Interval Set-Membership State Estimator for Discrete-Time LPV Systems

Stefan Krebs and Sören Hohmann

Abstract— **This publication is devoted to the design of an interval-based set-membership state estimator that can be applied to totally observable linear parameter-varying (LPV) systems. Hereto, a set-inversion procedure to determine state intervals that are consistent with a sequence of input and output values as well as two intersections with predicted values are used to reduce the pessimism of interval arithmetic. A numerical example illustrates the performance of the method. The benefit of this method is that it provides an a priori known accuracy of the result based on the assumption of unknown but bounded uncertainties.**

I. Introduction

Several approaches to describe uncertainties in a state estimation framework and to cope with those uncertainties have been proposed in the past. The majority of the well-known methods describe the uncertain disturbances as a result of stochastic processes. While there exist several approaches to deal with such uncertainties, the Kalman filter ([1]) and its variations (e.g. [2], [3]) are the most common ones. Besides of these approaches, set-based techniques have been developed which differ fundamentally from the aforementioned in the modelling of the uncertainties. Instead of describing them as a result of stochastic processes, only the boundedness of the uncertainties is assumed and different ways to describe the uncertainties are applied, such as zonotopes ([4]), ellipsoids ([5]), subpavings ([6]) and intervals ([7]). These set-based approaches are frequently motivated by safety-critical systems because they provide guaranteed bounds of the estimates ([8], [9]). According to [10], these set-based techniques can be roughly divided into two categories: interval observers (e.g. [11], [12]) that are based on the classical observer structure and set-membership estimators (SME) (e.g. [13], [10]) that are based on a predictor-corrector structure. Compared to interval observers, SME have a high computational effort due to the two-stage predictor-corrector structure and the usually applied complex geometric representation of the sets. A remedy for the high computational effort of SME is provided in [11] by presenting an SME for linear time-invariant systems which is based on an interval representation of the uncertainties. Another appreciable property of this approach is the provision of an a priori determinable accuracy of the estimation result. However, the practical applicability of the approach is limited because it does not cover parameters varying with time. This motivates the publication at hand which is devoted to the extension of the approach presented in [11] to LPV systems. To this, both, the prediction and the correction step, have to be modified to cover time-variant system matrices while preserving the reduction of pessimism propagation.

The rest of the publication is organized as follows. Initially, some mathematical fundamentals including matrix operations and interval arithmetic fundamentals are presented concisely. Secondly, the so far only textually defined problem is defined more precisely which is followed by the presentation of the main results. A numerical example and a conclusion complete the publication.

S. Krebs and S. Hohmann are with the Institute of Control Systems (IRS) at the Karlsruhe Institute of Technology (KIT), Germany {stefan.krebs, soeren.hohmann}@kit.edu

II. Mathematical Fundamentals

A. Matrix Calculus

Definition 1: $\mathbf{1}_a^{b,c}$ is a vertical concatenation of $b-c$ identity matrices of size a.

Definition 2: The ∞-norm of a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is defined by

$$\|\boldsymbol{A}\|_\infty := \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}|\,. \tag{1}$$

Definition 3: For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, we define $\boldsymbol{A}^+ := \max\{\mathbf{0}, \boldsymbol{A}\}$ and $\boldsymbol{A}^- := \min\{\mathbf{0}, \boldsymbol{A}\}$. The operators $\max(\cdot)$ and $\min(\cdot)$ are understood component-wise.

Definition 4: The operator $\prod_{i=a}^{b}$ which can be applied to a matrix $A_i \in \mathbb{R}^{n\times n}$ with $a \in \mathbb{Z}$ and $b \in \mathbb{Z}$ is defined by

$$\prod_{i=a}^{b} \boldsymbol{A}_i = \begin{cases} \boldsymbol{A}_a\boldsymbol{A}_{a+1}\ldots\boldsymbol{A}_{b-1}\boldsymbol{A}_b & \forall a\le b \\ \boldsymbol{A}_a\boldsymbol{A}_{a-1}\ldots\boldsymbol{A}_{b+1}\boldsymbol{A}_b & \forall a > b \end{cases}. \tag{2}$$

B. Interval Arithmetic

Definition 5: An interval $[x] = [\underline{x}, \overline{x}]$ is a connected subset of $\mathbb{R}$ ([14]). Its lower bound $\underline{x}$ and its upper bound $\overline{x}$ are defined by

$$\underline{x} = \mathrm{lb}([x]) := \sup\{a \in \mathbb{R}\cup\{-\infty,\infty\} \mid \forall x\in[x], a\le x\}, \tag{3a}$$

$$\overline{x} = \mathrm{ub}([x]) := \inf\{b \in \mathbb{R}\cup\{-\infty,\infty\} \mid \forall x\in[x], x\le b\}. \tag{3b}$$

The set of all real intervals is denoted as $\mathbb{IR}$.

Definition 6: An interval vector $[\boldsymbol{x}] = [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}] \in \mathbb{IR}^n$ is defined as the Cartesian product of n intervals:

$$[\boldsymbol{x}] := [x_1]\times[x_2]\times\cdots\times[x_n]\,. \tag{4}$$

A possible representation of $[\boldsymbol{x}]$ is

$$[\boldsymbol{x}] = \begin{pmatrix}\overline{x}_1 & \overline{x}_2 & \ldots & \overline{x}_n & \underline{x}_1 & \underline{x}_2 & \ldots & \underline{x}_n\end{pmatrix}^\mathsf{T}. \tag{5}$$

Definition 7: The width $w([x])$ of an interval $[x]$ is

$$w([x]) \coloneqq \overline{x} - \underline{x}. \tag{6}$$

Definition 8: The maximum width $W([\boldsymbol{x}])$ of an interval vector $[\boldsymbol{x}]$ is defined by

$$W([\boldsymbol{x}]) \coloneqq \max_{i \in \{1,\ldots,n\}} \left(w([x_i])\right). \tag{7}$$

Definition 9: The interval function $[\boldsymbol{f}] : \mathbb{IR}^n \to \mathbb{IR}^m$ is an inclusion function for $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ if $\boldsymbol{f}([\boldsymbol{x}]) \subset [\boldsymbol{f}]([\boldsymbol{x}])$ for all $[\boldsymbol{x}] \in \mathbb{IR}^n$.

Lemma 1: Given an interval vector $[\boldsymbol{x}] = [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$, $\boldsymbol{x} \in \mathbb{R}^n$ with $\boldsymbol{x} \in [\boldsymbol{x}]$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. Then

$$\boldsymbol{A}^+ \cdot \underline{\boldsymbol{x}} + \boldsymbol{A}^- \cdot \overline{\boldsymbol{x}} \leq \boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{A}^+ \cdot \overline{\boldsymbol{x}} + \boldsymbol{A}^- \cdot \underline{\boldsymbol{x}} \tag{8}$$

holds under consideration of definition 3 ([15]).

Remark 1: For further informations on interval arithmetic besides of this short overview, the reader is referred to [14].

III. Problem Statement

Let a discrete-time LPV system be given by

$$\boldsymbol{x}_{k+1} = \overbrace{\boldsymbol{A}(\boldsymbol{\theta}_k)}^{=:\boldsymbol{A}_k} \boldsymbol{x}_k + \overbrace{\boldsymbol{B}(\boldsymbol{\theta}_k)}^{=:\boldsymbol{B}_k} \boldsymbol{u}_k + \overbrace{\boldsymbol{E}(\boldsymbol{\theta}_k)}^{=:\boldsymbol{E}_k} \boldsymbol{d}_k, \tag{9a}$$

$$\boldsymbol{y}_k = \underbrace{\boldsymbol{C}(\boldsymbol{\theta}_k)}_{=:\boldsymbol{C}_k} \boldsymbol{x}_k + \underbrace{\boldsymbol{D}(\boldsymbol{\theta}_k)}_{=:\boldsymbol{D}_k} \boldsymbol{u}_k + \underbrace{\boldsymbol{F}(\boldsymbol{\theta}_k)}_{=:\boldsymbol{F}_k} \boldsymbol{v}_k \tag{9b}$$

with the state vector $\boldsymbol{x}_k \in \mathbb{R}^n$, the input vector $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, the output vector $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ and the parameter vector $\boldsymbol{\theta}_k \in \mathbb{R}^{n_\theta}$. The vectors $\boldsymbol{d}_k \in \mathbb{R}^{n_d}$ and $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ are disturbances influencing the state vector and the output vector, respectively. The system matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$, $\boldsymbol{E}_k$, $\boldsymbol{C}_k$, $\boldsymbol{D}_k$ and $\boldsymbol{F}_k$ are real-valued matrices of proper sizes. Before, the problem to be solved is presented in detail, five assumptions on system (9) are stated. The first assumption states that at runtime of the state estimator to be developed, the time-variant parameters are exactly known at every time step k whereas a priori only a time-invariant interval vector is known (assumption 2). The third assumption contains the statement that the only information given for the disturbances are time-invariant interval vectors while assumption 4 is the easily fulfillable assumption that an bounded interval for the initial state of (9) is known. An assumption on observability finishes this section.

Assumption 1: The parameter vector $\boldsymbol{\theta}_k$ and hence the system matrices are exactly known at every time step k.

Assumption 2: For the a priori (cf. assumption 1) unknown parameter vector $\boldsymbol{\theta}_k$, there exists an a priori known time-invariant bounded interval vector $[\boldsymbol{\theta}]$ with

$$\boldsymbol{\theta}_k \in [\boldsymbol{\theta}] = [\underline{\boldsymbol{\theta}}, \overline{\boldsymbol{\theta}}] \quad \forall k \geq 0. \tag{10}$$

Assumption 3: For the unknown time-variant disturbance vectors $\boldsymbol{d}_k$ and $\boldsymbol{v}_k$, there exist a priori known time-invariant bounded interval vectors $[\boldsymbol{d}]$ and $[\boldsymbol{v}]$ with

$$\boldsymbol{d}_k \in [\underline{\boldsymbol{d}}, \overline{\boldsymbol{d}}] \quad \forall k \geq 0, \tag{11a}$$

$$\boldsymbol{v}_k \in [\underline{\boldsymbol{v}}, \overline{\boldsymbol{v}}] \quad \forall k \geq 0. \tag{11b}$$

Assumption 4: There exists a known bounded interval vector $[\boldsymbol{x}_0]$ for which $\boldsymbol{x}_0 \in [\boldsymbol{x}_0]$ holds.

Assumption 5: System (9) is totally observable (cf. Definition 2 in [16]).

The aim of the paper at hand is to develop a set-membership state estimator that computes a bounded interval $[\boldsymbol{x}_k]$ guaranteeing

$$\boldsymbol{x}_k \in [\boldsymbol{x}_k] \quad \forall k \in \{1, \ldots, N\} \tag{12}$$

with N being the number of considered time steps. Moreover, we seek for an a priori known maximum interval width $W([\boldsymbol{x}_k])$ that can be upper bounded by a positive and real-valued constant c for $k \geq n-1$, i.e.

$$W([\boldsymbol{x}_k]]) \leq c \quad \forall k \in \{n-1, \ldots, N\}. \tag{13}$$

IV. Main Result

As typical for SME, the presented approach consists of a prediction step and a correction step. In the following two subsections, the basic principles of these steps are presented. Afterwards, the final algorithm is presented.

A. Prediction Step

The prediction step is based on the relation between the state $\boldsymbol{x}_s$ at time step s and the state $\boldsymbol{x}_k$ at time step k with $k > s$ which is given by using (9) as

$$\begin{aligned} {}^p\boldsymbol{x}_k = & \left(\prod_{i=s}^{k-1} \boldsymbol{A}_i\right) \boldsymbol{x}_s \\ & + \sum_{i=s+1}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \boldsymbol{B}_{i-1} \boldsymbol{u}_{i-1} \right) \\ & + \sum_{i=s+1}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \boldsymbol{E}_{i-1} \boldsymbol{d}_{i-1} \right) \end{aligned} \tag{14}$$

containing the Dirac delta function $\delta(z)$. The superscript p is used to highlight the result as a prediction. Extending (14) to uncertain states described by interval vectors and by applying (11a) yields

$$\begin{aligned} {}^p[\boldsymbol{x}_k] = & \left(\prod_{i=s}^{k-1} \boldsymbol{A}_i\right) [\boldsymbol{x}_s] \\ & + \sum_{i=s+1}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \boldsymbol{B}_{i-1} \boldsymbol{u}_{i-1} \right) \\ & + \sum_{i=s+1}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \boldsymbol{E}_{i-1} [\boldsymbol{d}] \right). \end{aligned} \tag{15}$$

The right side of this equation (15) is an inclusion function for (14).

Remark 2: This direct computation of ${}^p[\boldsymbol{x}_k]$ based on $[\boldsymbol{x}_s]$, $[\boldsymbol{d}]$ and $\boldsymbol{u}_i$ with $i \in \{s, \ldots, k-1\}$ avoids the wrapping effect that would occur in the case of a multiple evaluation of an inclusion function for (9a).

Remark 3: Although (15) implies that $\boldsymbol{A}_k$ has to be nonsingular, this is not the case because $\boldsymbol{A}_k^{-1}$ always appears within the expression $\boldsymbol{A}_k \boldsymbol{A}_k^{-1} = \boldsymbol{I}$.

B. Correction Step

If the system (9) it totally observable, its state at time step k can be computed by using at least $n-1$ future input and output values ([16]) by

$$\begin{aligned}{}^{o}[\boldsymbol{x}_k] = \boldsymbol{O}^{-1}_{(k:k+n-1)} \Big(& \boldsymbol{y}_{(k:k+n-1)} \\ & -\boldsymbol{O}_{u(k:k+n-1)}\boldsymbol{u}_{(k:k+n-1)} \\ & -\boldsymbol{O}_{d(k:k+n-1)}\left[\boldsymbol{d}_{(k:k+n-2)}\right] \\ & -\boldsymbol{O}_{v(k:k+n-1)}\left[\boldsymbol{v}_{(k:k+n-1)}\right]\Big)\end{aligned} \quad (16)$$

wherein the superscript o highlights that the resulting interval vector can be calculated due to the observability of the system. This equation is an inclusion function for the well-known function to determine $\boldsymbol{x}_k$ from future input and output values which has been extended to LPV systems under consideration of the interval vectors $[\boldsymbol{v}]$ and $[\boldsymbol{d}]$. The matrices and vectors in (16) are given by

$$\boldsymbol{O}_{(k:k+n-1)} = \begin{pmatrix} \boldsymbol{C}_k \\ \boldsymbol{C}_{k+1}\boldsymbol{A}_k \\ \boldsymbol{C}_{k+2}\boldsymbol{A}_{k+1}\boldsymbol{A}_k \\ \vdots \\ \boldsymbol{C}_{k+n-1}\boldsymbol{A}_{k+n-2}\boldsymbol{A}_{k+n-3}\dots\boldsymbol{A}_k \end{pmatrix}, \quad (17)$$

(18), (19),

$$\boldsymbol{O}_{v(k:k+n-1)} = \begin{pmatrix} \boldsymbol{F}_k & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_{k+1} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}_{k+2} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{F}_{k+n-1} \end{pmatrix}, \quad (20)$$

$$\boldsymbol{y}_{(k:k+n-1)} = \begin{pmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \boldsymbol{y}_{k+2} \\ \vdots \\ \boldsymbol{y}_{k+n-1} \end{pmatrix}, \quad (21)$$

$$\boldsymbol{u}_{(k:k+n-1)} = \begin{pmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \boldsymbol{u}_{k+2} \\ \vdots \\ \boldsymbol{u}_{k+n-1} \end{pmatrix}, \quad (22)$$

$$\left[\boldsymbol{d}_{(k:k+n-2)}\right] = \boldsymbol{1}^{n,1}_{n_d}\left[\boldsymbol{d}\right] \quad (23)$$

and

$$\left[\boldsymbol{v}_{(k:k+n-2)}\right] = \boldsymbol{1}^{n,0}_{n_v}\left[\boldsymbol{v}\right]. \quad (24)$$

As stated in [11], the interval vector ${}^{o}[\boldsymbol{x}_k]$ that can be computed for $k \geq n-1$ can be used to improve the result of the prediction ${}^{p}[\boldsymbol{x}_k]$ by intersecting both interval vectors:

$${}^{c}[\boldsymbol{x}_k] = {}^{p}[\boldsymbol{x}_k] \cap {}^{o}[\boldsymbol{x}_k]. \quad (25)$$

Now, all the basic relations needed to define the complete algorithm are given. In the next subsection, these basic relations are combined to present the interval set-membership state estimation algorithm.

C. Interval Set-Membership State Estimation Algorithm

Algorithm 1 illustrates the entire SME. The first for-loop of this algorithm contains the prediction of the initial interval vector $[\boldsymbol{x}_0]$ for the first $n-1$ time steps. A further improvement of the state enclosure based on (16) is not possible for these steps due to the missing future values for the input and the output. For $k \geq n$, these future input and output values are known for a time step $j = k-(n-1)$ which motivates the initial time shift in the second for-loop. Afterwards, (16) is evaluated for j and its result is intersected with a prediction for this time step that has been calculated in a previous iteration of this for-loop or the first for-loop, respectively. The resulting interval vector ${}^{c}[\boldsymbol{x}_j] = {}^{c}\left[\boldsymbol{x}_{k-(n-1)}\right]$ for the time step lying $n-1$ steps in the past is then predicted to the current time step k by evaluating (15) resulting in ${}^{cp}[\boldsymbol{x}_k]$. Finally, the result of a one step prediction of the result of the previous iteration or the first for-loop is calculated and intersected with ${}^{cp}[\boldsymbol{x}_k]$.

$$\begin{aligned}&\boldsymbol{O}_{u(k:k+n-1)} \\ &= \begin{pmatrix} \boldsymbol{D}_k & \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}_{k+1}\boldsymbol{B}_k & \boldsymbol{D}_{k+1} & \dots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{C}_{k+2}\boldsymbol{A}_{k+1}\boldsymbol{B}_k & \boldsymbol{C}_{k+2}\boldsymbol{D}_{k+1} & \dots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{C}_{k+n-1}\boldsymbol{A}_{k+n-2}\dots\boldsymbol{A}_{k+1}\boldsymbol{B}_k & \boldsymbol{C}_{k+n-1}\boldsymbol{A}_{k+n-2}\dots\boldsymbol{A}_{k+2}\boldsymbol{B}_{k+1} & \dots & \boldsymbol{C}_{k+n-1}\boldsymbol{B}_{k+n-2} & \boldsymbol{D}_{k+n-1} \end{pmatrix}\end{aligned} \quad (18)$$

$$\begin{aligned}&\boldsymbol{O}_{d(k:k+n-1)} \\ &= \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{C}_{k+1}\boldsymbol{E}_k & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{C}_{k+2}\boldsymbol{A}_{k+1}\boldsymbol{E}_k & \boldsymbol{C}_{k+2}\boldsymbol{E}_{k+1} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{C}_{k+n-1}\boldsymbol{A}_{k+n-2}\dots\boldsymbol{A}_{k+1}\boldsymbol{E}_k & \boldsymbol{C}_{k+n-1}\boldsymbol{A}_{k+n-2}\dots\boldsymbol{A}_{k+2}\boldsymbol{E}_{k+1} & \dots & \boldsymbol{C}_{k+n-1}\boldsymbol{E}_{k+n-2} \end{pmatrix}\end{aligned} \quad (19)$$

for $k := 1$ *to* $k := n-1$ **do**

$$
\begin{aligned}
{}^{p}[\boldsymbol{x}_k] :=& \left(\prod_{i=k-1}^{0} \boldsymbol{A}_i\right)[\boldsymbol{x}_0] \\
&+\sum_{i=1}^{k}\left(\left(\boldsymbol{A}_k^{-1}\left(\prod_{z=k}^{i}\boldsymbol{A}_z\right)\right)^{1-\delta(i-k)} \dots \right.\\
&\qquad \left. \boldsymbol{B}_{i-1}\boldsymbol{u}_{i-1}\right) \\
&+\sum_{i=1}^{k}\left(\left(\boldsymbol{A}_k^{-1}\left(\prod_{z=k}^{i}\boldsymbol{A}_z\right)\right)^{1-\delta(i-k)} \boldsymbol{E}_{i-1}[\boldsymbol{d}]\right)
\end{aligned}
\tag{26a}
$$

$$
[\boldsymbol{x}_k] := {}^{p}[\boldsymbol{x}_k] \tag{26b}
$$

end

for $k := n$ *to* $k := N$ **do**

$$
j := k-(n-1) \tag{26c}
$$

$$
\begin{aligned}
{}^{o}[\boldsymbol{x}_j] := \boldsymbol{O}^{-1}_{(j:j+n-1)}\Big(&\boldsymbol{y}_{(j:j+n-1)} \\
&-\boldsymbol{O}_{u(j:j+n-1)}\boldsymbol{u}_{(j:j+n-1)} \\
&-\boldsymbol{O}_{d(j:j+n-1)}\boldsymbol{1}^{n,1}_{n_d}[\boldsymbol{d}] \\
&-\boldsymbol{O}_{v(j:j+n-1)}\boldsymbol{1}^{n,0}_{n_v}[\boldsymbol{v}]\Big)
\end{aligned}
\tag{26d}
$$

$$
{}^{c}[\boldsymbol{x}_j] := {}^{p}[\boldsymbol{x}_j] \cap {}^{o}[\boldsymbol{x}_j] \tag{26e}
$$

$$
\begin{aligned}
{}^{cp}[\boldsymbol{x}_k] :=& \left(\prod_{i=k-1}^{j} \boldsymbol{A}_i\right){}^{c}[\boldsymbol{x}_j] \\
&+\sum_{i=j+1}^{k}\left(\left(\boldsymbol{A}_k^{-1}\left(\prod_{z=k}^{i}\boldsymbol{A}_z\right)\right)^{1-\delta(i-k)} \dots \right.\\
&\qquad \left. \boldsymbol{B}_{i-1}\boldsymbol{u}_{i-1}\right) \\
&+\sum_{i=j+1}^{k}\left(\left(\boldsymbol{A}_k^{-1}\left(\prod_{z=k}^{i}\boldsymbol{A}_z\right)\right)^{1-\delta(i-k)} \boldsymbol{E}_{i-1}[\boldsymbol{d}]\right)
\end{aligned}
\tag{26f}
$$

$$
{}^{p}[\boldsymbol{x}_k] := \boldsymbol{A}_{k-1}[\boldsymbol{x}_{k-1}] + \boldsymbol{B}_{k-1}\boldsymbol{u}_{k-1} + \boldsymbol{E}_{k-1}[\boldsymbol{d}] \tag{26g}
$$

$$
[\boldsymbol{x}_k] := {}^{cp}[\boldsymbol{x}_k] \cap {}^{p}[\boldsymbol{x}_k] \tag{26h}
$$

end

Algorithm 1: Interval set-membership state estimation

In figure 1, an overview of algorithm 1 for $k \geq n$ is displayed. The calculations at time step k are given in the green box while the necessary input, output and parameter values are depicted by the yellow box. Additionally, the results of the calculations at time steps $k-n+1$ and $k-1$ that are used at time step k are displayed in the brown boxes. However, for the sake of brevity, the calculations at these time step are not depicted in detail. To see how the relevant variables are calculated, they are highlighted in the same color in the calculations at time step k.

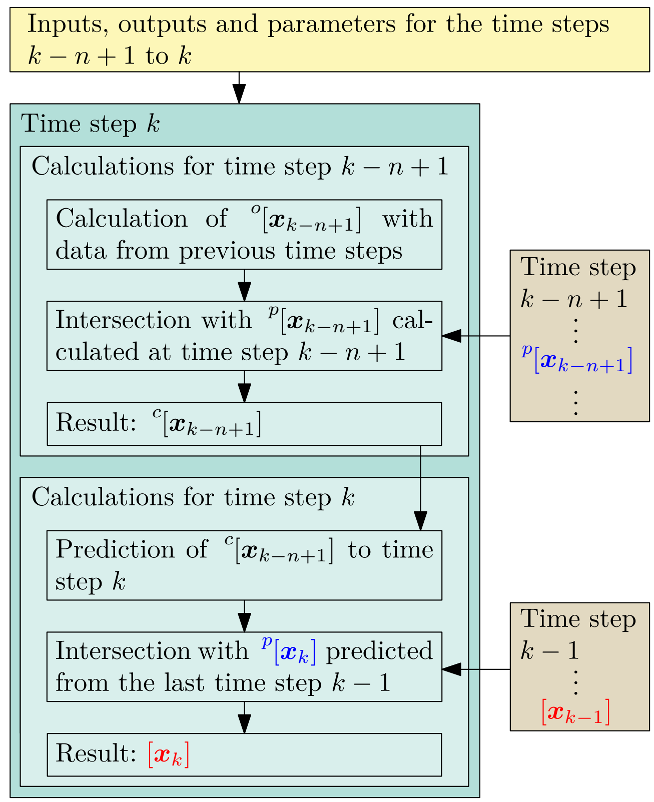

Fig. 1. Visualization of algorithm 1 for $k \geq n$

Based on this algorithm, we can can now state the central proposition of this publication.

Proposition 1: If assumptions 1 to 5 hold, then algorithm 1 provides interval vectors $[\boldsymbol{x}_k]$, $k \in \{1, 2, \dots, N\}$ with the following properties:

- The state of system (9) is always included in $[\boldsymbol{x}_k]$, i.e.

$$
\boldsymbol{x}_k \in [\boldsymbol{x}_k] \quad \forall k \geq 0. \tag{27}
$$

- For $k \geq n$, the width of the elements of the interval vector $[\boldsymbol{x}_k]$ is smaller than

$$W\left([\boldsymbol{x}_k]\right) \leq \max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\| \prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \|_\infty \right) \cdots$$
$$\max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\| \boldsymbol{O}_{(k-n+1:k)}^{-1} \boldsymbol{O}_{d(k-n+1:k)} \mathbf{1}_{n_d}^{n,1} \|_\infty \right) \cdot W\left([\boldsymbol{d}]\right)$$
$$+ \max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\| \prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \|_\infty \right) \cdots$$
$$\max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\| \boldsymbol{O}_{(k-n+1:k)}^{-1} \boldsymbol{O}_{v(k-n+1:k)} \mathbf{1}_{n_v}^{n,0} \|_\infty \right) \cdot W\left([\boldsymbol{v}]\right)$$
$$+ \max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\| \sum_{i=k-n+2}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \cdots \right. \right.$$
$$\left. \left. \boldsymbol{E}_{i-1} \right) \|_\infty \right) \cdot W\left([\boldsymbol{d}]\right). \tag{28}$$

Proof: The proof of proposition 1 is divided into two parts. Initially, (27) is proven to apply and afterwards (28) is proven to be valid.

To prove (27), the time steps $k \leq n-1$ and the time steps $k \geq n$ are analysed separately.

From (15) follows that (26a) is simply an inclusion function for (14) with $\boldsymbol{x}_s = \boldsymbol{x}_0$. Hence, taking into account assumption 3 yields (27) to be true for $k \leq n-1$.

For $k \geq n$, the state vector at a past time step j defined in (26c) can be computed by (16). Due to the fact that (16) respectively (26d) is an inclusion function for the equation to compute the state of an LPV system from future input and output values, its result is an overapproximation and thus includes the real state vector. The intersection of the resulting interval vector with the previous result of the for-loop or the first for-loop, respectively, in (26e) is not violating the inclusion property but improving the result with regard to the interval width if the rest of the proof for $k \geq n$ turns out well which shall be discussed a bit more in detail. For $k \leq 2(n-1)$, ${}^{p}[\boldsymbol{x}_j]$ is calculated by (26a) which has been already been proven to yield a valid result. Hence, if the rest of the algorithm, i.e. (26f), (26g) and (26h) is proven to guarantee (27), it is also proven that (26e) provides a result including $\boldsymbol{x}_j$ for all k. Thus, we proceed with the analysis of (26f) which is an inclusion function for the recursive evaluation of (9). Obviously the validity of this equation is thus given for the same reasons as explained for (26a). The following equation (26g) is an inclusion function for (9a), i.e. a one step prediction of the previous result of the for-loop or the first for-loop, respectively. Therefore, it can be concluded that if the previous result is including $\boldsymbol{x}_k$, then (26g) contains $\boldsymbol{x}_k$ as well. As already explained in the context of (26e), $\boldsymbol{x}_k \in [\boldsymbol{x}_k]$ has already been proven for $k \leq n-1$. Hence, if the last remaining equation to be analysed (26h) is proven to guarantee (27), then (26g) provides a valid estimate for all k. Because of (26h) only discarding inconsistent values from the interval vectors ${}^{cp}[\boldsymbol{x}_k]$ and ${}^{p}[\boldsymbol{x}_k]$, the proof of (27) is completed.

The proof of (28) is based on the neglection of (26h) and the intersection (26e) by using ${}^{c}[\boldsymbol{x}_j] = {}^{o}[\boldsymbol{x}_j]$. This means that an overapproximation of the maximum interval width of the interval vector $[\boldsymbol{x}_k]$ is calculated based on the assumption that the estimation is solely calculated based on future input and output values. Hence, (29) follows.

$$\begin{aligned}
[\boldsymbol{x}_k] &\subseteq {}^{cp}[\boldsymbol{x}_k] \\
&\subseteq \left(\prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \right) \boldsymbol{O}_{(k-n+1:k)}^{-1} \Big(\boldsymbol{y}_{(k-n+1:k)} \\
&\quad - \boldsymbol{O}_{u(k-n+1:k)} \boldsymbol{u}_{(k-n+1:k)} - \boldsymbol{O}_{d(k-n+1:k)} \mathbf{1}_{n_d}^{n,1} [\boldsymbol{d}] \\
&\quad - \boldsymbol{O}_{v(k-n+1:k)} \mathbf{1}_{n_v}^{n,0} [\boldsymbol{v}] \Big) \\
&\quad + \sum_{i=k-n+2}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \cdots \right. \\
&\quad \left. \boldsymbol{B}_{i-1} \boldsymbol{u}_{i-1} \right) \\
&\quad + \sum_{i=k-n+2}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \boldsymbol{E}_{i-1} [\boldsymbol{d}] \right)
\end{aligned} \tag{29}$$

In the following calculation of the interval width $w([\boldsymbol{x}_k])$, the terms highlighted in (29) do not contribute to the interval width because they do not include any interval-valued quantities. This leads to

$$w([\boldsymbol{x}_k]) \leq \mathrm{ub}(\boldsymbol{\alpha}_k) - \mathrm{lb}(\boldsymbol{\alpha}_k) + \mathrm{ub}(\boldsymbol{\beta}_k) - \mathrm{lb}(\boldsymbol{\beta}_k) \tag{30}$$

with

$$\boldsymbol{\alpha}_k = \left(\prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \right) \boldsymbol{O}_{(k-n+1:k)}^{-1} \cdots$$
$$\left(-\boldsymbol{O}_{d(k-n+1:k)} \mathbf{1}_{n_d}^{n,1} [\boldsymbol{d}] - \boldsymbol{O}_{v(k-n+1:k)} \mathbf{1}_{n_v}^{n,0} [\boldsymbol{v}] \right), \tag{31a}$$

$$\boldsymbol{\beta}_k = \sum_{i=k-n+2}^{k} \left(\left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \cdots \right.$$
$$\left. \boldsymbol{E}_{i-1} [\boldsymbol{d}] \right). \tag{31b}$$

By applying (7) and (10), an overapproximation of the interval width is

$$W([\boldsymbol{x}_k]) \leq \max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\left(\| \prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \|_\infty \right) \cdot W \Big(\boldsymbol{O}_{(k-n+1:k)}^{-1} \cdots \right.$$
$$\left. \left(-\boldsymbol{O}_{d(k-n+1:k)} \mathbf{1}_{n_d}^{n,1} [\boldsymbol{d}] - \boldsymbol{O}_{v(k-n+1:k)} \mathbf{1}_{n_v}^{n,0} [\boldsymbol{v}] \right) \Big) \right)$$
$$+ \max_{\boldsymbol{\theta}_k \in [\boldsymbol{\theta}]} \left(\| \sum_{i=k-n+2}^{k} \left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \cdots \right.$$
$$\left. \boldsymbol{E}_{i-1} \|_\infty \cdot W([\boldsymbol{d}]) \right) \tag{32a}$$

$$\leq \max_{\boldsymbol{\theta}_k\in[\boldsymbol{\theta}]} \left(\left(\left(\| \prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \|_\infty \right) \cdot \| \boldsymbol{O}^{-1}_{(k-n+1:k)} \cdots \right.\right.$$

$$\left. \boldsymbol{O}_{d(k-n+1:k)} \boldsymbol{1}^{n,1}_{n_d} \|_\infty \right) \cdot W\left([\boldsymbol{d}]\right)$$

$$+ \max_{\boldsymbol{\theta}_k\in[\boldsymbol{\theta}]} \left(\left(\left(\| \prod_{i=k-1}^{k-n+1} \boldsymbol{A}_i \|_\infty \right) \cdot \| \boldsymbol{O}^{-1}_{(k-n+1:k)} \cdots \right.\right.$$

$$\left. \boldsymbol{O}_{v(k-n+1:k)} \boldsymbol{1}^{n,0}_{n_v} \|_\infty \right) \cdot W\left([\boldsymbol{v}]\right)$$

$$+ \max_{\boldsymbol{\theta}_k\in[\boldsymbol{\theta}]} \left(\| \sum_{i=k-n+2}^{k} \left(\boldsymbol{A}_k^{-1} \left(\prod_{z=k}^{i} \boldsymbol{A}_z \right) \right)^{1-\delta(i-k)} \cdots \right.$$

$$\left. \boldsymbol{E}_{i-1} \|_\infty \right) \cdot W\left([\boldsymbol{d}]\right) \tag{32b}$$

As the product of two positive functions can be maximized by maximizing each function separately, (32b) leads to (28). ■

V. Numerical Example

To demonstrate the effectiveness of the method, it is applied to a numerical example defined by

$$\boldsymbol{A}_k = \begin{pmatrix} -0.4 & 1.2+\theta_k \\ 0 & 0.8 \end{pmatrix}, \tag{33a}$$

$$\boldsymbol{B} = \boldsymbol{E} = \begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}, \tag{33b}$$

$$\boldsymbol{C} = \begin{pmatrix} -0.5 & 1 \end{pmatrix}, \tag{33c}$$

$$D = F = 0.4. \tag{33d}$$

A. Scenario

The initial value $\boldsymbol{x}_0$ of (9) is chosen as $\boldsymbol{x}_0^\intercal = \begin{pmatrix} 1 & 2 \end{pmatrix}$ and the input is chosen as $u_k = \sin\left(4\pi\frac{k}{10000}\right)$. θ_k is a time-variant parameter with $\theta_k = \sin\left(2\pi\frac{k}{10000}\right)$. The disturbances are defined as constants with $v = 0.1$ and $d = 0.3$. For the uncertain quantities, the only a priori known information is $\boldsymbol{x}_0^\intercal \in \begin{pmatrix} [0.5, 1.5] & [1.5, 2.5] \end{pmatrix}$, $\theta_k \in [-1, 1]$, $v \in [0, 0.2]$ and $d \in [0.2, 0.4]$.

Finally, due to the fact that the rank of the observability matrix

$$\begin{pmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A}_k \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} & 1 \\ \frac{1}{5} & \frac{1}{5} - \frac{1}{2}\theta_k \end{pmatrix} \tag{34}$$

is equal to 2 for all $\theta_k \in [-1, 1]$, algorithm 1 can be applied.

B. Simulation Results

The simulation results that are obtained with the scenario defined in the previous subsection are depicted in figure 2. It can be seen that the real state values drawn with the black solid line are always bounded by the upper bound (blue dashed line) and the lower bound (red dash dotted line) of the estimation. The maximum interval width of $[x_1]$ is 1.62 and the maximum interval width of $[x_2]$ is 1.0. A comparison with the result of (28) which is 7.008 shows that the a priori known bound of the interval width is in the same range. Hence, the numerical simulation supports the theoretical results presented in proposition 1.

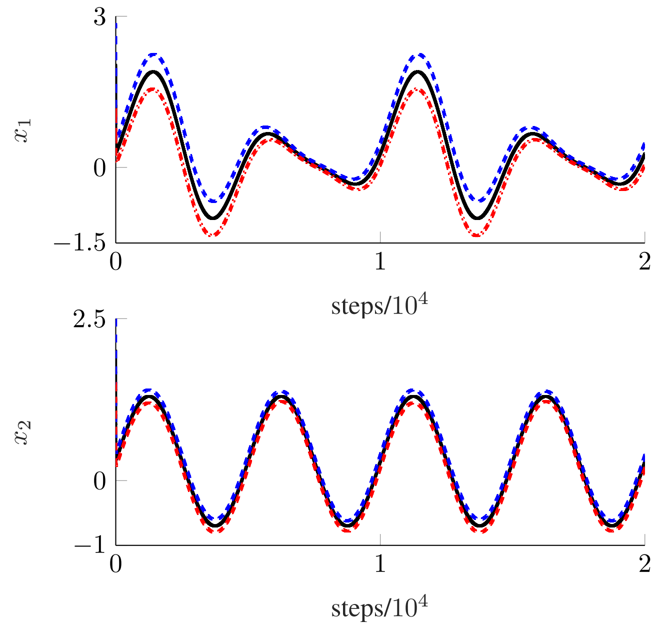

Fig. 2. Result of algorithm 1 for the example (blue dashed: upper bound of the estimation, red dash dotted: lower bound of the estimation, black solid: real value)

VI. Conclusion

Inspired by [11], the main contribution of this publication is the extension of an interval set-membership state estimation algorithm to LPV systems. In contrast to existing set-membership state estimation techniques for LPV systems, the presented approach can guarantee an a priori computable accuracy of the interval estimation. A possible benefit of this a priori information is that in a fault diagnosis environment, statements concerning the detectability of faults can be made a priori. A numerical simulation has shown that the method provides reasonable interval widths of the estimates even for a comparatively high uncertainty of the disturbances influencing the state vector and the output vector defined by ± 100 % and $\pm 33.\overline{3}$ % of the true values, respectively.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[2] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of the American Control Conference*, vol. 3, 1995, pp. 1628–1632.

[3] E. A. Wan and R. V. D. Merwe, "The unscented kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[4] C. Combastel, "A state bounding observer based on zonotopes," in *Proceedings of the European Control Conference*, 2003, pp. 2589–2594.

[5] F. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 22–28, 1968.

[6] M. Kieffer, I. Braems, E. Walter, and L. Jaulin, "Guaranteed set computation with subpavings," in *Scientific Computing, Validated Numerics, Interval Methods*. Springer, 2001, pp. 167–178.

[7] J. Gouzé, A. Rapaport, and M. Hadj-Sadok, "Interval observers for uncertain biological systems," *Ecological Modelling*, vol. 133, pp. 45 – 56, 2000.

[8] J. Heeks, E. Hofer, B. Tibken, and K. Thorwart, "An interval arithmetic approach for discrete time non-smooth uncertain systems with application to an anti-locking system," in *Proceedings of the American Control Conference*, vol. 3, 2002, pp. 1849–1854.

[9] A. Rauh, M. Kletting, H. Aschemann, and E. Hofer, "Interval methods for simulation of dynamical systems with state-dependent switching characteristics," in *Proceedings of the International Conference on Control Applications*, 2006, pp. 355–360.

[10] M. Kieffer and E. Walter, "Guaranteed nonlinear state estimator for cooperative systems," *Numerical Algorithms*, vol. 37, pp. 187–198, 2004.

[11] N. Meslem and N. Ramdani, "Forward-backward set-membership state estimator based on interval analysis," in *Proceedings of the American Control Conference*, 2018, pp. 5161–5166.

[12] S. Krebs, M. Pfeifer, S. Fugel, J. Weigold, and S. Hohmann, "Interval observer for LPV systems based on time-variant transformations," in *Proceedings of the Conference on Decision and Control*, 2016, pp. 4090–4096.

[13] N. Meslem, N. Ramdani, and Y. Candau, "Using hybrid automata for set-membership state estimation with uncertain nonlinear continuous-time systems," *Journal of Process Control*, vol. 20, no. 4, pp. 481 – 489, 2010.

[14] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.

[15] D. Efimov, L. Fridman, T. Raïssi, A. Zolghadri, and R. Seydou, "Interval estimation for LPV systems applying high order sliding mode techniques," *Automatica*, vol. 48, no. 9, pp. 2365 – 2371, 2012.

[16] M. Witczak, V. Puig, D. Rotondo, and P. Witczak, "A necessary and sufficient condition for total observability of discrete-time linear time-varying systems," *Proceedings of the IFAC World Congress*, vol. 50, no. 1, pp. 729 – 734, 2017.