CHALMERS | GÖTEBORGS UNIVERSITET

PREPRINT

Dynamic Regularization of Input Estimation Problems by Explicit Block Inversion

T PATRIK NORDBERG
IVAR GUSTAFSSON

Department of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
GÖTEBORG UNIVERSITY
Göteborg Sweden 2004

Preprint 2004:36

Dynamic Regularization of Input Estimation Problems by Explicit Block Inversion

T Patrik Nordberg
Ivar Gustafsson

CHALMERS | GÖTEBORGS UNIVERSITET

Mathematics
Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
Göteborg, June 2004

NO 2004:36
ISSN 0347-2809

Matematiska Vetenskaper
Göteborg 2004

Dynamic Regularization of Input Estimation Problems by Explicit Block Inversion

T Patrik Nordberg⋆ and Ivar Gustafsson⋆⋆

⋆ *Volvo Car Corporation and Department of Applied Mechanics,*
Chalmers University of Technology, SE-412 96 Göteborg, Sweden.
patrik.nordberg@me.chalmers.se
⋆⋆ *Numerical Analysis Group,*
Department of Mathematics Chalmers University of Technology and
Göteborg University.
ivar@math.chalmers.se

ABSTRACT

Input estimation problems in structural dynamics often lead to solving damped least squares problems. Explicit block inversion algorithms are developed to invert the associated upper triangular block Toeplitz matrix. Moreover, the inversion algorithms are used to introduce a new type of regularization technique. For optimal regularization parameters, it is demonstrated by a numerical example that the solution using the proposed regularization technique has an upper bound equal to zeroth-order Tikhonov regularization.

1 INTRODUCTION

Accurate knowledge of the operational loading conditions acting on a physical structure through its designated life is a crucial component in the design of all mechanical systems. In some cases, it is impossible to measure these forces directly. The structure may prohibit the use of a force transducer without it being destroyed or the transducer itself may alter the system properties. In these situations; when the force/input locations are inaccessible for direct measurements, indirect methods need to be utilized. Input estimation problems are difficult since frequently only a fraction of the potentially available response data is used to estimate all inputs. Moreover, measurement noise may render the identified inputs useless if it is not treated correctly.

Input estimation procedures dealing with transient input histories applied with known spatial distribution have received increasing attention, see [1], [2], [3], [4], [5], [6], [7] and [8], to mention just a few. Different methods are usually developed to accommodate a specific application.

Generally, input estimation problems are ill-posed. Treatment of these problems requires additional information about the solution sought. Regularization methods are commonly used to restrict the computed solution to be smooth and thereby consistent with complementary physical observations, see e.g. [9], [10], [11] and [12].

This paper focusses on the upper block triangular Toeplitz matrix, which is readily derived from the governing equations of motion. For a survey on techniques for solving systems with Toeplitz matrices see [13] and the references therein, cf also [14]. Here, the resulting regularized deconvolution problem is solved by explicit block inversion, which gives valuable insight to the exact sequence of operations performed in the inversion. The formulation is used to introduce a new type of regularization procedure. An extensive tutorial survey of numerical algorithms associated with these discrete regularized deconvolution problems is given in [15].

The paper is organized as follows. In Section 2, the input estimation problem is defined and the dynamic programming algorithm is outlined. A modified version of this algorithm is used in Section 3 to derive general formulations of the explicit inverse of the Toeplitz matrix. The advocated formulation with Markov parameters is used to introduce a new type of regularization procedure in Section 4. A numerical example is given to illustrate certain properties of the regualization procedure in Section 5. Finally, conclusions are drawn in Section 6.

2 PROBLEM DEFINITION

2.1 Theoretical preliminaries

Consider the continuous-time governing equations of motion for a causal, linear and time-invariant mechanical system with n_{dof} degrees of freedom

$$\boldsymbol{M}\ddot{\boldsymbol{q}}(t) + \boldsymbol{V}\dot{\boldsymbol{q}}(t) + \boldsymbol{K}\boldsymbol{q}(t) = \boldsymbol{f}(t) \tag{1}$$

where $\ddot{\boldsymbol{q}}(t)$, $\dot{\boldsymbol{q}}(t)$ and $\boldsymbol{q}(t)$ are vectors of accelerations, velocities and displacements, respectively. $\boldsymbol{M}$ denotes the symmetric and positive definite mass matrix; $\boldsymbol{V}$ and $\boldsymbol{K}$ denote positive semidefinite damping and stiffness matrices, respectively; $\boldsymbol{f}(t)$ is the excitation vector. Introducing the *state vector* $\boldsymbol{x}(t) = [\boldsymbol{q}(t)^{\mathrm{T}}\ \dot{\boldsymbol{q}}(t)^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{n_{\mathrm{s}}}$ where $n_{\mathrm{s}} = 2n_{\mathrm{dof}}$ and the dummy equation $\dot{\boldsymbol{q}}(t) = \dot{\boldsymbol{q}}(t)$, Eq. (1) can be rewritten in first order state-space form according to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_{\mathrm{c}}\boldsymbol{x}(t) + \boldsymbol{B}_{\mathrm{c}}\boldsymbol{u}(t) \stackrel{\mathrm{def}}{=} \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{V} \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{P}_{\mathrm{u}} \end{bmatrix} \boldsymbol{u}(t) \tag{2}$$

Here, $\boldsymbol{P}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{dof}} \times n_{\mathrm{i}}}$ is of full column rank and relates the excitation vector to n_{i} independent inputs $\boldsymbol{u}(t) \in \mathbb{R}^{n_{\mathrm{i}}}$ as

$$\boldsymbol{f}(t) = \boldsymbol{P}_{\mathrm{u}}\boldsymbol{u}(t) \tag{3}$$

In this paper, measured quantities $\hat{\boldsymbol{y}}(t) \in \mathbb{R}^{n_{\mathrm{o}}}$ are assumed to be accelerations which are directly related to $\dot{\boldsymbol{x}}(t)$. Thus, the output $\boldsymbol{y}(t) \in \mathbb{R}^{n_{\mathrm{o}}}$ relation becomes

$$\boldsymbol{y}(t) = \boldsymbol{C}_{\mathrm{a}}\dot{\boldsymbol{x}}(t) = \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}_{\mathrm{c}}\boldsymbol{x}(t) + \boldsymbol{C}_{\mathrm{a}}\boldsymbol{B}_{\mathrm{c}}\boldsymbol{u}(t) \tag{4}$$

Time discretization of the continuous-time state-space matrices $\boldsymbol{A}_{\mathrm{c}} \to \boldsymbol{A}$, $\boldsymbol{B}_{\mathrm{c}} \to \boldsymbol{B}$, and the input yields the discrete-time counterpart of Eq. (2) and (4) as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{5a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k \tag{5b}$$

with $\boldsymbol{C} \stackrel{\mathrm{def}}{=} \boldsymbol{C}_{\mathrm{a}}\boldsymbol{A}$ and $\boldsymbol{D} \stackrel{\mathrm{def}}{=} \boldsymbol{C}_{\mathrm{a}}\boldsymbol{B}$. $\boldsymbol{A}$ is the plant matrix, $\boldsymbol{B}$ the input influence matrix, $\boldsymbol{C}$ the output influence matrix and $\boldsymbol{D}$ the direct throughput matrix. The *direct* or *forward* problem, i.e. computing the system response from a known input sequence $\boldsymbol{u} \stackrel{\mathrm{def}}{=} [\boldsymbol{u}_N^{\mathrm{T}}\ \boldsymbol{u}_{N-1}^{\mathrm{T}}\ \dots\ \boldsymbol{u}_0^{\mathrm{T}}]^{\mathrm{T}}$ with initial states $\boldsymbol{x}_0$ at time $k = 0$, can be written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}^{k+1}\boldsymbol{x}_0 + \sum_{i=0}^{k} \boldsymbol{A}^i\boldsymbol{B}\boldsymbol{u}_{k-i} \tag{6a}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{A}^k\boldsymbol{x}_0 + \boldsymbol{D}\boldsymbol{u}_k + \sum_{i=0}^{k-1} \boldsymbol{C}\boldsymbol{A}^i\boldsymbol{B}\boldsymbol{u}_{k-1-i} \tag{6b}$$

for times $k = 0, \dots, N$, with the definition $\sum_{i=0}^{-1} \dots \stackrel{\mathrm{def}}{=} \boldsymbol{0}$. Equation (6b) can be written in the form

$$\boldsymbol{y}_k = \boldsymbol{H}_k^0\boldsymbol{x}_0 + \sum_{i=0}^{k} \boldsymbol{H}_i\boldsymbol{u}_{k-i} \tag{7}$$

where

$$\boldsymbol{H}_i = \begin{cases} \boldsymbol{D} & i = 0 \\ \boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{B} & i = 1, \dots, N \end{cases} \tag{8}$$

$$\boldsymbol{H}_i^0 = \boldsymbol{C}\boldsymbol{A}^i \qquad i = 0, \dots, N \tag{9}$$

Here, $\boldsymbol{H}_i \in \mathbb{R}^{n_{\mathrm{o}} \times n_{\mathrm{i}}}$ are the *Markov* (impulse response) parameters and $\boldsymbol{H}_i^0 \in \mathbb{R}^{n_{\mathrm{o}} \times n_{\mathrm{s}}}$ represents the influence from initial conditions $\boldsymbol{x}_0$ on the output at time step i. The output of the system for a given input can be solved in a straightforward manner if the parameters $\boldsymbol{H}_i$ and $\boldsymbol{H}_i^0$ are known (from analysis or experiments) for all time steps.

Equation (7) can be re-arranged into

$$\boldsymbol{y}_k^0 \stackrel{\mathrm{def}}{=} \boldsymbol{y}_k - \boldsymbol{H}_k^0\boldsymbol{x}_0 = \sum_{i=0}^{k} \boldsymbol{H}_i\boldsymbol{u}_{k-i} \tag{10}$$

where $\boldsymbol{y}_k^0$ is the output at time k, which has been compensated for effects of non-zero initial conditions. Expressing this relation for each discrete time $k = 0, \dots, N$ in block matrix form yields

$$\underbrace{\begin{bmatrix} \boldsymbol{H}_0 & \boldsymbol{H}_1 & \dots & \boldsymbol{H}_N \\ \boldsymbol{0} & \boldsymbol{H}_0 & \dots & \boldsymbol{H}_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{H}_0 \end{bmatrix}}_{\stackrel{\mathrm{def}}{=} \overline{\boldsymbol{H}}_0} \begin{bmatrix} \boldsymbol{u}_N \\ \boldsymbol{u}_{N-1} \\ \vdots \\ \boldsymbol{u}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_N^0 \\ \boldsymbol{y}_{N-1}^0 \\ \vdots \\ \boldsymbol{y}_0^0 \end{bmatrix} \tag{11}$$

The coefficient matrix $\overline{\boldsymbol{H}}_0$ is in upper block triangular Toeplitz form and consists of $(N+1)^2$ matrix blocks, each of dimension $n_{\mathrm{o}} \times n_{\mathrm{i}}$. The condition number of $\overline{\boldsymbol{H}}_0$ grows as its size increases, i.e. as the number of time steps N increases. Thus, for long time series, $\overline{\boldsymbol{H}}_0$ may become ill-conditioned. For extensive treatment of basic linear system theory, see [16].

2.2 Input estimation problem

The *inverse* or *input estimation* problem constitutes computation of the input sequence $\boldsymbol{u}$ for a known sequence of measurements $\hat{\boldsymbol{y}} \stackrel{\mathrm{def}}{=} [\hat{\boldsymbol{y}}_N^{\mathrm{T}}\ \hat{\boldsymbol{y}}_{N-1}^{\mathrm{T}}\ \dots\ \hat{\boldsymbol{y}}_0^{\mathrm{T}}]^{\mathrm{T}}$ compensated for effects of non-zero initial conditions $\hat{\boldsymbol{y}}^0 \stackrel{\mathrm{def}}{=} [(\hat{\boldsymbol{y}}_N^0)^{\mathrm{T}}\ (\hat{\boldsymbol{y}}_{N-1}^0)^{\mathrm{T}}\ \dots\ (\hat{\boldsymbol{y}}_0^0)^{\mathrm{T}}]^{\mathrm{T}} \stackrel{\mathrm{def}}{=} \hat{\boldsymbol{y}} - [(\boldsymbol{H}_N^0\boldsymbol{x}_0)^{\mathrm{T}}\ (\boldsymbol{H}_{N-1}^0\boldsymbol{x}_0)^{\mathrm{T}}\ \dots\ (\boldsymbol{H}_0^0\boldsymbol{x}_0)^{\mathrm{T}}]^{\mathrm{T}}$ of the outputs. Solving the input estimation problem in the least squares sense corresponds in the algebraic problem that minimizes the norm of the residual $\overline{\boldsymbol{H}}_0\boldsymbol{u} - \hat{\boldsymbol{y}}^0$ as

$$\min_{\boldsymbol{u}} \|\overline{\boldsymbol{H}}_0\boldsymbol{u} - \hat{\boldsymbol{y}}^0\|_2^2 \tag{12}$$

with the straightforward solution

$$\boldsymbol{u}_{\mathrm{LS}} = \overline{\boldsymbol{H}}_0^{+}\hat{\boldsymbol{y}}^0 \tag{13}$$

Here, $\overline{\boldsymbol{H}}_0^{+} \stackrel{\mathrm{def}}{=} [\overline{\boldsymbol{H}}_0^{\mathrm{T}}\overline{\boldsymbol{H}}_0]^{-1}\overline{\boldsymbol{H}}_0^{\mathrm{T}}$ is the Moore-Penrose pseudoinverse of $\overline{\boldsymbol{H}}_0$. For a case where n_{o}=n_{i}, $\overline{\boldsymbol{H}}_0$ will be a square matrix and $\overline{\boldsymbol{H}}_0^{+}$ becomes the regular matrix inverse $\overline{\boldsymbol{H}}_0^{-1}$. It follows that the existence of a unique solution requires that the block diagonal element $\boldsymbol{H}_0$ in $\overline{\boldsymbol{H}}_0$ must be of full column rank. Moreover, this implies that the entire block matrix $\overline{\boldsymbol{H}}_0$ is of full column rank and that there must be at least as many equations as there are unknowns, i.e. $n_o \geq n_i$. This will be assumed to hold throughout the rest of this paper. It should be noted that $\overline{\boldsymbol{H}}_0$ drops rank if at least one of the inputs does not have instant influence on any of the outputs, the so-called *non-collocated* input/output configuration. This problem can be overcome by reformulating the Markov parameters according to [8].

Equation (12) represents a discretization of an ill-posed problem. Consider the associated continuous-time convolution integral relation of Eq. (12) given as

$$\boldsymbol{y}(t) = \int_0^t \boldsymbol{H}(t-\tau)\boldsymbol{u}(\tau)d\tau \tag{14}$$

where $\boldsymbol{H}$ denotes the continuous-time impulse response kernel representation of $\overline{\boldsymbol{H}}_0$. In Eq. (14), $\boldsymbol{u}(t)$ does not depend continuously on the output $\boldsymbol{y}(t)$. The integral operator diminishes the effects of rapid oscillations (noise) in $\boldsymbol{u}(t)$, i.e. small changes (noise) in the outputs may correspond to large changes in the predicted inputs. In general, any attempt to solve Eq. (12) with Eq. (13) will produce meaningless results unless $\boldsymbol{u}$ is restricted. Regularization methods are frequently adopted to incorporate additional information of the sought solution, i.e. they impose restrictions on $\boldsymbol{u}$.

One of the most successful and widely known regularization methods is *Tikhonov regularization* or *damped least squares*. Tikhonov regularization was originally invented independently by Phillips [9] and by Tikhonov [10], cf also [11] and [12]. The restrictions on $\boldsymbol{u}$ are imposed by an *a priori* bound on $\|\boldsymbol{L}_i\boldsymbol{u}\|_2$ modifying Eq. (12) to

$$\min_{\boldsymbol{u}}\{\|\overline{\boldsymbol{H}}_0\boldsymbol{u} - \hat{\boldsymbol{y}}^0\|_2^2 + \lambda\|\boldsymbol{L}_i\boldsymbol{u}\|_2^2\} \tag{15}$$

where λ is a real regularization parameter that controls the balance between the restrictions on $\boldsymbol{u}$ and the minimization of the residual norm. $\boldsymbol{L}_i \in \mathbb{R}^{n_L \times n_i(N+1)}$, where $n_L \leq n_i(N+1)$, is typically a discrete approximation to the ith-order derivative operator. For example, *zeroth-order* Tikhonov regularization corresponds to choosing $\boldsymbol{L}_0 = \boldsymbol{I} \in \mathbb{R}^{n_i(N+1)\times n_i(N+1)}$ and *first-order* Tikhonov regularization corresponds to choosing

$$\boldsymbol{L}_1 = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & -\boldsymbol{I} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{I} & -\boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{n_i N \times n_i(N+1)}, \qquad \boldsymbol{I} \in \mathbb{R}^{n_i \times n_i}$$

except for a scaling factor. For simplicity, $\boldsymbol{L}_0$ will be utilized throughout the rest of this paper.

The regularization parameter λ is still to be determined. Denote the regularized solutions of Eq. (15) by $\boldsymbol{u}^\lambda$. A log-log plot of the residual norm and the norm of the solution, i.e. a log-log plot of $(\|\overline{\boldsymbol{H}}_0\boldsymbol{u}^\lambda - \hat{\boldsymbol{y}}^0\|_2, \|\boldsymbol{L}_i\boldsymbol{u}^\lambda\|_2)$ for $\lambda \in [0,\infty)$, will in general produce an L-shaped curve. The regularization parameter can be chosen in correspondence to the maximum curvature in this graph. Since the true solution $\hat{\boldsymbol{u}}$ is unknown, the optimal regularization parameter minimizing $\|\hat{\boldsymbol{u}} - \boldsymbol{u}^\lambda\|_2$ cannot be obtained in general. The use of L-curves to find near optimal regularization parameters is analyzed and advocated by Hansen in [12].

2.3 Dynamic programming algorithm

Dynamic programming (abbreviated DP) is a classical method for solving certain kinds of optimization problems, see [17]. In [2] the DP algorithm appears as a recursive and explicit algorithm that solves

$$\min_{\boldsymbol{u}}\{\|\sqrt{\boldsymbol{W}_D}(\boldsymbol{y} - \hat{\boldsymbol{y}})\|_2^2 + \lambda\|\boldsymbol{L}_i\boldsymbol{u}\|_2^2\} \tag{16}$$

for a given λ and with the output sequence $\boldsymbol{y} \stackrel{\text{def}}{=} [\boldsymbol{y}_N^T\ \boldsymbol{y}_{N-1}^T\ \dots\ \boldsymbol{y}_0^T]^T$ given by Eq. (5b). Equation (16) differs from Eq. (15) in $\boldsymbol{W}_D$ only. $\boldsymbol{W}_D$ is a block diagonal weighting matrix with $(N+1)$ equal, positive definite and symmetric block diagonal elements $\boldsymbol{W} \in \mathbb{R}^{n_o\times n_o}$. The derivation in [2] sets out from a discrete first order state-space formulation, see Eq. (5), but without the direct throughput matrix in Eq. (5b). Thus, it does not allow for direct coupling of the inputs and outputs. An extended derivation including the direct throughput matrix is given by Nordström in [18].

The DP algorithm constitutes two principle steps: a descending and an ascending sweep. The descending sweep establishes the input/output relationships at each discrete time, whereas the ascending sweep calculates the input that minimizes Eq. (16) using the relationships established during the descending sweep and state sequences given by Eq. (5a). The algorithm derived for zeroth-order Tikhonov regularization in [18] is outlined next, with minor modifications.

2.3.1 Descending sweep

Define the following matrices

$$\boldsymbol{E} \stackrel{\text{def}}{=} -(\boldsymbol{D}^T\boldsymbol{W}\boldsymbol{D} + \lambda\boldsymbol{I})^{-1}\boldsymbol{D}^T\boldsymbol{W} \tag{17}$$

$$\widetilde{\boldsymbol{W}} \stackrel{\text{def}}{=} (\boldsymbol{I} + \boldsymbol{D}\boldsymbol{E})^T\boldsymbol{W}(\boldsymbol{I} + \boldsymbol{D}\boldsymbol{E}) + \lambda\boldsymbol{E}^T\boldsymbol{E} \tag{18}$$

For time step $k = N$ calculate

$$\begin{gathered} \boldsymbol{R}_N = \boldsymbol{C}^T\widetilde{\boldsymbol{W}}\boldsymbol{C}, \quad \boldsymbol{s}_N = \boldsymbol{C}^T\widetilde{\boldsymbol{W}}\hat{\boldsymbol{y}}_N \\ \boldsymbol{V}_N = \boldsymbol{E}\boldsymbol{C}, \quad \boldsymbol{w}_N = -\boldsymbol{E}\hat{\boldsymbol{y}}_N \end{gathered} \tag{19}$$

For time steps $k = N-1, N-2, \ldots, 0$ calculate

$$\begin{aligned} \boldsymbol{G}_{k+1} &= (\boldsymbol{D}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{D} + \lambda\boldsymbol{I} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}_{k+1}\boldsymbol{B})^{-1} \\ \boldsymbol{V}_k &= -\boldsymbol{G}_{k+1}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{C} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}_{k+1}\boldsymbol{A}) \\ \boldsymbol{w}_k &= \boldsymbol{G}_{k+1}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{W}\hat{\boldsymbol{y}}_k + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{s}_{k+1}) \\ \boldsymbol{R}_k &= \boldsymbol{C}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{C} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{R}_{k+1}\boldsymbol{A} + (\boldsymbol{D}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{C} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}_{k+1}\boldsymbol{A})^{\mathrm{T}}\boldsymbol{V}_k \\ \boldsymbol{s}_k &= (\boldsymbol{C}^{\mathrm{T}}\boldsymbol{W} + \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{W})\hat{\boldsymbol{y}}_k + (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{V}_k)^{\mathrm{T}}\boldsymbol{s}_{k+1} \end{aligned} \tag{20}$$

This is known as the descending sweep, from which the matrices $\boldsymbol{V}_k$ and the vectors $\boldsymbol{w}_k$ need to be stored for later use in the ascending sweep.

2.3.2 Ascending sweep

For time steps $k = 0, 1, \ldots, N$ compute the optimal inputs according to

$$\begin{aligned} \boldsymbol{u}_k^{\mathrm{opt}} &= \boldsymbol{V}_k\boldsymbol{x}_k + \boldsymbol{w}_k \\ \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k^{\mathrm{opt}} \end{aligned} \tag{21}$$

for a given initial state $\boldsymbol{x}_0$. This is known as the ascending sweep.

3 EXPLICIT BLOCK INVERSION ALGORITHMS

The damped least squares problem given by Eq. (15) with $\boldsymbol{L}_i$ taken as $\boldsymbol{L}_0 = \boldsymbol{I}$, i.e. zeroth-order Tikhonov regularization, is equivalent to an expanded or enlarged formulation given as

$$\min_{\boldsymbol{u}} \left\| \begin{bmatrix} \overline{\boldsymbol{H}}_0 \\ \sqrt{\lambda}\boldsymbol{I} \end{bmatrix} \boldsymbol{u} - \begin{bmatrix} \hat{\boldsymbol{y}}^0 \\ \boldsymbol{0} \end{bmatrix} \right\|_2 \tag{22}$$

The solution to Eq. (22) may be formed in numerous ways, e.g. by use of QR decomposition, (truncated) singular value decomposition, or computation of an inverse matrix by forming the corresponding normal equations. For solution methods for this kind of enlarged system see [15], [19] and [20]. Moreover, extensive treatment of numerical methods for least squares problems is given in [21]. Here, explicit block inversion algorithms are derived in state-space matrix form, see Section 3.2.2 and Markov parameter form, see Sections 3.1 and 3.2.3. The algorithms give valuable insight to the exact sequence of operations in the matrix inversion. The explicit block inverses formulated in this section, are used to introduce a new regularization procedure, see Section 4.

3.1 Square system

Consider the special case $n_{\mathrm{o}} = n_{\mathrm{i}}$ of Eq. (11). As previously stated, $\overline{\boldsymbol{H}}_0$ will be square and $\overline{\boldsymbol{H}}_0^{+} \equiv \overline{\boldsymbol{H}}_0^{-1}$. The inverse can be written in explicit block form according to

$$\overline{\boldsymbol{H}}_0^{-1} \stackrel{\mathrm{def}}{=} \begin{bmatrix} \tilde{\boldsymbol{H}}_0 & \tilde{\boldsymbol{H}}_1 & \ldots & \tilde{\boldsymbol{H}}_N \\ \boldsymbol{0} & \tilde{\boldsymbol{H}}_0 & \ldots & \tilde{\boldsymbol{H}}_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \tilde{\boldsymbol{H}}_0 \end{bmatrix} \tag{23}$$

with block elements defined as

$$\tilde{\boldsymbol{H}}_0 \stackrel{\mathrm{def}}{=} \boldsymbol{H}_0^{-1} \tag{24a}$$

$$\tilde{\boldsymbol{H}}_k \stackrel{\mathrm{def}}{=} -\sum_{i=1}^{k} \tilde{\boldsymbol{H}}_{k-i}\boldsymbol{H}_i\boldsymbol{H}_0^{-1} \qquad k = 1, 2, \ldots, N \tag{24b}$$

Proof. The proof follows from the definition of the inverse. $\overline{\boldsymbol{H}}_0^{-1}\overline{\boldsymbol{H}}_0$ is upper triangular and given as

$$\overline{\boldsymbol{H}}_0^{-1}\overline{\boldsymbol{H}}_0 = \begin{bmatrix} \boldsymbol{I}_0 & \boldsymbol{I}_1 & \ldots & \boldsymbol{I}_N \\ \boldsymbol{0} & \boldsymbol{I}_0 & \ldots & \boldsymbol{I}_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{I}_0 \end{bmatrix}$$

where

$$\boldsymbol{I}_k = \sum_{i=0}^{k} \tilde{\boldsymbol{H}}_{k-i}\boldsymbol{H}_i \qquad k = 0, 1, \ldots, N \tag{25}$$

which can be verified by using Eqs. (11) and (23). Insert Eq. (24a) in Eq. (25) with $k = 0$

$$\boldsymbol{I}_0 = \tilde{\boldsymbol{H}}_0\boldsymbol{H}_0 = \boldsymbol{H}_0^{-1}\boldsymbol{H}_0 = \boldsymbol{I}$$

and Eq. (24b) in Eq. (25) with $k > 0$

$$\begin{aligned} \boldsymbol{I}_k &= \sum_{i=0}^{k} \tilde{\boldsymbol{H}}_{k-i}\boldsymbol{H}_i = \sum_{i=1}^{k} \tilde{\boldsymbol{H}}_{k-i}\boldsymbol{H}_i + \tilde{\boldsymbol{H}}_k\boldsymbol{H}_0 = \\ &= \sum_{i=1}^{k} \tilde{\boldsymbol{H}}_{k-i}\boldsymbol{H}_i - \sum_{i=1}^{k} \tilde{\boldsymbol{H}}_{k-i}\boldsymbol{H}_i\boldsymbol{H}_0^{-1}\boldsymbol{H}_0 = \boldsymbol{0} \end{aligned}$$

yielding

$$\overline{\boldsymbol{H}}_0^{-1}\overline{\boldsymbol{H}}_0 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \ldots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{I} \end{bmatrix}$$

i.e. $\overline{\boldsymbol{H}}_0^{-1}$ is the inverse of $\overline{\boldsymbol{H}}_0$ by definition. $\square$

An alternative explicit solution can be derived setting out from the discrete state-space formulation given by Eq. (5). Extracting $\boldsymbol{u}_k$ from Eq. (5b) and inserting it into Eq. (5a) yields the inverse representation

$$\boldsymbol{x}_{k+1} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{y}_k \tag{26a}$$
$$\boldsymbol{u}_k = -\boldsymbol{D}^{-1}\boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}^{-1}\boldsymbol{y}_k \tag{26b}$$

in which the input/output relation has been interchanged. The corresponding explicit inverse block elements are

$$\hat{\boldsymbol{H}}_0 \overset{\text{def}}{=} \boldsymbol{D}^{-1} \tag{27a}$$
$$\hat{\boldsymbol{H}}_k \overset{\text{def}}{=} -\boldsymbol{D}^{-1}\boldsymbol{C}(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C})^{k-1}\boldsymbol{B}\boldsymbol{D}^{-1} \qquad k = 1, 2, \ldots, N \tag{27b}$$

with the same matrix structure as Eq. (23). The two principally different approaches to the explicit inverses are mathematically equivalent.

It should be noted that $\boldsymbol{H}_0^{-1}$ does not exist when $n_\text{o} > n_\text{i}$. A mere substitution of $\boldsymbol{H}_0^{-1}$ (or $\boldsymbol{D}^{-1}$) to $\boldsymbol{H}_0^{+} = (\boldsymbol{H}_0^\text{T}\boldsymbol{H}_0)^{-1}\boldsymbol{H}_0^\text{T}$ (or $\boldsymbol{D}^{+} = (\boldsymbol{D}^\text{T}\boldsymbol{D})^{-1}\boldsymbol{D}^\text{T}$) in Eq. (24) (or Eq. (26)) will not produce the least squares solution of Eq. (11) in this case.

3.2 Regularized system

Consider the damped least squares problem given by Eq. (22) where $n_\text{o} \geq n_\text{i}$ and $\lambda \in [0, \infty)$. The problem satisfies the normal equations

$$(\overline{\boldsymbol{H}}_0^\text{T}\overline{\boldsymbol{H}}_0 + \lambda\boldsymbol{I})\boldsymbol{u}^\lambda = \overline{\boldsymbol{H}}_0^\text{T}\hat{\boldsymbol{y}}^0 \tag{28}$$

$\overline{\boldsymbol{H}}_0^\text{T}\overline{\boldsymbol{H}}_0 + \lambda\boldsymbol{I}$ is symmetric and positive definite, yielding a unique regularized solution $\boldsymbol{u}^\lambda$ to Eq. (28) according to

$$\boldsymbol{u}^\lambda = (\overline{\boldsymbol{H}}_0^\text{T}\overline{\boldsymbol{H}}_0 + \lambda\boldsymbol{I})^{-1}\overline{\boldsymbol{H}}_0^\text{T}\hat{\boldsymbol{y}}^0 \tag{29}$$

for a given $\lambda > 0$. The block elements in

$$\widehat{\boldsymbol{H}} \overset{\text{def}}{=} \begin{bmatrix} \widehat{\boldsymbol{H}}_{N,N} & \ldots & \widehat{\boldsymbol{H}}_{N,1} & \widehat{\boldsymbol{H}}_{N,0} \\ \vdots & \ddots & \vdots & \vdots \\ \widehat{\boldsymbol{H}}_{1,N} & \ldots & \widehat{\boldsymbol{H}}_{1,1} & \widehat{\boldsymbol{H}}_{1,0} \\ \widehat{\boldsymbol{H}}_{0,N} & \ldots & \widehat{\boldsymbol{H}}_{0,1} & \widehat{\boldsymbol{H}}_{0,0} \end{bmatrix} \overset{\text{def}}{=} (\overline{\boldsymbol{H}}_0^\text{T}\overline{\boldsymbol{H}}_0 + \lambda\boldsymbol{I})^{-1}\overline{\boldsymbol{H}}_0^\text{T} \tag{30}$$

can be derived explicitly by use of the DP algorithm, see Section 3.2.2. Here, the block elements are expressed in the state-space matrices. In Section 3.2.3, the block elements of

$$\breve{\boldsymbol{H}} \overset{\text{def}}{=} \begin{bmatrix} \breve{\boldsymbol{H}}_{N,N} & \ldots & \breve{\boldsymbol{H}}_{N,1} & \breve{\boldsymbol{H}}_{N,0} \\ \vdots & \ddots & \vdots & \vdots \\ \breve{\boldsymbol{H}}_{1,N} & \ldots & \breve{\boldsymbol{H}}_{1,1} & \breve{\boldsymbol{H}}_{1,0} \\ \breve{\boldsymbol{H}}_{0,N} & \ldots & \breve{\boldsymbol{H}}_{0,1} & \breve{\boldsymbol{H}}_{0,0} \end{bmatrix} \overset{\text{def}}{=} (\overline{\boldsymbol{H}}_0^\text{T}\overline{\boldsymbol{H}}_0 + \lambda\boldsymbol{I})^{-1} \tag{31}$$

are expressed in the Markov parameters. Note the indexing order of the block elements in $\widehat{\boldsymbol{H}}$ and $\breve{\boldsymbol{H}}$. The two principally different approaches in forming the explicit inverses; in state-space matrix form and in Markov parameter form, are mathematically equivalent.

3.2.1 Dynamic programming algorithm - modified

Three issues concerning the DP algorithm Eqs. (17)–(21) need to be addressed before the block elements of Eqs. (30) and (31) can be derived. These issues are:

- The weighting matrix $\boldsymbol{W}$ introduced in the DP algorithm needs to be accounted for.
- Effects of non-zero initial conditions on the measurements $\hat{\boldsymbol{y}}$ need to be compensated for.
- The explicit state sequence $\boldsymbol{x} \overset{\text{def}}{=} [\boldsymbol{x}_N^\text{T}\ \boldsymbol{x}_{N-1}^\text{T}\ \ldots\ \boldsymbol{x}_0^\text{T}]^\text{T}$ that the DP algorithm depends on, Eq. (21), needs to be embedded in the formulation.

A weighted version of Eq. (29) equivalent to the solution of the DP algorithm can be derived by a change of definitions according to

$$\boldsymbol{C} \rightarrow \boldsymbol{C}_\text{w} \overset{\text{def}}{=} \sqrt{\boldsymbol{W}}\boldsymbol{C} \qquad \boldsymbol{D} \rightarrow \boldsymbol{D}_\text{w} \overset{\text{def}}{=} \sqrt{\boldsymbol{W}}\boldsymbol{D} \qquad \hat{\boldsymbol{y}} \rightarrow \hat{\boldsymbol{y}}^\text{W} \overset{\text{def}}{=} \sqrt{\boldsymbol{W}_\text{D}}\hat{\boldsymbol{y}} \tag{32}$$

Furthermore, applying these changes to the DP algorithm eliminates $\boldsymbol{W}$ in Eqs. (17)–(20) since the weighting matrix is incorporated in the new definitions given by Eq. (32). Throughout the rest of this paper, the weighting matrix will be chosen as $\boldsymbol{W} = \boldsymbol{I}$ for simplicity, but without loss of generality since these definitions can be incorporated at any stage of the proceeding derivations.

The last two issues given above can be overcome by substituting $\hat{\boldsymbol{y}}$ in Eqs. (19) and (20) with $\hat{\boldsymbol{y}}^0$ and substituting Eq. (21) with

$$\boldsymbol{u}_k^\lambda = \boldsymbol{V}_k \sum_{p=1}^{k} \boldsymbol{A}^{p-1}\boldsymbol{B}\boldsymbol{u}_{k-p}^\lambda + \boldsymbol{w}_k \tag{33}$$

where $\sum_{p=1}^{0} \ldots \overset{\text{def}}{=} \boldsymbol{0}$. By the modifications introduced in this section, the modified DP algorithm will produce a mathematically equivalent solution to that of Eq. (29).

3.2.2 Explicit inverse expressed in state-space matrices

Consider the bottom block row of Eq. (29), with block element definitions according to Eq. (30), given as

$$\boldsymbol{u}_0^\lambda = \widehat{\boldsymbol{H}}_{0,N}\hat{\boldsymbol{y}}_N^0 + \ldots + \widehat{\boldsymbol{H}}_{0,1}\hat{\boldsymbol{y}}_1^0 + \widehat{\boldsymbol{H}}_{0,0}\hat{\boldsymbol{y}}_0^0 \tag{34}$$

It is also given by the modified DP algorithm as

$$\boldsymbol{u}_0^\lambda = \boldsymbol{w}_0 \tag{35}$$

where $\boldsymbol{w}_0$ is given in Eqs. (19) and (20) with $\sqrt{\boldsymbol{W}} = \boldsymbol{I}$ and $\hat{\boldsymbol{y}}$ substituted with $\hat{\boldsymbol{y}}^0$ according to Section 3.2.1. The bottom block row in Eq. (30) can be established by expanding $\boldsymbol{w}_0$ in Eq. (35) and comparing it to Eq. (34), yielding by identification of terms

$$\begin{aligned}
\widehat{\boldsymbol{H}}_{0,0} &= \boldsymbol{G}_1\boldsymbol{D}^{\mathrm{T}} \\
\widehat{\boldsymbol{H}}_{0,1} &= \boldsymbol{G}_1\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{V}_1^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}) \\
\widehat{\boldsymbol{H}}_{0,k} &= \boldsymbol{G}_1\boldsymbol{B}^{\mathrm{T}}(\textstyle\prod_{l=1}^{k-1}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{V}_l)^{\mathrm{T}})(\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}) \\
\widehat{\boldsymbol{H}}_{0,N} &= \boldsymbol{G}_1\boldsymbol{B}^{\mathrm{T}}(\textstyle\prod_{l=1}^{N-1}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{V}_l)^{\mathrm{T}})\boldsymbol{C}^{\mathrm{T}}\widetilde{\boldsymbol{W}}
\end{aligned} \tag{36}$$

where $k = 2, \ldots, N-1$. Proceeding in the same manner, the ith block row $0 < i < N$ is established by comparing

$$\boldsymbol{u}_i^\lambda = \widehat{\boldsymbol{H}}_{i,N}\hat{\boldsymbol{y}}_N^0 + \ldots + \widehat{\boldsymbol{H}}_{i,1}\hat{\boldsymbol{y}}_1^0 + \widehat{\boldsymbol{H}}_{i,0}\hat{\boldsymbol{y}}_0^0 \tag{37}$$

with

$$\boldsymbol{u}_i^\lambda = \boldsymbol{V}_i\sum_{p=1}^{i}\boldsymbol{A}^{p-1}\boldsymbol{B}\boldsymbol{u}_{i-p}^\lambda + \boldsymbol{w}_i \tag{38}$$

yielding

$$\begin{aligned}
\widehat{\boldsymbol{H}}_{i,j} &= \boldsymbol{V}_i\textstyle\sum_{p=1}^{i}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{i-p,j} \\
\widehat{\boldsymbol{H}}_{i,i} &= \boldsymbol{V}_i\textstyle\sum_{p=1}^{i}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{i-p,i} + \boldsymbol{G}_{i+1}\boldsymbol{D}^{\mathrm{T}} \\
\widehat{\boldsymbol{H}}_{i,i+1} &= \boldsymbol{V}_i\textstyle\sum_{p=1}^{i}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{i-p,i+1} + \boldsymbol{G}_{i+1}\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{V}_{i+1}^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}) \\
\widehat{\boldsymbol{H}}_{i,k} &= \boldsymbol{V}_i\textstyle\sum_{p=1}^{i}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{i-p,k} \\
&\quad + \boldsymbol{G}_{i+1}\boldsymbol{B}^{\mathrm{T}}(\textstyle\prod_{l=i+1}^{k-1}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{V}_l)^{\mathrm{T}})(\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{D}^{\mathrm{T}}) \\
\widehat{\boldsymbol{H}}_{i,N} &= \boldsymbol{V}_i\textstyle\sum_{p=1}^{i}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{i-p,N} \\
&\quad + \boldsymbol{G}_{i+1}\boldsymbol{B}^{\mathrm{T}}(\textstyle\prod_{l=i+1}^{N-1}(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{V}_l)^{\mathrm{T}})\boldsymbol{C}^{\mathrm{T}}\widetilde{\boldsymbol{W}}
\end{aligned} \tag{39}$$

where $j = 0, 1, \ldots, i-1$ and $k = i+2, i+3, \ldots, N-1$. The Nth block row is given by

$$\boldsymbol{u}_N^\lambda = \widehat{\boldsymbol{H}}_{N,N}\hat{\boldsymbol{y}}_N^0 + \ldots + \widehat{\boldsymbol{H}}_{N,1}\hat{\boldsymbol{y}}_1^0 + \widehat{\boldsymbol{H}}_{N,0}\hat{\boldsymbol{y}}_0^0 \tag{40}$$

or

$$\boldsymbol{u}_N^\lambda = \boldsymbol{V}_N\sum_{p=1}^{N}\boldsymbol{A}^{p-1}\boldsymbol{B}\boldsymbol{u}_{N-p}^\lambda + \boldsymbol{w}_N = \boldsymbol{V}_N\sum_{p=1}^{N}\boldsymbol{A}^{p-1}\boldsymbol{B}\boldsymbol{u}_{N-p}^\lambda - \boldsymbol{E}\hat{\boldsymbol{y}}_N^0 \tag{41}$$

Identification of terms yields

$$\begin{aligned}
\widehat{\boldsymbol{H}}_{N,j} &= \boldsymbol{V}_N\textstyle\sum_{p=1}^{N}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{N-p,j} \\
\widehat{\boldsymbol{H}}_{N,N} &= \boldsymbol{V}_N\textstyle\sum_{p=1}^{N}\boldsymbol{A}^{p-1}\boldsymbol{B}\widehat{\boldsymbol{H}}_{N-p,N} - \boldsymbol{E}
\end{aligned} \tag{42}$$

where $j = 0, 1, \ldots, N-1$. This concludes the derivation of the explicit block elements defined in Eq. (30). The block elements, expressed in state-space matrices, are given by Eqs. (36), (39), and (42). It can be verified that Eq. (30) becomes upper block triangular and with equivalent block elements as in Eq. (27) when $n_{\mathrm{o}} = n_{\mathrm{i}}$ and $\lambda = 0$.

3.2.3 Explicit inverse expressed in Markov parameters

By utilizing the results given in Section 3.2.2, i.e. Eqs. (36), (39), and (42), it is possible to express the block elements of Eq. (31) in Markov parameter form, see Eq. (8). Note that the $\boldsymbol{G}$-matrices in Eqs. (36) and (39) involve explicit inverses, see Eq. (20). A mathematically equivalent algorithm which redefines these matrices in Markov parameter form can be derived from Eqs. (17)–(20) using Eq. (8). The details of the derivation are very tedious and are therefore omitted. Consider $\boldsymbol{G}_{N-r}$, cf Eq. (20), given as

$$\begin{aligned}
\boldsymbol{G}_{N-r} &= (\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} + \lambda\boldsymbol{I} + \boldsymbol{R}_{N-r}^{\{0,0\}})^{-1} = (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I} + \boldsymbol{R}_{N-r}^{\{0,0\}})^{-1} \\
&\overset{\text{def}}{=} (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I} + \boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}_{N-r}\boldsymbol{B})^{-1}
\end{aligned} \tag{43}$$

where $r = 0, 1, \ldots, N-1$ and $\{0,0\}$ denotes an index pair given by row 1 of

$$\begin{array}{lllllllll}
\text{row } 1: & \{k,j\} & \rightarrow & \{0,0\} & & & & & \\
\text{row } 2: & \{k,j\} & \rightarrow & \{1,1\} & \{0,1\} & & & & \\
\text{row } 3: & \{k,j\} & \rightarrow & \{2,2\} & \{1,2\} & \{0,2\} & \{0,1\} & & \\
\text{row } 4: & \{k,j\} & \rightarrow & \{3,3\} & \{2,3\} & \{1,3\} & \{1,2\} & \{0,3\} & \{0,2\} & \{0,1\} \\
 & & \vdots & & & & & & \\
\text{row } N: & \{k,j\} & \rightarrow & \cdots & & & & &
\end{array}$$

which will be referred to as the *index ladder*. A new row in the index ladder is constructed in two steps. Consider the q:th row given as

$$\text{row } q\text{:} \quad \{k,j\} \;\rightarrow\; \{q-1,q-1\} \quad \cdots \quad \{0,1\}$$

where $0 < q \le N$. The next row $q+1$ is given by first adding $\{1,1\}$ to each index pair of row q and, secondly, the row is expanded by a decreasing set of index pairs according to

$$\text{row } q+1\text{:} \quad \{k,j\} \;\rightarrow\; \underbrace{\{q,q\} \quad \cdots \quad \{1,2\}}_{\text{row } q+\{1,1\}} \quad \underbrace{\{0,q\} \quad \{0,q-1\} \quad \cdots \quad \{0,1\}}_{\text{expansion}}$$

$\boldsymbol{R}_{N-r}^{\{0,0\}}$, cf Eq. (43), is given recursively for all r by a descending sweep starting the recursion with the set of matrices

$$\boldsymbol{R}_N^{\{k,j\}} = \boldsymbol{H}_{k+1}^{\mathrm{T}} \widetilde{\boldsymbol{W}} \boldsymbol{H}_{j+1} \tag{44}$$

for appropriate pairs of indices k and j given by row $r+1$ in the index ladder. $\widetilde{\boldsymbol{W}}$ is written in Markov parameter form as

$$\begin{aligned} \widetilde{\boldsymbol{W}} &= (\boldsymbol{I} + \boldsymbol{H}_0\boldsymbol{E})^{\mathrm{T}}(\boldsymbol{I} + \boldsymbol{H}_0\boldsymbol{E}) + \lambda \boldsymbol{E}^{\mathrm{T}}\boldsymbol{E} \\ \boldsymbol{E} &= -(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1}\boldsymbol{H}_0^{\mathrm{T}} \end{aligned} \tag{45}$$

If $r > 0$, the descending sweep continues to the top row of the index ladder as

$$\begin{aligned} \boldsymbol{R}_{N-s}^{\{k,j\}} = \;& \boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{H}_{j+1} + \boldsymbol{R}_{N-s+1}^{\{k+1,j+1\}} \\ & -(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_{k+1} + \boldsymbol{R}_{N-s+1}^{\{0,k+1\}})^{\mathrm{T}}\boldsymbol{G}_{N-s+1}(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_{j+1} + \boldsymbol{R}_{N-s+1}^{\{0,j+1\}}) \end{aligned} \tag{46}$$

for $s = 1, 2, \ldots, r$ with index pairs $\{k,j\}$ according to row $r-s+1$. Note that Eq. (44) constitutes computation of $(r+1)(r+2)/2$ unique $\boldsymbol{R}$-matrices. The procedure given above is repeated until $\boldsymbol{G}_1$ is established. This concludes the algorithm for establishing the $\boldsymbol{G}$-matrices in Markov parameter form.

Example. $\boldsymbol{G}_N$ and $\boldsymbol{G}_{N-1}$ are established next, to illustrate the algorithm. $\boldsymbol{G}_N$ is given by

$$\boldsymbol{G}_N = (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I} + \boldsymbol{R}_N^{\{0,0\}})^{-1}$$

according to Eq. (43). Since $r = 0$, the appropriate index pair on row r+1 = 1 of the index ladder is already $\{0,0\}$. Thus, Eq. (44) becomes

$$\boldsymbol{R}_N^{\{0,0\}} = \boldsymbol{H}_1^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_1$$

yielding

$$\boldsymbol{G}_N = (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I} + \boldsymbol{H}_1^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_1)^{-1}$$

$\widetilde{\boldsymbol{W}}$ is defined by Eq. (45). $\boldsymbol{G}_{N-1}$ is given by

$$\boldsymbol{G}_{N-1} = (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I} + \boldsymbol{R}_{N-1}^{\{0,0\}})^{-1}$$

With $r = 1$, the set of appropriate index pairs is $\{1,1\}$ and $\{0,1\}$. The recursion starts with the set of matrices given by Eq. (44) yielding

$$\boldsymbol{R}_N^{\{1,1\}} = \boldsymbol{H}_2^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_2 \qquad \boldsymbol{R}_N^{\{0,1\}} = \boldsymbol{H}_1^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_2$$

Equation (46) is used for a one-step recursion to the top (row 1) of the index ladder as

$$\boldsymbol{R}_{N-1}^{\{0,0\}} = \boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{H}_1 + \boldsymbol{R}_N^{\{1,1\}} - (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_1 + \boldsymbol{R}_N^{\{0,1\}})^{\mathrm{T}}\boldsymbol{G}_N(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_1 + \boldsymbol{R}_N^{\{0,1\}})$$

Thus, $\boldsymbol{G}_{N-1}$ is established as

$$\begin{aligned} \boldsymbol{G}_{N-1} = \;& (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I} + \boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{H}_1 + \boldsymbol{H}_2^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_2 \\ & -(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_1 + \boldsymbol{H}_1^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_2)^{\mathrm{T}}\boldsymbol{G}_N(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_1 + \boldsymbol{H}_1^{\mathrm{T}}\widetilde{\boldsymbol{W}}\boldsymbol{H}_2))^{-1} \qquad \square \end{aligned}$$

The explicit inverses $\widehat{\boldsymbol{H}}$ and $\breve{\boldsymbol{H}}$ differ in $\overline{\boldsymbol{H}}_0^{\mathrm{T}}$ only, i.e. $\widehat{\boldsymbol{H}} = \breve{\boldsymbol{H}}\overline{\boldsymbol{H}}_0^{\mathrm{T}}$ cf Eqs. (30) and (31). It is clearer to derive the the block elements of $\breve{\boldsymbol{H}}$ in Markov parameter form than the corresponding block elements of $\widehat{\boldsymbol{H}}$. To accomplish this, $\overline{\boldsymbol{H}}_0^{\mathrm{T}}$ needs to be extracted and eliminated from $\widehat{\boldsymbol{H}}$. Consider the block elements of the bottom row of $\widehat{\boldsymbol{H}}$ given by Eq. (36). Extraction and elimination of the corresponding block elements of $\overline{\boldsymbol{H}}_0^{\mathrm{T}}$ yields the bottom block row of $\breve{\boldsymbol{H}}$ as

$$\begin{aligned}
\breve{\boldsymbol{H}}_{0,0} &= \boldsymbol{G}_1 \\
\breve{\boldsymbol{H}}_{0,1}(\boldsymbol{V}_1^{\mathrm{T}}) &= \breve{\boldsymbol{H}}_{0,0}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{V}_1^{\mathrm{T}} \qquad \text{i.e. } \breve{\boldsymbol{H}}_{0,1} \text{ is a function of } \boldsymbol{V}_1^{\mathrm{T}} \\
\breve{\boldsymbol{H}}_{0,k}(\boldsymbol{V}_k^{\mathrm{T}}) &= \breve{\boldsymbol{H}}_{0,k-1}(\boldsymbol{V}_{k-1}^{\mathrm{T}} \equiv \boldsymbol{A}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}}) + \breve{\boldsymbol{H}}_{0,k-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{V}_k^{\mathrm{T}} \\
\breve{\boldsymbol{H}}_{0,N} &= (\breve{\boldsymbol{H}}_{0,N-1}(\boldsymbol{V}_{N-1}^{\mathrm{T}} \equiv \boldsymbol{A}^{\mathrm{T}}\boldsymbol{C}^{\mathrm{T}}) + \breve{\boldsymbol{H}}_{0,N-1}\boldsymbol{H}_1)\widetilde{\boldsymbol{W}}_{\mathrm{mod}}
\end{aligned} \tag{47}$$

for $k = 2, 3, \ldots, N-1$. $\widetilde{\boldsymbol{W}}_{\mathrm{mod}}$, where $\widetilde{\boldsymbol{W}} = \widetilde{\boldsymbol{W}}_{\mathrm{mod}}\boldsymbol{H}_0^{\mathrm{T}}$, is identified in Markov parameter form according to

$$\begin{aligned}
\widetilde{\boldsymbol{W}}_{\mathrm{mod}} \overset{\mathrm{def}}{=} \; & ((\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1}\boldsymbol{H}_0^{\mathrm{T}})^{\mathrm{T}}\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1} \\
& -2\boldsymbol{H}_0(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1} + \lambda\boldsymbol{H}_0(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1}(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1}
\end{aligned} \tag{48}$$

utilizing that $(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1}$ is symmetric. Consider Eq. (46) for $r = N-1$, i.e. while establishing $\boldsymbol{G}_1$, with the following definition

$$\begin{aligned}
\boldsymbol{R}_{N-s}^{\{k,j\}} = \; & \boldsymbol{H}_{k+1}^{\mathrm{T}}\boldsymbol{H}_{j+1} + \boldsymbol{R}_{N-s+1}^{\{k+1,j+1\}} \\
& \underbrace{-(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_{k+1} + \boldsymbol{R}_{N-s+1}^{\{0,k+1\}})^{\mathrm{T}}\boldsymbol{G}_{N-s+1}}_{\overset{\mathrm{def}}{=}\breve{\boldsymbol{R}}_s^{\{0,k+1\}}}(\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_{j+1} + \boldsymbol{R}_{N-s+1}^{\{0,j+1\}})
\end{aligned} \tag{49}$$

for $s = 1, 2, \ldots, N-1$. In computation, the $\breve{\boldsymbol{R}}$-matrices need to be stored during the descending sweep. Utilizing the definition given in Eq. (49) and the fact that the $\boldsymbol{G}$-matrices are symmetric, Eq. (47) can be rewritten in Markov parameter form by identification of terms as

$$\begin{aligned}
\breve{\boldsymbol{H}}_{0,0} &= \boldsymbol{G}_1 \\
\breve{\boldsymbol{H}}_{0,k} &= \textstyle\sum_{l=1}^{k} \breve{\boldsymbol{H}}_{0,l-1}\breve{\boldsymbol{R}}_k^{\{0,k-l+1\}} \\
\breve{\boldsymbol{H}}_{0,N} &= \textstyle\sum_{l=1}^{N} \breve{\boldsymbol{H}}_{0,l-1}\breve{\boldsymbol{H}}_{N-l+1}^{\mathrm{T}}\widetilde{\boldsymbol{W}}_{\mathrm{mod}}
\end{aligned} \tag{50}$$

for $k = 1, 2, \ldots, N-1$. Proceeding in the same manner, the ith block row $0 < i < N$

of Eq. (39) can be rewritten by identification of terms according to

$$\begin{aligned}
\breve{\boldsymbol{H}}_{i,j} &= \textstyle\sum_{p=1}^{i}(\breve{\boldsymbol{R}}_i^{\{0,p\}})^{\mathrm{T}}\breve{\boldsymbol{H}}_{i-p,j} \\
\breve{\boldsymbol{H}}_{i,i} &= \textstyle\sum_{p=1}^{i}(\breve{\boldsymbol{R}}_i^{\{0,p\}})^{\mathrm{T}}\breve{\boldsymbol{H}}_{i-p,i} + \overset{\triangle}{\boldsymbol{H}}_{i,i} \qquad \text{with } \overset{\triangle}{\boldsymbol{H}}_{i,i} \overset{\mathrm{def}}{=} \boldsymbol{G}_{i+1} \\
\breve{\boldsymbol{H}}_{i,k} &= \textstyle\sum_{p=1}^{i}(\breve{\boldsymbol{R}}_i^{\{0,p\}})^{\mathrm{T}}\breve{\boldsymbol{H}}_{i-p,k} + \overset{\triangle}{\boldsymbol{H}}_{i,k} \qquad \text{with } \overset{\triangle}{\boldsymbol{H}}_{i,k} \overset{\mathrm{def}}{=} \textstyle\sum_{l=i+1}^{k} \overset{\triangle}{\boldsymbol{H}}_{i,l-1}\breve{\boldsymbol{R}}_k^{\{0,k-l+1\}} \\
\breve{\boldsymbol{H}}_{i,N} &= \textstyle\sum_{p=1}^{i}(\breve{\boldsymbol{R}}_i^{\{0,p\}})^{\mathrm{T}}\breve{\boldsymbol{H}}_{i-p,N} + \sum_{l=i+1}^{N} \overset{\triangle}{\boldsymbol{H}}_{i,l-1}\boldsymbol{H}_{N-l+1}^{\mathrm{T}}\widetilde{\boldsymbol{W}}_{\mathrm{mod}}
\end{aligned} \tag{51}$$

for $j = 0, 1, \ldots, i-1$ and $k = i+1, i+2, \ldots, N-1$. Finally, identification of terms in Eq. (42) yields

$$\begin{aligned}
\breve{\boldsymbol{H}}_{N,j} &= \boldsymbol{E}\textstyle\sum_{p=1}^{N}\boldsymbol{H}_p\breve{\boldsymbol{H}}_{N-p,j} \\
\breve{\boldsymbol{H}}_{N,N} &= \boldsymbol{E}\textstyle\sum_{p=1}^{N}\boldsymbol{H}_p\breve{\boldsymbol{H}}_{N-p,N-1} + (\boldsymbol{H}_0^{\mathrm{T}}\boldsymbol{H}_0 + \lambda\boldsymbol{I})^{-1}
\end{aligned} \tag{52}$$

for $j = 0, 1, \ldots, N-1$ with $\boldsymbol{E}$ defined by Eq. (45). This concludes the derivation of the block elements in Eq. (31) in Markov parameter form given by Eqs. (50)–(52) and definitions therein. It should be stressed that only the upper or lower triangular part of the block inverse needs to be established, since $\breve{\boldsymbol{H}}$ of Eq. (31) is symmetric.

4 DYNAMIC TIKHONOV REGULARIZATION

Using the block inversion algorithm given in Section 3.2.3, it is possible to introduce a new type of regularization procedure here called *dynamic Tikhonov regularization*. The block matrices constituting $\breve{\boldsymbol{H}}$, given by Eqs. (50)–(52), are all functions of the scalar regularization parameter λ, i.e. $\breve{\boldsymbol{H}}(\lambda)$. Introduce the definition

$$\breve{\boldsymbol{H}}(\lambda \equiv \lambda_0 \mid \lambda_0) \overset{\mathrm{def}}{=} \begin{bmatrix} \breve{\boldsymbol{H}}_{N,N}(\lambda_0) & \ldots & \breve{\boldsymbol{H}}_{N,1}(\lambda_0) & \breve{\boldsymbol{H}}_{N,0}(\lambda_0) \\ \vdots & \ddots & \vdots & \vdots \\ \breve{\boldsymbol{H}}_{1,N}(\lambda_0) & \ldots & \breve{\boldsymbol{H}}_{1,1}(\lambda_0) & \breve{\boldsymbol{H}}_{1,0}(\lambda_0) \\ \breve{\boldsymbol{H}}_{0,N}(\lambda_0) & \ldots & \breve{\boldsymbol{H}}_{0,1}(\lambda_0) & \breve{\boldsymbol{H}}_{0,0}(\lambda_0) \end{bmatrix} \tag{53}$$

for a given zeroth-order Tikhonov regularization parameter λ_0. The corresponding regularized solution $\boldsymbol{u}^{\lambda}(\lambda_0 \mid \lambda_0)$ is given by

$$\boldsymbol{u}^{\lambda}(\lambda_0 \mid \lambda_0) \overset{\mathrm{def}}{=} \breve{\boldsymbol{H}}(\lambda \equiv \lambda_0 \mid \lambda_0)\,\overline{\boldsymbol{H}}_0^{\mathrm{T}}\hat{\boldsymbol{y}}^0 \tag{54}$$

which is equivalent to Eq. (29).

It is quite simple to establish a bounded and consecutive time step varying regularization $\lambda_0, \lambda_1, \ldots, \lambda_N$ which is an advantageous property of the explicit block inversion formulation presented in Section 3.2.3, cf also [22]. Starting with λ_1, recompute $N \times N$

block elements corresponding to time steps N to 1 of $\breve{\boldsymbol{H}}$ according to

$$\breve{\boldsymbol{H}}(\lambda \equiv \lambda_1 \mid \lambda_0, \lambda_1) \stackrel{\text{def}}{=} \begin{bmatrix} \breve{\boldsymbol{H}}_{N,N}(\lambda_1) & \dots & \breve{\boldsymbol{H}}_{N,1}(\lambda_1) & \breve{\boldsymbol{H}}_{N,0}(\lambda_0) \\ \vdots & \ddots & \vdots & \vdots \\ \breve{\boldsymbol{H}}_{1,N}(\lambda_1) & \dots & \breve{\boldsymbol{H}}_{1,1}(\lambda_1) & \breve{\boldsymbol{H}}_{1,0}(\lambda_0) \\ \breve{\boldsymbol{H}}_{0,N}(\lambda_0) & \dots & \breve{\boldsymbol{H}}_{0,1}(\lambda_0) & \breve{\boldsymbol{H}}_{0,0}(\lambda_0) \end{bmatrix} \tag{55}$$

for given regularization parameters λ_0 and λ_1. The regularized solution becomes

$$\boldsymbol{u}^\lambda(\lambda_1 \mid \lambda_0, \lambda_1) \stackrel{\text{def}}{=} \breve{\boldsymbol{H}}(\lambda \equiv \lambda_1 \mid \lambda_0, \lambda_1)\, \overline{\boldsymbol{H}}_0^{\mathrm{T}} \hat{\boldsymbol{y}}^0 \tag{56}$$

Proceeding in the same manner, $\breve{\boldsymbol{H}}$ is defined for the ith regularization parameter as

$$\breve{\boldsymbol{H}}(\lambda \equiv \lambda_i \mid \lambda_0, \dots, \lambda_i) \stackrel{\text{def}}{=} \begin{bmatrix} \breve{\boldsymbol{H}}_{N,N}(\lambda_i) & \dots & \breve{\boldsymbol{H}}_{N,i}(\lambda_i) & \dots & \breve{\boldsymbol{H}}_{N,0}(\lambda_0) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \breve{\boldsymbol{H}}_{i,N}(\lambda_i) & \dots & \breve{\boldsymbol{H}}_{i,i}(\lambda_i) & \dots & \breve{\boldsymbol{H}}_{1,0}(\lambda_0) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \breve{\boldsymbol{H}}_{0,N}(\lambda_0) & \dots & \breve{\boldsymbol{H}}_{0,i}(\lambda_0) & \dots & \breve{\boldsymbol{H}}_{0,0}(\lambda_0) \end{bmatrix} \tag{57}$$

for $i = 1, 2, \dots, N$ with corresponding regularized solutions

$$\boldsymbol{u}^\lambda(\lambda_i \mid \lambda_0, \dots, \lambda_i) \stackrel{\text{def}}{=} \breve{\boldsymbol{H}}(\lambda \equiv \lambda_i \mid \lambda_0, \dots, \lambda_i)\, \overline{\boldsymbol{H}}_0^{\mathrm{T}} \hat{\boldsymbol{y}}^0 \tag{58}$$

For optimal choices of the regularization parameters $\lambda_0^{\text{opt}}, \dots, \lambda_N^{\text{opt}}$ the solutions will be bounded according to

$$\begin{aligned} \| \boldsymbol{u}^\lambda(\lambda_0^{\text{opt}} \mid \lambda_0^{\text{opt}}) - \boldsymbol{u}_{\text{dT}}^\lambda \|_2 \; &\geq \| \boldsymbol{u}^\lambda(\lambda_1^{\text{opt}} \mid \lambda_0^{\text{opt}}, \lambda_1^{\text{opt}}) - \boldsymbol{u}_{\text{dT}}^\lambda \|_2 \geq \; \dots \; \geq \\ &\geq \| \boldsymbol{u}^\lambda(\lambda_N^{\text{opt}} \mid \lambda_0^{\text{opt}}, \dots, \lambda_N^{\text{opt}}) - \boldsymbol{u}_{\text{dT}}^\lambda \|_2 = \; \boldsymbol{0} \end{aligned} \tag{59}$$

where $\boldsymbol{u}_{\text{dT}}^\lambda \stackrel{\text{def}}{=} \boldsymbol{u}^\lambda(\lambda_N^{\text{opt}} \mid \lambda_0^{\text{opt}}, \dots, \lambda_N^{\text{opt}})$. It should be noted, that the solution given by the dynamic Tikhonov regularization procedure $\boldsymbol{u}_{\text{dT}}^\lambda$ is not equivalent to

$$\boldsymbol{u}^\lambda(\boldsymbol{\lambda}_{\text{D}}) = (\overline{\boldsymbol{H}}_0^{\mathrm{T}} \overline{\boldsymbol{H}}_0 + \boldsymbol{\lambda}_{\text{D}})^{-1} \overline{\boldsymbol{H}}_0^{\mathrm{T}} \hat{\boldsymbol{y}}^0 \tag{60}$$

where $\boldsymbol{\lambda}_{\text{D}}$ denotes a block diagonal matrix with $\lambda_N \boldsymbol{I}, \dots, \lambda_0 \boldsymbol{I}$ as diagonal blocks. Next, Eq. (59) is verified by a numerical example.

5 NUMERICAL EXAMPLE

A 20 degree of freedom mass chain, depicted in Fig. 1, is used in the following numerical example. The spring stiffness is k=1 N/m and all masses are equal; m_i=1 kg for $i = 1, \dots, 20$. Finally, 0.1% stiffness-proportional viscous damping is added to the model. The resonance frequencies of the system span from $f_{\min} = 0.0122$ Hz to $f_{\max} = 0.317$ Hz. The system is excited by two identical transient forces $\hat{u}_6(t)$ and $\hat{u}_{14}(t)$ applied to masses m_6 and m_{14} respectively. The non-zero part of the excitation is defined according to

$$\hat{u}_6(t) = \hat{u}_{14}(t) = (1 - \cos 2\pi f_0 t) \sin 6\pi f_0 t \qquad 0 < t < \frac{1}{f_0} \tag{61}$$

where $f_0 = 0.06$ Hz. The excitation is shown in Fig. 2. The analysis is carried out for approximately 33 seconds using $N + 1 = 101$ time steps with a sampling frequency f_{samp} of 3 Hz.

Consider the *collocated* case of sensor placement, i.e. all inputs have instant and distinguishable influence on the output, yielding two accelerometers attached to masses m_6 and m_{14}. Fictitious measurement data $\hat{\boldsymbol{y}}^0$ is generated by solving the forward problem given by Eq. (11) and artificially adding noise to the calculated response $\boldsymbol{y}^0$. The initial states $\boldsymbol{x}_0$ are taken as zero and the input $\boldsymbol{u}$ is taken as the discrete-time counterpart of Eq. (61), denoted $\hat{\boldsymbol{u}}$. The measurement data is established for sensor i and time step $k = 0, \dots, N$ according to

$$\hat{y}_{i,k}^0 = y_{i,k}^0 (1 + 0.02 \cdot N(0,1)) + 0.01 \cdot \max(|\boldsymbol{y}_i^0|) \cdot N(0,1) \qquad i = 6, 14 \tag{62}$$

i.e. Gaussian white noise with zero mean and unit variance is added as an overall level of 1% of the maximum absolute value added to 2% relative to each individual value. The input estimation is carried out using an otherwise ideal system description. Moreover, the system is fully observable and controllable, which means that all eigenmodes can be detected and excited by the used sensor/input configuration.

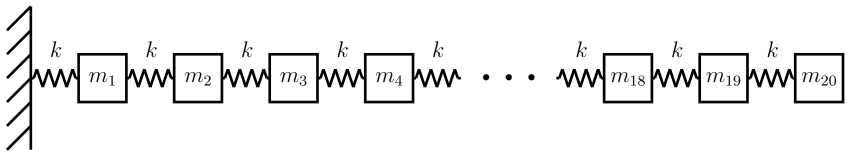

Figure 1: Mass chain with 20 degrees of freedom.

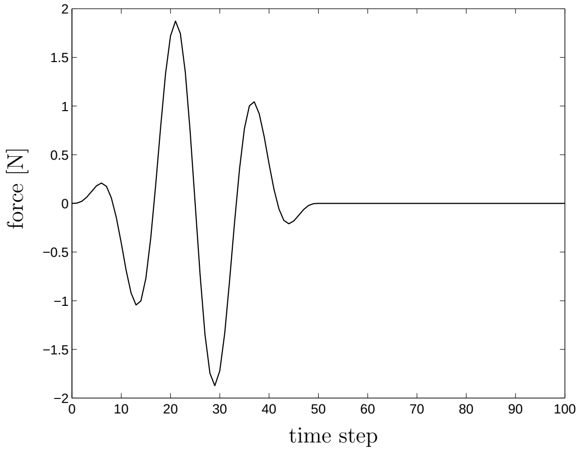

Figure 2: Transient excitation force applied to masses m_6 and m_{14}.

5.1 Classic Tikhonov vs. dynamic Tikhonov

To quantify deviations of the ith optimal regularized solution $\boldsymbol{u}^\lambda(\lambda_i^{\mathrm{opt}} \mid \lambda_0^{\mathrm{opt}}, \dots, \lambda_i^{\mathrm{opt}})$ to the true or sought solution $\hat{\boldsymbol{u}}$, an error measure $\epsilon_\%$ is defined as

$$\epsilon_\%(\lambda_i^{\mathrm{opt}}) \stackrel{\mathrm{def}}{=} 100 \cdot \| \boldsymbol{u}^\lambda(\lambda_i^{\mathrm{opt}} \mid \lambda_0^{\mathrm{opt}}, \dots, \lambda_i^{\mathrm{opt}}) - \hat{\boldsymbol{u}} \|_2 \, / \, \| \hat{\boldsymbol{u}} \|_2 \tag{63}$$

for $i = 0, 1, \dots, N$. The optimal regularized solutions are computed using the true solution and in accordance with the procedure given in Section 4 for a realization of Eq. (62). Typical results for the error measure given by Eq. (63) are plotted in Fig. 3. The corresponding optimal regularization parameters are given in Fig. 4. The circles in the two figures mark the results for optimal zeroth-order Tikhonov regularization. Note that relatively small changes in the first half of the regularization parameters produce a greater decrease of the error measure than the relatively larger changes in the following half of parameters. In general, it is intuitively clear that the first regularization parameters have the greatest influence on the error measure. Moreover, Fig. 3 verifies the bounds given by Eq. (59).

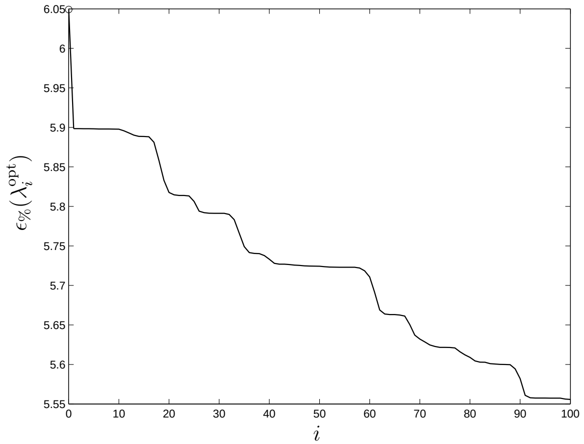

Figure 3: Error for optimal choices of dynamic Tikhonov regularization parameters. The first data point, marked by a circle, corresponds to optimal result of zeroth-order Tikhonov regularization.

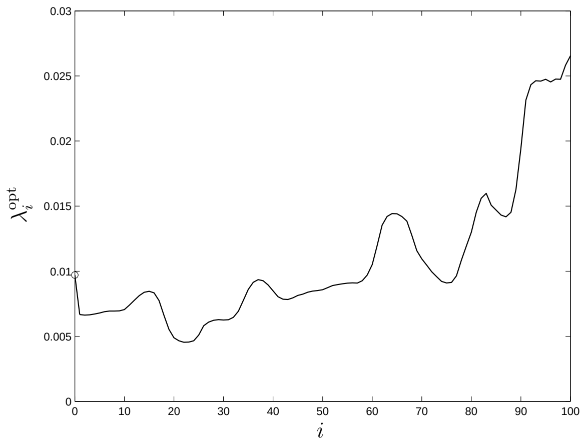

Figure 4: Optimal dynamic Tikhonov regularization parameters. The first data point, marked by a circle, corresponds to the optimal zeroth-order Tikhonov regularization parameter λ_0^{opt}.

6 CONCLUSIONS

Explicit block inversion formulations have been derived to solve damped least squares problems. The dynamic programming algorithm is modified to enable the derivation of the block inverse in state-space matrix form. Using this formulation, the block inverse is identified in Markov parameter form. The two formulations are mathematically equivalent. However, the Markov parameters (impulse response parameters) can be determined directly from physical vibration tests or from an analytical model, which is advantageous.

The inversion algorithms give valuable insight to the exact sequence of operations performed. This knowledge is utilized to introduce a new type of regularization procedure called dynamic Tikhonov regularization. The regularization parameters in dynamic Tikhonov regularization have the attractive feature of being intimately connected to the time steps. For optimal regularization parameters, the zeroth-order Tikhonov solution is an upper bound of the regularized solution which is illustrated by a numerical example.

7 ACKNOWLEDGEMENTS

Financial support was given by Volvo Car Corporation and VINNOVA within its Swedish National Road Vehicle Program. Patrik expresses his gratitude to Prof. Thomas Abrahamsson at the Department of Applied Mechanics, Chalmers University of Technology, and Dr. Mikael Fermér at Volvo Car Corporation for their support and encouragement. Patrik would also like thank Lic. Eng. Lars Nordström at the Department of Applied Mechanics for hours of fruitful discussions.

REFERENCES

[1] T. G. Carne, V. I. Bateman, and R. L. Mayes. Force reconstruction using a sum of weighted accelerations technique. *Proceedings of the 10th International Modal Analysis Conference*, pages 291–298, 1992.

[2] D. M. Trujillo and H. R. Busby. *Practical Inverse Analysis in Engineering*. CRC Press LLC, Boca Raton, Florida, 1997.

[3] J. F. Doyle. A wavelet deconvolution method for impact force identification. *Experimental Mechanics*, 37:403–408, 1997.

[4] G. G. Genaro and D. A. Rade. Input force identification in the time domain. *Proceedings of the 16th International Modal Analysis Conference*, pages 124–129, 1998.

[5] T. H. T. Chan, L. Yu, S. S. Law, and T. H. Yung. Moving force identification studies, I: Theory. *Journal of Sound and Vibration*, 247:59–76, 2001.

[6] A. D. Steltzner, D. C. Kammer, and P. Milenkovic. A time domain method for estimating forces applied to an unrestrained structure. *Journal of Vibration and Acoustics*, 123:524–532, 2001.

[7] L. J. L. Nordström and T. P. Nordberg. A critical comparison of time domain load identification methods. *Proceedings of the Sixth International Conference on Motion and Vibration Control*, 2:1151–1156, 2002.

[8] L. J. L. Nordström and T. P. Nordberg. A time delay method to solve non-collocated input estimation problems. *To Appear in Mechanical Systems and Signal Processing*, 18:1469–1483, 2004.

[9] D. L. Phillips. A technique for the numerical solution of certain equations of the first kind. *J. ACM*, 9:84–97, 1962.

[10] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[11] A. N. Tikhonov and Y. Arsenin. *Solution of Ill-posed Problems*. Wiley, 1977.

[12] P. C. Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Review*, 34:561–580, 1992.

[13] G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press, third edition, 1996.

[14] L. Eldén. An efficient algorithm for the regularization of ill-conditioned least squares problems with triangular Toeplitz matrix. *SIAM J. Sci. Statist. Comput.*, 5:229–236, 1984.

[15] P. C. Hansen. Deconvolution and regularization with Toeplitz matrices. *Numerical Algorithms*, 29:323–378, 2002.

[16] T. Kailath. *Linear Systems*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1980.

[17] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[18] L. J. L. Nordström. *Load Identification Using Time Domain Methods*. Licentiate of Engineering Thesis, Chalmers University of Technology, Göteborg, Sweden, 2003.

[19] Å. Björck and L. Eldén. *Methods in Numerical Algebra for Ill-posed Problems.* Report LiTH–MAT–R–1979–33, Department of Mathematics, Linköping University, Sweden, 1979.

[20] T. P. Nordberg and I. Gustafsson. Using QR factorization and SVD to solve input estimation problems in structural dynamics. *Submitted for International Publication*, 2004.

[21] Å. Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, 1996.

[22] P. K. Lamm and L. Eldén. Numerical solution of first-kind Volterra equations by sequential Tikhonov regularization. *SIAM J. Numer. Anal.*, 34(4):1432–1450, 1997.