



Parameter identification of a linear wave equation from experimental boundary data

Christophe Roman, Francesco Ferrante, Christophe Prieur

► To cite this version:

Christophe Roman, Francesco Ferrante, Christophe Prieur. Parameter identification of a linear wave equation from experimental boundary data. IEEE Transactions on Control Systems Technology, 2021, 29 (5), pp.2166-2179. 10.1109/TCST.2020.3032714 . hal-02985612v1

HAL Id: hal-02985612

<https://hal.science/hal-02985612v1>

Submitted on 13 Nov 2020 (v1), last revised 7 Feb 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parameter identification of a linear wave equation from experimental boundary data

Christophe Roman^{*†}, Francesco Ferrante^{*}, and Christophe Prieur^{*}

Abstract—Parameter identification of a drill-string is studied. The system is modeled as a hyperbolic system with dynamical boundary conditions. The considered model is a wave equation with spatial dependent elasticity and viscous damping terms. The identification problem is recast as an optimization problem over an infinite dimensional space. The developed approach ensures that the estimates of the parameters lie in a given set. A gradient descent-based algorithm is proposed to generate parameters estimates based on experimental data. A thorough comparative study with more classical algorithms is presented.

Index Terms—Parameter identification, Lagrange multiplier, inverse problems, adjoint method, wave equation, infinite dimensional systems.

I. INTRODUCTION

A. Background, motivation, and setup description

FOR many decades the control problem associated with drillstring torsional vibrations has been considered in the control community, see [20]. For more than twenty years, researchers model the drillstring behavior as a wave partial differential equation (PDE), see [24]. In this paper an experimental setup is considered to study these vibrations. The identification problem associated with the model is necessary to fit the parameters with the experimental observation. It is also useful for the diagnosis of drilling system. In order words, one estimates the parameters of the model and therefore one provides a physical description of the drillstring. Note that the wave equation has been also used to model other physical behaviors, as for example overhead crane [4], [6]. In short, we address the parameter identification of a one-dimensional wave equation with space-depending velocity coefficient and distributed viscous terms subject to second order dynamical boundary conditions.

In Figure 1, a schematic view of the experimental setup is given. See also Figure 2, where the actual platform considered in this paper is depicted. In this paper, we consider a configuration in which the bit is not in contact

with the medium, i.e., the so-called bit-off bottom configuration. It prevents boundary frictions that may interfere with the identification of the system dynamics. The platform is composed of two actuators: a motor for actuating the drill itself and a motor ensuring vertical displacement of the bit. Those are denoted by M_2 and M_3 in Figure 1. As shown in Figure 1, the experimental platform is equipped with four sensors. A torque sensor and an angular position sensor located at the bit and denoted by S_1 . A speed sensor for M_2 denoted S_2 . Angular position and speed sensors for M_3 , the latter denoted by S_3 . A force sensor denoted by S_4 , which measures the weight-on-the-bit. The parameters m_1 and m_2 represent the moments of inertia at the two ends of the drill-string, while $a(\cdot)$ represents the distributed elasticity of the drill-string. For the identification problem, the system input is the current U driving the motor M_2 . Electrical dynamics are neglected, that is, we assume that the current driving the motor is proportional to the torque delivered by the motor.

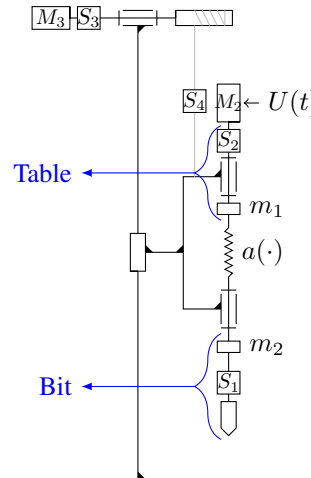


Fig. 1. Schematic view of the platform.

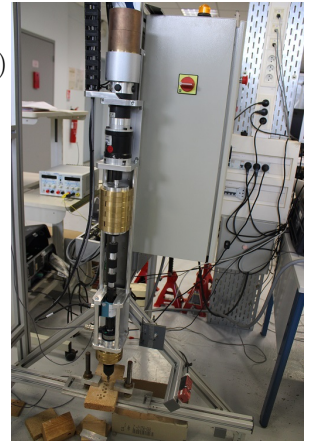


Fig. 2. Picture of the experimental platform.

B. Literature Review on Wave Equation Inverse Problems

In contrast with early lumping methods, where the dynamics is first discretized and then the inverse problem is solved on the associated finite-dimensional system (e.g. [17], [7], [27]), we focus on late lumping methods. This way the discretization for the numerical solver is considered in the latest stage. Note also that we restrict the literature review to offline parameter estimation procedures, as it is the scope here.

^{*} Francesco Ferrante and Christophe Prieur are with Univ. Grenoble Alpes, CNRS, Grenoble-INP, GIPSA-lab, F-38000, Grenoble, France (name.lastname@gipsa-lab.fr).

[†] Christophe Roman is with Aix Marseille Univ, Université de Toulon, CNRS, LIS (UMR 7020), Avenue Escadrille Normandie-Niemen, F-13397 Marseille Cedex 20, France (christophe.roman@lis-lab.fr). During this work he also has been with Institut FEMTO-ST UMR61742, Univ. Bourgogne Franche-Comté, ENSMM, CNRS, F-25000, Besançon, France

Acknowledgment: This work has been supported by ANR-15-CE23-0008 MACS-DRILL, and by MIAI@Grenoble Alpes, (ANR- 19-P3IA-0003)

In [28] the one dimensional wave is considered to have a space depending velocity and subject to Neumann actuation, the space interval considered is semi-infinite, i.e., $[0, +\infty)$. In this setting, the author used the fact that, for the coefficient inverse problem with boundary impulse input, the first order Fréchet derivative of inverse problem with respect to the parameters is invertible. Therefore, Newton's method is used to solve the inverse problem. However, computing this derivative is computationally costly.

In [13], the uniqueness of the solution to a parameter identification inverse problem for the wave equation is proven. This result is difficult to apply to the wave equation under consideration.

In [10], an approach for parameter identification of a wave equation is proposed. In this work, the second order derivative of the distributed state is assumed to be known. This approach shares some features with the one we propose here. However, since for the problem under consideration, the experimental data are extremely noisy, getting a reliable estimate of the second derivative is challenging.

In [26], a method for the estimation of the distributed elastic parameters of the wave equation is suggested. This approach is based on a gradient descent algorithm, but it assumes that the parameters are artificially time-dependent. This is not the case in our work.

A method based on the principle of homotopy for velocity estimation of a two-dimensional wave is presented in [9]. This homotopy method has been proven to be a powerful tool to solve nonlinear inverse problems. However, the applicability of this methodology requires the computation of the Fréchet derivative of the operator associated with the parameters of the inverse problem. This operation is in general computationally costly. This motivates the approach proposed in this paper, for which the computation of that derivative is numerically less costly.

More recently, in [1], an inverse problem for the linear wave equation subject to Dirichlet's and Robin's boundary conditions is considered. Using the method of characteristics, the authors identify both the domain length and the Robin's parameters. Unfortunately the method of characteristic cannot be applied in the setting considered in this paper, mainly because of the dynamical boundary conditions.

C. Contribution

As opposed to the references cited so far, in this paper we focus on the solution to an inverse problem for a wave equation with second order dynamical boundary conditions.

The objective is to find the best parameters which fit real data to the considered wave equation. To solve this problem, we propose the following methodology:

- We start by computing the first variations of the cost function associated with the inverse problem of offline parameter estimation for the wave equation;
- As numerical simulations show that the considered problem is non-convex, we provide a new gradient descent algorithm;

For the procedure to be relevant we show the well-posedness of the considered wave equations and the identifiability of the parameters.

The paper is organized as follows. The wave equation along with the inverse problem considered is presented in Section II. Some mathematical results are stated in Section III. Section IV presents the algorithms proposed in the paper that are instrumental to solve the identification problem. Our approach is applied on experimental results in Section V.

D. Notation

Given $u: [0, 1] \times \mathbb{R}^+ \rightarrow \mathbb{R}$, we denote the partial derivative of u with respect to its first variable values at $(x, t) \in [0, 1] \times \mathbb{R}^+$ by $u_x(x, t)$. Similarly we denote the partial derivative of u with respect to its second variable evaluated at $(x, t) \in [0, 1] \times \mathbb{R}^+$ by $u_t(x, t)$. Let I be a measure space, $L^2(I; \mathbb{R})$ denotes the class of equivalence of square integrable functions from I to $\mathbb{R}$. For the sake of simplicity, $L^2([0, 1]; \mathbb{R})$ is abusively denoted by $L^2(0, 1)$. Furthermore H^1 denotes the Sobolev space $W^{1,2}$, i.e.,

$$u \in H^1 \Leftrightarrow u \in L^2, \quad u' \in L^2, \quad (1)$$

in which u' denotes the weak derivative of u . Two scalar products are used in this paper. The scalar product associated with the parameter space is

$$p, \hat{p} \in L^2(0, 1) \times L^2(0, 1) \times \mathbb{R}^5, \quad (2)$$

$$\begin{aligned} \langle p, \hat{p} \rangle = & \int_0^1 p_1(x) \hat{p}_1(x) dx + \int_0^1 p_2(x) \hat{p}_2(x) dx + p_3 \hat{p}_3 \\ & + p_4 \hat{p}_4 + p_5 \hat{p}_5 + p_6 \hat{p}_6 + p_7 \hat{p}_7. \end{aligned} \quad (3)$$

Its associated norm is $\|p\| = \sqrt{\langle p, p \rangle}$. The other one is dedicated to the state space. It is used to prove the well-posedness of the evolution equation. It is presented in the sequel.

II. PROBLEM STATEMENT

We consider the following wave equation

$$\begin{cases} u_{tt}(x, t) = (a(x)u_x(x, t))_x - q(x)u_t(x, t), & (4a) \end{cases}$$

$$\begin{cases} m_1 u_{tt}(1, t) = \sigma U(t) - a(1)u_x(1, t) - q_1 u_t(1, t), & (4b) \end{cases}$$

$$\begin{cases} m_2 u_{tt}(0, t) = a(0)u_x(0, t) - q_2 u_t(0, t), & (4c) \end{cases}$$

$$\begin{cases} u(\cdot, 0) = 0, \quad u_t(\cdot, 0) = 0, & (4d) \end{cases}$$

in which $a(\cdot)$ stands for the elastic rigidity. In particular, the square root of a is the space-dependent velocity, that is assumed to be strictly positive. The term $q(\cdot)$ is a distributed viscous term. The scalars $m_1, m_2 > 0$ stand for the ratio of masses with respect to constant in-domain linear inertia. The scalars q_1, q_2 are boundary viscous terms. The scalar σ is an input scaling coefficient. The system input is U and is considered to be given.

The vector of measurements is

$$y(t) := \begin{bmatrix} u(1, t) \\ u(0, t) \end{bmatrix}, \quad t \in [0, T], \quad (5)$$

where T stands for the final time of the experiment.

The vector of parameters to be identified is denoted by

$$p := [a(\cdot), q(\cdot), m_1, m_2, q_1, q_2, \sigma]. \quad (6)$$

Let us consider the following operators

$$\Sigma_u : p \mapsto u, \quad (7)$$

and

$$\Sigma_y : u \mapsto y. \quad (8)$$

The domains of the above variables are omitted and their definitions are presented in Section III-B. The operator Σ_u maps the parameters p to the state trajectory u for $t \in [0, T]$. The operator Σ_y maps the state trajectory u to the output trajectory y for $t \in [0, T]$. The parameter T represents the length of the horizon over which data are available.

We are now in a position to formulate the considered parameter identification problem as the following inverse problem:

Problem 2.1: Given y , find p such that

$$(\Sigma_y \circ \Sigma_u)(p) = y. \quad (9)$$

To solve Problem 2.1, we recast it into an optimization problem of a suitable cost function. Such a cost function captures the error between the measurement y and its estimate $\hat{y}$, which is obtained using an estimate of the parameters. More explicitly, the vector of estimated parameter writes:

$$\hat{p} := [\hat{a}(\cdot), \hat{q}(\cdot), \hat{m}_1, \hat{m}_2, \hat{q}_1, \hat{q}_2, \hat{\sigma}]. \quad (10)$$

The estimated output is

$$\hat{y}(t) := \begin{bmatrix} \hat{u}(1, t) \\ \hat{u}(0, t) \end{bmatrix}, \quad (11)$$

and $\hat{u} = \Sigma_u(\hat{p})$. Inspired by [19], let us define the following cost function:

$$J(p_{\mathcal{L}}) = J_1(y, \hat{y}) + J_2(p_{\mathcal{L}}), \quad (12)$$

in which

$$J_1(y, \hat{y}) := \frac{1}{2} \int_0^T [(u(0, t) - \hat{u}(0, t))^2 + (u(1, t) - \hat{u}(1, t))^2] dt, \quad (13)$$

$$J_2(p_{\mathcal{L}}) := \int_0^T \left(\int_0^1 \lambda(x, t) (\hat{u}_{tt}(x, t) - (\hat{a}(x) \hat{u}_x(x, t))_x + \hat{q}(x) \hat{u}_t(x, t)) dx + \lambda_1(t) (\hat{m}_1 \hat{u}_{tt}(1, t) + \hat{a}(1) \hat{u}_x(1, t) + \hat{q}_1 \hat{u}_t(1, t) - \hat{\sigma} U(t)) + \lambda_2(t) (\hat{m}_2 \hat{u}_{tt}(0, t) - \hat{a}(0) \hat{u}_x(0, t) + \hat{q}_2 \hat{u}_t(0, t)) \right) dt, \quad (14)$$

The terms λ , λ_1 , and λ_2 are the Lagrange multipliers corresponding to the constraint $\hat{u} = \Sigma_u(\hat{p})$. The vector $p_{\mathcal{L}}$ is defined as

$$p_{\mathcal{L}} = [y, \hat{u}, \hat{p}, \lambda, \lambda_1, \lambda_2]. \quad (15)$$

The term J_1 corresponds to the error between the estimated output and the actual measurements. The term J_2 is for the constraint $\hat{u} = \Sigma_u(\hat{p})$.

The identification problem we consider can be turned into the following optimization problem.

Problem 2.2: Given y , find $\hat{p}$ solution to the following optimization problem

$$\min_{\hat{p}} J(p_{\mathcal{L}}). \quad (16)$$

The following property is direct

$$p = \hat{p} \Rightarrow J(p_{\mathcal{L}}) = 0, \quad (17)$$

whereas the next one

$$J(p_{\mathcal{L}}) = 0 \Rightarrow p = \hat{p}, \quad (18)$$

is related to the uniqueness of solution to the inverse problem (9), which is equivalent to the identifiability of (4). More information about inverse problems for partial differential equations can be found in [11]. See also [16], [8], for recent contributions on parameter identification of infinite dimensional systems. Next, we present our approach to solve (16).

A. Outline of the proposed methodology

With the objective of solving (16), we determine the derivative of J with respect to the system parameters to be estimated. Specifically, we determine the first variation of J defined as:

$$\delta J(p_{\mathcal{L}}, \delta p_{\mathcal{L}}) = J(p_{\mathcal{L}} + \delta p_{\mathcal{L}}) - J(p_{\mathcal{L}}) + O(\|\delta p_{\mathcal{L}}\|^2). \quad (19)$$

We can interpret $\delta p_{\mathcal{L}}$ as a test function. In our case, and under suitable conditions that are given in the sequel, the first variation is linear with respect to the estimated parameters' variation. Then, by relying on (19), the search of local minima of (12) can be performed by employing a gradient decent algorithm. To achieve this goal, there are two main difficulties to be tackled. First, one should make sure that the algorithm does not get stuck in a local minimum. This is achieved by one the descent algorithms we propose in Section IV-C. Second, the algorithm needs to ensure the nonnegativity of parameters estimates; this is done augmenting the cost function by a Tikhonov's term.

III. MAIN RESULTS

This section illustrates how the gradient of the cost functional defined in (12) can be determined analytically and assesses the well-posedness of the considered wave equation. Parameters identifiability issues are discussed at the end of this section.

A. Computation of the Gradient of the Cost Functional

The following result provides, under suitable conditions, an expression of the derivative of the cost functional.

Theorem 3.1: Assume that:

- (i) $\hat{u} = \Sigma_u(\hat{p})$,
- (ii) $\lambda = \Sigma_\lambda(\hat{p}, \hat{u})$,
- (iii) $\lambda_1(t) = \lambda(1, t)$,
- (iv) and $\lambda_2(t) = \lambda(0, t)$,

where $\lambda = \Sigma_\lambda(\hat{p}, \hat{u})$ is the solution to the following system

$$\begin{cases} \lambda_{tt}(x, t) - (\lambda_x(x, t)\hat{u}_x(x, t))_x - \lambda_t(x, t)\hat{q}(x) = 0, & (20a) \\ \hat{m}_1\lambda_{tt}(1, t) = -\hat{a}(1)\lambda_x(1, t) + \hat{q}_1\lambda_t(1, t) \\ \quad + (u(1, t) - \hat{u}(1, t)), & (20b) \\ \hat{m}_2\lambda_{tt}(0, t) = \hat{a}(0)\lambda_x(0, t) + \hat{q}_2\lambda_t(0, t) \\ \quad + (u(0, t) - \hat{u}(0, t)), & (20c) \\ \lambda(\cdot, T) = \lambda_t(\cdot, T) = 0. & (20d) \end{cases}$$

Then, the first variation of $J(p_{\mathcal{L}})$ defined in (12) is given by

$$\delta J(p_{\mathcal{L}}, \delta p_{\mathcal{L}}) = \langle dJ_{1-2}, \delta \hat{p} \rangle, \quad (21)$$

in which

$$dJ_{1-2}(x) := \begin{bmatrix} \int_0^T \lambda_x(x, t)\hat{u}_x(x, t)dt \\ \int_0^T \lambda(x, t)\hat{u}_t(x, t)dt \\ - \int_0^T \lambda_t(1, t)\hat{u}_t(1, t)dt \\ - \int_0^T \lambda_t(0, t)\hat{u}_t(0, t)dt \\ \int_0^T \lambda(1, t)\hat{u}_t(1, t)dt \\ \int_0^T \lambda(0, t)\hat{u}_t(0, t)dt \\ - \int_0^T \lambda(1, t)U(t)dt \end{bmatrix}, \quad (22)$$

is the derivative of the cost functional and belongs to the parameter space defined in (2). $\square$

Remark 1: Notice that the adjoint system (20) represents a wave equation backward in time. The proof of Theorem 3.1 is given in Appendix A. $\circ$

Once the derivative of the cost function is available, one can use gradient descent technique to find a minimum of the cost functional. If the cost functional is convex this minimum is unique. In general, parameters identification from noisy data gives rise to a ill-posed problems. A common technique used to solve ill-posed problems is the so-called Tikhonov regulation. This main idea underlying this technique consists of introducing some additional constraints to achieve the convergence of the optimization algorithm. In our setting, when using experimental data, the use of a gradient descent algorithm may give rise to negative values for the estimates of the parameter. This is not consistent with the physical description of the system (see [22, Chapter 9] for more details). To overcome this drawback, we add constraints on the minimization problem to penalize the cost function when parameter estimates get close to zero. There exist several methods to perform constrained optimization (see e.g. [3] and references within). In this paper, to limit the resulting

computational complexity, we consider the following additional penalty term when parameters get close to zero:

Proposition 3.1: Consider

$$\begin{aligned} J_3(\hat{p}) := & c_a \int_0^1 e^{\frac{1}{\hat{a}(x)}} dx + c_q \int_0^1 e^{\frac{1}{\hat{q}(x)}} dx \\ & + c_{m_1} e^{\frac{1}{\hat{m}_1}} + c_{m_2} e^{\frac{1}{\hat{m}_2}} + c_{q_1} e^{\frac{1}{\hat{q}_1}} \\ & + c_{q_2} e^{\frac{1}{\hat{q}_2}} + c_\sigma e^{\frac{1}{\hat{\sigma}}}. \end{aligned} \quad (23)$$

The first variation of J_3 is

$$\delta J_3(\hat{p}) = \langle dJ_3, \delta \hat{p} \rangle, \quad (24)$$

in which

$$dJ_3(x) := \begin{bmatrix} c_a \left(-\frac{1}{\hat{a}(x)^2}\right) e^{\frac{1}{\hat{a}(x)}} \\ c_q \left(-\frac{1}{\hat{q}(x)^2}\right) e^{\frac{1}{\hat{q}(x)}} \\ c_{m_1} \left(-\frac{1}{\hat{m}_1^2}\right) e^{\frac{1}{\hat{m}_1}} \\ c_{m_2} \left(-\frac{1}{\hat{m}_2^2}\right) e^{\frac{1}{\hat{m}_2}} \\ c_{q_1} \left(-\frac{1}{\hat{q}_1^2}\right) e^{\frac{1}{\hat{q}_1}} \\ c_{q_2} \left(-\frac{1}{\hat{q}_2^2}\right) e^{\frac{1}{\hat{q}_2}} \\ c_\sigma \left(-\frac{1}{\hat{\sigma}^2}\right) e^{\frac{1}{\hat{\sigma}}} \end{bmatrix}. \quad (25)$$

$\square$

The proof of this result follows from a direct computation of the first variation of J_3 .

By combining Theorem 3.1 and Proposition 3.1, one obtain the following result.

Corollary 3.1: Let

$$J_r(p_{\mathcal{L}}) := J_1(y, \hat{y}) + J_2(p_{\mathcal{L}}) + J_3(\hat{p}), \quad (26)$$

in which J_1 , J_2 , and J_3 are defined in (13), (14), and (23). Assume that items (i)-(iv) in Theorem 3.1 hold. Then the first variation of J_r is equal to

$$\delta J_r(p_{\mathcal{L}}) = \langle dJ_{1-2} + dJ_3, \delta \hat{p} \rangle. \quad (27)$$

$\square$

B. Well-posedness of considered wave equations

In this section, we state that the systems (4) and (20) are well-posed. Therefore, the formal computation of the cost J defined in (12) is meaningful.

Let us consider the following wave equation

$$\begin{cases} v_{tt}(x, t) = (a(x)v_x(x, t))_x - q(x)v_t(x, t), & (28a) \\ m_1 v_{tt}(1, t) = U_1(t) - a(1)v_x(1, t) - q_1 v_t(1, t), & (28b) \\ m_2 v_{tt}(0, t) = a(0)v_x(0, t) - q_2 v_t(0, t) + U_2(t), & (28c) \\ v(\cdot, 0) = v_0, \quad v(\cdot, 0) = v_{t0}, & (28d) \end{cases}$$

and observe that, up to a change of variable, both systems (4), and (20) can be rewritten as (28). More specifically, (4) corresponds to (28) with $U_1(t) = U(t)$ and $U_2(t) = 0$, while (20) corresponds to (28) with $U_1(t) = u(1, T-t) - \hat{u}(1, T-t)$

$t)$ and $U_2(t) = u(1, T - t) - \hat{u}(1, T - t)$. Let us consider the following space

$$\mathbf{H} := H^1(0, 1) \times L^2(0, 1) \times \mathbb{R}^2. \quad (29)$$

It is straightforward to check that $\mathbf{H}$ is an Hilbert space for the following scalar product, $\forall z, g \in \mathbf{H}$,

$$\langle z, g \rangle_{\mathbf{H}} := \int_0^1 [z_1(x)g_1(x) + a(x)z'_1(x)g'_1(x) + z_2(x)g_2(x)]dx + m_1 z_3 g_3 + m_2 z_4 g_4. \quad (30)$$

We recall that $a(\cdot)$, m_1 , and m_2 are strictly positive. Let us define the following unbounded operator,

$$\forall z := \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \in \mathcal{D}(\mathcal{A}), \quad \mathcal{A}z := \begin{bmatrix} -z_2 \\ -(az'_1)' + qz_2 \\ \frac{a(1)z'_1(1) + q_1 z_3}{m_1} \\ \frac{q_2 z_4 - a(0)z'_1(0)}{m_2} \end{bmatrix}, \quad (31)$$

where

$$\mathcal{D}(\mathcal{A}) := \left\{ z \in H^2(0, 1) \times H^1(0, 1) \times \mathbb{R}^2; \right. \\ \left. z_3 = z_2(1), \quad z_4 = z_2(0) \right\}. \quad (32)$$

and consider the following abstract Cauchy problem:

$$\begin{cases} \dot{z}(t) + \mathcal{A}z(t) = \mathcal{U}(t) \\ z(0) = z_0, \end{cases} \quad (33a)$$

$$(33b)$$

where

$$\mathcal{U}(t) := \begin{bmatrix} 0 & 0 & U_1(t) & U_2(t) \end{bmatrix}^T. \quad (34)$$

The abstract Cauchy problem (33) is an alternative representation of (28). In particular, the regularity of z and $v(\cdot, t)$ is illustrated in the result given next.

Theorem 3.2: Consider $\mathbf{H}$, $\mathcal{A}$, and $\mathcal{D}(\mathcal{A})$ defined respectively in (29), (31), and (32) with $m_1, m_2 \in (0, \infty)$, $q_1, q_2 \in \mathbb{R}$. The following items hold:

(a) Consider

- (i) $U_1, U_2 \in C^1([0, \infty); \mathbb{R})$,
- (ii) $a \in H^1([0, 1]; (0, \infty))$,
- (iii) $q \in L^2([0, 1]; \mathbb{R})$.

For all initial data $z_0 \in \mathcal{D}(\mathcal{A})$, there exists a unique solution to the abstract problem (33), $z(t) \in \mathcal{D}(\mathcal{A})$, $\forall t \geq 0$. Moreover, denoting $v = z_1$, one has:

$$\begin{aligned} v &\in C^0([0, \infty); H^2(0, 1)) \cap C^1([0, \infty); H^1(0, 1)), \\ v(1, \cdot), v(0, \cdot) &\in C^2([0, \infty); \mathbb{R}). \end{aligned}$$

(b) Consider

- (i) $U_1, U_2 \in L^2([0, \infty); \mathbb{R})$,
- (ii) $a \in L^2([0, 1]; (0, \infty))$,
- (iii) $q \in L^2([0, 1]; \mathbb{R})$.

For all initial data $\mathcal{X}_0 \in \mathbf{H}$, there exists a mild solution $z(t) \in \mathbf{H}$ to the abstract problem (33) given by

$$z(t) = S(t)z_0, \quad (35)$$

in which S is the C_0 -semigroup generated by the unbounded operator $\mathcal{A}$. Moreover, it holds

$$v \in C^0([0, \infty); H^1(0, 1)) \cap C^1([0, \infty); L^2(0, 1)).$$

□

C. Identifiability of the parameters

This section deals with the identifiability of the parameters from boundary measurements. To establish this property, we rely on the uniqueness of the solutions to the considered inverse problem. As a first step towards this goal, next we show that parameters are identifiable from state measurements.

Theorem 3.3: System (33) is state identifiable from non-zero initial conditions. More precisely, if $u - \hat{u} = 0$, $(u_0, u_1) \neq (0, 0)$, then $\hat{p} = p$. Moreover, (33) is state identifiable for zero initial conditions under excitation. Namely, if $u - \hat{u} = 0$, $(u_0, u_1) = (0, 0)$, $\sigma \neq 0$, $\exists t_0 \in [0, T]$, $U(t_0) \neq 0$, then $\hat{p} = p$. □

The proof of Theorem 3.3 is given in Appendix C.

Definition 3.1: Let $z_0 \in \mathbf{H}$ and $y_0(\cdot, z_0)$ be the output of (33) from the initial condition z_0 with zero input. System (33) is said to be observable if $z_0 \neq \tilde{z}_0$ implies that there exists $t_1 \in [0, \infty)$ such that $y_0(t, z_0) \neq y_0(t, \tilde{z}_0)$.

The result given next provides some insights on the observability of (33).

Theorem 3.4: System (33) is observable from the output y defined in (5). □

The proof of Theorem 3.4 is given in Appendix D.

Building upon the above results, the following corollary can be established.

Corollary 3.2: Under the assumptions of Theorems 3.3 and 3.4, the parameters of the wave equation are identifiable from the position measurement at any boundary. More precisely

$$J(p_{\mathcal{L}}) = 0 \Rightarrow \hat{p} = p. \quad (36)$$

□

Proof : The proof is a direct application of Theorems 3.3 and 3.4. In particular, by building upon the following obvious implication

$$J = 0 \Rightarrow y - \hat{y} = 0 \quad (37)$$

from Theorem 3.4 one gets

$$y - \hat{y} = 0 \Rightarrow u = \hat{u}. \quad (38)$$

Finally, from Theorem 3.3, it holds

$$u = \hat{u} \Rightarrow p = \hat{p}. \quad (39)$$

This concludes the proof. ■

IV. GRADIENT DESCENT ALGORITHMS

In this part, we present three algorithms that can be used to carry out the proposed parameter identification procedure. These algorithms are based on gradient descent methods; see [23] for an introduction on these methods. The first two are not used on the experimental data but only on the numerical simulations; see Section IV-D for more details on that.

By discretizing (21), one gets

$$J(p_{\mathcal{L}}[k+1]) - J(p_{\mathcal{L}}[k]) = \langle dJ, \Delta\hat{p}[k] \rangle. \quad (40)$$

Thus, the update of the parameter estimation follows

$$\hat{p}[k+1] = \Delta\hat{p}[k] + \hat{p}[k]. \quad (41)$$

All optimization problems can be reduced to the computation of $\Delta\hat{p}[k]$ such that $J(p_{\mathcal{L}}[k+1]) - J(p_{\mathcal{L}}[k]) < 0$. Several sophisticated optimization algorithms exist to achieve this goal. In particular, accelerated/momentum algorithm, e.g. Heavy ball method and Nesterov acceleration (see, e.g., [5]). In this paper, first we present the common gradient descent algorithm. Then, an algorithm referred to as *Adagrad* is illustrated. Finally, we suggest a new descent strategy, this third algorithm is referred to as *Geograd*. A comparative study of the behavior of these three algorithms is given in Section IV-D through numerical simulations for the considered wave equation.

A. Batch gradient descent [23]

For the batch gradient descent, the update of the parameters is

$$\Delta\hat{p}[k] = -\alpha dJ. \quad (42)$$

in which α is a positive constant. Note that in this case (40) becomes

$$J(p_{\mathcal{L}}[k+1]) - J(p_{\mathcal{L}}[k]) = -\alpha \|dJ\|^2. \quad (43)$$

B. Adagrad algorithm [23]

The idea behind the Adagrad algorithm is to adapt α to each parameter. The update law is

$$\Delta\hat{p}[k] = -\frac{\alpha}{\sqrt{G[k] + \varepsilon}} dJ, \quad (44)$$

in which

$$\begin{cases} G[k+1] = \mu G[k] + (1-\mu) \text{diag}(dJ)^2, \\ G[0] = 0. \end{cases} \quad (45a) \quad (45b)$$

In (44), ε is a term to avoid the division by zero (equal to 10^{-8} in this paper), $\text{diag}(dJ)$ is the diagonal matrix representation of the vector dJ , μ is taken equal to 1 for the Adagrad algorithm, otherwise the algorithm with $\mu \neq 1$ is actually referred to as RMSprop (see [23]).

C. The suggested heuristic: Geograd

The idea of the algorithm we suggested is to impose the decay rate of the cost function to decrease geometrically. Namely

$$J(p_{\mathcal{L}}[k+1]) = \kappa J(p_{\mathcal{L}}[k]), \quad (46)$$

in which $\kappa \in (0, 1)$. This is associated with the following update law

$$\Delta_g \hat{p}[k] = -\frac{(1-\kappa)J(p_{\mathcal{L}}[k])dJ}{\|dJ\|^2}. \quad (47)$$

Similarly as to the Adagrad algorithm presented in Section IV-B, one can avoid any division by zero by adding a positive term in the denominator of (47). We suggest an alternative, which is to bound $\Delta_g \hat{p}[k]$ using the following function

$$N(x) = \frac{x}{1 + \varepsilon \|x\|_{\infty}}. \quad (48)$$

For $x \in \mathbb{R}$ it holds $N(x) \in [-\frac{1}{\varepsilon}, \frac{1}{\varepsilon}]$. Defining $\Delta\hat{p}[k] = N(\Delta_g \hat{p}[k])$ yields to

$$\Delta\hat{p}[k] = -\frac{(1-\kappa)J(p_{\mathcal{L}}[k])dJ}{\|dJ\|^2 + \varepsilon(1-\kappa)J\|dJ\|_{\infty}}. \quad (49)$$

What is interesting about this update law is the fact that ε defines the interval where the step belongs, and κ is the decay rate of the cost function we desire. When dJ goes to zero and J does not, the algorithm can leave the minimum (or maximum). Moreover, if the global minimum is not zero, the algorithm jump endlessly.

Remark 2: When considering multiple parameters, it is important to tune the different values of ε for each parameter. This can be done by taking ε to be a vector, with the only caveat of interpreting the division in (49) in an element-wise fashion. $\circ$

Next, we present the classical descent algorithm based on the adjoint system. This is presented in Algorithm 1. Note however that Algorithm 1 might not converge when using Geograd algorithm. This is typically the case when one has no a priori knowledge of the cost function (e.g. it can be nonconvex, with nonconvex local minima,...). To overcome this problem, we provide an alternative algorithm, i.e., Algorithm 2 that is structurally stable. In other words, the algorithm imposes that $J[\hat{p}[i]] \leq J[\hat{p}[0]]$ and that the set of global minima is Lyapunov stable. The idea is simple: at each iteration we test if $J[\hat{p}[i]] \leq J[\hat{p}[0]]$ if not, we rescale the step parameter $\Delta\hat{p}$ by dividing it by two and recompute $J[i]$. This structurally imposes Lyapunov stability with J as Lyapunov functional. The main drawback of having imposed structural stability is that at least a test at each iteration is required. This leads to a computationally greedier algorithm.

D. Parameter estimations on simulated data

This section illustrates the estimation of the parameter $a(\cdot)$ on simulated data, when all the remaining parameters are known. This allows us to compare the proposed approach

Algorithm 1: Adjoint method for finding a minimum

Input : A set of measured data Y and the associate input U

Output: An optimized value of the parameters $\hat{p}[b]$

- 1 We define two functions corresponding to solving the forward and backward systems;
- 2 **Function** Σ_u **is**
- 3 $\hat{u} = \Sigma_u(\hat{p}, U);$ */* Forward system */*
- 4 return $\hat{u};$
- 5 **Function** Σ_λ **is**
- 6 $\lambda = \Sigma_\lambda(\hat{p}, Y, \hat{u});$ */* Backward system */*
- 7 return $\lambda;$
- 8 We set an initial guess $\hat{p}[0]$ and a fixed number of iteration $n > 1$;
- 9 **for** $i \leftarrow 0$ **to** $n - 1$ **do**
- 10 $\hat{u}[i] = \Sigma_u(\hat{p}[i], U);$
- 11 We compute the value of the cost $J[i];$
- 12 $\lambda[i] = \Sigma_\lambda(\hat{p}[i], Y, \hat{u}[i]);$
- 13 According to the chosen strategy (batch gradient, Adagrad, Geograd,...)
- 14 We compute the value of the step $\Delta\hat{p}[i];$
- 15 $p[i + 1] = \Delta\hat{p}[i] + p[i];$ */* We update. */*
- 16 We get the argmin of the cost $b = \operatorname{argmin}(J);$
- 17 return $\hat{p}[b];$

with respect to existing other approaches focusing on the estimation of $a(\cdot)$, e.g. [26] and [18]. Numerical simulations are carried out via a simple finite difference scheme in space and a classic ODE solver for the time.

Remark 3: Notice that the strength of proposed methodology is that the computation of the derivative of the cost function does not depend on the numerical scheme. On the other hand, the use of more sophisticated numerical schemes might lead to better results. $\circ$

The input is taken as

$$U(t) = e^{-400(t-0.1)^2}. \quad (50)$$

In this numerical simulation, we estimate $a(\cdot)$, which is selected as

$$a(x) = 1 + e^{-25(x-0.5)^2}. \quad (51)$$

The number of spatial points for the estimation and adjoint system is $N = 100$. The number of spatial points, for the estimated parameter a , is $N_0 = 600$. The time horizon is $T = 3$, the number of time points for all systems is $M = 3000$. We consider the case where only measurements at $x = 1$ are available, yielding the adjoint system (20) with the input $u(1, t) - \hat{u}(1, t)$. The other parameters are known and equal to $q(\cdot) = q_1 = q_2 = 0$, $m_1 = m_2 = \sigma = 1$.

Algorithm 2: Adjoint method for finding a minimum with enforced stability

Input : A set of measured data Y and the associate input U

Output: An optimized value of the parameters $\hat{p}[b]$

- 1 We set an initial guess $\hat{p}[0]$, a fixed number of iteration $n > 1$ and a fixed number $M > 1$ of loops to enforce the stability;
- 2 $\hat{u}[0] = \Sigma_u(\hat{p}[0], U);$
- 3 We compute the value of the cost $J[0];$
- 4 $\lambda[0] = \Sigma_\lambda(\hat{p}[0], Y, \hat{u}[0]);$
- 5 We compute the value of the step $\Delta\hat{p}[0];$
- 6 **for** $i \leftarrow 1$ **to** $n - 1$ **do**
- 7 $p[i] = \Delta\hat{p}[i - 1] + p[i - 1];$
- 8 $\hat{u}[i] = \Sigma_u(\hat{p}[i], U);$
- 9 We compute the value of the cost $J[i];$
- 10 $m = 1;$
- 11 **while** $J[i] > J[0] \ \& \ m < M$ **do**
- 12 $\Delta\hat{p}[i - 1] = \Delta\hat{p}[i - 1]/2;$
- 13 $p[i] = \Delta\hat{p}[i - 1] + p[i - 1];$
- 14 $\hat{u}[i] = \Sigma_u(\hat{p}[i], U);$
- 15 We compute the new value of the cost $J[i];$
- 16 $m = m + 1;$
- 17 **if** $m \leq M$ **then**
- 18 break; */* Security break */*
- 19 $\lambda[i] = \Sigma_\lambda(\hat{p}[i], Y, \hat{u}[i]);$
- 20 According to the chosen strategy (batch gradient, Adagrad, Geograd,...)
- 21 We compute the value of the step $\Delta\hat{p}[i];$
- 22 We get the argmin of the cost $b = \operatorname{argmin}(J);$
- 23 return $\hat{p}[b];$

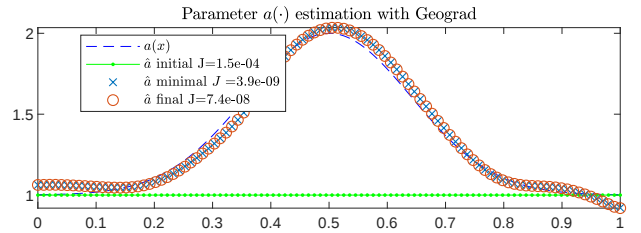


Fig. 3. Estimation of parameter $a(\cdot)$ using the algorithm (49) for 200 iterations.

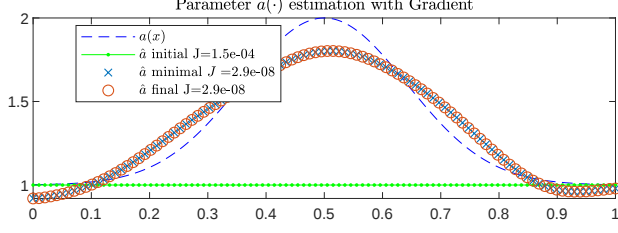


Fig. 4. Estimation of parameter $a(\cdot)$ using the algorithm (42) for 200 iterations.

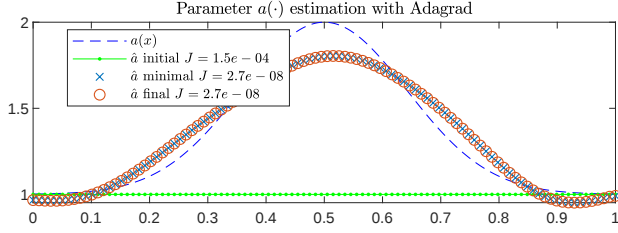


Fig. 5. Estimation of parameter $a(\cdot)$ using the algorithm (44) for 200 iteration.

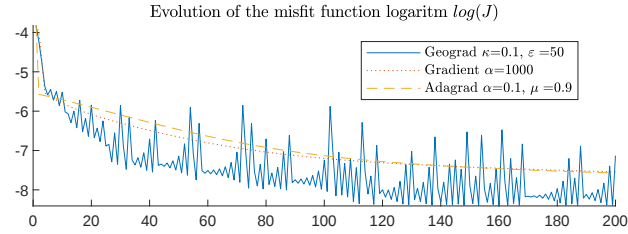


Fig. 6. Evolution of the cost function for Figures 3, 4, and 5 in function of iterations.

In Figure 6 the evolution of the cost function logarithm is displayed with respect to the iteration. Considering the minimum value, Geograd algorithm is outperforming both other algorithms see Figure 3. With this set of experiment we can conclude that the optimization problem considered in this part is nonconvex. Indeed let us compare Figure 3 to Figure 4 or Figure 5 final estimation. Adagrad and Gradient have a smaller misfit value than Geograd algorithm, but the estimation of $a(\cdot)$ is better for Geograd algorithm even for its last value.

Comparing our result with respect to [26] and [18], we obtain a closer result for the estimation of $a(\cdot)$ with in less iteration.

V. EXPERIMENTAL RESULTS

In this section we illustrate the application of the proposed methodology on the experimental platform. The experimental data are obtained from a step response of 1.7 magnitude.

It corresponds to the largest input that experimental setup can tolerate. The measured output is the position at both ends, in other words, the table position and the bit position, see Figure 1. The position of the bit is directly measured by an incremental encoder, while the position of the table is obtained by direct integration of speed measurements provided by a tachometer. In the experiments, measurements sampling rate is set to 1 ms.

Specifically, we execute the proposed estimation algorithm on data obtained for a step response with amplitude 1.7 Ampere between $t \in [0, 5]$ s. The procedure is applied for two initial guesses of the parameters. The values of the initial guesses are given respectively in Table I and Table II. We consider the cost function in Corollary 3.1. The regulation parameter values (penalty terms) are given in Table III. The Geograd tuning parameters are given in Table III. The number of spatial points used for the numerical scheme is $N = 60$. The time evolution of the input U is given in Figure 7.

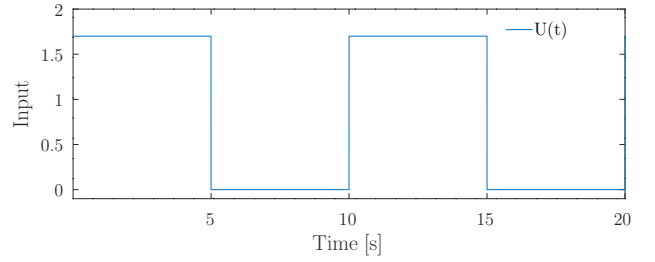


Fig. 7. Input applied to the system.

The velocity responses of the actual system and of their estimates are depicted in Figures 10 and 11 for the first setup of initial parameters, and in Figures 17 and 18 for the second setup of initial parameters. The associated position responses are given in Figures 13, 14, 20, and 21. The parameter estimation associated with these responses are given in Figures 8, 9, 15, and 16, and also in Tables I and xII. For each initial guess, the proposed estimation algorithm has been applied using data associated with two intervals of time $[0, T]$, one with $T = 5$ and one for $T = 10$. The time evolution of the cost function for different iterations of the algorithms is given in Figures 14 and 21. For all these simulations, Algorithm 2 has been used.

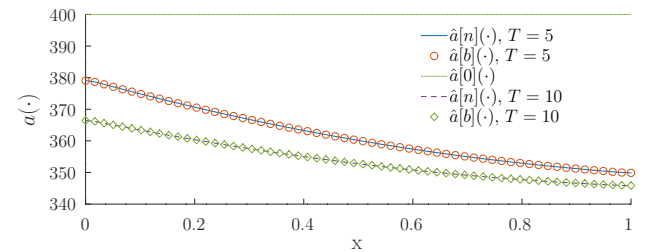


Fig. 8. Estimation of parameter $a(\cdot)$, for initial guess 1.

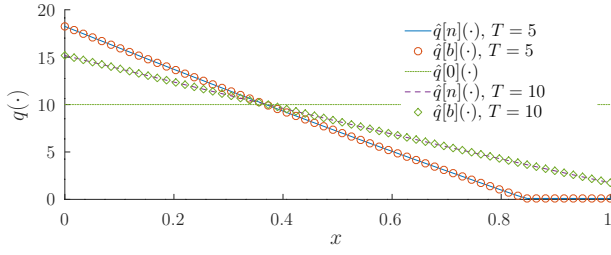


Fig. 9. Estimation of parameter $q(\cdot)$, for initial guess 1.

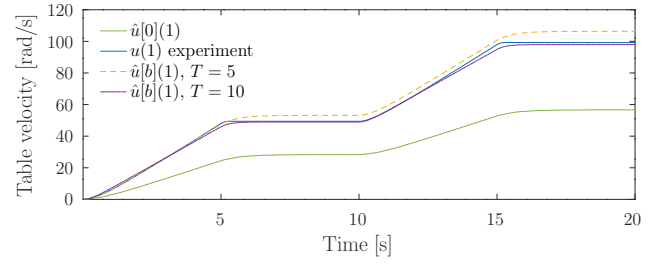


Fig. 13. Real data on the position of the table and identified system responses, for initial guess 1.

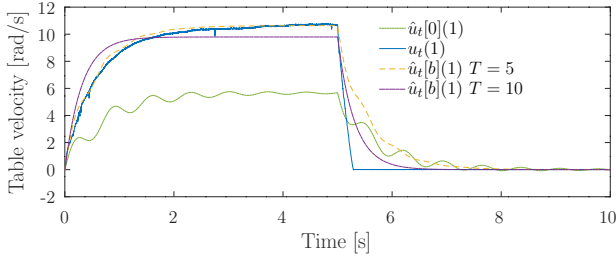


Fig. 10. Real data on the velocity of the table and identified system responses, for initial guess 1.

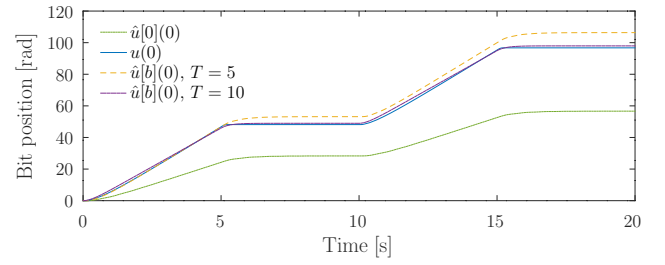


Fig. 14. Bit position real data and identified system responses, for initial guess 1.

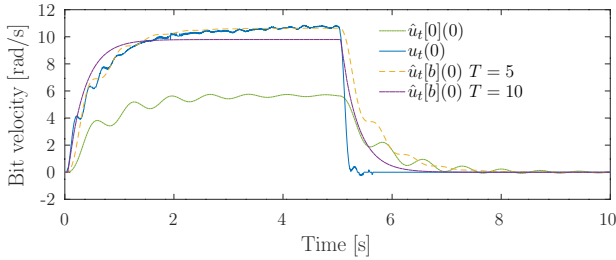


Fig. 11. Bit velocity real data and identified system responses, for initial guess 1.

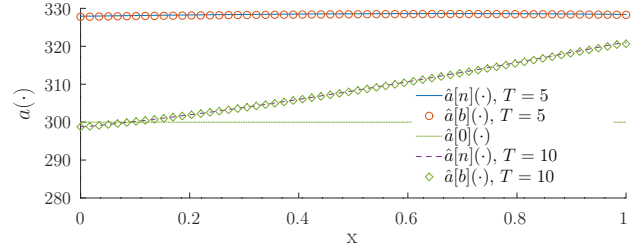


Fig. 15. Estimation of parameter $a(\cdot)$, for initial guess 2.

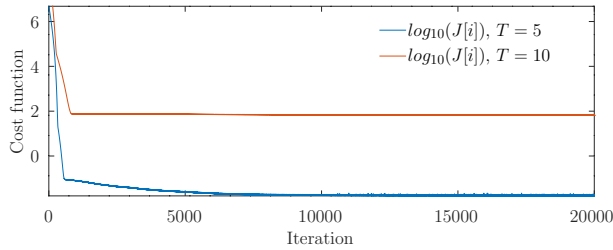


Fig. 12. Evolution of the cost function versus the procedure iteration, for initial guess 1.

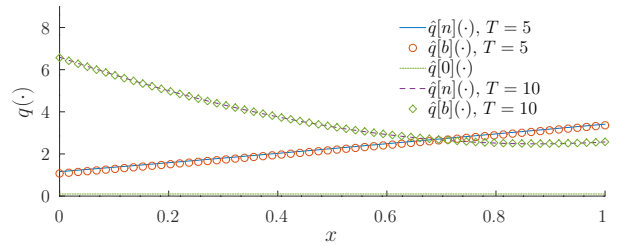


Fig. 16. Estimation of parameter $q(\cdot)$, for initial guess 2.

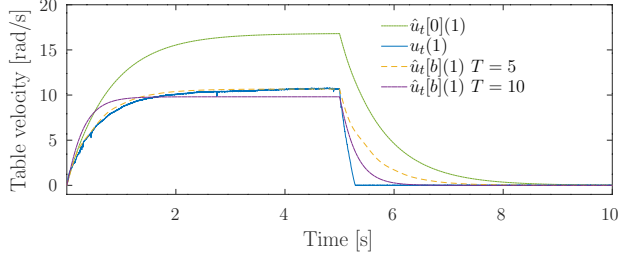


Fig. 17. Real data on the velocity of the table and identified system responses, for initial guess 2.

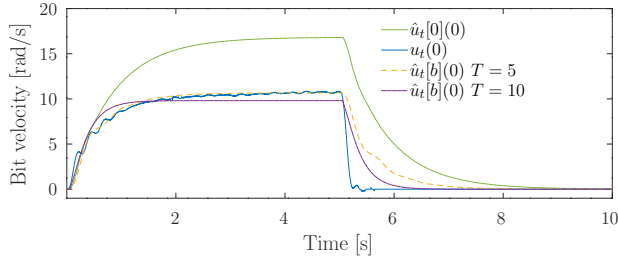


Fig. 18. Bit velocity real data and identified system responses, for initial guess 2.

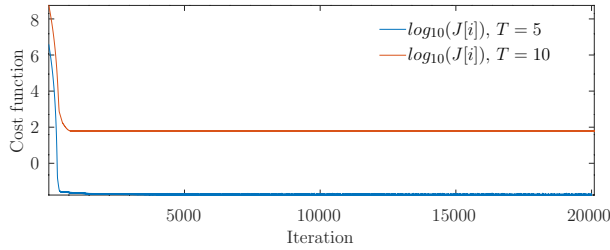


Fig. 19. Evolution of the cost function versus the procedure iteration, for initial guess 2.

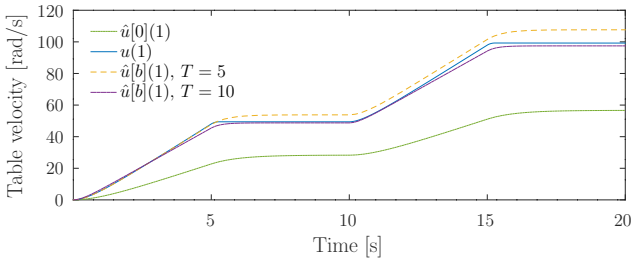


Fig. 20. Real data on the position of the table and identified system responses, for initial guess 2.

Initial guess 1		$t \in [0, 5]$		$t \in [0, 10]$	
	[0]	[b]	[n]	[b]	[n]
$\hat{a}[i](\cdot)$	400	Fig 8		Fig 8	
$\hat{q}[i](\cdot)$	10	Fig 9		Fig 9	
$\hat{m}_1[i]$	10	10.6	10.6	8.0	8.0
$\hat{m}_2[i]$	10	3.9	3.9	0.1	0.1
$\hat{q}_1[i]$	10	0.1	0.1	1.7	1.7
$\hat{q}_2[i]$	10	18.2	18.2	15.2	15.2
$\hat{\sigma}[i]$	100	162.2	162.2	145.4	145.4

TABLE I
PARAMETER ESTIMATION VALUES FOR DIFFERENT ITERATION, INITIAL [0], MINIMAL COST [b], AND FINAL [n].

Initial guess 2		$t \in [0, 5]$		$t \in [0, 10]$	
	[0]	[b]	[n]	[b]	[n]
$\hat{a}[i](\cdot)$	300	Fig 8		Fig 15	
$\hat{q}[i](\cdot)$	0.1	Fig 9		Fig 16	
$\hat{m}_1[i]$	15	11.9	11.9	11.2	11.2
$\hat{m}_2[i]$	1	2.7	2.8	0.1	0.1
$\hat{q}_1[i]$	0.1	3.4	3.4	2.6	2.6
$\hat{q}_2[i]$	20	21.0	20.0	26.5	26.5
$\hat{\sigma}[i]$	200	166.5	167.4	189.0	189.2

TABLE II
PARAMETER ESTIMATION VALUES FOR DIFFERENT ITERATION, INITIAL [0], MINIMAL COST [b], AND FINAL [n].

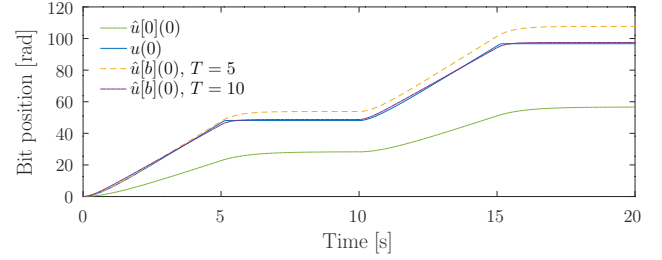


Fig. 21. Bit position real data and identified system responses, for initial guess 2.

ε_a	1	c_a	10^{-10}
ε_q	100	c_q	10^{-10}
ε_{m_1}	100	c_{m_1}	10^{-10}
ε_{m_2}	100	c_{m_2}	10^{-10}
ε_{q_1}	100	c_{q_1}	10^{-10}
ε_{q_2}	100	c_{q_2}	10^{-10}
ε_σ	1	c_σ	0

$$\varepsilon = [\varepsilon_a \mathbb{1}_N \quad \varepsilon_q \mathbb{1}_N \quad \varepsilon_{m_1} \quad \varepsilon_{m_2} \quad \varepsilon_{q_1} \quad \varepsilon_{q_2} \quad \varepsilon_\sigma], \quad \kappa = 0.5.$$

where $\mathbb{1}_N$ is a vector full of 1 of length N .

TABLE III
GEOGRAD TUNING PARAMETER VALUES, AND REGULARIZATION COROLLARY 3.1 PARAMETER VALUES.

The position responses in Figures 13, 14, 20, and 21 for $t \in [10, 20]$ are not used in the estimation procedure. These data validates the estimated model a posteriori. For $t \in [0, 5]$ or $t \in [0, 10]$ we see that the proposed procedure allows to fit the trajectory of the considered model with experimental data. Then for $t \in [10, 20]$ we see that the trajectory of the model also fit with validation data. Experimental results show that the proposed algorithm yields an accurate model

of the actual system, despite the presence of nonlinear unmodeled dynamics (these nonlinearities are visible in Figure 10: the response from 0 to 10 rad/s is not symmetric compared to the one from 10 to 0). Moreover our estimation is numerically very efficient, even without performing heavy and sophisticated data filtering. The only used filters are hardware and inherent to the experiment.

VI. CONCLUSION

In this paper, we proposed an algorithm for parameter identification of a wave equation from experimental boundary data. The proposed approach relies on the use of calculus of variation and functional analysis tools. In particular, we recast the parameter identification problem as a minimization of a misfit functional. This misfit functional represents the square of an L^2 error between data and the boundary responses of a specific wave equation. The resulting (infinite dimensional) optimization is solved via an iterative algorithm that can jump between local minima. Despite its simplicity, the proposed algorithm was shown to be very efficient for solving the parameter identification problem in numerical simulations and on actual experimental data. As a future work, we plant to explore the use of on nonconvex optimization algorithms as the so-called stochastic gradient descent algorithm; see [23], [12].

APPENDIX

A. Calculus of variations and proof of Theorem 3.1

Before proving Theorem 3.1, let us first consider an example of calculus of variations.

Example A.1: Let us consider the following scalar evolution system (see [15]).

$$\dot{y}(t) = ay(t) + bu(t), \quad y(0) = y_0. \quad (52)$$

Let

$$J(y, u, \hat{y}, \lambda, \hat{a}, \hat{b}) = \int_0^T (y(t) - \hat{y}(t))^2 dt + \int_0^T \lambda(t)(\dot{\hat{y}}(t) - \hat{a}\hat{y}(t) - \hat{b}u(t))dt. \quad (53)$$

The definition of the first variation of J with respect to (estimate parameters) $\hat{y}$, λ , $\hat{a}$ and $\hat{b}$ is

$$\begin{aligned} \delta J(y, u, \hat{y}, \delta \hat{y}, \lambda, \delta \lambda, \hat{a}, \delta \hat{a}, \hat{b}, \delta \hat{b}) = \\ J(y, u, \hat{y} + \delta \hat{y}, \lambda + \delta \lambda, \hat{a} + \delta \hat{a}, \hat{b} + \delta \hat{b}) \\ - J(y, u, \hat{y}, \lambda, \hat{a}, \hat{b}) + O(\|\delta \hat{y}\|^2 + \|\delta \lambda\|^2 + \|\delta \hat{a}\|^2 + \|\delta \hat{b}\|^2). \end{aligned} \quad (54)$$

One computes, using the previous definition and one integration by parts, that

$$\begin{aligned} \delta J = & \int_0^T -(y(t) - \hat{y}(t))^2 dt \\ & + \int_0^T \delta \lambda(t)(\dot{\hat{y}}(t) - \hat{a}\hat{y}(t) - \hat{b}u(t))dt \\ & + \int_0^T \delta \hat{y}(t)(-\dot{\lambda}(t) - \hat{a}\delta \hat{y}(t))dt + [\lambda(t)\delta \hat{y}(t)]_{t=0}^T \\ & + \int_0^T \lambda(t)(-\hat{y}(t)\delta \hat{a} - u(t)\delta \hat{b}). \end{aligned} \quad (55)$$

As the initial condition is fixed $\hat{y}(0) = y_0$, then we do not study the resulting variation, i.e., $\delta \hat{y}(0) = 0$. Therefore one gets that if

- $\dot{\hat{y}}(t) = \hat{a}\hat{y}(t) + \hat{b}u(t)$, $\hat{y}(0) = y_0$,
- $\dot{\lambda}(t) = -\hat{a}\lambda(t) - (y(t) - \hat{y}(t))$, $\lambda(T) = 0$,

then the first variation of J is

$$\begin{aligned} \delta J = & \int_0^T \lambda(t)(-\hat{y}(t)\delta \hat{a} - u(t)\delta \hat{b})dt \\ = & \left\langle dJ, \begin{bmatrix} \delta \hat{a} \\ \delta \hat{b} \end{bmatrix} \right\rangle. \end{aligned} \quad (56)$$

$$\text{with } dJ = \begin{bmatrix} \int_0^T -\lambda(t)\hat{y}(t)dt \\ \int_0^T -\lambda(t)u(t)dt \end{bmatrix}. \quad \star$$

Similar arguments are used to obtain the following proof. Abusively we called the first variation simply the variation.

Proof of Theorem 3.1: Let us compute the variation of J defined in (12)

$$\delta J = \delta J_1 + \delta J_2. \quad (57)$$

From this point we drop the obvious dependence of the different variables. One gets

$$\begin{aligned} \delta J_1 = & \int_0^T -((u(0, t) - \hat{u}(0, t))\delta \hat{u}(0, t) \\ & + (u(1, t) - \hat{u}(1, t))\delta \hat{u}(1, t))dt. \end{aligned} \quad (58)$$

Note that $\delta u(\cdot, t) = 0$. One computes the variations of J_2

$$\begin{aligned} \delta J_2 = & \int_0^T \left(\int_0^1 [(\hat{u}_{tt} - (\hat{a}\hat{u}_x)_x + \hat{q}\hat{u}_t)\delta \lambda + \lambda\delta \hat{u}_{tt} \right. \\ & - \lambda\delta(\hat{a}\hat{u}_x)_x + \lambda\hat{q}\delta \hat{u}_t + \lambda\hat{u}_t\delta \hat{q}]dx + (\hat{m}_1\hat{u}_{tt}(1, t) \\ & + \hat{a}(1)\hat{u}_x(1, t) + \hat{q}_1\hat{u}_t(1, t) - \hat{\sigma}U)\delta \lambda_1 \\ & + \lambda_1(\hat{m}_1\delta \hat{u}_{tt}(1, t) + \hat{u}_{tt}(1, t)\delta \hat{m}_1 + \hat{a}(1)\delta \hat{u}_x(1, t) \\ & + \hat{u}_x(1, t)\delta \hat{a}(1) + \hat{q}_1\delta \hat{u}_t(1, t) + \hat{u}_t(1, t)\delta \hat{q}_1 - U\delta \hat{\sigma}) \\ & + (\hat{m}_2\hat{u}_{tt}(0, t) - \hat{a}(0)\hat{u}_x(0, t) + \hat{q}_2\hat{u}_t(0, t))\delta \lambda_2 \\ & + \lambda_2(\hat{m}_2\delta \hat{u}_{tt}(0, t) + \hat{u}_{tt}(0, t)\delta \hat{m}_2 - \hat{a}(0)\delta \hat{u}_x(0, t) \\ & - \hat{u}_x(0, t)\delta \hat{a}(0) + \hat{q}_2\delta \hat{u}_t(0, t) + \hat{u}_t(0, t)\delta \hat{q}_2) \Big) dt. \end{aligned} \quad (59)$$

Using integrations by parts, it yields

$$\int_0^T \lambda \delta \hat{u}_{tt} dt = [\lambda \delta \hat{u}_t - \lambda_t \delta \hat{u}]_{t=0}^T + \int_0^T \lambda_{tt} \delta \hat{u} dt, \quad (60)$$

$$\int_0^T \lambda \hat{q} \delta \hat{u}_t dt = [\lambda \hat{q} \delta \hat{u}]_{t=0}^T + \int_0^T -(\lambda \hat{q})_t \delta \hat{u} dt, \quad (61)$$

$$\begin{aligned} \int_0^T \lambda_1 \delta \hat{u}_{tt}(1, t) dt &= [\lambda_1 \delta \hat{u}_t(1, t) - \dot{\lambda}_1 \delta \hat{u}(1, t)]_{t=0}^T \\ &+ \int_0^T \ddot{\lambda}_1 \delta \hat{u}(0, t) dt, \end{aligned} \quad (62)$$

$$\begin{aligned} \int_0^T \lambda_1 \hat{q}_1 \hat{u}_t(1, t) dt &= [\lambda_1 \hat{q}_1 \delta \hat{u}(1, t)]_{t=0}^T \\ &+ \int_0^T -(\lambda_1 \hat{q}_1)_t \delta \hat{u}(1, t) dt, \end{aligned} \quad (63)$$

$$\begin{aligned} \int_0^1 -\lambda \delta (\hat{u}_x)_x dx &= [-\lambda \delta \hat{u}_x]_{x=0}^1 + \int_0^1 \lambda_x \delta \hat{u}_x dx \\ &= [-\lambda \delta \hat{u}_x + \lambda_x \hat{u} \delta \hat{u}]_{x=0}^1 \\ &+ \int_0^1 \lambda_x \hat{u}_x \delta \hat{u} dx + \int_0^1 -(\lambda_x \hat{u})_x \delta \hat{u} dx. \end{aligned} \quad (64)$$

Therefore one obtains

$$\begin{aligned} \delta J_2 &= \int_0^T \left(\int_0^1 [(\hat{u}_{tt} - (\hat{u}_x)_x + \hat{q} \hat{u}_t) \delta \lambda + \lambda \hat{u}_t \delta \hat{q} \right. \\ &+ \lambda_x \hat{u}_x \delta \hat{u} + (\lambda_{tt} - (\lambda_x \hat{u})_x - (\lambda \hat{q})_t) \delta \hat{u}] dx \\ &+ (\hat{m}_1 \hat{u}_{tt}(1, t) + \hat{a}(1) \hat{u}_x(1, t) + \hat{q}_1 \hat{u}_t(1, t) - \hat{\sigma} U) \delta \lambda_1 \\ &+ (\hat{m}_2 \hat{u}_{tt}(0, t) - \hat{a}(0) \hat{u}_x(0, t) + \hat{q}_2 \hat{u}_t(0, t)) \delta \lambda_2 \\ &+ \lambda_1 (\hat{u}_{tt}(1, t) \delta \hat{m}_1 + \hat{a}(1) \delta \hat{u}_x(1, t) + \hat{u}_x(1, t) \delta \hat{a}(1) \\ &+ \hat{u}_t(1, t) \delta \hat{q}_1 - U \delta \hat{\sigma}) \\ &+ \lambda_2 (\hat{u}_{tt}(0, t) \delta \hat{m}_2 - \hat{a}(0) \delta \hat{u}_x(0, t) - \hat{u}_x(0, t) \delta \hat{a}(0) \\ &+ \hat{u}_t(0, t) \delta \hat{q}_2) + (\hat{m}_1 \ddot{\lambda}_1 \\ &- (\lambda_1 \hat{q}_1)_t \delta \hat{u}(1, t) + (\hat{m}_2 \ddot{\lambda}_2 - (\lambda_2 \hat{q}_2)_t) \delta \hat{u}(0, t) \Big) dt \\ &+ \int_0^1 \left[\lambda \delta \hat{u}_t - \lambda_t \delta \hat{u} + \lambda \hat{q} \delta \hat{u} \right]_{t=0}^T dx \\ &+ \int_0^T \left[-\lambda \hat{a} \delta u_x + \lambda_x \hat{u} \delta \hat{u} \right]_{x=0}^1 dt \\ &+ \left[\lambda_1 \hat{m}_1 \delta \hat{u}_t(1, t) - \dot{\lambda}_1 \hat{m}_1 \delta \hat{u}(1, t) + \lambda_1 \hat{q}_1 \delta \hat{u}(1, t) \right. \\ &+ \left. \lambda_2 \hat{m}_2 \delta \hat{u}_t(0, t) - \dot{\lambda}_2 \hat{m}_2 \delta \hat{u}(0, t) + \lambda_2 \hat{q}_2 \delta \hat{u}(0, t) \right]_{t=0}^T. \end{aligned} \quad (65)$$

Note that due to the fact that the initial conditions are known $\delta \hat{u}(\cdot, 0) = \delta \hat{u}_t(\cdot, 0) = 0$. The idea is to get the variations of $J = J_1 + J_2$ with respect to the variation of the parameters. Therefore we enforce the factor of other variation to be equal to zero. In other words, taking the factor of $\delta \lambda$, $\delta \lambda_1$, and $\delta \lambda_2$ to be equal to zero is equivalent to $\hat{u} = \Sigma_u(\hat{p})$. Moreover taking the factor of $\delta \hat{u}$, $\delta \hat{u}_x(1, t)$,

$\delta \hat{u}_x(0, t)$, $\delta \hat{u}_t(\cdot, t)$, and $\delta \hat{u}(\cdot, t)$ equals to zero is equivalent to

$$\begin{cases} \lambda_{tt}(x, t) - (\lambda_x(x, t) \hat{a}(x))_x - (\lambda(x, t) \hat{q}(x))_t = 0, & (66a) \\ \lambda_1(t) \hat{a}(1) - \lambda(1, t) \hat{a}(1) = 0, & (66b) \\ -\lambda_2(t) \hat{a}(0) + \lambda(0, t) \hat{a}(0) = 0, & (66c) \\ \lambda(\cdot, T) = \lambda_t(\cdot, T) = 0. & (66d) \end{cases}$$

Imposing the factor of $\delta \hat{u}(1, t)$, $\delta \hat{u}_t(1, T)$, and $\delta \hat{u}(1, T)$ to be zero is equivalent to

$$\begin{cases} 0 = -(u(1, t) - \hat{u}(1, t)) + \lambda_x(1, t) \hat{a}(1) \\ \quad + \hat{m}_1 \ddot{\lambda}_1(t) - (\lambda_1(t) \hat{q}_1)_t, & (67a) \end{cases}$$

$$\begin{cases} \lambda_1(T) = 0, & (67b) \end{cases}$$

$$\begin{cases} \dot{\lambda}_1(T) = 0. & (67c) \end{cases}$$

Ensuring that the factor of $\delta \hat{u}(0, t)$, $\delta \hat{u}_t(0, T)$, and $\delta \hat{u}(0, T)$ is zero is equivalent to

$$\begin{cases} 0 = -(u(0, t) - \hat{u}(0, t)) - \lambda_x(0, t) \hat{a}(0) \\ \quad + \hat{m}_2 \ddot{\lambda}_2(t) - (\lambda_2(t) \hat{q}_2)_t, & (68a) \end{cases}$$

$$\begin{cases} \lambda_2(T) = 0, & (68b) \end{cases}$$

$$\begin{cases} \dot{\lambda}_2(T) = 0. & (68c) \end{cases}$$

Consider conjointly (66)-(68), these systems are equivalent to

$$(i) \quad \lambda = \Sigma_\lambda(\hat{p}, \hat{u}) \quad (\lambda \text{ solution to (20)}),$$

$$(ii) \quad \lambda_1(t) = \lambda(1, t),$$

$$(iii) \quad \text{and } \lambda_2(t) = \lambda(0, t).$$

The condition $\lambda_1(t) = \lambda(1, t)$ implies that the factor of $\hat{a}(1)$ is equal to zero. Moreover $\lambda_2(t) = \lambda(0, t)$ implies that the factor of $\hat{a}(0)$ is equal to zero. Then the variation of J can be written as

$$\delta J = \left\langle \begin{bmatrix} \int_0^T \lambda_x(x, t) \hat{u}_x(x, t) dt \\ \int_0^T \lambda(x, t) \hat{u}_t(x, t) dt \\ \int_0^T \lambda_1(t) \hat{u}_{tt}(1, t) dt \\ \int_0^T \lambda_2(t) \hat{u}_{tt}(0, t) dt \\ \int_0^T \lambda_1(t) \hat{u}_t(1, t) dt \\ \int_0^T \lambda_2(t) \hat{u}_t(0, t) dt \\ - \int_0^T \lambda_1(t) U(t) dt \end{bmatrix}, \begin{bmatrix} \delta \hat{a}(x) \\ \delta \hat{q}(x) \\ \delta \hat{m}_1 \\ \delta \hat{m}_2 \\ \delta \hat{q}_1 \\ \delta \hat{q}_2 \\ \delta \hat{\sigma} \end{bmatrix} \right\rangle. \quad (69)$$

Using integrations by parts on the factor of $\delta \hat{m}_1$ and $\delta \hat{m}_2$ together with (4d), (67b), and (68b) one gets (22). This concludes the proof of Theorem 3.1. ■

B. Proof of Theorem 3.2

The idea of the proof is to decompose the operator $\mathcal{A}$ defined in (31) into one m-accretive part and a remaining part, then to perform a bijective change of variable which cancels the remaining part. Finally, we conclude using the following theorem

Theorem A.1: [Inhomogenous Abstract Problem, [Theorem 7.10 on [2] p. 198]]

Consider, in a Banach space E , the problem

$$\begin{cases} \frac{dX}{dt}(t) + \mathcal{A}X(t) = f(t), & (70a) \\ X(0) = X_0. & (70b) \end{cases}$$

Assume that $\mathcal{A}$ is m -accretive (equivalent to maximal monotone in a Hilbert space). Then for every $X_0 \in \mathcal{D}(\mathcal{A})$ and every $f \in C^1([0, T]; \mathbb{E})$, there exists a unique solution X to (70) with

$$X \in C^1([0, T]; \mathbb{E}) \cap C([0, T]; \mathcal{D}(\mathcal{A})). \quad (71)$$

Now consider the following operator

$$\forall z \in \mathcal{D}(\mathcal{G}), \quad \mathcal{G}z = \begin{bmatrix} -z_2 \\ -(az'_1)' + z_2 - z_1 \\ \frac{a(1)z'_1(1)}{m_1} \\ -\frac{a(0)z'_1(0)}{m_2} \end{bmatrix}, \quad (72)$$

and the following matrix

$$\mathcal{H} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -q - 1 & 0 & 0 \\ 0 & 0 & -\frac{q_1}{m_1} & 0 \\ 0 & 0 & 0 & -\frac{q_2}{m_2} \end{bmatrix}. \quad (73)$$

The domain of $\mathcal{G}$ is equal to the domain of $\mathcal{A}$. One gets

$$\mathcal{A} = \mathcal{G} + \mathcal{H}. \quad (74)$$

$\mathcal{G}$ is the m -accretive part, this is established in the following lemma

Lemma A.1: *The linear unbounded operator $\mathcal{G}$ defined in (72) is maximal monotone on $\mathbb{H}$ defined in (29).* $\square$

Proof: Considering the definition of the scalar product (30), one gets

$$\begin{aligned} \langle z, \mathcal{G}z \rangle_{\mathbb{H}} &= \int_0^1 [z_1 z_2 - z'_1 z'_2 + z_2 (-(az'_1)' + z_2 - z_1)] dx \\ &\quad + a(1)z_3 z'_1(1) - a(0)z_4 z'_1(0), \end{aligned} \quad (75)$$

using the fact that $z \in \mathcal{D}(\mathcal{A})$, one obtains

$$\langle z, \mathcal{G}z \rangle_{\mathbb{H}} = \int_0^1 z_2^2 dx \geq 0. \quad (76)$$

Thus the operator $\mathcal{G}$ is monotone (see [2] Chapter 7 on Page 181). In addition if we establish that

$$R(I + \mathcal{G}) = \mathbb{H}, \quad (77)$$

then the operator $\mathcal{G}$ is maximal monotone (see [2] Chapter 7 on Page 181, R stands for the range of the operator). Let $y \in \mathbb{H}$, we have to solve

$$z \in \mathcal{D}(\mathcal{A}), \quad z + \mathcal{G}z = y, \quad (78)$$

which means that

$$z_1 - z_2 = y_1, \quad (79)$$

$$z_2 - (az'_1)' + z_2 - z_1 = y_2, \quad (80)$$

$$m_1 z_3 + a(1)z'_1(1) = m_1 y_3, \quad (81)$$

$$m_2 z_4 - a(0)z'_1(0) = m_2 y_4, \quad (82)$$

using the fact that $z \in \mathcal{D}(\mathcal{A})$ one gets

$$z_1 - (az'_1)' = 2y_1 + y_2, \quad (83)$$

$$a(1)z'_1(1) + m_1 z_1(1) = m_1(y_3 + y_1(1)), \quad (84)$$

$$-a(0)z'_1(0) + m_2 z_1(0) = m_2(y_4 + y_1(0)). \quad (85)$$

This is a classical stationary problem (e.g. see [2]) with Robin's boundaries conditions, using standard result (as done in [2] p. 226 Example 6) one gets that as $2y_1 + y_2 \in L^2(0, 1)$, (83)-(85) has a unique solution $z_1 \in H^2(0, 1)$. Now one can check that the element $z = (z_1, z_2, z_3, z_4)$ with

$$\begin{cases} z_1 \text{ is solution to (83)-(85),} \end{cases} \quad (86a)$$

$$\begin{cases} z_2 = z_1 - y_1, \end{cases} \quad (86b)$$

$$\begin{cases} z_3 = y_3 + \frac{-a(1)z'_1(1)}{m_1}, \end{cases} \quad (86c)$$

$$\begin{cases} z_4 = y_4 + \frac{a(0)z'_1(0)}{m_2}, \end{cases} \quad (86d)$$

satisfies (79)-(82). Moreover using (83)-(85) on (86) one gets that z satisfying (86) is in $\mathcal{D}(\mathcal{A})$. $\blacksquare$

Now, we are ready to state the proof of the well-posedness of (33). Note that the fact that G is maximal monotone implies that $\mathcal{D}(\mathcal{A})$ is dense in $\mathbb{H}$ (i.e., $\overline{\mathcal{D}(\mathcal{A})} = \mathbb{H}$).

Proof of Theorem 3.2: Using the bijective change of variable

$$z_e(t) = z(t)e^{\mathcal{H}t}, \quad (87)$$

z is solution to (33) is equivalent to, $z_e \in \mathcal{D}(\mathcal{A})$ is solution to

$$\begin{cases} \frac{d}{dt} z_e(t) + \mathcal{G}z_e(t) = \mathcal{U}(t)e^{\mathcal{H}t}, \end{cases} \quad (88a)$$

$$\begin{cases} z_e(0) = z_0, \end{cases} \quad (88b)$$

where $\mathcal{H}$ is defined in (73) and $\mathcal{G}$ is defined in (72).

From Lemma A.1, using Theorem A.1 on (88), and the change of variable (87), one establishes (i). Using argument of density of $\mathcal{D}(\mathcal{A})$ in $\mathbb{H}$, and C_0 -semigroups theory one obtains (ii).

Note that we refer the reader to [14], [21] for the notion mild solutions. Moreover part of the proof are inspired from [4] and [6] which in turn originates from [25]. $\blacksquare$

C. State identifiability

We only prove the second assertion (ii) of Theorem 3.3, the other item can be established by following similar arguments.

Proof of Theorem 3.3 (ii):

The proof is based on showing the converse. Consider that

$$\hat{p} \neq p. \quad (89)$$

We study the dynamics of the difference,

$$\tilde{u}(x, t) = u(x, t) - \hat{u}(x, t). \quad (90)$$

In the abstract form, we get

$$\begin{cases} \dot{\tilde{z}}(t) + \mathcal{A}(p)\tilde{z}(t) - \mathcal{A}(\hat{p})\tilde{z}(t) = \mathcal{U}(p, t) \\ \quad - \mathcal{U}(\hat{p}, t), \end{cases} \quad (91a)$$

$$\begin{cases} \tilde{z}(0) = 0. \end{cases} \quad (91b)$$

One gets using the linearity of the operator $\mathcal{A}$:

$$\dot{\tilde{z}}(t) + \tilde{\mathcal{A}}(\hat{p}, p)z(t) + \mathcal{A}(\hat{p})\tilde{z}(t) = \tilde{\mathcal{U}}(\sigma, \hat{\sigma}, t). \quad (92)$$

By continuity of the C_0 -semigroup associated with the difference dynamics, we have that

$$\tilde{z} = 0 \Leftrightarrow \dot{\tilde{z}}(t) = 0, \quad \forall t \in (0, T]. \quad (93)$$

First consider the case where $\sigma = \hat{\sigma}$, one gets

$$\tilde{U}(\sigma, \sigma, t) = 0. \quad (94)$$

We recall that $\sigma \neq 0$, this implies that $z \neq 0$, i.e., there exist several time t_i such that $z(t_i) \neq 0$. We take the first one, and denote it t_1 , by continuity one gets

$$\dot{\tilde{z}}(t_1) + \tilde{\mathcal{A}}(\hat{p}, p)z(t_1) = 0, \quad (95)$$

therefore, $z = 0$ is equivalent to $\tilde{\mathcal{A}}(\hat{p}, p) := \mathcal{A}(p) - \mathcal{A}(\hat{p}) = 0$. It follows

$$\tilde{\mathcal{A}}(\hat{p}, p) = 0 \Leftrightarrow \hat{p} = p. \quad (96)$$

This concludes the first case.

Now consider that $\sigma \neq \hat{\sigma}$, we still have that $z \neq 0$ as $\sigma \neq 0$ and $\exists t_0 \in [0, T]$, $U(t_0) \neq 0$. We consider the first time such that U is not equal to zero, we denote it t_2 , by continuity of the C_0 -semigroups, we have

$$\dot{\tilde{z}}(t_2) = \tilde{\mathcal{U}}(\sigma, \hat{\sigma}, t_2). \quad (97)$$

It is easy to show that $\tilde{\mathcal{U}}(\sigma, \hat{\sigma}, t_2) := \mathcal{U}(\sigma, t_2) - \mathcal{U}(\hat{\sigma}, t_2) = 0$ is equivalent to $\sigma = \hat{\sigma}$. This concludes the second and last case, and also the proof of Theorem 3.3. ■

D. Boundary observability

Proof of Theorem 3.4: Pick two solutions u and $\tilde{u}$ to (28) and assume that $y(t) - \tilde{y}(t) = 0$, $\forall t \in [0, \infty)$, where y is defined in (5). Let us consider the dynamics of the difference $\tilde{u}(x, t) = u(x, t) - \tilde{u}(x, t)$ which is driven by the following over-determined system of PDEs

$$\begin{cases} \tilde{u}_{tt}(x, t) = (a(x)\tilde{u}_x(x, t))_x - q(x)\tilde{u}_t(x, t), & (98a) \end{cases}$$

$$\begin{cases} m_1\tilde{u}_{tt}(1, t) = -a(1)\tilde{u}_x(1, t) - q_1\tilde{u}_t(1, t), & (98b) \end{cases}$$

$$\begin{cases} m_2\tilde{u}_{tt}(0, t) = a(0)\tilde{u}_x(0, t) - q_2\tilde{u}_t(0, t), & (98c) \end{cases}$$

$$\begin{cases} \tilde{u}(1, t) = 0, & (98d) \end{cases}$$

$$\begin{cases} \tilde{u}(0, t) = 0. & (98e) \end{cases}$$

Next, we prove that the unique solution to the above system is zero. To this end, we consider the associated first order hyperbolic system. Define

$$f(x, t) = \begin{bmatrix} \tilde{u}_t(x, t) + \sqrt{a(x)}\tilde{u}_x(x, t) \\ \tilde{u}_t(x, t) - \sqrt{a(x)}\tilde{u}_x(x, t) \end{bmatrix} \quad (99)$$

where $a(\cdot) > 0$. Since $\tilde{u}$ is solution to (98), then f is solution to the following system

$$\begin{aligned} f_t(x, t) &= \begin{bmatrix} \sqrt{a(x)} & 0 \\ 0 & -\sqrt{a(x)} \end{bmatrix} f_x(x, t) + B(x)f(x, t) \\ f(0, t) &= 0 \\ f(1, t) &= 0 \end{aligned} \quad (100)$$

where $B(x) \in \mathbb{R}^{2 \times 2}$. Let us consider the following invertible change of variable: $g(x, t) := e^{-\int_0^x B(s)ds} f(x, t)$. Then, (100) turns into

$$\begin{cases} g_t(x, t) = \begin{bmatrix} \sqrt{a(x)} & 0 \\ 0 & -\sqrt{a(x)} \end{bmatrix} g_x(x, t) & (101a) \end{cases}$$

$$\begin{cases} g(0, t) = 0 & (101b) \end{cases}$$

$$\begin{cases} g(1, t) = 0 & (101c) \end{cases}$$

This shows that, since $g(\cdot, 0) = 0$, $g \equiv 0$. This concludes the proof. ■

REFERENCES

- [1] V. Bacchelli, D. Pierotti, S. Micheletti, and S. Perotto, "Parameter identification for the linear wave equation with robin boundary condition," *Journal of Inverse and Ill-posed Problems*, 05 2018.
- [2] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.
- [3] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Mathematical Programming*, vol. 89, no. 1, pp. 149–185, Nov 2000.
- [4] F. Conrad and A. Mifdal, "Strong stability of a model of an overhead crane," *Control and Cybernetics*, vol. 27, no. 3, pp. 363–374, 1998.
- [5] S. Cyrus, B. Hu, B. Van Scoy, and L. Lessard, "A robust accelerated optimization algorithm for strongly convex functions," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 1376–1381.
- [6] B. d'Andréa-Novet, F. Boustany, and F. Conrad, "Control of an overhead crane: Stabilization of flexibilities," *Boundary Control and Boundary Variation*, pp. 1–26, 1992.
- [7] H. Fu and B. Han, "A wavelet multiscale method for the inverse problems of a two-dimensional wave equation," *Inverse Problems in Science and Engineering*, vol. 12, no. 6, pp. 643–656, 2004.
- [8] Ha, J. and Gutman, S., "Identifiability for linearized sine-gordon equation," *Math. Model. Nat. Phenom.*, vol. 8, no. 1, pp. 106–121, 2013.
- [9] B. Han, H. S. Fu, and Z. Li, "A homotopy method for the inversion of a two-dimensional acoustic wave equation," *Inverse Problems in Science and Engineering*, vol. 13, no. 4, pp. 411–431, 2005.
- [10] D. Hua, Z. Zhe-min, and X. Shou-ze, "A new approach to inverse problems of wave equations," *Applied Mathematics and Mechanics*, vol. 11, no. 12, pp. 1113–1117, Dec 1990.
- [11] V. Isakov, *Inverse problems for partial differential equations*. Springer, 2006, vol. 127.
- [12] P. Jain and P. Kar, *Non-convex Optimization for Machine Learning*. now, 2017.
- [13] L. Jijun and W. Yuanming, "On uniqueness of an inverse problem for a 1-d wave equation from transmission data," *SIAM Journal on Applied Mathematics*, vol. 57, no. 1, pp. 195–204, 1997.
- [14] T. Kato, *Perturbation theory for linear operators*. Springer Science & Business Media, 2013, vol. 132.
- [15] M. Kot, *A First Course in the Calculus of Variations*. American Mathematical Society, 2014.
- [16] D. Lesnic, L. Elliott, and D. Ingham, "Identifiability of distributed parameters for high-order quasi-linear differential equations," *Journal of Inverse and Ill-Posed Problems*, vol. 8, 01 2000.
- [17] W. Liao, "A computational method to estimate the unknown coefficient in a wave equation using boundary measurements," *Inverse Problems in Science and Engineering*, vol. 19, no. 6, pp. 855–877, 2011.
- [18] S.-W. Na and L. F. Kallivokas, "On the inverse problem of soil profile reconstruction: a comparison of time-domain approaches," *Computational Mechanics*, vol. 42, no. 6, p. 921, May 2008.
- [19] V. T. Nguyen, D. Georges, and G. Besançon, "Calculus of variations approach for state and parameter estimation in switched 1d hyperbolic pdes," *Optimal Control Applications and Methods*, vol. 39, no. 3, pp. 1182–1201, 2018.
- [20] P. A. Patil and C. Teodoriu, "A comparative review of modelling and controlling torsional vibrations and experimentation using laboratory setups," *Journal of Petroleum Science and Engineering*, vol. 112, pp. 227–238, 2013.

- [21] A. Pazy, *Semigroups of linear operators and applications to partial differential equations*. Springer Science & Business Media, 2012, vol. 44.
- [22] C. Roman, “Boundary control of a wave equation with in-domain damping,” Theses, Université Grenoble Alpes, Aug. 2018.
- [23] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.
- [24] B. Saldivar, S. Mondié, S.-I. Niculescu, H. Mounier, and I. Boussaada, “A control oriented guided tour in oilwell drilling vibration modeling,” *Annual Reviews in Control*, vol. 42, pp. 100 – 113, 2016.
- [25] M. Slemrod, “Feedback stabilization of a linear control system in hilbert space with an a priori bounded control,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 3, pp. 265–285, Sep 1989.
- [26] M. Tadi, “Evaluation of the elastic property based on boundary measurement,” *Acta Mechanica*, vol. 129, no. 3, pp. 231–241, Sep 1998.
- [27] L. Wang, Z. Wang, and Z. Qian, “A meshfree method for inverse wave propagation using collocation and radial basis functions,” *Computer Methods in Applied Mechanics and Engineering*, vol. 322, pp. 311 – 350, 2017.
- [28] G. Q. Xie, “A new iterative method for solving the coefficient inverse problem of the wave equation,” *Communications on Pure and Applied Mathematics*, vol. 39, no. 3, pp. 307–322, 1986.



Christophe Prieur graduated in Mathematics from the Ecole Normale Supérieure de Cachan, France in 2000. He received the Ph.D degree in 2001 in Applied Mathematics from the Université Paris-Sud, France. He is currently a senior researcher of the CNRS (since 2011). His current research interests include nonlinear control theory, hybrid systems, and control of partial differential equations. He was the Program Chair of the 9th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2013) and of the 14th European Control Conference (ECC 2015). He has been a member of the IEEE-CSS Conference Editorial Board and an associate editor for IMA J. Mathematical Control and Information, for the IEEE Trans. on Automatic Control and European J. of Control. He is currently a member of the EUCA-CEB, an associate editor for the IEEE Trans. on Control Systems Technology, a senior editor for the IEEE Control Systems Letters, and an editor for the IMA Journal of Mathematical.



Christophe Roman was born in Val-des-près, France, in 1991. He graduated in applied physics and electrical engineering from the Ecole Normale Supérieure de Cachan, France in 2015. He received the Ph. D degree in Automatic from the University Grenoble Alpes in 2018. After that he held a post-doc position at the same laboratory during one year. He worked a year at Femto-st, at Besançon, France. He is currently associate professor at Aix-Marseille University and member of the Laboratory Informatique and system. His main

research works are the study of infinite-dimensional dynamical system: identification, observation, output feedback stabilization, boundary control, adaptive and robust control, and their experimental applications.



Francesco Ferrante is an assistant professor of Control Systems at the University of Grenoble Alpes and a member of the Grenoble Images Speech Signal and Control Laboratory. He received in 2010 a “Laurea degree” (BSc) in Control Engineering from University “Sapienza” in Rome, Italy and in 2012 a “Laurea Magistrale” degree (MSc) cum laude in Control Engineering from University “Tor Vergata” in Rome, Italy. During 2014, he held a visiting scholar position at the Department of Computer Engineering, University

of California Santa Cruz. In 2015, he received a PhD degree in control theory from “Institut supérieur de l’aéronautique et de l’espace” (SUPAERO) Toulouse, France. From November 2015 to August 2016, he was a post-doctoral fellow at the Department of Electrical and Computer Engineering, Clemson University, USA. From August 2016 to September 2017, he held a position as postdoctoral scientist at the Hybrid Systems Laboratory (HSL) at the University of California at Santa Cruz, USA. His research interests are in the general area of systems and control with a special focus on hybrid systems, observer design, and the application of convex optimization in systems and control. He is the ex-aequo recipient of the “Best Ph.D. Thesis Award 2017” from the Research Cluster on Modeling, Analysis, and Management of Dynamical Systems (GDR-MACS) of the French National Council of Scientific Research (CNRS). He currently serves as an Associate Editor for the European Journal of Control, the IMA journal of Journal of Mathematical Control and Information, and the conference editorial boards of the IEEE Control Systems Society and the European Control Association.