Model Updating for Nonlinear Systems with Stability Guarantees ⋆

Farhad Ghanipoor [a], Carlos Murguia [a,b], Peyman Mohajerin Esfahani [c], Nathan van de Wouw [a]

[a] *Mechanical Engineering Department, Eindhoven University of Technology, The Netherlands*

[b] *Engineering Systems Design, Singapore University of Technology and Design, Singapore*

[c] *University of Toronto, Canada and Delft University of Technology, The Netherlands*

Abstract

To improve the predictive capacity of system models in the input-output sense, this paper presents a framework for model updating via learning of modeling uncertainties in locally and globally Lipschitz nonlinear systems. First, we introduce a method to extend an existing known model with an uncertainty model so that stability of the extended model is guaranteed in the sense of set invariance and input-to-state stability. To achieve this, we provide two tractable semi-definite programs. These programs allow obtaining optimal uncertainty model parameters for both locally and globally Lipschitz nonlinear models, given uncertainty and state trajectories. Subsequently, in order to extract this data from the available input-output trajectories, we introduce a filter that incorporates an approximated internal model of the uncertainty and asymptotically estimates uncertainty and state realizations. This filter is also synthesized using semi-definite programs with guaranteed robustness with respect to uncertainty model mismatches, disturbances, and noise. Numerical simulations for a large data-set of a roll plane model of a vehicle illustrate the effectiveness and practicality of the proposed methodology in improving model accuracy, while guaranteeing stability.

Key words: unmodeled dynamics; machine learning; input-to-state stability; set invariance; ultra local model;

1 Introduction

Dynamical systems modeling has been a key problem in many engineering and scientific fields, such as biology, physics, chemistry, and transportation. When modeling dynamical systems, it requires well-established principles of physics and known prior system properties (e.g., stability and set-invariance) [1]. However, for many complex practical systems, we only tend to have partial knowledge of the physics governing their dynamics [2]. Even in cases where accurate physics-based models are established, as, e.g., in robotics, there exist inevitable (both parametric and non-parametric) uncertainties that impact the model's predictive accuracy.

For a class of nonlinear dynamical systems, this paper assumes a prior system model is available (however, the results are also applicable when no prior model is available). It focuses on learning models for uncertainties while guaranteeing the stability of extended models (prior models plus uncertainty representations), given available input-output data. This problem contrasts with black box modeling approaches, such as Neural Networks (NNs), because we incorporate prior relationships derived from first principles into both the modeling and learning framework. Moreover, this problem differs from typical grey-box identification problems because, in our case, a prior model with known parameters is available. Off-the-shelf grey-box system identification methods can cope with a subset of the problem under discussion, where no prior model is available. The problem considered here enables the characterization of system representations by comprising both prior (physics-based) and uncertainty learned models. In what follows, a comparison of our approach with some related existing literature on hybrid modeling (based on

⋆ Corresponding author F. Ghanipoor.

Email addresses: f.ghanipoor@tue.nl (Farhad Ghanipoor), C.G.Murguia@tue.nl, murguia_rendon@sutd.edu.sg (Carlos Murguia), P.MohajerinEsfahani@tudelft.nl (Peyman Mohajerin Esfahani), N.v.d.Wouw@tue.nl (Nathan van de Wouw).

both first-principles and data) is provided.

Existing Literature: Our approach augments a known (physics-based) model by a black-box model used as a correction term, see, e.g., [3, 4] and references therein. Such generic approach is also taken by Quaghebeur et al. in [5], where an NN model is added 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, simulating the hybrid model at each iteration during the training process is necessary. This approach, depending on the size of the NN model, may be more computationally intensive compared the method proposed here, which eliminates the requirement of simulating the model in every iteration. Moreover, the main drawback of the method in [5] is the assumption that the initial state of the dynamic system is known, which may not hold in practical applications. This assumption is dropped in the method proposed here. Furthermore, methods such as those in [5] do not provide stability guarantees, which are a core feature of our approach.

Our approach is fundamentally different from existing Physics-Informed (PI) learning techniques where standard black box models (such as NNs) are trained constrained to satisfy physics-based relations [6–8]. Yazdani et al. in [6] use this technique to construct so-called Physics Informed Neural Networks (PINNs). Although this method provides a NN as a system model (as well as parameters for physics-based models), it does not give a closed-form expression for the uncertainty in known physics-based models (i.e., it does not account for modeling mismatches due to unmodeled physics). Furthermore, parameters of both physics-based models and NNs are learned simultaneously, which increases the computational burden.

The approach proposed in this paper also differs from the so-called Sparse Identification of Nonlinear Dynamics (SINDy) scheme [9]. SINDy assumes full knowledge of system states and their time derivatives. This approach has demonstrated accurate performance in sparse model identification of complex nonlinear systems [10–12]. However, SINDy not only requires full-state measurement but also requires the derivative of states to be known. Although the state derivatives can be approximated numerically if the complete state is known, most numerical methods are noise sensitive. Furthermore, the requirement of full-state measurements is a strong assumption for most dynamical systems to begin with. In our work, we do not require measurements of the full-state and its time derivative. The proposed algorithms need input-output data only.

Another important advantage of the proposed method is guaranteeing the stability of the extended nonlinear model (i.e., the model consisting of the prior known model and the uncertainty model). The identification of stable models has (mainly) been widely studied in the context of discrete LTI systems [13, 14]. For further results on uncertainty learning for LTI systems, refer to [15]. However, the identification of nonlinear stable models is still under study, mainly focusing on identifying the complete dynamics in a black box fashion. For instance, kernel- or Koopman-based methods that enforce some form of model stability during the learning process are proposed in [16, 17] for the autonomous case and for the non-autonomous case in [18]. Moreover, there are results that aim to enforce model stability in other types of models, such as recurrent equilibrium network models [19] and Lur'e-type models [20, 21]. However, it is important to note that none of these methods can be directly applied to the specific problem we are addressing here, given the *prior known model*.

In this paper, we propose a framework for model updating via learning modeling uncertainties in (physics-based) models applicable to locally Lipschitz nonlinear systems. We first focus on learning uncertainty models, assuming that some realizations of input, estimated uncertainty, and estimated state are given. During uncertainty learning, we guarantee that trajectories of the extended model (i.e., prior known model plus uncertainty model) belong to a given invariant set for locally Lipschitz extended models, or ensure input-to-state stability for globally Lipschitz extended models. This is achieved by formulating the problem as a constrained supervised learning problem.

One key challenge in this problem involves the introduction of stability constraints, which is tackled using Lyapunov-based tools. The stability criteria usually result in an optimization problem that is non-convex. We address this challenge by proposing convex approximated programs for both locally and globally Lipschitz models. This convexification, referred to as the *cost modification approach*, involves a change of variables in the cost function (Theorems 1 and 2 for locally and globally Lipschitz models, respectively). For the sake of completeness and comparison, we also provide a sequential algorithm that alternates the use of some variables in the optimization problem to convexity the program. However, we show in the numerical section that (as it is known in related literature [22, Sec. 6.5]) that initializing this algorithms is challenging, which further strengthens the importance of the results provided here. We refer to the above mentioned sequential algorithm as the method of coordinate descent.

After addressing the challenge of non-convexity, the paper proceeds to discuss the practical implementation of the framework. It outlines a method for estimating uncertainty and state trajectories using input-output data and the prior known model. In this context, uncertainty is considered as an unknown input affecting the system dynamics and the estimation of uncertainty and state

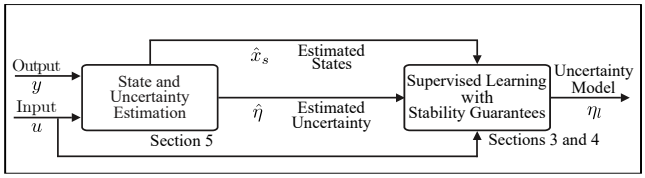

Fig. 1. Overview of the MUNSyS methodology.

trajectories is achieved using robust state and unknown input observers [23, 24]. For a schematic overview of the proposed methodology, Model Updating for Nonlinear Systems with Stability guarantees (MUNSyS), see Figure 1.

In summary, the main contributions of this paper are as follows:

(a) ***Practically Applicable Framework:*** The proposed framework for learning modeling uncertainties in locally (and globally) Lipschitz nonlinear models only requires input-output data, in contrast to methods based on SINDy [9, 11].
(b) ***Stability Guarantee of Extended Models:*** The proposed framework guarantees the following: 1. For locally Lipschitz extended models, trajectories of the extended model (i.e., prior known model plus uncertainty model) belong to a given invariant set; 2. For globally Lipschitz models, it ensures Input-to-State Stability (ISS).
(c) ***Convex Approximations for Non-Convex Programs:*** Methods for both locally and globally Lipschitz models are proposed to offer tractable convex approximations of formulated non-convex optimization problems for uncertainty model learning, while ensuring stability guarantees.

This paper generalizes the preliminary results published in the conference paper [15]. In comparison to [15], which is applicable to LTI systems only, we present results for a broader class of systems, considering both locally and globally Lipschitz nonlinear systems. The results in [15] are a subclass of the results provided for globally Lipschitz nonlinear systems (see Theorem 2). Moreover, we would like to clarify that [23] focuses on fault estimation, while this paper addresses model updating by proposing a method that separates uncertainty and state estimation from model fitting. Although different methods could be used for uncertainty and state estimation, we choose to develop a particular method here, based on the work in [23], such that the paper is fully self-contained.

Notation: The set of nonnegative real numbers is represented by the symbol $\mathbb{R}^+$. The identity matrix of size $n \times n$ is denoted as I_n or simply I when the context specifies n. Similarly, matrices of dimensions $n \times m$ comprising only zeros are denoted as $0_{n\times m}$ or 0 when the dimensions are clear. The first and second time-derivatives of a vector x are denoted as $\dot{x}$ and $\ddot{x}$, respectively. For the r^{th}-order time-derivatives of vector x, the notation $x^{(r)}$ is employed. A positive definite matrix is symbolized by $X \succ 0$, and positive semi-definite matrices are indicated by $X \succeq 0$. Similarly, $X \prec 0$ is used for negative definite matrices, and $X \preceq 0$ for negative semi-definite matrices. The Hadamard (element-wise) power of $n \in \mathbb{N}$ for a matrix X is denoted by $X^{\circ n}$. The same notation is used for vectors. The $\exp(x)$ shows the element-wise exponential function. The notation $(x_1, \ldots, x_n)$ signifies the column vector composed of elements $x_1, \ldots, x_n$, and this notation is also used when the components x_i are vectors. Both Euclidean norm and the matrix norm induced by Euclidean norm are represented by the notation $||\cdot||$. The infinity-norm is denoted as $||\cdot||_\infty$. We employ the notation $\mathcal{L}_2(0,T)$ (or simply $\mathcal{L}_2$) to represent vector-valued functions $z : [0,T] \to \mathbb{R}^k$ satisfying $||z(t)||^2_{\mathcal{L}_2} := \int_0^T ||z(t)||^2 dt < \infty$. For a vector-valued signal z defined for all $t \geq 0$, the $\mathcal{L}_\infty$ norm is denoted as $||z||_{\mathcal{L}_\infty} := \sup_{t\geq 0} ||z(t)||$. For a differentiable function $V : \mathbb{R}^n \to \mathbb{R}$, the row-vector of partial derivatives is denoted as $\frac{\partial V}{\partial x}$, and $\dot{V}(x)$ denotes the total derivative of $V(x)$ with respect to time (i.e., $\frac{\partial V}{\partial x}\frac{dx}{dt}$). The trace of a matrix W is denoted as $tr(W)$. A continuous function $\bar{\alpha} : [0,a) \to [0,\infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\bar{\alpha}(0) = 0$. A continuous function $\bar{\beta} : [0,a) \times [0,\infty) \to [0,\infty)$ is said to belong to class $\mathcal{KL}$ if, for each fixed s, the mapping $\bar{\beta}(r,s)$ belongs to class $\mathcal{K}$ with respect to r and, for each fixed r, the mapping $\bar{\beta}(r,s)$ is decreasing with respect to s and $\bar{\beta}(r,s) \to 0$ as $s \to \infty$. Time dependencies are often omitted for simplicity in notation.

2 Problem Formulation

Consider the nonlinear system

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

where $x_s \in \mathbb{R}^n$, $y_s \in \mathbb{R}^m$, and $u \in \mathbb{R}^l$ are system state, measured output and known input vectors, respectively. Function $g : \mathbb{R}^{n_{v_g}} \times \mathbb{R}^l \to \mathbb{R}^{n_g}$ is a known nonlinear vector field, and function $\eta : \mathbb{R}^{n_{v_\eta}} \times \mathbb{R}^l \to \mathbb{R}^{n_\eta}$ denotes 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 typically with high-frequency content (e.g., related to measurement noise). Known matrices $(A, B_u, S_g, V_g, S_\eta, V_\eta, B_\omega, C, D_\nu)$ are of appropriate dimensions, with $n, m, l, n_{v_g}, n_g, n_{v_\eta}, n_\eta, n_\omega, m_\nu \in \mathbb{N}$. Moreover, we assume that the system admits an equilibrium point, which, without loss of generality, is assumed to be at the origin for $u = 0$.

In the following, we state the problem we aim to solve at a high abstraction level.

Problem 1 (Uncertainty Learning with Stability Guarantees) *Consider system* (1) *with known input and output signals, u and y_s. We aim to learn a data-based model for the uncertainty function $\eta(\cdot)$ so that the extended model composed of the known part of* (1) *and the learned uncertainty model are "stable". The objective is to construct a more accurate system model (at least applicable to trajectories close to the training data set).*

We first assume the availability of a labeled data-set containing input, estimated uncertainty, and estimated state realizations. A method to obtained this data from input-output trajectories is given in Section 5. In what follows, we outline the problem settings for both locally and globally Lipschitz systems.

2.1 Problem Settings

We consider two model classes. For both model classes, the extended model (for the system in (1)) is of the form:

$$\begin{aligned} \dot{x} &= Ax + B_u u + S_g g(V_g x, u) \\ &\quad + S_{\eta_l} \eta_l(V_\eta x, u), \\ \eta_l(V_\eta x, u) &:= \Theta_l V_\eta x + B_l u + \Theta_n h(V_\eta x, u), \end{aligned} \tag{2}$$

where $x \in \mathbb{R}^n$ is model state and function $\eta_l : \mathbb{R}^{n_{v_\eta}} \times \mathbb{R}^l \to \mathbb{R}^{n_{\eta_l}}$ is the uncertainty model that is parameterized by Θ_l, B_l and Θ_n. Function $h : \mathbb{R}^{n_{v_\eta}} \times \mathbb{R}^l \to \mathbb{R}^{n_h}$ is a given nonlinear vector filed. This function contains the vector of basis functions that serve as candidates for the nonlinearities in the uncertainty model. Design matrices are collected as $\theta = (\Theta_l, B_l, \Theta_n)$ and have appropriate dimensions, $n_{\eta_l}, n_h \in \mathbb{N}$.

2.1.1 Cost Function

Recall that, for now, we assume that estimated uncertainty and state realizations are available. Define the data vector corresponding to the i-th sample in time as

$$d_i := \begin{bmatrix} \hat{x}_i^\top V_\eta^\top & u_i^\top & \hat{\eta}_i^\top & h(V_\eta \hat{x}_i, u_i)^\top \end{bmatrix}^\top,$$

where $\hat{x}_i, u_i$, and $\hat{\eta}_i$ represent the state estimate, input, and uncertainty estimate, respectively. Vector $h(V_\eta \hat{x}_i, u_i)$ corresponds to the evaluated nonlinearity in (2) at the given realizations (of state estimation and input). Given N samples of the data vector defined above, define the data matrix D as follows:

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

Further consider the error vector between the uncertainty model and its (given) estimate as

$$e_i := \eta_l(V_\eta \hat{x}_i, u_i) - \hat{\eta}_i = T d_i$$

with

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

For learning, we use the following quadratic cost function to be minimized by properly selecting Θ_l, B_l, and Θ_n:

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

In the following sections, for two model classes (two types of basis functions in $h(\cdot)$ and known nonlinearities in $g(\cdot)$), we provide stability constraints and formulate the constrained supervised learning of uncertainty models as optimization problems.

2.1.2 Locally Lipschitz Model Class

We first consider locally Lipschitz nonlinearities in (2), for both the basis functions $h(\cdot)$ and the known vector field $g(\cdot)$. Note that given bounded estimated state and input trajectories, we can always find ellipsoids that bound these trajectories. Hereafter, the system state and input trajectories are referred to as state and input sets, respectively.

Note that the set of all trajectories that can be generated by system in (1) is not known a priori. We only have finite data realizations of the estimated state in response to some input trajectories. Given this, we seek models (2) that generate state trajectories "close" to the available data set in response to inputs "close" to the known input trajectories. To this end, we embed estimated state data and input trajectories in some known ellipsoidal sets, $\mathcal{E}_{sys}$ and $\mathcal{E}_u$, respectively. This embedding can be obtained efficiently, e.g., exploiting results in [25, Sec. 2.2.4].

Having these ellipsoids $\mathcal{E}_{sys}$ and $\mathcal{E}_u$, to induce closeness of trajectories between system training data and model trajectories, we seek to enforce during learning that all trajectories generated by the model, in response to all input trajectories contained in $\mathcal{E}_u$, are contained in some known ellipsoidal set $\mathcal{E}_{inv}$. If we enforce the latter, and $\mathcal{E}_{inv} \subseteq \mathcal{E}_{sys}$, then we can guarantee that the model produces trajectories close to the training date set (close in the sense of set inclusion within $\mathcal{E}_{sys}$), see Figure 2.

Let the ellipsoidal set containing the state estimates training data be of the form:

$$\mathcal{E}_{sys} := \left\{ x_s \mid x_s^\top F x_s \leq 1 \right\}, \tag{6}$$

with $F \succeq 0$ of appropriate dimensions, and the input set of the form:

$$\mathcal{E}_u := \left\{ u \mid u^\top U u \leq 1 \right\}, \tag{7}$$

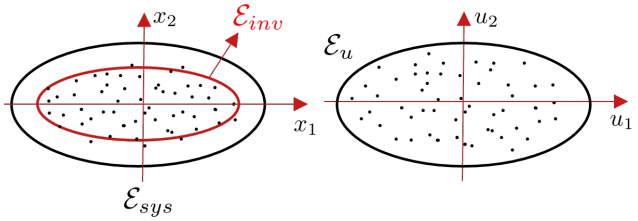

Fig. 2. Illustration of different (ellipsoidal) sets for the system state $\mathcal{E}_{sys}$, input $\mathcal{E}_u$, and model state $\mathcal{E}_{inv}$.

with $U \succeq 0$ of appropriate dimension.

Because we seek to restrict model trajectories to the sets $\mathcal{E}_u$ and $\mathcal{E}_{sys}$, and consider models in (2) with Lipschitz nonlinearities, we only require them to be Lipschitz within these sets. We formally state this in the following assumption.

Assumption 1 (Locally Lipschitz Nonlinearities) *The functions $g(V_g x, u)$ and $h(V_\eta x, u)$ in (2) are locally Lipschitz in $\mathcal{E}_u$ and $\mathcal{E}_{sys}$, i.e., there exist known positive constants $l_{g_x}, l_{g_u}, l_{h_x}$ and l_{h_u} satisfying*

$$\begin{aligned}\|g(V_g x_1, u_1) - g(V_g x_2, u_2)\| \leq\, & l_{g_x}\|V_g(x_1 - x_2)\| \\ & + l_{g_u}\|u_1 - u_2\|, \\ \|h(V_\eta x_1, u_1) - h(V_\eta x_2, u_2)\| \leq\, & l_{h_x}\|V_\eta(x_1 - x_2)\| \\ & + l_{h_u}\|u_1 - u_2\|, \end{aligned} \tag{8}$$

for all $x_1, x_2 \in \mathcal{E}_{sys} \subset \mathbb{R}^n$ and all $u_1, u_2 \in \mathcal{E}_u \subset \mathbb{R}^l$.

In what follows, we formulate conditions to guarantee that model trajectories of (2), with locally Lipschitz nonlinearities, belong, in forward time, to an ellipsoidal invariant set:

$$\mathcal{E}_{inv} := \left\{x \mid x^\top P x \leq 1\right\}, \tag{9}$$

guaranteeing $\mathcal{E}_{inv} \subseteq \mathcal{E}_{sys}$, for some positive definite matrix P. Note that if $\mathcal{E}_{sys}$ is too small, it might not be feasible to find an $\mathcal{E}_{inv}$ that is both a forward invariant set for the model and a subset of $\mathcal{E}_{sys}$. In such scenarios, $\mathcal{E}_{sys}$ should be enlarged. Conditions to ensure the latter can be formulated through Lyapunov-based stability tools and the $\mathcal{S}$-procedure [25, Sec. 2.6.3]. For the sake of readability, these conditions are derived in Appendix A.1. There, it is shown that if the following conditions hold, the ellipsoid $\mathcal{E}_{inv}$ is a forward invariant set for (2) and $\mathcal{E}_{inv} \subseteq \mathcal{E}_{sys}$:

$$\begin{bmatrix} \Delta + \beta P & P(B_u + S_{\eta_l} B_l) & 0 \\ \star & (l_{g_u} + \bar{l}_{h_u})I - \alpha U & 0 \\ \star & \star & (\alpha - \beta) I \end{bmatrix} \preceq 0, \tag{10a}$$

$$\begin{bmatrix} \gamma F - P & 0 \\ * & -\gamma + 1 \end{bmatrix} \preceq 0, \tag{10b}$$

$$l_{h_x}\|\Theta_n\| \leq \bar{l}_{h_x}, \tag{10c}$$

where the involved components in the above equation are defined as

$$\begin{aligned}\Delta :=\, & A^\top P + PA + V_\eta^\top \Theta_l^\top S_{\eta_l}^\top P + P S_{\eta_l} \Theta_l V_\eta \\ & + (l_{g_x} + l_{g_u}) P S_g S_g^\top P + (\bar{l}_{h_x} + \bar{l}_{h_u}) P S_{\eta_l} S_{\eta_l}^\top P \\ & + l_{g_x} V_g^\top V_g + \bar{l}_{h_x} V_\eta^\top V_\eta, \end{aligned} \tag{10d}$$

$$\bar{l}_{h_u} := \bar{l}_{h_x} \frac{l_{h_u}}{l_{h_x}}, \tag{10e}$$

and $\bar{l}_{h_x}, \alpha, \beta, \gamma$ are adjustable parameters. In fact, $\bar{l}_{h_x}$ characterizes the size of a set within which Θ_n is enforced to reside.

Note that the condition in (10a) is not convex in the invariant set shape matrix P and uncertainty model parameters $\theta = (\Theta_l, B_l, \Theta_n)$, resulting in a non-convex optimization problem for uncertainty learning with invariance guarantees. Now, we can state the non-convex optimization problem for the locally Lipschitz model class, aiming to find a tractable convex solution later.

Problem Setting 1 (Locally Lipschitz Model Class) *Consider a given data-set D of input, estimated uncertainty, and state realizations and let Assumption 1 be satisfied. Further consider given ellipsoidal sets $\mathcal{E}_{sys}$ and $\mathcal{E}_u$, as introduced in (6) and (7), respectively, containing the available input and state estimates data sets. Find the optimal parameters $\theta = (\Theta_l, B_l, \Theta_n)$ of the uncertainty model $\eta_l(\cdot)$ in (2) (with locally Lipschitz nonlinearities) that minimizes the cost function J in (5) constrained to the ellipsoid $\mathcal{E}_{inv}$ in (9) being a forward invariant set for the extended system model in (2). In other words, solve the non-convex optimization problem:*

$$\begin{aligned} \min_{P, \Theta_l, B_l, \Theta_n, \alpha, \gamma} \quad & J \\ \text{s.t.} \quad & (10) \text{ and } P \succ 0. \end{aligned} \tag{11}$$

In what follows, we formulate an analogue problem for the globally Lipschitz model class (that is assuming that the nonlinearities in (2) are globally Lipschitz).

Remark 1 (Comparison of Locally and Globally Lipschitz Model Classes) *The main drawback of the globally Lipschitz model class is that it covers a smaller class of nonlinear systems compared to the locally Lipschitz class. However, we can enforce a stronger stability property (input-to-state stability [26]) for globally Lipschitz nonlinearities. Notably, for this model class, knowledge of the system state and input sets is no longer required. Furthermore, in the derivation of stability conditions for globally Lipschitz case, the need for a sufficient approximation (see S-procedure tools in Appendix A.1) is eliminated. This elimination relaxes the conservatism typically introduced by the sufficient condition. Moreover, the absence of this condition reduces the number of tuning parameters required for optimizing the globally Lipschitz class. The above arguments motivate to consider both model classes.*

2.1.3 Globally Lipschitz Model Class

For this model class, the nonlinearities in (2) are assumed globally Lipschitz.

Assumption 2 (Globally Lipschitz Nonlinearities) *The functions $g(V_g x, u)$ and $h(V_\eta x, u)$ in (2) are globally Lipschitz, i.e., there exist known positive constants $l_{g_x}, l_{g_u}, l_{h_x}$ and l_{h_u} satisfying*

$$\begin{aligned}\|g(V_g x_1, u_1) - g(V_g x_2, u_2)\| \leq\, & l_{g_x}\|V_g(x_1 - x_2)\| \\ & + l_{g_u}\|u_1 - u_2\|, \\ \|h(V_\eta x_1, u_1) - h(V_\eta x_2, u_2)\| \leq\, & l_{h_x}\|V_\eta(x_1 - x_2)\| \\ & + l_{h_u}\|u_1 - u_2\|,\end{aligned} \tag{12}$$

for all $x_1, x_2 \in \mathbb{R}^n$ and all $u_1, u_2 \in \mathbb{R}^l$.

In what follows, we formulate conditions to ensure Input-to-State Stability (ISS) of the extended model in (2) satisfying Assumption 2. In the following definition, we introduce the notion of ISS for the extended model [26].

Definition 1 (Input-to-State Stability) The extended model (2) is said to be ISS with respect to the input $u(t)$ if there exist a class $\mathcal{KL}$ function $\bar{\beta}(\cdot)$ and a class $\mathcal{K}$ function $\bar{\alpha}(\cdot)$ such that for any initial state $x(t_0)$ and any bounded input $u(t)$, the solution $x(t)$ of (2) exists for all finite $t \geq t_0$ and satisfies

$$\|x(t)\| \leq \bar{\beta}\left(\|x\left(t_0\right)\|, t - t_0\right) + \bar{\alpha}(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\|). \tag{13}$$

For readability, ISS conditions for the model in (2) satisfying Assumption 2 are derived in Appendix A.2. It is shown that if the condition (10c) together with the following condition hold,

$$\Delta \prec 0, \tag{14}$$

where Δ is as defined in (10d), then, the extended model in (2) with globally Lipschitz nonlinearities is ISS with respect to the input $u(t)$.

Now, we can state the non-convex optimization problem for the globally Lipschitz model class, aiming to find a tractable convex approximation later.

Problem Setting 2 (Globally Lipschitz Model Class) *Consider a given data-set D of input, estimated uncertainty, and state realizations and let Assumption 2 be satisfied. Find the optimal parameters $\theta = (\Theta_l, B_l, \Theta_n)$ of the uncertainty model $\eta_l(\cdot)$ in (2) (with globally Lipschitz nonlinearities) that minimizes the cost function J in (5), such that the extended system model in (2) is ISS with respect to input $u(t)$. In other words, solve the non-convex optimization problem:*

$$\begin{aligned}\min_{P, \Theta_l, B_l, \Theta_n} \quad & J \\ \text{s.t.} \quad & \text{(10c)}, (14), P \succ 0.\end{aligned} \tag{15}$$

The challenge in Problem Settings 1 and 2 is that stability constraints are not convex, resulting in non-convex optimization problems. To tackle this challenge, we provide two approximate convex solutions for both problem settings. The approach for convexification is refereed to as the *cost modification approach*. As an alternative solution, we provide a procedure to solve the original non-convex problems via coordinate descent.

3 Approximate Convex Programs

We convexify the stability constraints by a change of variables and rewrite the cost function in (5) in them. First, we provide the approximate solution for Problem Setting 1, followed by the approximate solution for Problem Setting 2.

3.1 Locally Lipschitz Model Class

The following theorem formalizes the associated convex optimization problem as an approximation to the non-convex optimization problem in (11).

Theorem 1 (Stable Locally Lipschitz Model Learning) *Consider system (1), a given data-set D of input, estimated uncertainty, and state realizations. In addition, consider given ellipsoidal sets $\mathcal{E}_{sys}$ and $\mathcal{E}_u$, as introduced in (6) and (7) with shape matrices $F \succeq 0$ and $U \succeq 0$, respectively. Further consider the extended system model in (2), under Assumption 1 with Lipschitz constants $l_{g_u}, l_{g_x}, l_{h_u}$ and l_{h_x}. Consider the following*

convex program:

$$\begin{aligned}
\min_{P,S,R,Z,W,\alpha,\gamma} \quad & tr(W) \\
\text{s.t.} \quad & \begin{bmatrix} M_{11} & M_{12} & 0 & M_{14} & M_{15} \\ \star & M_{22} & 0 & 0 & 0 \\ \star & \star & M_{33} & 0 & 0 \\ \star & \star & \star & -I & 0 \\ \star & \star & \star & \star & -I \end{bmatrix} \preceq 0, && \text{(16a)} \\
& \begin{bmatrix} \bar{l}_{h_x} I & l_{h_x} Z^\top \\ * & \bar{l}_{h_x}(2\mu_1 P - \mu_1^2 I) \end{bmatrix} \succeq 0, && \text{(16b)} \\
& \begin{bmatrix} \gamma F - P & 0 \\ * & -\gamma + 1 \end{bmatrix} \preceq 0, \\
& \begin{bmatrix} 2\mu_2 P & \tilde{T}\tilde{D}^\top & \mu_2 I \\ * & I & 0 \\ * & * & W \end{bmatrix} \succeq 0, && \text{(16c)} \\
& P \succ 0, \quad \alpha, \gamma \geq 0
\end{aligned}$$

with the involved matrices defined as follows:

$$\begin{aligned}
M_1 &:= A^\top P + PA + V_\eta^\top S^\top + SV_\eta + l_{g_x} V_g^\top V_g \\
&\quad + \bar{l}_{h_x} V_\eta^\top V_\eta, \quad M_{11} := M_1 + \beta P, \\
M_{12} &:= PB_u + R, \quad M_{22} := (l_{g_u} + \bar{l}_{h_u}) I - \alpha U, \\
M_{14} &:= \sqrt{l_{g_x} + l_{g_u}} P S_g, \quad M_{15} := \sqrt{\bar{l}_{h_x} + \bar{l}_{h_u}} P, \\
M_{33} &:= (\alpha - \beta) I, \quad \tilde{T} := \begin{bmatrix} S & R & -P & Z \end{bmatrix},
\end{aligned} \tag{16d}$$

for given positive scalars $\bar{l}_{h_x}, \beta, \mu_1, \mu_2$, and $\bar{l}_{h_u}$ as defined in (10e)*, where $\tilde{D}$ is the Cholesky decomposition of the data matrix D in* (3) *(i.e., $D = \tilde{D}^\top \tilde{D}$), and the remaining matrices are the known parts of the system dynamics in* (1)*. Denote part of the optimizers of* (16) *as $P^\star$, $S^\star$, $R^\star, Z^\star$, and $W^\star$. Then, the following parameters of the extended model* (2)*, $S_{\eta_l} = I, \Theta_l = \Theta_l^\star = {P^\star}^{-1} S^\star, B_l = B_l^\star = {P^\star}^{-1} R^\star$ and $\Theta_n = \Theta_n^\star = {P^\star}^{-1} Z^\star$ guarantee that the ellipsoid $\mathcal{E}_{inv}$ in* (9) *with $P = P^\star$ is a forward invariant set for the extended system model in* (2) *and that $\mathcal{E}_{inv} \subseteq \mathcal{E}_{sys}$. In addition, it holds that the cost J of* (11) *satisfies $J \leq tr(W^\star)$. As such,* (16) *represents an approximate convexified version of the problem in* (11)*.*

Proof: The proof can be found in Appendix A.3. ■

3.2 Globally Lipschitz Model Class

The subsequent theorem formalizes the convex optimization problem, serving as an approximation to the non-convex optimization problem in (15).

Theorem 2 (Stable Globally Lipschitz Model Learning) *Consider system* (1)*, a given data-set D of input, estimated uncertainty, and state realizations. In addition, consider the extended system model in* (2)*, under Assumption 2 with Lipschitz constants $l_{g_u}, l_{g_x}, l_{h_u}$ and l_{h_x}. Consider the following convex program:*

$$\begin{aligned}
\min_{P,S,R,Z,W} \quad & tr(W) \\
\text{s.t.} \quad & \begin{bmatrix} M_1 & M_{14} & M_{15} \\ \star & -I & 0 \\ \star & \star & -I \end{bmatrix} \prec 0, && \text{(17a)} \\
& \begin{bmatrix} \bar{l}_{h_x} I & l_{h_x} Z^\top \\ * & \bar{l}_{h_x}(2\mu_1 P - \mu_1^2 I) \end{bmatrix} \succeq 0, && \text{(17b)} \\
& \begin{bmatrix} 2\mu_2 P & \tilde{T}\tilde{D}^\top & \mu_2 I \\ * & I & 0 \\ * & * & W \end{bmatrix} \succeq 0, \quad P \succ 0
\end{aligned}$$

with $M_1, M_{14}, M_{15}, \tilde{T}$ as defined in (16d)*, for given positive scalars $\bar{l}_{h_x}, \mu_1$, and μ_2, where $\tilde{D}$ is the Cholesky decomposition of the data matrix D in* (3)*, and the remaining matrices are the known parts of the system dynamics in* (1)*. Denote part of the optimizers of* (17) *as $P^\star$, $S^\star$, $R^\star, Z^\star$, and $W^\star$. Then, the following parameters of the extended model* (2)*, $S_{\eta_l} = I, \Theta_l = \Theta_l^\star = {P^\star}^{-1} S^\star, B_l = B_l^\star = {P^\star}^{-1} R^\star$ and $\Theta_n = \Theta_n^\star = {P^\star}^{-1} Z^\star$ guarantee that the extended model in* (2) *is ISS with respect to input $u(t)$. In addition, it holds that the cost J of* (15) *satisfies $J \leq tr(W^\star)$. As such,* (17) *represents an approximate convexified version of the problem in* (15)*. As a special case, when the Lipschitz constants are zero (no nonlinearity) the conditions in* (17a) *and* (17b) *reduces to*

$$A^\top P + PA + V_\eta^\top S^\top + SV_\eta \prec 0. \tag{17c}$$

Proof: The proof follows the line of reasoning of the proof of Theorem 1 and it is omitted here due to space constraints. ■

Remark 2 (Surrogate Convex Optimizations) *We remark that the semi-definite programs in* (16) *and* (17) *are not equivalent to the non-convex optimization problems in* (11) *and* (15)*, respectively (i.e., these are convex approximations). This approximation is due to setting $S_{\eta_l} = I$ and using two sufficient conditions (two lower bounds) in the derivation of LMI conditions in each of the convex programs (see details at the end of the proof of Theorem 1). 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.*

In the following section, an alternative solution for solving the non-convex problems in Problem Settings 1 and 2 is provided using a procedure based on the technique of coordinate descent.

4 Coordinate Descent Approach

A classical approach to provide a suboptimal solution for nonconvex optimization problems with a convexity structure in each variable (or coordinates) is coordinate descent. This is an iterative optimization algorithm that updates each coordinate at a time while holding the others fixed. This procedure repeats in a periodic manner, and is particularly useful when updates along individual or a subset of coordinates are easy to compute. This is indeed the structure for the problems (11) and (15). For more details about the algorithm and its different variants, we refer to [22, Sec. 6.5]. For the given non-convex optimization problems, we can fix Lyapunov function parameter P or uncertainty model parameters $\theta = (\Theta_l, B_l, \Theta_n)$. The former transforms the non-convex optimization problem(s) in (11) (or in (15)) to feasibility problem(s) and the latter to convex optimization sub-problem(s). Then, this transformed problems can be solved iteratively until the convergence of the cost(s) of the convex optimization sub-problem(s).

Next, an algorithm is presented as a sub-optimal solution for the non-convex optimization problem(s) in (11) (or in (15)). Steps 1 and 2 in this algorithm can be swapped to create a dual algorithm. In the loop below, initialization for θ is necessary, whereas for the dual loop, initialization for P is required. The algorithm consists of three steps:

Step 1: Fix θ and find a feasible solution for the transformed feasibility problems in (11) (or (15));
Step 2: Fix P as the feasible solution from Step 1 and solve the transformed sub-convex problem(s) in (11) (or (15)).
Step 3: Redo Steps 1 and 2 until the cost of the transformed sub-convex problem(s) in Step 2 converge. Denote part of the optimizers of the transformed sub-convex problem(s) with $\theta = \theta^\star$ that provides the uncertainty model parameters. These uncertainty parameters guarantee stability of the extended model in (2).

Remark 3 (Coordinate Descent and Convex Approximations Comparison) *The effectiveness of this coordinate descent method is recognized to rely on the initialization [22, Sec. 6.5].For example, in the simulation results in Section 6, no other initialization was found except the solution of the convex approximation optimizations provided in Section 3, highlighting the significance of the proposed convex programs. Moreover, depending on the optimization problem and initialization, the coordinate descent method might require many iterations for convergence, making this method expensive in terms of computations. On the other hand, the drawbacks of the convex approximation programs are that they require some tuning and, due to conservatism in their derivation, their solutions might be more sub-optimal (i.e., have a larger cost) than the coordinate descent method.*

In the results so far, 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.

5 Uncertainty and State Estimation

The following filter is designed for uncertainty and state estimation:

$$\begin{cases} \dot{z} = f(z, u, y_s; \psi), \\ \hat{\eta} = \phi_1(z, y_s; \psi), \\ \hat{x}_s = \phi_2(z, y_s; \psi), \end{cases} \tag{18a}$$

where $z \in \mathbb{R}^{n_z}$ is the internal state of the filter with $n_z \in \mathbb{N}$. Functions $f : \mathbb{R}^{n_z} \times \mathbb{R}^l \times \mathbb{R}^m \to \mathbb{R}^{n_z}$, $\phi_1 : \mathbb{R}^{n_z} \times \mathbb{R}^m \to \mathbb{R}^{n_\eta}$, and $\phi_2 : \mathbb{R}^{n_z} \times \mathbb{R}^m \to \mathbb{R}^n$ characterize the filter structure, ψ denotes design parameters. Moreover, $\hat{x}_s$ is the estimate of the state x_s of (1) and $\hat{\eta}$ is the estimate of η in (1). Inspired from observer-based approaches, we design $f(\cdot)$ and $\phi_i(\cdot), i = 1, 2$, as follows:

$$\begin{aligned} f(z, u, y_s; \psi) &= Nz + Gu + Ly_s \\ &\quad + MS_{g_a} g(V_{g_a} \hat{x}_a + H(y_s - C_a \hat{x}_a), u), \\ \phi_i(z, y_s; \psi) &= \bar{C}_i(z - Ey_s), \end{aligned} \tag{18b}$$

with $\hat{x}_a = z - Ey_s$, filter order $n_z = n + rn_\eta$, where r is a positive integer selected by the user following the instruction provided later, and the remaining matrices defined as

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

$$\begin{aligned} N &:= MA_a - KC_a, \qquad M := I + EC_a, \\ G &:= MB_{u_a}, \qquad\qquad L := K(I + C_a E) - MA_a E, \\ A_a &:= \begin{bmatrix} A & S_\eta & 0 \\ 0 & 0 & I_{d_n} \\ 0 & 0 & 0 \end{bmatrix}, \quad B_{u_a} := \begin{bmatrix} B_u \\ 0 \end{bmatrix}, \\ S_{g_a} &:= \begin{bmatrix} S_g^\top & 0 \end{bmatrix}^\top, \qquad V_{g_a} := [\, V_g \;\; 0\,], \\ C_a &:= \begin{bmatrix} C & 0 \end{bmatrix}, \end{aligned} \tag{18c}$$

with $d_n := (r-1)n_\eta$. The matrices E, K, and H are filter gains to be designed which can be collected as $\psi = \{E, K, H\}$. The filter gains in ψ are determined via an optimization problem to satisfy performance guarantees, as discussed later.

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

$$e_d := \hat{x}_d - x_d, \tag{19}$$

where $x_d := (\eta, x_s)$. Note that according to (18b), $\hat{x}_d$ that represents the variables to be estimated is $\hat{x}_d = \bar{C}_a \hat{x}_a$ with

$$\bar{C}_a := \begin{bmatrix} \bar{C}_1^\top & \bar{C}_2^\top \end{bmatrix}^\top. \tag{20}$$

The estimation error dynamics can be written as follows [27, Sec. 6.4]:

$$\begin{cases} \dot{e} = Ne + MS_{g_a}\delta g - MB_{\omega_a}\omega_a + B_{\nu_a}\nu_a, \\ e_d = \bar{C}_a e, \end{cases} \tag{21a}$$

where $e \in \mathbb{R}^{n_z}$ is an auxiliary state to formulate the error dynamics and the involved components in the above equation are defined as

$$\begin{aligned} \delta g &:= g(V_{g_a}\hat{x}_a + H(y_s - C_a\hat{x}_a), u) - g(V_{g_a}x_a, u), \\ B_{\omega_a} &:= \begin{bmatrix} B_\omega & 0 \\ 0 & 0 \\ 0 & I_{n_\eta} \end{bmatrix}, \quad \omega_a := \begin{bmatrix} \omega \\ \eta^{(r)} \end{bmatrix}, \\ B_{\nu_a} &:= \begin{bmatrix} KD_\nu & -ED_\nu \end{bmatrix}, \quad \nu_a := \begin{bmatrix} \nu^\top(t) & \dot{\nu}^\top(t) \end{bmatrix}^\top. \end{aligned} \tag{21b}$$

The selection of r in (18) can be performed by starting with $r = 1$ and increasing it iteratively until the norm of the desired error in (19) of the filter provided below does not decrease further. For further details on the interpretation of r, refer to [23].

In what follows, we provide an optimization problem with LMI to design the gains of the proposed uncertainty-state estimator in (18) (i.e., $\psi = \{E, K, H\}$). Consider the system (1), the uncertainty-state estimator in (18), under Assumption 2 with Lipschitz constant l_{g_x}. Solve the optimization problem with LMI

$$\begin{aligned} \min_{\Pi, \bar{R}, \bar{Q}, H, \rho, \sigma} \quad & \rho \\ \text{s.t.} \quad & \begin{bmatrix} X_{11} & X_{12} \\ * & -I \end{bmatrix} \prec 0, \\ & \begin{bmatrix} L_{11} & -(\Pi + \bar{R}C_a)B_{\omega_a} & X_{12} \\ * & -\rho a I & 0 \\ * & * & -I \end{bmatrix} \preceq 0, \\ & \begin{bmatrix} X_{11} & H_{12} & 0 & X_{12} \\ * & -b^2 I & T_\nu^\top H^\top & 0 \\ * & * & -I & 0 \\ * & * & * & -I \end{bmatrix} \preceq 0, \\ & \begin{bmatrix} \Pi & \bar{C}_a^\top \\ * & \sigma I \end{bmatrix} \succeq 0, \quad \Pi \succ 0, \quad \sigma \leq \sigma_{max} \end{aligned} \tag{22}$$

with the involved matrices defined as follows:

$$\begin{aligned} X_{11} &:= A_a^\top \Pi + A_a^\top C_a^\top \bar{R}^\top - C_a^\top \bar{Q}^\top + \Pi A_a + \bar{R}C_a A_a \\ &\quad - \bar{Q}C_a + l_{g_x}(V_{g_a}^\top V_{g_a} - V_{g_a}^\top H C_a - C_a^\top H^\top V_{g_a}), \\ X_{12} &:= \begin{bmatrix} \sqrt{2l_{g_x}}\big((\Pi + \bar{R}C_a)S_{g_a}\big)^\top \\ \sqrt{l_{g_x}}(HC_a)^\top \end{bmatrix}^\top, \\ B_{\omega_a} &:= \begin{bmatrix} B_\omega & 0 \\ 0 & 0 \\ 0 & I_{n_\eta} \end{bmatrix}, \quad L_{11} := X_{11} + a\bar{C}_a^\top \bar{C}_a, \\ H_{12} &:= [\, \bar{Q}D_\nu \;\; -\bar{R}D_\nu \,], \quad T_\nu := [\, D_\nu \;\; 0 \,], \end{aligned}$$

given scalars $a, \sigma_{max} > 0$, b, matrix $\bar{C}_a$ as in (20), and the remaining matrices as in (18c) and (1). Denote the optimizers as $\Pi^\star$, $\bar{R}^\star$, $\bar{Q}^\star$, $H^\star$, $\rho^\star$ and $\sigma^\star$. Then, we have the optimal parameters of (18) as $\psi = \psi^\star$, namely, $\{E^\star = {\Pi^\star}^{-1}\bar{R}^\star, K^\star = {\Pi^\star}^{-1}\bar{Q}^\star, H^\star\}$.

Remark 4 (Performance Guarantees) *The estimation error of the filter defined in* (19)*, governed by the dynamics in* (18)*, can be seen as the image of the input* $(\omega, \eta^{(r)}, \nu, \dot{\nu})$ *under a mapping induced by the filter dynamics. The main theoretical results presented here, together with the optimization program in* (22)*, establish important regularity properties for this mapping, typically expressed as input–output gains under certain metrics, as follows:*
i) Stability: *The estimation error dynamics in* (21) *is input-to-state stable in the sense of [27, Def. 1]*;
ii) Disturbance Attenuation: *For* $\nu = 0$*, the* $\mathcal{L}_2$*-gain of the mapping* $(\omega, \eta^{(r)}) \mapsto e_d$ *in* (21) *is upper bounded*

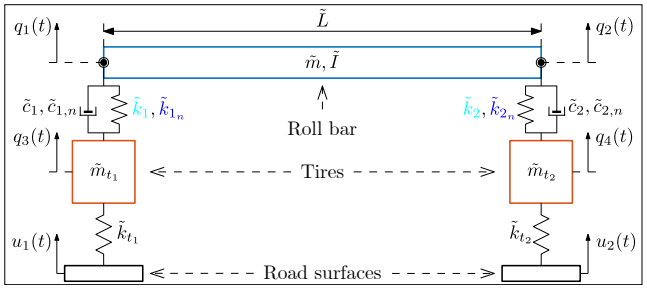

Fig. 3. Roll plane system schematic.

by $\sqrt{\rho^\star}$ *, i.e.,*

$$\|e_d\|_{\mathcal{L}_2} \leq \sqrt{\rho^\star}\|(\omega, \eta^{(r)})\|_{\mathcal{L}_2}; \tag{23}$$

iii) Noise Rejection: *For* $(\omega, \eta^{(r)}) = 0$*, the* $\mathcal{L}_2 - \mathcal{L}_\infty$ *induced gain of the mapping* $(\nu, \dot{\nu}) \mapsto e_d$ *in* (21) *is upper bounded by* $\sqrt{b\sigma^\star}$*, i.e.,*

$$\|e_d\|_{\mathcal{L}_2} \leq \sqrt{b\sigma^\star}\|(\nu, \dot{\nu})\|_{\mathcal{L}_\infty}. \tag{24}$$

It should be noted that the guarantees in (23) *and* (24) *remain valid even if* $||(w, \eta^{(r)})||_{\mathcal{L}_2}$ *or* $\|(\nu, \dot{\nu})\|_{\mathcal{L}_\infty}$ *is not bounded; however, in such cases the statement is trivial, since the upper bound becomes unbounded. We refer the interested readers to [27, Prop. 1] and [23, Thm. 1] for further details.*

Note that the solution we have provided for uncertainty-state estimation is applicable to globally Lipschitz nonlinear systems (i.e., Assumption 2 is satisfied). However, for locally Lipschitz nonlinear systems, a similar unknown input-state estimator proposed in [28] can be used.

6 Case Study

In this section, we evaluate the proposed methods via a case study. We assess the obtained model accuracy for the considered class of nonlinear systems in the sense of output accuracy. The case study is a four-degree-of-freedom roll plane maneuver of a vehicle with nonlinear suspension, as considered in [29, 30]. See Figure 3 for a schematic of this system. The equations of motion and the system parameters are given in Appendix A.4. The nonlinearity of the system is rooted from nonlinear stiffness and damping between the roll bar and both tires. As available measured outputs, we have the relative displacement and relative velocity between the roll bar and both tires (i.e., $y_{s_1} = q_1 - q_3, y_{s_2} = q_2 - q_4, y_{s_3} = \dot{q}_1 - \dot{q}_3, y_{s_4} = \dot{q}_2 - \dot{q}_4$, see Figure 3). The system inputs are the position of the road surfaces (u_1 and u_2 in Figure 3). The training and test data sets are generated by providing different system inputs, as described in Section 6.1.

For the above-mentioned system, we have (unknown) parametric uncertainty in the linear stiffness between the roll bar and tires (i.e., uncertainty in the linear parameters $\tilde{k}_1, \tilde{k}_2$ shown in Figure 3), and the nonlinear stiffnesses (i.e., $\tilde{k}_{1_n}, \tilde{k}_{2_n}$ shown in Figure 3) are completely unknown. The problem is to find a data-based model for the mentioned uncertainties and guarantee stability of the extended model (i.e., known model together with uncertainty model). The proposed methods (three algorithms) presume that a (training) data-set of inputs, estimated uncertainties, and estimated state realizations is given (using uncertainty and state estimators discussed in Section 5). The results of uncertainty and state estimation for one training data-set are provided in [27, Sec. A.6].

Following the proposed method for this problem, since the nonlinear stiffness is unknown, first we assume that we have basis functions as given by $h(\cdot)$ in (2) that are representative of the unknown nonlinearity (the proposed method is evaluated for a range of basis functions in Section 6.3).

6.1 Training and Test Data-Sets

The training and test data-sets are generated via different system inputs. System inputs comprise combinations of sinusoidals at different frequency ranges (to cover different range of inputs). This type of multi-sine signal is usual for system identification of the considered case study [30, Sec. 3.3]. The system input $u = (u_1, u_2)$ can be written as

$$u_1 = \frac{\max \tilde{\alpha}_i}{\sum_{i=1}^{\tilde{n}} \tilde{\alpha}_i} \sum_{i=1}^{\tilde{n}} \tilde{\alpha}_i \sin\left(\tilde{\omega}_i t + \tilde{\phi}_i\right), \quad t \in [0, t_f]$$
$$u_2 = 0,$$

where the parameters

$$(\tilde{\alpha}_i)_{i=1}^{\tilde{n}}, (\tilde{\omega}_i)_{i=1}^{\tilde{n}}, \left(\tilde{\phi}_i\right)_{i=1}^{\tilde{n}},$$

and $\tilde{n}$ are randomly drawn from uniform distributions in the intervals $[0.01, 0.1]$ m, $[0.6\pi, 3\pi]$ rad/s, $[0, 0.94\pi]$ rad, and $[2, 10]$, respectively. The parameter t_f indicates the final time of the simulation. Five training data sets and 1000 test data sets of inputs are drawn from the aforementioned system input distributions.

6.2 Model Accuracy

Next, we compare the accuracy of the uncertainty models trained via the proposed methods. We compare the results against the true system model used to generate the data. We fit three extended models using the known physics-based model and the learned uncertainty models via the proposed methods described in Sections 3 and

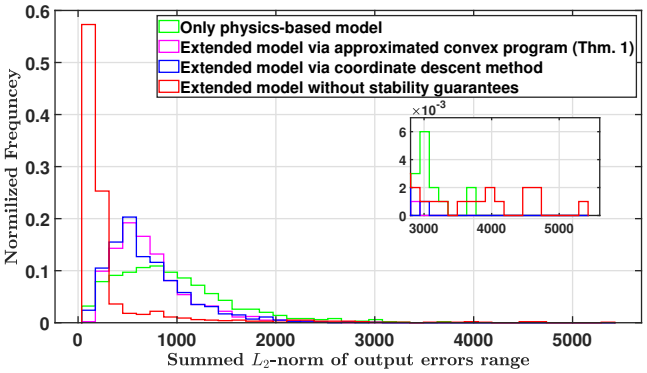

Fig. 4. Histogram for comparison of system outputs and the outputs of different models with cubic basis function (i.e., $h = (V_\eta x)^{\circ 3}$) for the test data.

4. Additionally, we examine a scenario where no stability guarantees for the extended model is considered (i.e., only by minimizing the cost function in (5)). Thus, for the test data-set, we compare the system output with the outputs of three extended models and a model that is not extended and neglects uncertainties (i.e., only consist physics-based model).

Given the large number of test data-sets (1000 test data-sets), this comparison is depicted via the histogram shown in Figure 4. The horizontal axis is a metric for the output error computed via $\sum_{t=0}^{t_f=20} \|y_s(t) - Cx(t)\|$. The histogram presents the distribution of output errors for the test data by dividing it into intervals and displaying the frequency of occurrences within each interval. In the results shown in Figure 4, the basis function representing the true system is a cubic function (i.e., $h = (V_\eta x)^{\circ 3}$ in (2)). For results considering basis functions other than the cubic function, see Section 6.3. In the histogram, results from four models are shown: 1. The green plot represents results from the physics-based model alone; 2. The magenta plot represents results using the extended model trained via the approximated convex optimization in Theorem 1; 3. The blue plot represents results using the extended model trained via the coordinate descent method; 4. The red plot represents results using the extended model trained by minimizing the cost in (5) without stability guarantees. From Figure 4, one can observe that both learning strategies (the results in blue and magenta) substantially improve model quality compared to the baseline without a learned uncertainty model (green). The coordinate descent method is initialized with the solution of the convex program. The performance gain from magenta to blue is relatively small, indicating that most of the improvement is already achieved by the convex approximation. Other initializations, such as setting the parameters of the uncertainty model $\theta = (\Theta_l, B_l, \Theta_n)$ to zero or fixing the invariance set shape matrix P as a constant multiplied by the identity matrix, did not yield a feasible solution. This further highlights the significance of the proposed convex approximation.

Table 1
Mean of RMSEs for different models.

Model \ Output entry	first [m]	second [m]	third [m/s]	forth [m/s]
Only physics-based model	0.054	0.054	0.389	0.383
Extended model without stability guarantees	0.015	0.016	0.130	0.134
Extended model via method in Thm. 1	0.042	0.042	0.316	0.285
Extended model via coordinate descent method	0.038	0.039	0.294	0.285

Moreover, for further comparisons of different models, the mean Root Mean Square Error (RMSE) for each output entry, computed across all 1000 test samples, is presented in Table 1. As can be seen, the RMSE decreases across all output entries when using the proposed extended models, demonstrating their strong performance and reliability.

Furthermore, to provide a quantitative comparison with existing literature on model updating, we compare the proposed method with the neural network (NN)-based approach presented in [5], which updates the model by adding a fully connected feedforward NN to a physics-based model. We train two updated models for the studied example: one with 50 training epochs and a NN with two hidden layers of size 12 and tangent hyperbolic activation functions, and another with 1000 training epochs and a NN with two hidden layers of size 20 and tangent hyperbolic activation functions. The performance of these models on the test data is shown in the histogram in Figure 5, where the cyan plot corresponds to the first setting and the black plot to the second. Since this NN-based method does not guarantee model stability, we compare it to our proposed method without stability guarantees.

As shown in the figure, the hybrid physics-NN model with the first setting outperforms the proposed method. However, its training time exceeds 6 hours, whereas our

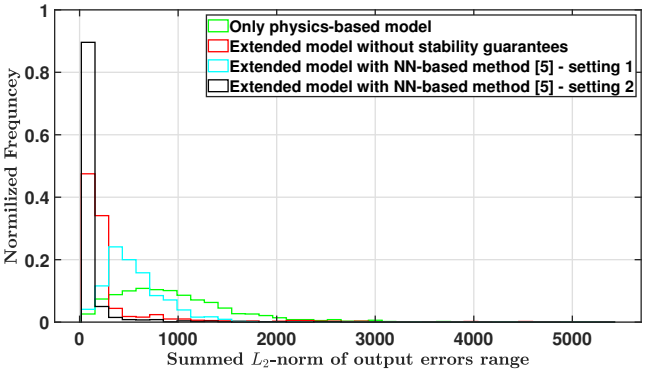

Fig. 5. Histogram comparing the true system outputs with model outputs obtained using the proposed method and the physics-NN model from [5], evaluated on the test data.

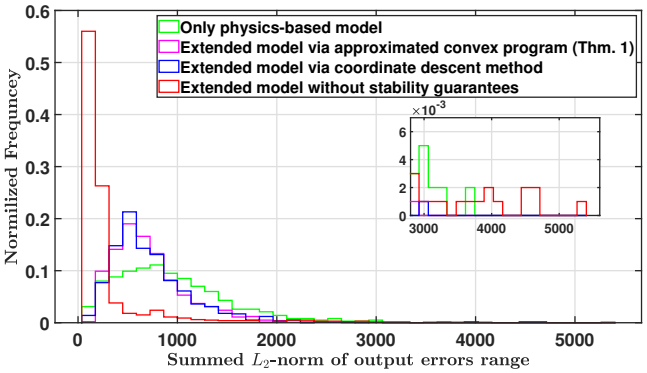

Fig. 6. Histogram for comparison of system outputs and the outputs of different models with basis function $h = [(V_\eta x)^{\circ 2^\top}, (V_\eta x)^{\circ 3^\top}]^\top$ for the test data.

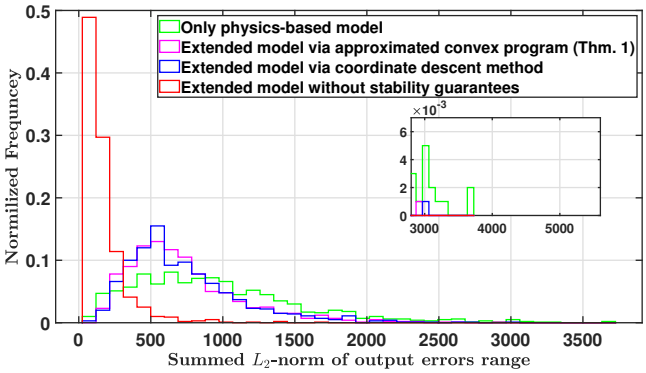

Fig. 7. Histogram for comparison of system outputs and the outputs of different models with basis function $h = [(V_\eta x)^{\circ 2^\top}, \exp(V_\eta x)^\top, (V_\eta x)^{\circ 3^\top}]^\top$ for the test data.

method (even when enforcing stability guarantees) requires less than 10 seconds to train. Notably, the hybrid model with the second setting performs worse than our proposed method and still requires over 120 seconds of training time. All models were trained on the same hardware setup to ensure a fair comparison: a laptop equipped with an Intel Core i7-9750H CPU @ 2.60GHz and 16 GB RAM, using high-performance power settings. These results demonstrate the effectiveness of our method in achieving high performance with significantly lower computational cost, at least for the studied example.

Remark 5 (The Price of Stability) *Imposing the stability constraint turns out to deteriorate the expected output errors (compare the average of the red histogram with the blue and magenta histograms in Figure 4). The root cause of this can be due to the fact that the stability constraints make the resulting optimization program nonconvex. Namely, we have to resort to a sub-optimal solution among all possible models considered (the magenta and blue histograms in Figure 4), whereas ignoring this condition allows us to stay in a convex setting where we can find the global optimal model (the red histogram). It is, however, worth noting that this better performance at the average level may come at the cost of some rare instances in the tail of the histogram (see the zoom window in Figure 4). In this light, one can see this average performance degradation as "the price of stability" to ensure that any model we estimate meets our prior stability assumption.*

6.3 Effect of Basis Functions

Figures 6, 7, and 8, similar to Figure 4, illustrate the histogram for the models' output errors. Unlike Figure 4, the basis functions (i.e., $h(\cdot)$ in (2)) are not same as the true nonlinearity in the system. Now, the basis functions are $h = [(V_\eta x)^{\circ 2^\top}, (V_\eta x)^{\circ 3^\top}]^\top$, $h = [(V_\eta x)^{\circ 2^\top}, \exp(V_\eta x)^\top, (V_\eta x)^{\circ 3^\top}]^\top$, $h = [\sin(V_\eta x)^\top$, $\cos(V_\eta x)^\top, \exp(V_\eta x)^\top, (V_\eta x)^{\circ 2^\top}, (V_\eta x)^{\circ 4^\top}, (V_\eta x)^{\circ 6^\top}]^\top$ in Figures 6, 7, and 8, respectively. The results in these three figures indicate that the selection of the basis function does not significantly affect the accuracy of the extended model, especially for the results with stability guarantees. That is, having a model that uses the "wrong" basis function is still better that using the model that does not account for uncertainty (at least in the considered use case).

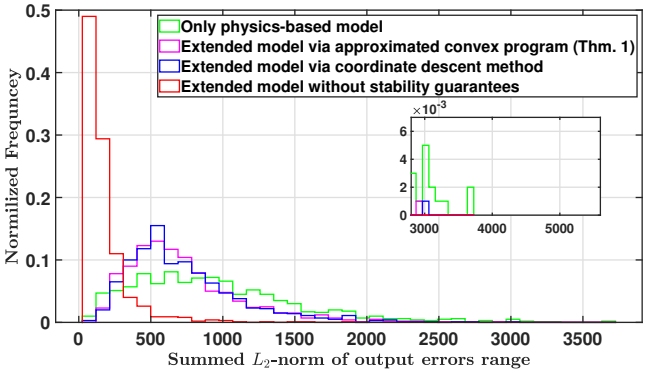

Fig. 8. Histogram for comparison of system outputs and the outputs of different models with basis function $h = [\sin(V_\eta x)^\top, \cos(V_\eta x)^\top, \exp(V_\eta x)^\top, (V_\eta x)^{\circ 2^\top}, (V_\eta x)^{\circ 4^\top}, (V_\eta x)^{\circ 6^\top}]^\top$ for the test data.

7 Conclusion

This paper has proposed a framework for model updating via learning modeling uncertainties in locally (and globally) Lipschitz nonlinear models. Moreover, we have ensured the stability of the extended model, which consists of a prior known model and the learned uncertainty model. The proposed framework includes two steps. First, we have assumed that uncertainty and state estimates are known. Under this assumption, we have introduced two semi-definite programs: Theorem 1 for locally (and Theorem 2 for globally) Lipschitz nonlinear

models, to learn uncertainty models while ensuring the stability of the extended model (invariant set and ISS for locally and globally Lipschitz nonlinear models, respectively). Second, we have proposed a filter, designed using the semi-definite program outlined in (22), to estimate uncertainty and state, based on the known prior model and input-output data. Simulations for a large data-set of a roll plane model of a vehicle have demonstrated the performance and potential of the proposed approach. Future work could include addressing the suboptimality of the convexification method presented in this paper for locally Lipschitz systems (i.e., Theorem 1).

A Appendices

A.1 Invariant set conditions for model (2)

Let $W(x) = x^\top P x$ with $P \succ 0$ be a Lyapunov function candidate. We need to derive the derivative of the Lyapunov function candidate along the trajectories of the extended system model (2). Before that, first, we impose the condition $\mathcal{E}_{inv} \subseteq \mathcal{E}_{sys}$ on the model invariant set and the system state set as defined in (9) and (6), respectively. By doing so, later, the Lipschitz property of the nonlinearities of extended system model (2) can be used in deriving a tractable condition for $\dot{W} \leq 0$. It is shown in [25, Sec. 3.7.1] that the condition $\mathcal{E}_{inv} \subseteq \mathcal{E}_{sys}$ can be written as (10b) by applying $\mathcal{S}$-procedure.

From (2) and the Lipschitz conditions for the known nonlinearity $g(\cdot)$ and the uncertainty model nonlinearity $h(\cdot)$ in (8) (together with that zero is an equilibrium point of system for $u = 0$), it follows that

$$\begin{aligned}\dot{W}(x) \leq & x^\top \left(A^\top P + PA + V_\eta^\top \Theta_l^\top S_{\eta_l}^\top P + P S_{\eta_l} \Theta_l V_\eta\right) x \\ & + 2\|x^\top P S_g\|(l_{g_x}\|V_g x\| + l_{g_u}\|u\|) \\ & + 2\|x^\top P S_{\eta_l}\|\|\Theta_n\|(l_{h_x}\|V_\eta x\| + l_{h_u}\|u\|) \\ & + 2x^\top P(B_u + S_{\eta_l} B_l)u.\end{aligned} \tag{A.1}$$

See [27] for more details of (A.1) derivation. Now, by imposing the condition on parameters of the nonlinear part of uncertainty model given in (10c) and then a specific form of the Cauchy–Schwarz inequality an upper bound for the right-hand-side of (A.1) can be found as follows:

$$\begin{aligned}\dot{W}(x) \leq & x^\top \left(A^\top P + PA + V_\eta^\top \Theta_l^\top S_{\eta_l}^\top P + P S_{\eta_l} \Theta_l V_\eta\right) x \\ & + (l_{g_x} + l_{g_u})\|x^\top P S_g\|^2 + l_{g_x}\|V_g x\|^2 + l_{g_u}\|u\|^2 \\ & + (\bar{l}_{h_x} + \bar{l}_{h_u})\|x^\top P S_{\eta_l}\|^2 + \bar{l}_{h_x}\|V_\eta x\|^2 + \bar{l}_{h_u}\|u\|^2 \\ & + 2x^\top P(B_u + S_{\eta_l} B_l)u \\ = & x^\top \Delta x + 2x^\top P(B_u + S_{\eta_l} B_l)u + (l_{g_u} + \bar{l}_{h_u})u^\top u\end{aligned} \tag{A.2}$$

with $\bar{l}_{h_u}$ as defined in (10e) and Δ as defined in (10d). See [27] for more details of (A.2) derivation. Therefore, from (A.2), it follows that

$$\dot{W}(x) \leq x^\top \Delta x + 2x^\top P(B_u + S_{\eta_l} B_l)u + (l_{g_u} + \bar{l}_{h_u})u^\top u. \tag{A.3}$$

If we can find a P such that $\dot{W} \leq 0$ whenever $u \in \mathcal{E}_u$ (as defined in (7)) and $W \geq 1$ along the trajectories of (2); Then, the ellipsoid $\mathcal{E}_{inv}$ (as defined in (9)) is an invariant set of (2). To this end, consider the vector $\begin{bmatrix} x^\top u^\top 1 \end{bmatrix}^\top$. Using (A.2), the condition $\dot{W} \leq 0$ can be stated as

$$E_1 = \begin{bmatrix} \Delta & P(B_u + S_{\eta_l} B_l) & 0 \\ \star & (l_{g_u} + \bar{l}_{h_u})I & 0 \\ \star & \star & 0 \end{bmatrix} \preceq 0. \tag{A.4}$$

Similarly, the condition (7) can be written as

$$F_1 = \begin{bmatrix} 0 & 0 & 0 \\ \star & U & 0 \\ \star & \star & -1 \end{bmatrix} \preceq 0. \tag{A.5}$$

Finally, the condition $W(x) \geq 1$ can be restated as

$$G_1 = \begin{bmatrix} -P & 0 & 0 \\ \star & 0 & 0 \\ \star & \star & 1 \end{bmatrix} \preceq 0. \tag{A.6}$$

Therefore, for constructing an invariant set of trajectories, we require (A.4) ($\dot{W} \leq 0$) to hold when (A.5) and (A.6) hold. To satisfy these conditions, we can apply $\mathcal{S}$-procedure [25, Sec. 2.6.3], which states that there should exist non-negative constants α and β such that the following inequality holds:

$$E_1 - \alpha F_1 - \beta G_1 \preceq 0. \tag{A.7}$$

The condition above can be written as (10a). Note that in the derivation of (10a), we impose the condition on Lipschitz set condition in (10b) and the norm constraint on parameters of nonlinear part of uncertainty model in (10c). Therefore, the condition (10a) together with (10b), and (10c) imply that the ellipsoid $\mathcal{E}_{inv}$ (as defined in (9)) is an invariant set of the extended model (2).

A.2 ISS conditions for model (2)

Let $W(x) := x^\top P x$ be an ISS Lyapunov function candidate. Similar to the proof in Appendix A.1, given the globally Lipschitz property of the model nonlinearities in (12) and the imposed condition on parameters of nonlinear part of uncertainty model in (10c), the time-derivative of the candidate ISS Lyapunov function, $\dot{W}$,

can be written in the form of (A.2). The rest of the proof follows the line of reasoning of the proof of Proposition 1 in [23], which for brevity is omitted here. It is shown in [23] for a Lyapunov derivative inequality similar to (A.2) that, in order to conclude ISS property of a dynamics, Δ has to be negative definite. Note that in the derivation of Δ in (A.2), we impose the condition on the parameters of nonlinear part of uncertainty model in (10c). Therefore, the condition $\Delta \prec 0$ together with (10c) imply that the extended model (2) is ISS with respect to input u.

A.3 Proof of Theorem 1

We show that the non-convex optimization problem in (11) can be convexified as in (16). First, to make the problem tractable, we set the $S_{\eta_l} = I$ in extended system model (2). Then, we convexify the non-convex conditions in (11) as follows.

Consider the stability constraint (10a). By applying changes of variables as $S := P\Theta_l$, $R := PB_l$ and the Schur complement, the stability constraint in (10a) is equivalent to (16a). Favorably, by the introduced change of variables, the stability constraint becomes convex (an LMI). However, since the cost J in (5) is a function of $\Theta_l = P^{-1}S$ and $B_l = P^{-1}R$, the cost is not convex in S, R and P, after the changes of variables. Therefore, we have to convexify the cost to arrive at a convex optimization problem formulation. For scalar cost function J in (5), we have the following equality:

$$J = \sum_{i=1}^{N} d_i^\top T^\top T d_i = \sum_{i=1}^{N} tr(d_i^\top T^\top T d_i).$$

Due to cyclic property of the trace operator, the above cost can be written as

$$J = tr(T^\top T \sum_{i=1}^{N} d_i d_i^\top).$$

By the definition of the matrix D in (3) and the cyclic property of the trace, we have

$$J = tr(TDT^\top).$$

Now, we can write the epigraph form of the optimization problem in (11) as follows:

$$\begin{aligned} \min_{P,S,R,\Theta_n,W,\alpha,\gamma} \quad & tr(W) \\ \text{s.t.} \quad & (16a), (10b), (10c), \\ & P \succ 0, \alpha, \gamma \geq 0, \\ & tr(TDT^\top) \leq tr(W). \end{aligned} \tag{A.8}$$

Due to monotonicity of trace, the last constraint in (A.8) can be transformed to the following constraint:

$$TDT^\top \preceq W.$$

Note that even with the application of the above transformation, the optimization remains equivalent to (A.8) [25, p.8]. By applying the Schur complement to the above inequality, we have

$$\begin{bmatrix} W & T\tilde{D}^\top \\ \star & I \end{bmatrix} \succeq 0.$$

Note that by construction, the data matrix D is always symmetric and positive semi-definite. Therefore, its Cholesky decomposition (i.e., $D = \tilde{D}^\top \tilde{D}$) always exists. By applying the congruence transformation of $\text{diag}(P, I)$ to the above inequality, we obtain the following equivalent inequality:

$$\begin{bmatrix} PWP & \tilde{T}\tilde{D}^\top \\ \star & I \end{bmatrix} \succeq 0. \tag{A.9}$$

Note that in the above inequality to introduce $\tilde{T} := PT = \begin{bmatrix} S & R & -P & P\Theta_n \end{bmatrix}$, the change of variable as $Z := P\Theta_n$ is applied. Now, by substituting the lower bound $2\mu_2 P - \mu_2^2 W^{-1}$, with given positive scalar μ_2, for PWP and applying the Schur complement, the LMI in (16c) is obtained.

Consider another non-convex condition (upper bound on the parameters of nonlinear part of uncertainty model) as (10c). Based on [31, sec. 2.11], the condition (10c) can be written as

$$\begin{bmatrix} \bar{l}_{h_x} I & l_{h_x} \Theta_n^\top \\ \star & \bar{l}_{h_x} I \end{bmatrix} \succeq 0. \tag{A.10}$$

By applying the congruence transformation of $\text{diag}(I, P)$ to the above inequality, we obtain the following equivalent inequality

$$\begin{bmatrix} \bar{l}_{h_x} I & l_{h_x} Z^\top \\ \star & \bar{l}_{h_x} P^2 \end{bmatrix} \succeq 0.$$

Now, by substituting the lower bound $2\mu_1 P - \mu_1^2 I$, with given positive scalar μ_1, for P^2, the LMI in (16b) is obtained.

In conclusion, instead of the non-convex optimization problem in (11), we provide an approximation of that problem in the form of the semi-definite program in (16). We remark that the approximation arises from initially

setting $S_{\eta l} = I$ at the beginning of the proof. Additionally, we use the lower bound of $2\mu_2 P - \mu_2^2 W^{-1}$ for PWP in the derivation of the LMI in (16c), and the lower bound of $2\mu_1 P - \mu_1^2 I$ for P^2 in the derivation of the LMI in (16b).

A.4 Roll Plane System Description

The equations of motion of the roll plane system are given in [29, app. B.2]. Here, we provide the state space representation. By defining the state vector $x_s = [x_{s_1}, x_{s_2}, \ldots, x_{s_7}, x_{s_8}]^\top := [q_1, q_2, q_3, q_4, \dot{q}_1, \ldots, \dot{q}_4]^\top$, where q_i and $\dot{q}_i$ are the displacements (as shown in Figure 3) and the corresponding velocities, the system dynamics can be described in the form of (1), where we have

$$A = \begin{bmatrix} 0 & I \\ -\tilde{M}^{-1}\tilde{K} & -\tilde{M}^{-1}\tilde{C} \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ \tilde{M}^{-1}\tilde{K}_u \end{bmatrix},$$

$$S_g = S_\eta = \begin{bmatrix} 0 \\ -\tilde{M}^{-1}\tilde{S} \end{bmatrix}, \quad g(\cdot) = 0.2\tilde{c}_{1,n} tanh(V_g x_s)$$

$$D_\nu = I, \quad B_\omega = 0, \quad \eta(\cdot) = \delta\tilde{k}_1 V_\eta x_s + \tilde{k}_{1_n}(V_\eta x_s)^{\circ 3},$$

$$C = \begin{bmatrix} \tilde{C} & 0 \\ 0 & \tilde{C} \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$V_g = \begin{bmatrix} 0 & 10\tilde{C} \end{bmatrix}, \quad V_\eta = \begin{bmatrix} \tilde{C} & 0 \end{bmatrix},$$

$$\tilde{M} = \begin{bmatrix} \frac{\tilde{m}}{2} & \frac{\tilde{m}}{2} & 0 & 0 \\ \frac{-\tilde{I}}{\tilde{L}} & \frac{\tilde{I}}{\tilde{L}} & 0 & 0 \\ 0 & 0 & \tilde{m}_{t_1} & 0 \\ 0 & 0 & 0 & \tilde{m}_{t_2} \end{bmatrix},$$

$$\tilde{K} = \begin{bmatrix} \tilde{k}_1 & \tilde{k}_2 & -\tilde{k}_1 & -\tilde{k}_2 \\ \frac{-\tilde{L}}{2}\tilde{k}_1 & \frac{\tilde{L}}{2}\tilde{k}_2 & \frac{\tilde{L}}{2}\tilde{k}_1 & \frac{-\tilde{L}}{2}\tilde{k}_2 \\ -\tilde{k}_1 & 0 & \tilde{k}_1 + \tilde{k}_{t_1} & 0 \\ 0 & -\tilde{k}_2 & 0 & \tilde{k}_2 + \tilde{k}_{t_2} \end{bmatrix},$$

$$\tilde{C} = \begin{bmatrix} \tilde{c}_1 & \tilde{c}_2 & -\tilde{c}_1 & -\tilde{c}_2 \\ \frac{-\tilde{L}}{2}\tilde{c}_1 & \frac{\tilde{L}}{2}\tilde{c}_2 & \frac{\tilde{L}}{2}\tilde{c}_1 & \frac{-\tilde{L}}{2}\tilde{c}_2 \\ -\tilde{c}_1 & 0 & \tilde{c}_1 & 0 \\ 0 & -\tilde{c}_2 & 0 & \tilde{c}_2 \end{bmatrix},$$

$$\tilde{K}_u = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \tilde{k}_{t_1} & 0 \\ 0 & \tilde{k}_{t_2} \end{bmatrix}, \quad \tilde{S} = \begin{bmatrix} 1 & 1 \\ \frac{-\tilde{L}}{2} & \frac{\tilde{L}}{2} \\ -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The parameter values are provided in [27, Sec. A.5].

Acknowledgements

This research is partially supported by the Dutch Research Council (NWO) Perspectief Digital Twin Programme, project number P18-03, sub-project 4.3, the Ministry of Education (MOE), Singapore, under its Academic Research Fund (AcRF) Tier 1 grant, funded through the SUTD Kickstarter Initiative (SKI 2021_06_02), and the European Research Council (ERC) under the grant TRUST-949796.

References

[1] Ian R Manchester, Max Revay, and Ruigang Wang. Contraction-based methods for stable identification and robust machine learning: a tutorial. In *IEEE Conference on Decision and Control (CDC)*, pages 2955–2962, 2021.

[2] Amirhassan Abbasi and C Nataraj. Physics-informed machine learning for uncertainty reduction in time response reconstruction of a dynamic system. *IEEE Internet Computing*, 26(4):35–44, 2022.

[3] Mariane Yvonne Schneider, Ward Quaghebeur, Sina Borzooei, Andreas Froemelt, Feiyi Li, Ramesh Saagi, Matthew J Wade, Jun-Jie Zhu, and Elena Torfs. Hybrid modelling of water resource recovery facilities: status and opportunities. *Water Science and Technology*, 85(9):2503–2524, 2022.

[4] William Bradley, Jinhyeun Kim, Zachary Kilwein, Logan Blakely, Michael Eydenberg, Jordan Jalvin, Carl Laird, and Fani Boukouvala. Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering*, 166:107898, 2022.

[5] Ward Quaghebeur, Ingmar Nopens, and Bernard De Baets. Incorporating unmodeled dynamics into first-principles models through machine learning. *IEEE Access*, 9:22014–22022, 2021.

[6] Alireza Yazdani, Lu Lu, Maziar Raissi, and George Em Karniadakis. Systems biology informed deep learning for inferring parameters and hidden dynamics. *PLoS Computational Biology*, 16(11):e1007575, 2020.

[7] Mitchell Daneker, Zhen Zhang, George Em Karniadakis, and Lu Lu. Systems biology: Identifiability analysis and parameter identification via systems-biology-informed neural networks. In *Computational Modeling of Signaling Networks*, pages 87–105. Springer, 2023.

[8] Karan Taneja, Xiaolong He, QiZhi He, Xinlun Zhao, Yun-An Lin, Kenneth J Loh, and Jiun-Shyan Chen. A feature-encoded physics-informed parameter identification neural network for musculoskeletal systems. *Journal of Biomechanical Engineering*, 144(12):121006, 2022.

[9] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[10] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[11] Kathleen Champion, Peng Zheng, Aleksandr Y Aravkin, Steven L Brunton, and J Nathan Kutz. A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, 8:169259–169271, 2020.

[12] Jean-Christophe Loiseau and Steven L Brunton. Constrained sparse galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.

[13] Seth L Lacy and Dennis S Bernstein. Subspace identification with guaranteed stability using constrained optimization. *IEEE Transactions on Automatic Control*, 48(7):1259–1263, 2003.

[14] Loris Di Natale, Muhammad Zakwan, Philipp Heer, Giancarlo Ferrari Trecate, and Colin N Jones. Simba: System identification methods leveraging backpropagation. *Acailable at arXiv:2311.13889*, 2023.

[15] Farhad Ghanipoor, Carlos Murguia, Peyman Mohajerin Esfahani, and Nathan Van de Wouw. Uncertainty learning for LTI systems with stability guarantees. In *European Control Conference (ECC)*, pages 2568–2573, 2024.

[16] Mohammad Khosravi and Roy S Smith. Nonlinear system identification with prior knowledge on the region of attraction. *IEEE Control Systems Letters*, 5(3):1091–1096, 2020.

[17] Mohammad Khosravi. Representer theorem for learning koopman operators. *IEEE Transactions on Automatic Control*, 2023.

[18] MF Shakib, Roland Tóth, AY Pogromsky, A Pavlov, and Nathan van de Wouw. Kernel-based learning of stable nonlinear state-space models. In *IEEE Conference on Decision and Control (CDC)*, pages 2897–2902, 2023.

[19] Max Revay, Ruigang Wang, and Ian R Manchester. Recurrent equilibrium networks: Unconstrained learning of stable and robust dynamical models. In *IEEE Conference on Decision and Control (CDC)*, pages 2282–2287, 2021.

[20] MF Shakib, AY Pogromsky, Alexey Pavlov, and Nathan van de Wouw. Fast identification of continuous-time lur'e-type systems with stability certification. *IFAC-PapersOnLine*, 52(16):227–232, 2019.

[21] Max Revay, Ruigang Wang, and Ian R Manchester. Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness. *IEEE Transactions on Automatic Control*, 2023.

[22] Dimitri Bertsekas. *Convex optimization algorithms*. Athena Scientific, 2015.

[23] Farhad Ghanipoor, Carlos Murguia, Peyman Mohajerin Esfahani, and Nathan van de Wouw. Robust fault estimators for nonlinear systems: An ultra-local model design. *Automatica*, 171:111920, 2025.

[24] Farhad Ghanipoor, Carlos Murguia, Peyman Mohajerin Esfahani, and Nathan van de Wouw. Ultra local nonlinear unknown input observers for robust fault reconstruction. In *IEEE Conference on Decision and Control (CDC)*, pages 918–923, 2022.

[25] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[26] Eduardo D Sontag et al. On the input-to-state stability property. *European Journal of Control*, 1(1):24–36, 1995.

[27] Farhad Ghanipoor, Carlos Murguia, Peyman Mohajerin Esfahani, and Nathan van de Wouw. Model updating for nonlinear systems with stability guarantees. *Available at arXiv:2406.06116*, 2024.

[28] Farhad Ghanipoor, Carlos Murguia, Peyman Mohajerin Esfahani, and Nathan van de Wouw. Linear fault estimators for nonlinear systems: An ultra-local model design. *IFAC-PapersOnLine*, 56(2):11693–11698, 2023.

[29] Bas M Kessels, Rob HB Fey, and Nathan van de Wouw. Real-time parameter updating for nonlinear digital twins using inverse mapping models and transient-based features. *Nonlinear Dynamics*, 111(11):10255–10285, 2023.

[30] Emmanuel Blanchard. *Polynomial chaos approaches to parameter estimation and control design for mechanical systems with uncertain parameters*. PhD thesis, Virginia Tech, 2010.

[31] Ryan James Caverly and James Richard Forbes. Lmi properties and applications in systems, stability, and control theory. *Available at arXiv:1903.08599*, 2019.