

Margolis, D. L., Yang, W. C., 1985. Bond graph models for fluid networks using modal approximation. Journal of Dynamic Systems Measurement and Control-Transactions of the ASME 107 (3), 169–175.

Pudar, R. S., Liggett, J. A., 1992. Leaks in pipe networks. Journal of Hydraulic Engineering, ASCE 118 (7), 1031–1046.

Rieutord, E., Blanchard, A., 1979. Pulsating viscoelastic pipe flow - water-hammer. Journal of Hydraulic Research, IAHR 17 (3), 217–229, in French.

Stephens, M. L., 2008. Transient response analysis for fault detection and pipeline wall condition assessment in field water transmission and distribution pipelines and networks. PhD, The University of Adelaide.

Suo, L., Wylie, E. B., 1989. Impulse response method for frequency-dependent pipeline transients. Journal of Fluids Engineering, ASME 111 (December), 478–483.

Triverio, P., Grivet-Talocia, S., Nakhla, M. S., Canavero, F. G., Achar, R., 2007. Stability, causality, and passivity in electrical interconnect models. (IEEE) Transactions on Advanced Packaging 30 (4), 795–808.

Vardy, A. E., Brown, J. M. B., 2003. Transient turbulent friction in smooth pipe flows. Journal of Sound and Vibration 259 (5, January), 1011–1036.

Vardy, A. E., Brown, J. M. B., 2004. Transient turbulent friction in fully-rough pipe flows. Journal of Sound and Vibration 270, 233–257.

Vardy, A. E., Brown, J. M. B., 2007. Approximation of turbulent wall shear stresses in highly transient pipe flows. Journal of Hydraulic Engineering-ASCE 133 (11), 1219–1228.

Vítkovský, J. P., 2001. Inverse analysis and modelling of unsteady pipe flow: Theory, applications and

experimental verification. PhD thesis, Adelaide University.

Vítkovský, J. P., Stephens, M. L., Bergant, A., Lambert, M. F., Simpson, A. R., 2004. Efficient and accurate calculation of zielke and vardy-brown unsteady friction in pipe transients. In: 9th International Conference on Pressure Surges. Vol. 2. BHR Group Ltd, Chester, UK, pp. 405–419.

Wylie, E. B., Streeter, V. L., 1993. Fluid Transients in Systems. Prentice-Hall Inc., Englewood Cliffs, New Jersey, USA.

Yang, W. C., Tobler, W. E., 1991. Dissipative modal approximation of fluid transmission-lines using linear friction model. Journal of Dynamic Systems Measurement and Control-Transactions of the ASME 113 (1), 152–162.

Zecchin, A. C., 2010. Laplace-domain analysis of fluid line networks with applications to time-domain simulation and system parameter identification. Doctor of Philosophy Dissertation, The University of Adelaide.

Zecchin, A. C., Simpson, A. R., Lambert, M. F., White, L. B., Vitkovsky, J. P., 2009a. Frequency-domain modeling of transients in pipe networks with compound nodes using a Laplace-domain admittance matrix. Journal of Hydraulic Engineering, ASCE Accepted.

Zecchin, A. C., Simpson, A. R., Lambert, M. F., White, L. B., Vitkovsky, J. P., 2009b. Transient modeling of arbitrary pipe networks by a Laplace-domain admittance matrix. Journal of Engineering Mechanics, ASCE 135 (6), 538–547.

Zielke, W., 1968. Frequency-dependent friction in transient pipe flow. Journal of Basic Engineering, ASME 90 (1), 109–115.

8. Acronyms

I/O input/output

ILT inverse Laplace transform

IPREM impulse response method

LSF laminar-steady-friction

LT Laplace transform

LUF laminar-unsteady-friction

MOC method of characteristics

NILT numerical inverse Laplace transform

PDE partial differential equation

TSF turbulent-steady-friction

TUF turbulent-unsteady-friction

VE viscoelastic