2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

# Uncertainty Learning for LTI Systems with Stability Guarantees

Farhad Ghanipoor[1], Carlos Murguia[1], Peyman Mohajerin Esfahani[2], and Nathan van de Wouw[1]

*Abstract*— **We present a framework for learning of modeling uncertainties in Linear Time Invariant (LTI) systems to improve the predictive capacity of system models in the input-output sense. First, we propose a methodology to extend the LTI model with an uncertainty model. The proposed framework guarantees stability of the extended model. To achieve this, two semi-definite programs are provided that allow obtaining optimal uncertainty model parameters, given state and uncertainty data. Second, to obtain this data from available input-output trajectory data, we introduce a filter in which an internal model of the uncertainty is proposed. This filter is also designed via a semi-definite program with guaranteed robustness with respect to uncertainty model mismatches, disturbances, and noise. Numerical simulations are presented to illustrate the effectiveness and practicality of the proposed methodology in improving model accuracy, while guaranteeing model stability.**

## I. INTRODUCTION

Modeling dynamical systems is crucial across various engineering and scientific fields. It is essential to incorporate established principles, such as known physics, and leverage prior knowledge, such as stability, for effective modeling [1], [2].

For a class of linear uncertain dynamical systems, this paper focuses on learning models for uncertainties while guaranteeing stability of extended models (prior models plus uncertainty characterization), given available input-output data. This problem contrasts with black box modelling approaches, e.g., using Neural Networks (NNs) or Gaussian Processes (GPs), as we incorporate prior relations that come from first-principles into the modeling and learning scheme. Moreover, this problem differs from identifying a full model in a gray box fashion, as a prior model with known parameters is given. However, such problems can be a subclass of the problem we consider here with no prior model. Below, some related existing literature is provided.

*Existing Literature:* Our approach, augments a known physics-based model by a black-box model used as a correction term. Such generic approach is also taken by Quaghebeur et al. in [3], who add an NN model to a known physics-based model with unknown parameters. This approach allows maintaining the basic structure of the model that comes from first principles, which improves interpretability. However, it requires simulating the hybrid model at each iteration during the training process. This approach is clearly more computationally expensive compared to the proposed method, which alleviates the need for simulating the model in every iteration. Furthermore, the main drawback of this method is that it assumes that the initial state of the dynamic system is known or at least it requires measuring all the states (full-state measurement) of the true dynamical system.

Furthermore, our approach offers stability guarantees for the extended LTI model (i.e., the model consisting of the known physics-based model and the uncertainty model). The identification of stable LTI models has (mainly) been studied in the context of discrete systems [4]–[6]. For instance, in [7], the authors provide convex constraints to ensure incremental stability for linear non-autonomous discrete-time models and some nonlinear models such as recurrent neural networks. Additionally, there exist studies that have explored model identification with asymptotic stability guarantees for discrete LTI systems using subspace identification methods [8], [9]. Subspace identification methods involve obtaining an estimate of state sequence or extended observability matrix, followed by solving a least squares problem to estimate the model parameters. One way to ensure the asymptotic stability of the model is to add stability constraints to the least squares optimization [10]. This addition results in a convex linear program with mixed equality, quadratic, and semi-definite constraints. Moreover, there exist some studies such as [11], [12] which provide non-parametric model identification with stability guarantees for discrete and continuous LTI systems using kernel-based approaches. It can be observed that stable LTI model identification has a rich literature, with a focus on identifying the complete dynamics. In this context, our focus is on identifying the missing elements (uncertainty) in known physics-based models.

In this paper, we propose a framework for learning of modeling uncertainties in physics-based models applicable to Linear Time Invariant systems (LTIs). We first focus on fitting uncertainty models, assuming that some realizations of input, (estimated) uncertainty, and (estimated) state are given, while guaranteeing asymptotic stability of the extended model (i.e., known physics-based model plus uncertainty model). This is achieved by formulating the problem as a constraint supervised learning problem.

One key challenge in this problem is the introduction of stability constraints, which is addressed using Lyapunov-based tools. The stability criteria typically lead to a non-convex optimization problem. We tackle this challenge by proposing two different approaches:

1) **Cost Modification:** The first approach involves a change of variables, resulting in cost function being

[1] Farhad Ghanipoor, Carlos Murguia, and Nathan van de Wouw are with the Mechanical Engineering Department, Eindhoven University of Technology, Eindhoven, The Netherlands. Emails: f.ghanipoor@tue.nl, C.G.Murguia@tue.nl, and N.v.d.Wouw@tue.nl.

[2] Peyman Mohajerin Esfahani is with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. Email: P.MohajerinEsfahani@tudelft.nl.

rewritten in terms of these new variables (Theorem 1).

2) **Constraint Modification:** The second approach introduces a sufficient condition to fulfill the stability constraint by solving a convex program (Theorem 2).

Having addressed the non-convexity challenge, the paper proceeds to discuss the practical implementation of the framework. Specifically, it outlines a method for estimating uncertainty and state trajectories using input-output data and the known physics-based model (Proposition 1).

**Notation:** The symbol $\mathbb{R}^+$ denotes the set of nonnegative real numbers. The $n \times n$ identity matrix is denoted by $I_n$ or simply $I$ if $n$ is clear from the context. Similarly, $n \times m$ matrices composed of only zeros are denoted by $0_{n\times m}$ or simply 0 when their dimensions are clear. First and second time-derivatives of a vector $x$ are expressed as $\dot{x}$ and $\ddot{x}$, respectively. For $r^{th}$-order time-derivatives of a vector $x$, the notation $x^{(r)}$ is adopted. A positive definite matrix is denoted by $X \succ 0$ and positive semi-definite matrices are denoted by $X \succeq 0$. Similarly, for a negative definite $X \prec 0$ is used, and $X \preceq 0$ for negative semi-definite matrices. The imaginary unit $j$ is defined by $j^2 = -1$. For a transfer function $T(s)$, with $s \in \mathbb{C}$, $\sigma_{\max}(T(s))$ denotes the maximum singular value, and $T^H(s)$ represents the Hermitian transpose. The notation col$[x_1, \ldots, x_n]$ stands for the column vector composed of the elements $x_1, \ldots, x_n$. This notation is also used when the components $x_i$ are vectors. For a differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ we denote by $\frac{\partial V}{\partial x}$ the row-vector of partial derivatives and by $\dot{V}(x)$ the total derivative of $V(x)$ with respect to time (i.e., $\frac{\partial V}{\partial x}\frac{dx}{dt}$). The notation $tr(W)$ stands for trace of a matrix $W$. We often omit time dependencies for notation simplicity.

## II. Problem Formulation

Consider the system

$$\begin{cases} \dot{x}_s = Ax_s + B_u u + S_\eta \eta(x_s, u) + B_\omega \omega, \\ y_s = Cx_s + D_\nu \nu, \end{cases} \tag{1}$$

where $t \in \mathbb{R}^+$, $x_s \in \mathbb{R}^n$, $y_s \in \mathbb{R}^m$, and $u \in \mathbb{R}^l$ are time, system state, measured output and known input vectors, respectively, and function $\eta : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^{n_\eta}$ is unknown modeling uncertainty. Signals $\omega : \mathbb{R}^+ \to \mathbb{R}^{n_\omega}$ and $\nu : \mathbb{R}^+ \to \mathbb{R}^{m_\nu}$ are unknown bounded disturbances; the former with unknown frequency range and the latter with high-frequency content (e.g., related to measurement noise). Known matrices $(A, B_u, S_\eta, B_\omega, C, D_\nu)$ are of appropriate dimensions, with $n, m, l, n_\eta, n_\omega, m_\nu \in \mathbb{N}$. Matrix $S_\eta$ is used to indicate in which equation(s) the uncertainty $\eta$ appears explicitly.

We aim to fit a data-based model for the uncertainty (i.e., $\eta(\cdot)$ in (1)) using a supervised learning method, while guaranteeing model stability, with the goal of constructing a more accurate system model (valid at least for trajectories close to the training data set). The proposed methods in Section III assume that a data-set (labeled data) of input, (estimated) uncertainty, and (estimated) state realizations are given. This assumption can be considered as another problem for which a solution is provided in Section IV. In what follows, we formulate the problem of uncertainty model learning with stability guarantees.

For the system in (1), consider the following LTI model

$$\begin{aligned} \dot{x} &= Ax + B_u u + S_{\eta_l}\eta_l(x, u), \\ \eta_l(x, u) &:= \Theta_l x + B_l u, \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^n$ is model state and function $\eta_l : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^{n_{\eta_l}}$ is the uncertainty model that is parameterized by $\Theta_l, B_l$. Matrices $(\Theta_l, B_l, S_{\eta_l})$ are of appropriate dimensions, with $n_{\eta_l} \in \mathbb{N}$. Matrix $S_{\eta_l}$, similar to $S_\eta$ in (1) shows explicit appearance of the uncertainty model $\eta_l$ in the right-hand side and could be different from $S_\eta$.

Next, we define a cost function for supervised learning and the stability constraint.

### A. Cost Function

Recall that in this section, we presume that (estimated) uncertainty and state realizations are given. Let us define the following (given) $i$-th sample (in time) data vector $d_i := \begin{bmatrix} \hat{x}_i^\top & u_i^\top & \hat{\eta}_i^\top \end{bmatrix}^\top$, where $\hat{x}_i, u_i$, and $\hat{\eta}_i$ correspond to given $i$-th realizations of state estimation, input, and uncertainty estimation, respectively. Given $N$ samples of data realizations, define the data matrix $D$ as follows:

$$D := \sum_{i=1}^{N} d_i d_i^\top. \tag{3}$$

Further, define the error vector between the uncertainty model and its (given) estimation as $e_i := \eta_l(\hat{x}_i, u_i) - \hat{\eta}_i = T d_i$ with

$$T := \begin{bmatrix} \Theta_l & B_l & -I \end{bmatrix}. \tag{4}$$

Then, we define the following quadratic cost function to be minimized to identify $\Theta_l$ and $B_l$:

$$J := \sum_{i=1}^{N} e_i^\top e_i = \sum_{i=1}^{N} d_i^\top T^\top T d_i. \tag{5}$$

### B. Stability Constraint

We aim to formulate a constraint to satisfy asymptotic stability of the model in (2) via Lyapunov-based stability analysis.

Consider the quadratic function $V(x) = x^\top P x$ for a positive definite matrix $P \succ 0$. If we can find a $P$ such that $\dot{V} < 0$ for $u = 0$ along trajectories of (2); then, the model in (2) is asymptotically stable. The condition $\dot{V} < 0$, for $u = 0$ can be stated as

$$(A + S_{\eta_l}\Theta_l)^\top P + P(A + S_{\eta_l}\Theta_l) \prec 0, \tag{6}$$

or equivalently, by applying the congruence transformation of $Q := P^{-1}$, (6) can be written as

$$(A + S_{\eta_l}\Theta_l)Q + Q(A + S_{\eta_l}\Theta_l)^\top \prec 0. \tag{7}$$

Note that the asymptotic stability conditions above requires the linear matrix of the dynamics $A + S_{\eta_l}\Theta_l$ to be Hurwitz. For a nonzero $u$, this condition implies Input-to-State Stability (ISS) of the model in (2) [13, Col. 5.2], for any $B_u$ and $B_l$. Now, we can state the problem we seek to solve.

**Problem 1 (Uncertainty Model Learning with Stability Guarantee)** *Consider a given data-set of input and (estimated) uncertainty and state realizations. Find the optimal parameters* $\Theta_l$ *and* $B_l$ *of uncertainty model* $\eta_l(\cdot)$ *of the form in* (2) *that minimizes the cost function* $J$ *in* (5)*, such that the system model in* (2) *is asymptotically stable (i.e., respecting the constraint in* (6) *or* (7)*). In other words, find the optimal parameters of the following optimization problem:*

$$\begin{aligned} \min_{P,\Theta_l,B_l} \quad & J \\ \text{s.t.} \quad & (A+S_{\eta_l}\Theta_l)^\top P + P(A+S_{\eta_l}\Theta_l) \prec 0, \\ & P \succ 0. \end{aligned} \tag{8}$$

In what follows, we provide an approximate solution to Problem 1 guaranteeing the stability of the system model described in (2).

## III. Approximate Solution to Problem 1

The challenge is that the stability condition (6) that appears in the optimization problem (8) is not convex in $P$ and $\Theta_l$. Therefore, in what follows we provide two approaches to convexify the optimization problem.

### A. Cost Modification Approach

First, we convexify the stability constraint by a change of variable and rewrite the cost function in (5) in terms of this new variable. The following theorem formalizes the associated convex optimization problem obtained via this approach (which can be considered an approximation to the problem in (8)).

**Theorem 1 (Stable Model Learning with Modified Cost)** *Consider system* (1)*, a given data-set of input and (estimated) uncertainty and state realizations. In addition, consider the uncertainty model of the form in* (2)*. Consider the following convex program:*

$$\begin{aligned} \min_{P,S,R,W} \quad & tr(W) \\ \text{s.t.} \quad & A^\top P + PA + S^\top + S \prec 0, && \text{(9a)} \\ & \begin{bmatrix} 2P & \tilde{T}\tilde{D}^\top & I \\ * & I & 0 \\ * & * & W \end{bmatrix} \succeq 0, && \text{(9b)} \\ & P \succ 0 \end{aligned}$$

*with given* $A$ *related to the known part of the system dynamics in* (1)*,* $\tilde{T} := \begin{bmatrix} S & R & -P \end{bmatrix}$*, and* $\tilde{D}$ *the Cholesky decomposition of the data matrix* $D$ *defined in* (3) *(i.e.,* $D = \tilde{D}^\top\tilde{D}$*). Denote the optimizers of* (9) *as* $P^\star$*,* $S^\star$*,* $R^\star$*, and* $W^\star$*. Then, the following parameters of the model* (2)*,* $S_{\eta_l} = I, \Theta_l = \Theta_l^\star = {P^\star}^{-1}S^\star, B_l = B_l^\star = {P^\star}^{-1}R^\star$ *guarantee asymptotic stability of the model in* (2)*. In addition, it holds that the cost* $J$ *of* (8) *satisfies* $J \leq tr(W)$*; as such* (9) *represents an approximate convexified problem of the problem in* (8)*.*

***Proof:*** See [14, Thm. 1]. ■

**Remark 1 (Surrogate Convex Optimization with Modified Cost)** *We remark that the semi-definite program in* (9) *is not equivalent to the non-convex optimization problem in* (8) *(i.e., it is a convex approximation) due to setting* $S_{\eta_l} = I$ *and using a sufficient condition (a lower bound) in the derivation of the LMI in* (9b)*. Although by letting* $S_{\eta_l} = I$*, we do not use the known structure of uncertainty, this makes the problem tractable. Note that here, we do not use knowledge of uncertainty structure.*

Next, we follow a different approach to formulate an alternative surrogate (approximate) convex optimization problem for Problem 1.

### B. Constraint Modification Approach

Instead of changing the model-related variable ($\Theta_l$) in the stability constraint (7) (or in its equivalent (6)), we formulate a sufficient condition (an upper bound) for the stability constraint (7) which is linear in all the optimization parameters in order to convexify the optimization problem (8). The following theorem formalizes this approach.

**Theorem 2 (Stable Model Learning with Modified Constraint)** *Consider system* (1)*, a given data-set of input and (estimated) uncertainty and state realizations and the uncertainty model of the form in* (2)*. Consider the following convex program:*

$$\begin{aligned} \min_{Q,\Theta_l,B_l,W} \quad & tr(W) \\ \text{s.t.} \quad & \begin{bmatrix} AQ + QA^\top & S_\eta\Theta_l + \bar{\gamma}Q \\ \star & -2\bar{\gamma}I \end{bmatrix} \prec 0, && \text{(10a)} \\ & tr(TDT^\top) \leq tr(W), && \text{(10b)} \\ & Q \succ 0 \end{aligned}$$

*with given Hurwitz* $A$*,* $S_\eta$ *related to known parts of the system in* (1)*, positive scalar* $\bar{\gamma}$*, and* $D$ *and* $T$ *as defined in* (3)*, and* (4)*, respectively. Denote the optimizers of* (10) *as* $Q^\star$*,* $\Theta_l^\star$*,* $B_l^\star$*, and* $W^\star$*. Then, the following parameters of the model* (2)*,* $S_{\eta_l} = S_\eta, \Theta_l = \Theta_l^\star, B_l = B_l^\star$ *guarantee asymptotic stability of the model in* (2)*. In addition, it holds that the cost* $J$ *of* (8) *satisfies* $J \leq tr(W)$*.*

***Proof:*** See [14, Thm. 2]. ■

**Remark 2 (Surrogate Convex Optimization with Modified Constraint)** *Similar to Theorem 1, the semi-definite program in* (10) *is a convex approximation of the non-convex optimization problem in* (8) *since the stability constraint* (10a) *is a sufficient condition for asymptotic stability of the model in* (2)*. Note that, a disadvantage of Theorem 2 compared to Theorem 1 is that to ensure the feasibility of the semi-definite problem in Theorem 2, the known* $A$ *matrix of the system in* (1) *has to be Hurwitz. On the other hand, unlike Theorem 1, Theorem 2 uses the knowledge of uncertainty structure by setting* $S_{\eta_l} = S_\eta$*, which is potentially beneficial.*

In the above, we assumed that state and uncertainty realizations are available, which is, in practice typically not the case. In what follows, we present a solution for uncertainty and state estimation based on only input and output data.

## IV. Uncertainty and State Estimation

First, we formulate the uncertainty and state estimation problem before providing a solution for that problem. Consider system in (1) and the required assumptions as below to ensure that the problem is well-posed.

**Assumption 1 (Regularity)** *The following assumptions are required to ensure the regularity of the uncertainty and state estimation problem, which stand throughout this section:*

- **State and Input Boundedness:** *The state variable $x_s(t)$ and the input $u(t)$ remain bounded in some compact region of interest.*
- $\mathcal{C}^r$ **Uncertainty Vector:** *The uncertainty vector $\eta(x_s(t), u(t))$ in (1) is $r$ times differentiable with respect to time, i.e., the time derivatives $\eta^{(1)}(x_s(t), u(t))$, $\eta^{(2)}(x_s(t), u(t))$, ... ,$\eta^{(r)}(x_s(t), u(t))$ exist and are continuous, and $\eta^{(r)}(x_s(t), u(t))$ is uniformly bounded.*
- **Disturbance Boundedness:** *The disturbance vector $\omega(t)$ in (1) is bounded uniformly in $t$.*
- $\mathcal{C}^1$ **Measurement Noise:** *The measurement noise vector $\nu(t)$ in (1) is bounded uniformly in $t$ and differentiable, i.e., the total derivative with respect to time $\dot{\nu}(t)$ exists, is continuous, and bounded uniformly in $t$.*

We assume input $u$ and measured output $y_s$ vector-valued signals in (1) are available. The following filter is designed for uncertainty and state estimation:

$$\begin{cases} \dot{z} = h(z, u, y_s; \theta), \\ \hat{\eta} = \phi_1(z, y_s; \theta), \\ \hat{x}_s = \phi_2(z, y_s; \theta), \end{cases} \tag{11}$$

where $z \in \mathbb{R}^{n_z}$ is the internal state of the filter with $n_z \in \mathbb{N}$. Functions $h : \mathbb{R}^{n_z} \times \mathbb{R}^l \times \mathbb{R}^m \to \mathbb{R}^{n_z}$, $\phi_1 : \mathbb{R}^{n_z} \times \mathbb{R}^m \to \mathbb{R}^{n_\eta}$, and $\phi_2 : \mathbb{R}^{n_z} \times \mathbb{R}^m \to \mathbb{R}^n$ characterize the filter structure, $\theta$ denotes design parameters.

Define $\hat{x}_d := \mathrm{col}[\hat{\eta}, \hat{x}_s]$ (representing the estimate of both the uncertainty and the state) and its estimation error as

$$e_d := \hat{x}_d - x_d, \tag{12}$$

where $x_d := \mathrm{col}[\eta, x_s]$. The error dynamics of the filter is given later as a linear system and it is shown that $e_d = e_d(\omega, \eta^{(r)}, \nu, \dot{\nu})$. With this, we can state the uncertainty and state estimation problem.

**Problem 2 (Uncertainty and State Estimation)** *Consider the system* (1) *with known input and output signals, $u(t)$ and $y_s(t)$, and the uncertainty-state estimator filter* (11). *For given $r$, design the filter parameters $\theta$ such that the following properties are guaranteed:*
**1) Stability:** *The estimation error dynamics is input-to-state stable with respect to the perturbation input $(\omega, \eta^{(r)}, \nu, \dot{\nu})$;*
**2) Disturbance Attenuation:** *The $H_\infty$-norm of the transfer function from col$[\omega, \eta^{(r)}]$ to $e_d$ in* (12) *is bounded by some known $\lambda > 0$;*
**3) Noise Rejection:** *The $H_2$-norm of the transfer function from col$[\nu, \dot{\nu}]$ to $e_d$ in* (12) *is bounded by some known $\gamma > 0$.*

For a more formal formulation of the aforementioned problem, refer to [14]. Before presenting the solution for Problem 2, we discuss the uncertainty-state estimator filter architecture, in what follows.

### A. Ultra Local Uncertainty Representation

Under Assumption 1, it is discussed in [14] that by considering Taylor series, the actual internal state-space representation of $\eta$ in (1) is as follows:

$$\begin{cases} \dot{\zeta}_j = \zeta_{j+1}, \qquad 0 < j < r, \\ \dot{\zeta}_r = \eta^{(r)}, \\ \eta = \zeta_1, \end{cases} \tag{13}$$

where $\zeta_j \in \mathbb{R}^{n_\eta}$. In the following, to design the uncertainty-state estimator we augment the system state, $x_s(t)$, with the states of the actual uncertainty internal state $\zeta_j(t), j \in \{1, \ldots, r\}$, and augment the system dynamics in (1) with (13). We then design a linear filter (observer) for the augmented system to simultaneously estimate $x_s$ and $\zeta_j$. We remark that proper selection of the number of the uncertainty derivatives, $r$, added to the uncertainty internal representation (13) is problem-dependent, see [15] for discussion on selection of $r$.

### B. Augmented Dynamics

Based on the uncertainty internal representation in (13) introduced above, define the augmented state $x_a := \mathrm{col}[x_s, \zeta_1, \zeta_2, \ldots, \zeta_r]$, and rewrite the augmented dynamics using (1) and (13) as

$$\begin{cases} \dot{x}_a = A_a x_a + B_{u_a} u_a + B_{\omega_a} \omega_a, \\ y_s = C_a x_a + D_\nu \nu, \end{cases} \tag{14a}$$

$$A_a := \begin{bmatrix} A & S_\eta & 0 \\ 0 & 0 & I_{d_n} \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{u_a} := \begin{bmatrix} B_u^\top & 0 \end{bmatrix}^\top, \quad u_a := u,$$
$$B_{\omega_a} := \begin{bmatrix} B_\omega & 0 \\ 0 & 0 \\ 0 & I_{n_\eta} \end{bmatrix}, \quad \omega_a := \begin{bmatrix} \omega \\ \eta^{(r)} \end{bmatrix}, \quad C_a := \begin{bmatrix} C & 0 \end{bmatrix} \tag{14b}$$

with $d_n := (r-1)n_\eta$.

### C. Uncertainty-State Estimator

In this section, considering the uncertainty-state estimator general structure in (11), inspired from observer-based approaches, we consider $h(\cdot)$ and $\phi_i(\cdot), i = 1, 2$, as

$$\begin{aligned} h(z, u, y_s; \theta) =& Nz + Gu + Ly_s, \\ \phi_i(z, y_s; \theta) =& \bar{C}_i(z - Ey_s), \end{aligned} \tag{15a}$$

with $\hat{x}_a = z - Ey_s$, filter state $z \in \mathbb{R}^{n_z}$, $n_z = n + rn_\eta$,

$$\bar{C}_1 := \begin{bmatrix} 0 & I_{n_\eta} & 0 \end{bmatrix}, \quad \bar{C}_2 := \begin{bmatrix} I_n & 0 \end{bmatrix},$$

and matrices $(N, G, L)$ defined as

$$\begin{aligned} N &:= MA_a - KC_a, \quad M := I + EC_a, \\ G &:= MB_a, \qquad\qquad L := K(I + C_aE) - MA_aE. \end{aligned} \tag{15b}$$

Matrices $E$ and $K$ are filter gains to be designed which can be collected as $\theta = (E, K)$. Note that according to (15a), the part of the augmented state, $x_a$, that we use to reconstruct uncertainty and state signals is $\bar{C}_a x_a$ with

$$\bar{C}_a := \begin{bmatrix} \bar{C}_1^\top & \bar{C}_2^\top \end{bmatrix}^\top. \tag{16}$$

In the following section, we analyze the estimator error dynamics.

### D. Uncertainty-State Estimator Error Dynamics

Consider the augmented state estimate $\hat{x}_a$ and let us define estimation error as $e := \hat{x}_a - x_a = z - x_a - Ey_s$. Then, given the algebraic relations in (15b), the estimation error dynamics can be written as

$$\dot{e} = Ne - MB_{\omega_a}\omega_a + \begin{bmatrix} KD_\nu & -ED_\nu \end{bmatrix} \begin{bmatrix} \nu \\ \dot{\nu} \end{bmatrix}.$$

Define $\nu_a := \mathrm{col}[\nu, \dot{\nu}]$, $e_d := \bar{C}_a e$ with $\bar{C}_a$ as in (16), and $B_{\nu_a} := \begin{bmatrix} KD_\nu & -ED_\nu \end{bmatrix}$. Then, the estimation error dynamics is given by

$$\begin{cases} \dot{e} = Ne - MB_{\omega_a}\omega_a + B_{\nu_a}\nu_a, \\ e_d = \bar{C}_a e. \end{cases} \tag{17}$$

Define the transfer matrices

$$\begin{aligned} T_{e_d\omega_a}(s) &:= -\bar{C}_a(sI - N)^{-1} MB_{\omega_a}, \\ T_{e_d\nu_a}(s) &:= \bar{C}_a(sI - N)^{-1} B_{\nu_a}, \end{aligned} \tag{18}$$

where $T_{e_d\omega_a}(s)$ and $T_{e_d\nu_a}(s)$, with $s \in \mathcal{C}$, denote the corresponding transfer matrices from $\omega_a$ and $\nu_a$, both to $e_d$, respectively. Now, we can restate Problem 2 in a more formal way.

### E. Uncertainty-State Estimator Design

In the following proposition, we provide the solution of Problem 2 as a semi-definite problem, where we seek to minimize the $H_\infty$-norm of $T_{e_d\omega_a}(s)$ for an acceptable upper bound on the $H_2$-norm of $T_{e_d\nu_a}(s)$ (there exist a trade-off between these two norms, see [16], [17]). Moreover, we add the Input-to-State Stability (ISS) constraint with respect to filter error dynamics input $\mathrm{col}[\omega_a, \nu_a]$ to this program to enforce that stability of the resulting estimation filter.

**Proposition 1 (Estimator Design)** *Consider the system* (1)*, the augmented dynamics* (14)*, the uncertainty-state estimator* (11) *with $h(\cdot)$ and $\phi(\cdot)$ as defined in* (15)*, the corresponding estimation error dynamics* (17)*, and the transfer functions* (18)*. Consider the following convex program:*

$$\begin{aligned} \min_{\Pi, F, H, Z, \lambda, \gamma} \quad & \lambda \\ \text{s.t.} \quad & \bar{S} + \epsilon I \preceq 0, \\ & \begin{bmatrix} \bar{S} & -(\Pi + FC_a)B_{\omega_a} & \bar{C}_a^\top \\ * & -\lambda I & 0 \\ * & * & -\lambda I \end{bmatrix} \prec 0, \\ & \begin{bmatrix} \bar{S} & HD_\nu & -FD_\nu \\ * & -\gamma I & 0 \\ * & * & -\gamma I \end{bmatrix} \prec 0, \\ & \begin{bmatrix} \Pi & \bar{C}_a^\top \\ * & Z \end{bmatrix} \succ 0, \\ & \Pi \succ 0, \\ & \gamma - \mathrm{trace}(Z), \gamma, \lambda > 0, \\ & \gamma \leq \gamma_{max} \end{aligned}$$

*with* $\bar{S} := A_a^\top \Pi + A_a^\top C_a^\top F^\top - C_a^\top H^\top + \Pi A_a + FC_aA_a - HC_a$, *given* $\epsilon, \gamma_{max} > 0$, $\bar{C}_a$ *in* (16)*, and the remaining matrices as defined in* (14b)*. Denote the optimizers as* $\Pi^\star, F^\star, H^\star, Z^\star, \lambda^\star$, *and* $\gamma^\star$*. Then, the optimal parameters in* (15) $\theta = \theta^\star = \{E^\star = \Pi^{\star^{-1}} F^\star, K^\star = \Pi^{\star^{-1}} H^\star\}$ *guarantee the following properties:*

1) *The estimation error dynamics in* (17) *is ISS and the ISS-gain from input* $\mathrm{col}[\omega_a, \nu_a]$ *to the estimation error is upper bounded by* $2\|\Pi^\star \begin{bmatrix} (I + E^\star C_a)B_{\omega_a} & -K^\star D_\nu & E^\star D_\nu \end{bmatrix}\|\epsilon^{-1}$.
2) $\|T_{e_d\omega_a}\|_\infty := \sup_{\alpha \in \mathbb{R}^+} \sigma_{\max}(T_{e_d\omega_a}(i\alpha))$ *is upper bounded by* $\lambda^\star$.
3) $\|T_{e_d\nu_a}\|_{H_2} = \sqrt{\frac{1}{2\pi}\,\mathrm{trace} \int_{-\infty}^{\infty} T_{e_d\nu_a}(i\alpha) T_{e_d\nu_a}^H(i\alpha)\, \mathrm{d}\alpha}$ *is upper bounded by* $\gamma^\star$.

***Proof:*** See [18, Thm. 1]. ■

## V. Simulation Results

In this section, we evaluate the proposed method using a two-mass-spring-damper system. By defining the state vector $x_s = [x_{s_1}, x_{s_2}, x_{s_3}, x_{s_4}]^\top := [q_1, \dot{q}_1, q_2, \dot{q}_2]^\top$, where $q_i$ and $\dot{q}_i$ are the displacement and velocity of the $i-$th mass, respectively, the system dynamics can be described as:

$$\begin{cases} \dot{x}_s = Ax_s + B_u u + S_\eta \eta(x_s), \\ \eta(x_s) = \Theta_a x_s, \\ y_s = Cx_s + D_\nu \nu, \end{cases} \tag{20}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_1+k_2}{m_1} & -\frac{c_1+c_2}{m_1} & \frac{k_2}{m_1} & \frac{c_2}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_2}{m_2} & \frac{c_2}{m_2} & -\frac{k_2}{m_2} & -\frac{c_2}{m_2} \end{bmatrix}, \quad D = I,$$

$$B_u = \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix}, \quad \Theta_a = \begin{bmatrix} -\frac{\delta k_1 + \delta k_2}{m_1} & \frac{\delta k_2}{m_2} \\ 0 & 0 \\ \frac{\delta k_2}{m_1} & -\frac{\delta k_2}{m_2} \\ 0 & 0 \end{bmatrix}^\top,$$

$$S_\eta = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^\top, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

and constants $m_i$, $k_i$, and $c_i$ are the mass, stiffness, and viscous coefficient of the $i$-th mass, spring and damper,

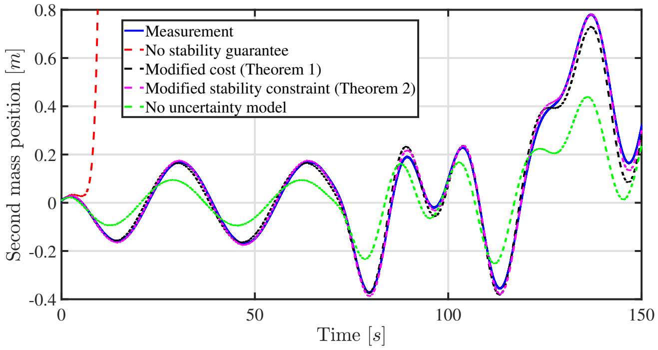


Fig. 1. Comparison of second output of system and different models.

TABLE I
RMSEs OF DIFFERENT MODELS AND SYSTEM OUTPUTS.

| RMSE [m] | First mass position | Second mass position |
|---|---|---|
| No uncertainty model | 0.0296 | 0.1272 |
| Cost modification approach | 0.0085 | 0.0303 |
| Constraint modification approach | 0.0077 | 0.0117 |

respectively. The uncertainty is due to the uncertainty on the springs stiffness which are captured by $\delta k_i$ for the $i$-th spring. Input $u$ is the force which applies to the first mass. The parameters values are: $m_1 = 4\ kg, m_2 = 3\ kg, k_1 = 2\ N/m, k_2 = 1.5\ N/m, c_1 = 3.4\ Ns/m, c_2 = 3.8\ Ns/m, \delta k_1 = 0.25k_1, \delta k_2 = -0.2k_2$. For simulation, we set initial conditions as $x_s(0) = [0.01, 0.01, 0.01, 0.01]^T$.

For the above-mentioned system, $\Theta_a$ is unknown, and three estimations for it are trained using the proposed methods without imposing stability constraints. Then, for a test data-set, we have compared the output of system with extended models (which consist of the known model plus one of the uncertainty models) and also with the model without any uncertainty model. Figure 1 depicts this comparison for the second mass position. It can be seen that the result with uncertainty model which is trained without any stability constraint is unstable, see red dashed line. This shows that considering stability condition while learning a model for a stable system is indeed necessary. Figure 1 also shows that using the learning strategy proposed in this paper, model quality is significantly improved compared to the model without learned uncertainty model.

Furthermore, for better comparison, the Root Mean Square Errors (RMSEs) of error of each (stable) model (difference of model and system outputs) are given in Table I. As the results show, the constraint modification approach (Theorem 2) outperforms for this example. Note that we cannot generalize better performance of constraint modification approach in comparison with cost modification approach since we only show the results for one case study here.

## VI. Conclusion

This paper proposes a framework for learning of modeling uncertainties in linear time invariant models. Simulations of a two-mass-spring-damper system demonstrate the proposed approach's effectiveness and potential. Future work could include extending the method to nonlinear systems.

## Acknowledgment

This publication is part of the project Digital Twin project 4.3 with project number P18-03 of the research programme Perspectief which is (mainly) financed by the Dutch Research Council (NWO).

## References

[1] M. Revay, R. Wang, and I. R. Manchester, "Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness," *IEEE Transactions on Automatic Control*, 2023.
[2] A. Abbasi and C. Nataraj, "Physics-informed machine learning for uncertainty reduction in time response reconstruction of a dynamic system," *IEEE Internet Computing*, vol. 26, no. 4, pp. 35–44, 2022.
[3] W. Quaghebeur, I. Nopens, and B. De Baets, "Incorporating unmodeled dynamics into first-principles models through machine learning," *IEEE Access*, vol. 9, pp. 22 014–22 022, 2021.
[4] J. Umenberger, J. Wågberg, I. R. Manchester, and T. B. Schön, "Maximum likelihood identification of stable linear dynamical systems," *Automatica*, vol. 96, pp. 280–292, 2018.
[5] M. M. Tobenkin, I. R. Manchester, J. Wang, A. Megretski, and R. Tedrake, "Convex optimization in identification of stable non-linear state space models," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 7232–7237.
[6] M. H. Abbasi, L. Iapichino, W. Schilders, and N. van de Wouw, "A data-based stability-preserving model order reduction method for hyperbolic partial differential equations," *Nonlinear Dynamics*, vol. 107, no. 4, pp. 3729–3748, 2022.
[7] I. R. Manchester, M. Revay, and R. Wang, "Contraction-based methods for stable identification and robust machine learning: a tutorial," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2955–2962.
[8] T. Van Gestel, J. A. Suykens, P. Van Dooren, and B. De Moor, "Identification of stable models in subspace identification by using regularization," *IEEE Transactions on Automatic control*, vol. 46, no. 9, pp. 1416–1420, 2001.
[9] D. N. Miller and R. A. De Callafon, "Subspace identification with eigenvalue constraints," *Automatica*, vol. 49, no. 8, pp. 2468–2473, 2013.
[10] S. L. Lacy and D. S. Bernstein, "Subspace identification with guaranteed stability using constrained optimization," *IEEE Transactions on automatic control*, vol. 48, no. 7, pp. 1259–1263, 2003.
[11] G. Pillonetto and G. De Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.
[12] M. Scandella, M. Mazzoleni, S. Formentin, and F. Previdi, "Kernel-based identification of asymptotically stable continuous-time linear dynamical systems," *International Journal of Control*, vol. 95, no. 6, pp. 1668–1681, 2022.
[13] H. K. Khalil, "Nonlinear systems third edition," *Prentice Hall*, vol. 115, 2002.
[14] F. Ghanipoor, C. Murguia, P. Mohajerin Esfahani, and N. van de Wouw, "Uncertainty learning for lti systems with stability guarantees," *available at arXiv:2310.20568*, 2023.
[15] ——, "Ultra local nonlinear unknown input observers for robust fault reconstruction," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 918–923.
[16] M. Khosrowjerdi, R. Nikoukhah, and N. Safari-Shad, "Fault detection in a mixed $h_2/h_\infty$ setting," *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 1063–1068, 2005.
[17] P. P. Khargonekar, M. A. Rotea, and E. Baeyens, "Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ filtering," *International Journal of Robust and Nonlinear Control*, vol. 6, no. 4, pp. 313–330, 1996.
[18] F. Ghanipoor, C. Murguia, P. Mohajerin Esfahani, and N. van de Wouw, "Linear fault estimators for nonlinear systems: An ultra-local model design," *available at arXiv:2211.06214*, 2022.