Data-driven Input Reconstruction and Experimental Validation

Jicheng Shi, Yingzhao Lian*, and Colin N. Jones

***Abstract*— This paper addresses a data-driven unknown input reconstruction (UIR) problem based on the Willems' Fundamental Lemma, where the unknown inputs are estimated recursively by the UIR only based on the output measurements. A recursive UIR and a moving-horizon UIR are developed based respectively on Lyapunov conditions and Luenberger-observer-type feedback, and their asymptotic convergence properties are studied. An experimental study is presented demonstrating the efficacy of the moving-horizon UIR for estimating the occupancy of a building on the EPFL campus via measured carbon dioxide levels.**

I. INTRODUCTION

Input reconstruction estimates unknown inputs based on measured states/outputs, which finds broad application in system supervision, sensor fault detection, and robust control [1]–[3]. This problem is of particular interest when the real-time/online measurement of inputs is not affordable or is privacy-sensitive. For example, a critical factor in predicting the evolution of the thermal state of a building is the number of occupants, but this value can often not be measured directly by cameras or Wi-Fi due to privacy or cost concerns. Instead, an indirect estimation is commonly deployed based on the measurement of indoor CO2 levels [4]. Another important example is the cutting force of machine tools, whose measurement is only feasible with a dedicated laboratory setup [5].

The input reconstruction problem has been studied in a model-based setup, and various methods have been proposed based on the unknown input observer (UIO) [6], [7], optimal filters [8], the generalized inverse approach [9] and sliding mode observers [5]. UIO is of special interest in our study, and most methods fall into two categories in the model-based setup. In one class of methods, system states are measured or estimated, and are further used to reconstruct the unknown input by matrix inversion [6] or matrix pencil decomposition [1]. In another category of methods, states and unknown inputs are estimated concurrently, and this estimate can achieve finite step convergence [9].

In practice, the model of the targeted system is usually not available. Instead of running a system identification procedure, the Willems' fundamental lemma offers a direct characterization of the system responses with an informative historical dataset [10]. This characterization provides a convenient interface to data-driven methods, and has been deployed in output prediction [11] and in controller design [12]–[17].

We applly Willems' fundamental lemma to enable direct input reconstruction with historical data. A similar setup was studied in [18], where the system states are assumed

Still weird if there are people who use input reconstruction method to mean something else (do they also assume only output measurements?) ... how about "unknown input observer from outputs" UIO^2 or UIOO ?
or "input reconstruction method from outputs" IRO ?
as these would be close to standard, but with a slight "O" emphasis?

known, and the unknown input reconstructor (UIR), instead of the unknown input observer (UIO). In this paper, we propose two design schemes of a data-driven stable UIR.

The word 'estimator' and 'observer' are generally synonyms... I'm not sure that people will see that this change of word means that we're not measuring the state... I think we might need a different name, or just call it an UIO and emphasize that it doesn't need state measurement

'UIE' exists in some paper but not well-known. 'UIO' is more popular but always indicates the state estimation. In some highly cited paper using they directly call it input reconstruction method. Then I think we can call it unknown input reconstructor (UIR)

In the following, Section II reviews output prediction based on the Willems' Fundamental Lemma, alongside the statement of the UIR problem. The design of a stable data-driven recursive UIR is proposed in Section III, followed by its moving-horizon counterpart in Section IV. The proposed UIRs are validated in Section V by simulations and an occupancy estimation experiment in a real-world building, followed by a conclusion in Section VI.

Notation: $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ denotes an identity matrix. The number of columns and rows of a matrix M are denoted respectively by n_M and m_M such that $M \in \mathbb{R}^{n_M\times m_M}$. Accordingly, $\mathrm{Null}(M)$ denotes its null space. $M^g := \{X|MXM = M\}$ is the set of generalized inverses of matrix M. A strictly positive definite matrix M is denoted by $M \succ 0$. The dimension of a vector s denoted by n_s. Given an ordered sequence of vectors $\{s_t, s_{t+1}, \ldots, s_{t+L}\}$, its vectorization is denoted by $s_{t:t+L} = [s_t^\top, \ldots, s_{t+L}^\top]^\top$.

II. PRELIMINARIES

A discrete-time linear time-invariant (LTI) system, dubbed $\mathcal{B}(A, B, C, D)$, is defined by

$$x_{t+1} = Ax_t + Bu_t\,, y_t = Cx_t + Du_t\,, \tag{1}$$

State that author one received support from NCCR and author two from RISK...

oject has received support from the Swiss National Science Foun-er the RISK project (Risk Aware Data-Driven Demand Response), ber 200021 175627, and and under the NCCR Automation project, ement 51NF40_180545

. corresponding author

JS, YL and CNJ are with Automatic Laboratory, Ecole Polytechnique Federale de Lausanne, Switzerland. {jicheng.shi, yingzhao.lian, colin.jones}@epfl.ch

whose states, inputs and outputs are denoted by $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_y}$ respectively. The order of the system is defined as $\mathbf{n}(\mathcal{B}(A,B,C,D)) := n_x$. The lag of the system $\mathbf{l}(\mathcal{B}(A,B,C,D))$ is defined as the smallest integer ℓ with which its observability matrix $\mathcal{O}_\ell := \left[C^\top, (CA)^\top, \ldots, \left(CA^{\ell-1}\right)^\top\right]^\top$ has rank n_x. The set of all vectors $[u_{1:L}; y_{1:L}]$, where $[u_{1:L}; y_{1:L}]$ is an L-step I/O trajectory generated by the system $\mathcal{B}(A,B,C,D)$, is denoted by $\mathcal{B}_L(A,B,C,D)$.

Definition 1: A Hankel matrix of depth L constructed by a vector-valued signal sequence $s := \{s_i\}_{i=1}^T$, $s_i \in \mathbb{R}^{n_s}$ is

$$\mathfrak{H}_L(s) := \begin{bmatrix} s_1 & s_2 & \ldots & s_{T-L+1} \\ s_2 & s_3 & \ldots & s_{T-L+2} \\ \vdots & \vdots & & \vdots \\ s_L & s_{L+1} & \ldots & s_T \end{bmatrix} .$$

Given a sequence of input-output measurements $\{u_{\mathbf{d},i}, y_{\mathbf{d},i}\}_{i=1}^T$, the input sequence is called *persistently exciting* of order L if $\mathfrak{H}_L(u_\mathbf{d})$ is full row rank. By building the following stacked Hankel matrix

$$\mathfrak{H}_L(u_\mathbf{d}, y_\mathbf{d}) := \begin{bmatrix} \mathfrak{H}_L(u_\mathbf{d}) \\ \mathfrak{H}_L(y_\mathbf{d}) \end{bmatrix} ,$$

we state **Willems' Fundamental Lemma** as

Lemma 1: [10, Theorem 1] Consider a controllable linear system $\mathcal{B}(A,B,C,D)$ and assume $\{u_{\mathbf{d},i}\}_{i=1}^T$ is persistently exciting of order $L + \mathbf{n}(\mathcal{B}(A,B,C,D))$. The condition $\text{colspan}(\mathfrak{H}_L(u_\mathbf{d}, y_\mathbf{d})) = \mathcal{B}_L(A,B,C,D)$ holds.

There's a few places where you're mixing up mathfrak and mathcal. For example, I think that one of the mathfrak's in the lemma above should be a mathcal? (JS: done.)

In the rest of the paper, the subscript $_\mathbf{d}$ marks a data point from the historical dataset collected offline, and L is reserved for the length of the system response.

The characterization of system response by Lemma 1 is used to develop a data-driven output prediction [11], [12]. In [11], the N_{pred}-step output prediction $\bar{y}_{t+1:t+N_{pred}}$ driven by an N_{pred}-step predicted output $u_{t+1:t+N_{pred}}$ is given by the solution to the following equations at time t:

$$\begin{bmatrix} \mathfrak{H}_{L,init}(u_\mathbf{d}) \\ \mathfrak{H}_{L,init}(y_\mathbf{d}) \\ \mathfrak{H}_{L,pred}(u_\mathbf{d}) \end{bmatrix} g = \begin{bmatrix} u_{t-N_{init}+1:t} \\ y_{t-N_{init}+1:t} \\ u_{t+1:t+N_{pred}} \end{bmatrix} \tag{2a}$$

$$\mathfrak{H}_{L,pred}(y_\mathbf{d}) =: \bar{y}_{t+1:t+N_{pred}} . \tag{2b}$$

Two sub-Hankel matrices of the output are defined by

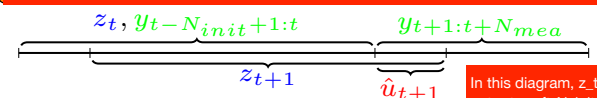
$$\mathfrak{H}_L(y_\mathbf{d}) = \begin{bmatrix} \mathfrak{H}_{L,init}(y_\mathbf{d}) \\ \mathfrak{H}_{L,pred}(y_\mathbf{d}) \end{bmatrix} , \tag{3}$$

and each of them is of depth N_{init} and N_{pred} respectively, such that $N_{init} + N_{pred} = L$. Similarly, the Hankel matrices $\mathfrak{H}_{L,init}(u_\mathbf{d})$ and $\mathfrak{H}_{L,pred}(u_\mathbf{d})$ are constructed. Last but not least, the solution to (2) is well-defined if $N_{init} \geq \mathbf{l}(\mathcal{B}(A,B,C,D))$. Specifically, this condition implies that $\{u_{t-N_{init}+1:t}, y_{t-N_{init}+1:t}\}$, the N_{init}-step input output sequences preceding the current point of time, can uniquely determine the underlying state x_t. Readers are referred to [11] for more details.

A. Problem Statement and Inspiration

Change the time index. Maybe use an example of t=0. Use N_{mea} rather than N_{mea}

The colors here are pretty bad. Can you pick a set of good colors to use for the equations below? (Perhaps to match the plots at the end of the paper?) These colors are from the 1990's when red green and blue was all that monitors could do ;)



In this diagram, z_t looks like it's the outputs y_{t-N_init+1:t} A bit confusing. Possible drawing on the next page

Fig. 1: Diagram of input estimation at time $t+1$. Input estimate $\hat{u}_t$ after $y_{t+N_{mea}}$ is measured at time $t+N_{mea}$. Differently, out is computed immediately by (2), if given u_{t+1}.

Recall the LTI dynamics (1), and we assume that we have an offline I/O dataset $\{u_\mathbf{d}, y_\mathbf{d}\}$. During online operation the inputs are not mea this work studies the r inputs from the measure estimate is generated (UIR), whose structure

For estimation it's fairly usual to write the dynamics as z_t = Az_{t-1} + B d_{t-1}
ut = Cz_t

This is because we would normally state that we're trying to estimate the input at the current time t, rather than at t+1.

$$\begin{aligned} z_{t+1} &= A_{UIR} z_t + B_{UIR} d_t , \\ \hat{u}_t &= [\mathbf{0}\ \mathbf{I}_{n_u}] z_t , \end{aligned} \tag{4}$$

where $z_t := [\hat{u}_{t-N_{init}+1}^\top, \ldots, \hat{u}_t^\top]^\top$ is a vectorized N_{init}-step unknown input estimate, and $d_t := y_{t-N_{init}+1:t+N_{mea}}$ is the sequence of output measurements, as depicted in Figure 1. We leave the discussion about N_{mea} to Section III. Furthermore, a UIR is **stable** if $\lim_{t\to\infty} \hat{u}_t - u_t \to 0$ for any initial guess z_0. Note that since z_t is the sequence of N_{init}-step unknown input estimates, it is reasonable to design an observable canonical form based UIR, such that the recursive estimator only updates the last unknown input in z_t (i.e. $\hat{u}_t$), and we term a UIR of this form a recursive UIR (R-UIR). Otherwise, it is called a moving-horizon UIR (MH-UIR).

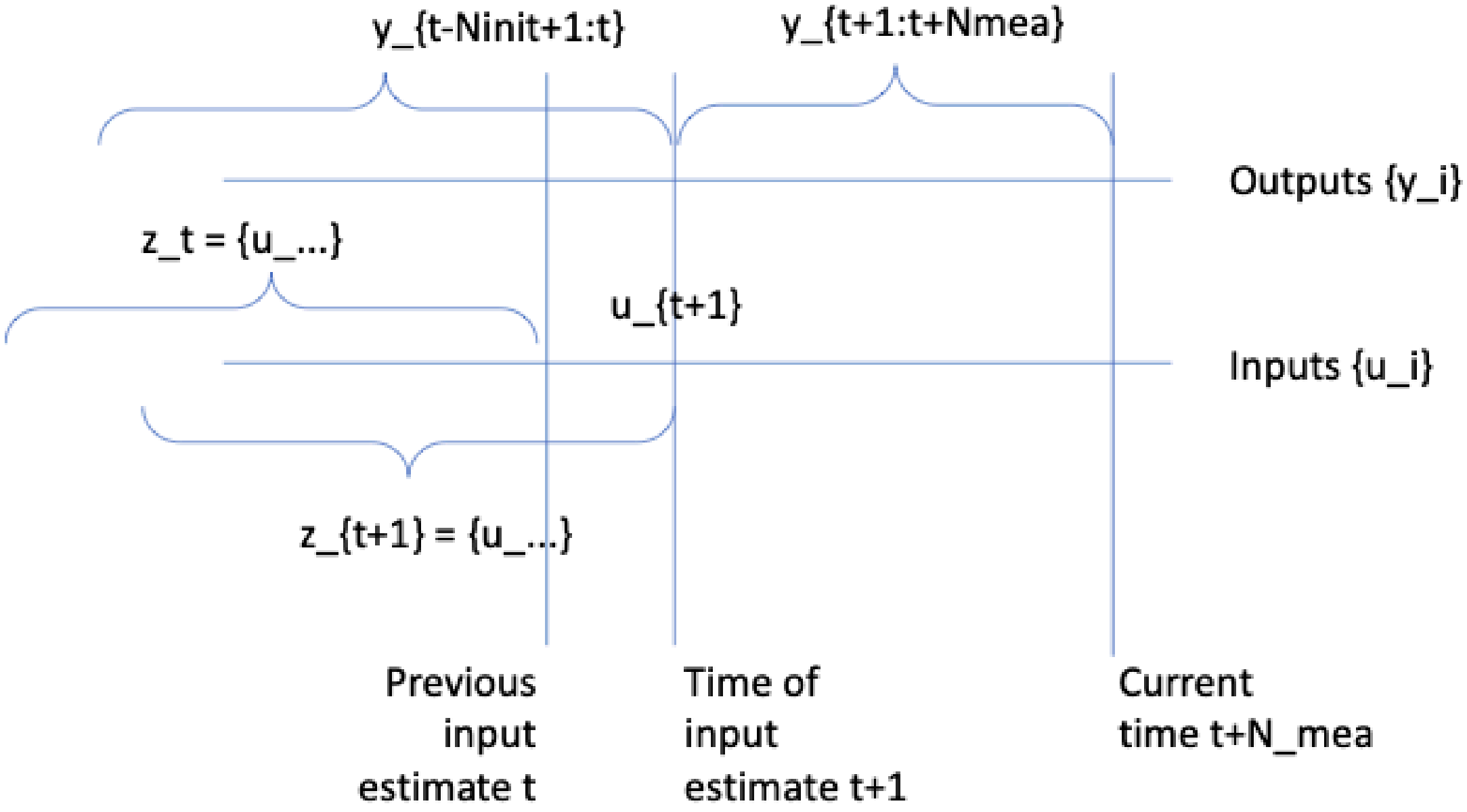
I'm not sure that the names recursive and moving-horizon really make the most sense here... perhaps something more like recursive and moving-horizon for these two concepts, since the ideas are very similar to a standard recursive estimator vs a moving-horizon estimator?
(JS: agree)

The goal of this work is to design the UIR components (i.e. A_{UIR} and B_{UIR}) directly from data $\{u_\mathbf{d}, y_\mathbf{d}\}$. Inspired by the data-driven output prediction (2), it is reasonable to formulate a similar data-driven input estimation scheme:

$$\begin{bmatrix} \mathfrak{H}_{L,init}(u_\mathbf{d}) \\ \mathfrak{H}_{L,init}(y_\mathbf{d}) \\ \mathfrak{H}_{L,mea}(y_\mathbf{d}) \end{bmatrix} g = \begin{bmatrix} u_{t-N_{init}+1:t} \\ y_{t-N_{init}+1:t} \\ y_{t+1:t+N_{mea}} \end{bmatrix} \tag{5a}$$

$$\mathfrak{H}_{L,est}(u_\mathbf{d}) g =: \bar{u}_{t+1} , \tag{5b}$$

where sub-Hankel matrices $\mathfrak{H}_{L,init}(u_\mathbf{d})$, $\mathfrak{H}_{L,init}(y_\mathbf{d})$, $\mathfrak{H}_{L,mea}(y_\mathbf{d})$, $\mathfrak{H}_{L,mea}(u_\mathbf{d})$ follow a similar splitting definition


y_{t-Ninit+1:t}
y_{t+1:t+Nmea}
Outputs {y_i}
z_t = {u_...}
u_{t+1}
Inputs {u_i}
z_{t+1} = {u_...}
Previous input estimate t
Time of input estimate t+1
Current time t+N_mea

in (3) such that $N_{init} + N_{mea} = L$, and $\mathfrak{H}_{L,est}(u_{\mathbf{d}})$ denotes the first n_u rows of $\mathfrak{H}_{L,mea}(u_{\mathbf{d}})$. However, this scheme (5) is numerically not implementable, as input measurements in (5a) are not available. The key idea of this work is to fit this scheme (5) into the general UIR structure (4).

Remark 1: In the rest of the paper, $\bar{u}_{t+1}$ indicates the input estimate by (5) given the actual $u_{t-N_{init}+1:t}$. $\hat{u}_{t+1}$ denotes the input estimate by (4) and z_{t+1}.

III. Data-Driven R-UIR

In this section, we present the proposed R-UIR and its design approach. We will first summarize its standard formulation and its stability property in Theorem 1, whose proof is later accomplished by Lemma 2, Lemma 3 and Lyapunov conditions. The design of a standard R-UIR in Theorem 1 suffers from NP-hardness, and so we offer a tractable reformulation by LMI tightening.

The key idea of the R-UIR fomulation is to substitute $u_{t-N_{init}+1:t}$ in (5a) by its recursive input estimate $\hat{u}_{t-N_{init}+1:t} =: z_t$ in the UIR (4). For the sake of clarity, the notation in (5) is simplified using the notation

$$H := \begin{bmatrix} \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,mea}(u_{\mathbf{d}}) \end{bmatrix}, \quad b := \begin{bmatrix} u_{t-N_{init}+1:t} \\ y_{t-N_{init}+1:t} \\ y_{t+1:t+N_{mea}} \end{bmatrix}, \quad H_u := \mathfrak{H}_{L,est}(u_{\mathbf{d}})$$

We state the set of data-driven R-UIR candidates by

$$\mathcal{U}_R := \left\{ \begin{array}{l|l} A_{UIR}, & A_{UIR} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{(N_{init}-1)n_u} \\ & H_u G_u \end{bmatrix}, \\ B_{UIR} & B_{UIR} = \begin{bmatrix} \mathbf{0} \\ H_u G_y \end{bmatrix}, \ \forall\, [G_u, G_y] = G \in H^g \end{array} \right\} \quad (6)$$

where G_u and G_y partitions any generalized inverse G, and respectively consists of $N_{init} n_u$ and $(N_{init} + N_{mea}) n_y$ columns.

Assumption 1: The historical input signals $\{u_{\mathbf{d},i}\}_{i=1}^{T}$ are persistently exciting of order $N_{init} + N_{mea} + \mathbf{n}(\mathcal{B}(A, B, C, D))$.

Theorem 1 (Stability of R-UIR): Let condition

$$\mathrm{Null}(H) \subseteq \mathrm{Null}(H_u)\,, \quad (7)$$

and Assumption 1 hold. A UIR in $\mathcal{U}_R$ is stable if and only if the following optimization problem is feasible

Just state as a feasibility problem, rather than minimize 0
i.e., the set of stable UIR is given by {(Auir,Buir) in Ur | exists W >= 0 [xxx]}

$$\begin{aligned} \underset{W \succ 0, A_{UIR}, B_{UIR}}{\text{minimize}} \quad & 0 \\ \text{s.t. } & A_{UIR}\,, B_{UIR} \in \mathcal{U}_R \quad (8a) \\ & \begin{bmatrix} W & A_{UIR} W \\ W A_{UIR}^\top & W \end{bmatrix} \succ 0\,. \quad (8b) \end{aligned}$$

The rest of this subsection will show how (6) and Thereom 1 are developed. Assumption 1 ensures that the Hankel matrix H constructed by $\{u_{\mathbf{d}}, y_{\mathbf{d}}\}$ is sufficiently informative such that Lemma 1 guarantees that $b \in \mathrm{colspan}(H)$. Therefore, the solution set to $Hg = b$ is non-empty, and it can be characterized as

$$\mathcal{T}(b) := \{g | g = Gb + \nu,\ G \in H^g,\ \nu \in \mathrm{Null}(H)\}\,. \quad (9)$$

Accordingly, $\bar{u}_{t+1}$ in (5b) is given by

$$\bar{u}_{t+1} = \{H_u g | g \in \mathcal{T}(b)\}\,. \quad (10)$$

However, the solution set (9) is not a singleton and therefore $\bar{u}_{t+1}$ is not necessarily unique. To ensure uniqueness, we give the following lemma.

Lemma 2: If Assumption 1 holds, the set (10) is a singleton if and only if the condition (7) holds.

Proof: ($\Rightarrow$) For any solutions $g_1,\ g_2 \in \mathcal{T}(b)$ to $Hg = b$, we have $Hg_1 - Hg_2 = H(g_1 - g_2) = 0$, which indicates $(g_1 - g_2) \in \mathrm{Null}(H)$ Therefore, by $\mathrm{Null}(H) \subseteq \mathrm{Null}(H_u)$, $H_u g_1 - H_u g_2 = 0$. Due to the arbitrariness of g_1 and g_2, u_t defined in (10) is a singleton. ($\Leftarrow$) For any $G \in H^g$, $\nu \in \mathrm{Null}(H)$, $H_u(Gb + \nu) - H_u Gb = 0$ because $\hat{u}_t$ by (10) is a singleton. This indicates $H_u \nu = 0, \forall \nu \in \mathrm{Null}(H)$ and therefore $\mathrm{Null}(H) \subseteq \mathrm{Null}(H_u)$. ∎

If the condition (7) holds, the effect of null space $\mathrm{Null}(H)$ in $\mathcal{T}(b)$ can be neglected. Next, by substituting $u_{t-N_{init}+1:t}$ in (10) by z_t, we formulate the input reconstruction by:

$$\forall\, G \in H^g,\ \hat{u}_{t+1} := H_u G [z_t^\top\ d_t^\top]^\top. \quad (11)$$

It is not difficult to show that its observable canonical form results in the set of data-driven R-UIR candidates (6).

Remark 2: The choices of N_{mea} for output measurement depends on the properties of matrices $\{B, C, D\}$ in (1), which intuitively reflects how soon all the entries of inputs can affect the output. For example, if D is full column rank, the effect from the input to output is instantaneous and thus N_{mea} can be set to one. A model-based discussion about N_{mea} can be found in [6] and [19]. The condition (7) in Lemma 2 gives a data-driven criterion of N_{mea} selection, which intuitively states that the variation in input will always change the output, as any $g \notin \mathrm{Null}(H_u)$ is not in $\mathrm{Null}(H)$.

Note that an R-UIR defined by a random element in set $\mathcal{U}_R$ (6) is not necessarily stable. To find a stable R-UIR, we first characterize its stability by the following lemma.

Lemma 3: Let Assumption 1 and condition (7) hold, a UIR in $\mathcal{U}_R$ is stable if and only if A_{UIR} is Schur.

Proof: If Assumption 1 and condition (7) holds, Lemma 1 and 2 guarantee that $\bar{u}_{t+1} = u_{t+1}$ in (10). Therefore $\forall\ A_{UIR},\ B_{UIR} \in \mathcal{U}_R$, (10) is equivalent to

$$\begin{aligned} u_{t-N_{init}+2:t+1} &= A_{UIR} u_{t-N_{init}+1:t} + B_{UIR} d_t\,, \\ u_t &= [\mathbf{0}\ \mathbf{I}_{n_u}] u_{t-N_{init}+1:t}\,. \end{aligned}$$

Thus, we have

$$\begin{aligned} \lim_{t\to\infty} \hat{u}_t - u_t &= \lim_{t\to\infty} [\mathbf{0}\ \mathbf{I}_{n_u}] A_{UIR} (z_{t-1} - u_{t-N_{init}:t-1}) \\ &= \lim_{t\to\infty} [\mathbf{0}\ \mathbf{I}_{n_u}] A_{UIR}^t (z_0 - u_{-N_{init}+1:0}). \end{aligned}$$

The above equation converges to 0 if and only if A_{UIR} is Schur stable, and we conclude the proof. ∎

The Schur stability criterion can be validated via a semidefinite program [20, Chapter 3.3] as

$$A_{UIR} \text{ SCHUR STABLE} \iff \left\{ \begin{array}{l} \exists\, W \succ 0 \\ \begin{bmatrix} W & A_{UIR} W \\ W A_{UIR}' & W \end{bmatrix} \succ 0. \end{array} \right. \quad (12)$$

It naturally leads to the feasibility problem (8), and we complete the proof of Theorem 1.

A. Design of data-driven R-UIR

Theorem 1 simultaneously gives the design of a data-driven R-UIR by the feasibility problem (8). However, this optimization problem is NP-hard due to the bilinear matrix (BMI) inequality (8b) [21]. In the rest of this section, we will tighten this BMI into a tractable linear matrix inequality (LMI) [22] and characterize the set of generalized inverse H^g in U_R via singular value decomposition (SVD).

1) Characterization of Generalized Inverse: Denote the SVD of matrix H by $H = U\begin{bmatrix} S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} V^\top$, with $S \in \mathbb{R}^{n_S \times n_S}$ containing all the positive singular values. Then the generalized inverse is characterized by

$$H^g = \left\{ G \middle| \begin{array}{l} V\left(\begin{bmatrix} S^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + F\right) U^\top \\ F \in \mathbb{R}^{m_H \times n_H},\ [I_{n_S}\ \mathbf{0}] F \begin{bmatrix} I_{n_S} \\ \mathbf{0} \end{bmatrix} = \mathbf{0} \end{array} \right\}, \quad (13)$$

where F is a any matrix of shape H whose upper-left block of size $n_S \times n_S$ is zero. For the sake of clarity, we characterize an element in H^g by $G(F)$ such that

$$G(F) := V\left(\begin{bmatrix} S^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + F\right) U^\top .$$

The set (13) is indeed the set of generalized inverse as

$$\begin{aligned} HG(F)H &= U\begin{bmatrix} S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\left(\begin{bmatrix} S^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + F\right)\begin{bmatrix} S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} V^\top \\ &= U\begin{bmatrix} S & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} V^\top = H \end{aligned}$$

2) LMI Tightening: Before going into details, we would first intuitively explain the idea behind the design procedure. Recall the idea behind Lemma 2, we can see that the design of A_{UIR} lies in the selection of the null space of the matrix H such that the set (10) is still unique, and the matrix A_{UIR} is Schur stable. Hence, we only need to focus on the null space of H, which motivates the following LMI reformulation. Based on the characterization of H^g in (13), any A_{UIR} in our feasible set $\mathcal{U}_R$ is accordingly parametrized by matrix F such that

$$\begin{aligned} A_{UIR}(F) &= \begin{bmatrix} \mathbf{0} & \mathbf{I}_{(N_{init}-1)n_u} \\ H_u V(\begin{bmatrix} S^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + F) U^\top \begin{bmatrix} \mathbf{I}_{N_{init}n_u} \\ \mathbf{0} \end{bmatrix} \end{bmatrix} \\ &= N_1 + N_2 F N_3, \\ N_2 :&= \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{n_u} \end{bmatrix} H_u V, \quad N_3 = U^\top \begin{bmatrix} \mathbf{I}_{N_{init}n_u} \\ \mathbf{0} \end{bmatrix} \\ N_1 :&= \begin{bmatrix} \mathbf{0} & \mathbf{I}_{(N_{init}-1)n_u} \\ & \mathbf{0} \end{bmatrix} + N_2 \begin{bmatrix} S^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} N_3 , \end{aligned} \quad (14)$$

To enable the LMI reformulation, we define

$$T_1 = [\mathbf{0}\ \ \mathbf{I}_{n_H - n_S}] \in \mathbb{R}^{(n_H - n_S)\times n_H} , \quad (15)$$

and we denote $r = \text{rank}(T_1 N_3)$. Regarding the definition of generalized inverse and (14), the operation $T_1 N_3$ selects the components in U related to $\text{Null}(H)$. Followed by this, we define $T_2 = [\mathbf{I}_r\ \mathbf{0}]E \in \mathbb{R}^{r\times(n_H - n_S)}$, where E is the multiplication of elementary operations to execute Gauss-Jordan Elimination for $T_1 N_3$. In summary, the operation $T_2 T_1 N_3$ generates the subspace of U related to the $\text{Null}(H)$, and based on the aforementioned discussion, the design of A_{UIR} lies within this space, which leads to the following LMI tightening.

Lemma 4: The BMI constraint (8b) is satisfied if $\exists\ N \in \mathbb{R}^{m_H \times r}, M \in \mathbb{R}^{r\times r}$ and $W \in \mathbb{R}^{N_{init}nu \times N_{init}nu} \succeq 0$ such that $F = NM^{-1}T_2T_1$ and

$$\begin{cases} \begin{bmatrix} W & N_1W + \\ & N_2NT_2T_1N_3 \\ WN_1' + & \\ (N_2NT_2T_1N_3)' & W \end{bmatrix} \succ 0 & \text{(16a)} \\ T_2T_1N_3W = MT_2T_1N_3 & \text{(16b)} \end{cases}$$

Proof: The first n_S columns of F are zeros, because $F = NM^{-1}T_2T_1$ and $T_1 = [\mathbf{0}\quad \mathbf{I}_{n_H - n_S}]$. Hence, F satisfies (13) and gives a generalized inverse $G(F)$.

The rest of the proof is similar to [22, Theorem 1]. By definition of T_2, matrix $T_2T_1N_3$ is full row rank. The left-hand-sied of condition (16b) is therefore full rank as $W \succ 0$, which further ensures that M is also full rank. Therefore, M^{-1} exists and $T_2T_1N_3 = M^{-1}T_2T_1N_3W$. Then we get (8a) from (16a) by

$$\begin{aligned} N_1W + N_2NT_2T_1N_3 &= N_1W + N_2NM^{-1}T_2T_1N_3W \\ &\overset{(a)}{=} N_1W + N_2FN_3W = A_{UIR}(F)W , \end{aligned}$$

where (a) follows $F = NM^{-1}T_2T_1$, and we summarize the proof. ∎

Finally, we summarize the design of data-driven R-UIR into the following LMI feasibility problem:

$$\underset{W\succ 0, M, N}{\text{minimize}}\ 0$$

s.t.

$$\begin{bmatrix} W & A_{UIR}W \\ WA_{UIR}' & W \end{bmatrix} \succ 0,\ T_2T_1N_3W = MT_2T_1N_3$$
$$\begin{bmatrix} W & (WN_1 + N_2NT_2T_1N_3) \\ (WN_1 + N_2NT_2T_1N_3)^\top & W \end{bmatrix} \succ 0$$

The A_{UIR} , $B_{UIR} \in \mathcal{U}_R$ is constructed by setting $G \in H^g$ to $G(F)$ with $F = NM^{-1}T_2T_1$.

IV. Data-Driven MH-UIR

Recall that an R-UIR only updates the most recent unknown input in z_{t+1}, i.e. $\hat{u}_{t+1}$. Similar to the concept used in Luenberger observer [23], the key idea behind a data-driven MH-UIR is to enable the correction update of the $\hat{u}_{t-N_{init}+2:t+1}$ estimate, i.e. z_{t+1}, by the error between the actual measurement of y_{t+1} and its data-driven predictive estimate $\hat{y}_{t+1}$.

Recall the data-driven prediction problem in Section II, we replace $u_{t-N_{init}+1:t}$ by z_t and define following matrices for the sake of clarity,

$$\hat{H} := \begin{bmatrix} \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,est}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \end{bmatrix}, \; H_y := \begin{bmatrix} \mathbf{I}_{n_y} & \mathbf{0} \end{bmatrix} \mathfrak{H}_{L,mea}(y_{\mathbf{d}})$$

$$\hat{b} := \begin{bmatrix} \hat{u}_{t-N_{init}+1:t} \\ \hat{u}_{t+1} \\ y_{t-N_{init}+1:t} \end{bmatrix} \overset{(a)}{=} \begin{bmatrix} z_t \\ H_u(G_u z_t + G_y d_t) \\ [I_{n_y N_{init}} \; \mathbf{0}] d_t \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} I & \mathbf{0} \\ H_u G_u & H_u G_y \\ \mathbf{0} & [I_{n_y N_{init}} \; \mathbf{0}] \end{bmatrix}}_{P(G)} \begin{bmatrix} z_t \\ d_t \end{bmatrix}, \; \forall \, [G_u \; G_y] = G \in H^g$$

where (a) follows (11) and $P(G)$ is introduced for the sake of compactness. Then, similar to (11), the corresponding output prediction $\hat{y}_{t+1}$ is defined by

$$\begin{aligned} \forall \, \hat{G} \in \hat{H}^g, \; \hat{y}_{t+1} :=& H_y \hat{G} \hat{b} \\ =& H_y \hat{G} P(G) \begin{bmatrix} z_t \\ d_t \end{bmatrix} \end{aligned} \tag{19}$$

Under Assumption 1 and $N_{init} \geq \mathbb{l}(\mathcal{B}(A,B,C,D))$, the Fundamental Lemma 1 and Lemma 2 guarantees this equality holds for the actual output y_{t+1} with respect to the actual but unknown previous input sequence $u_{t-N_{init}+1:t}$

$$\forall \, \hat{G} \in \hat{H}^g, \; y_{t+1} = H_y \hat{G} P(G) \begin{bmatrix} u_{t-N_{init}+1:t} \\ d_t \end{bmatrix} \tag{20}$$

Following a Luenberger observer style design, the observer will have the following structure with $\hat{A}_{UIR}, \; \hat{B}_{UIR} \in \mathcal{U}_R$:

$$z_{t+1} = \hat{A}_{UIR} z_t + \hat{B}_{UIR} d_t + L(y_{t+1} - \hat{y}_{t+1}) \, ,$$

where $L \in \mathbb{R}^{n_u N_{init} \times n_y}$ is a design parameter, $\hat{y}_{t+1}$ is given in (19) and y_{t+1} is always an entry of d_t as $N_{mea} \geq 1$ with

$$y_{t+1} = \overbrace{\underbrace{\mathbf{0}}_{(a)} \; \mathbf{I}_{n_y} \; \underbrace{\mathbf{0}}_{(b)}}^{T_y} d_t \, .$$

The term (a) is of $N_{init} n_y$ columns and the term (b) is of $(N_{mea} - 1) n_y$ columns, and this linear mapping is denoted by T_y. Hence, $\forall \; \hat{A}_{UIR}, \; \hat{B}_{UIR} \in \mathcal{U}_R, \; \hat{G} \in \hat{H}^g, \; G \in H^g$, the components of a data-driven MH-UIR can be written as:

$$A_{UIR} = \hat{A}_{UIR} - L H_y \hat{G} P(G) \begin{bmatrix} \mathbf{I}_{n_y N_{init}} \\ \mathbf{0} \end{bmatrix} \tag{21a}$$

$$B_{UIR} = \hat{B}_{UIR} + L T_y - L H_y \hat{G} P(G) \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_{n_y (N_{init} + N_{mea})} \end{bmatrix} \tag{21b}$$

The following Theorem summarizes the stability of a data-driven MH-UIR.

Theorem 2 (Stability of MH-UIR): Let Assumption 1 and condition (2) hold for any $\hat{A}_{UIR} \, , \hat{B}_{UIR} \in \mathcal{U}_R, \; \hat{G} \in \hat{H}^g, \; G \in H^g$, the data-driven MH-UIR in (21) is stable if $\hat{A}_{UIR} - L H_y \hat{G} P(G) \begin{bmatrix} \mathbf{I}_{n_y N_{init}} \\ \mathbf{0} \end{bmatrix}$ is Schur stable.

Proof: Similar to the proof of Lemma 3, the actual unknown input sequence satisfies the dynamics

$$\begin{aligned} u_{t-N_{init}+2:t+1} &= \hat{A}_{UIR} u_{t-N_{init}+1:t} + \hat{B}_{UIR} d_t \, , \\ u_t &= [\mathbf{0} \; \mathbf{I}_{n_u}] u_{t-N_{init}+1:t} \, , \end{aligned}$$

Therefore, we have

$$\begin{aligned} & \lim_{t \to \infty} \hat{u}_t - u_t \\ \overset{(a)}{=} & \lim_{t \to \infty} [\mathbf{0} \; \mathbf{I}_{n_u}] A_{UIR} (z_{t-1} - u_{t-N_{init}:t-1}) \\ = & \lim_{t \to \infty} [\mathbf{0} \; \mathbf{I}_{n_u}] A_{UIR}^t (z_0 - u_{-N_{init}+1:0}) \, , \end{aligned}$$

where (a) follows the equations (20) and (21). The above equation converges to 0 if and only if A_{UIR} is Schur stable, and we conclude the proof. ∎

In conclusion, the design process and operation of R-UIR and MH-UIR are summarized in Algorithm 1.

Algorithm 1 Design a data-driven UIR

Given historical signals $\{u_{\mathbf{d},i}, y_{\mathbf{d},i}\}_{i=1}^{T}$.

1) Choose a large N_{init}, try $N_{mea} = 1, 2, ..., max_{N_{mea}}$ until the condition (7) holds.
2) Build the UIR in the form (4) by either:
   a) R-UIR: Compute G by either (8) or (16). Compute the components in (6).
   b) MH-UIR: Choose any $G \in H^g$, $\hat{G} \in \hat{H}^g$. Design L such that the stability condition in Theorem 2 holds. Compute the components in (21).
3) From $t = 0$, choose arbitrary z_0 and repeatedly compute (4) to output $\hat{u}_t$.

Remark 3: The design methods by Lyapunov condition (8), LMI formulation (16) and MH-UIR in Theorem (21) do not guarantee the existence of a data-driven UIR for any system. The existence problem of a UIR has been explored in a model-based setup, which shows that the existence is related to the system dynamic $\mathcal{B}(A,B,C,D)$ [1], [6]. However, the existence problem within a data-driven setup is still unclear and remains for future work.

Remark 4: In comparison with the data-driven R-UIR, we observed that the data-driven MH-UIR is more robust to measurement noise contaminated data, because it does not require any construction of the null space, which may be sensitive to the measurement noise [24].

V. Simulation and Experimental Validation

A. Simulation

We consider the following **unstable** LTI dynamics:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{ccc|cc} 0.9 & 1.4 & 0.2 & 0.5 & 1.0 \\ 0.5 & 1.5 & 1.5 & 0.9 & 0.3 \\ 1.6 & 0.6 & 0.4 & 0.4 & 0.3 \\ \hline 1.5 & 1.0 & 1.4 & \multicolumn{2}{c}{\multirow{2}{*}{$\gamma \begin{bmatrix} 2.0 & 0.8 \\ 1.4 & 1.4 \end{bmatrix}$}} \\ 0.6 & 0.3 & 0.3 & & \end{array}\right]$$

Recall remark 2, different γ results in different N_{mea}, and we consider $\gamma = 1$ with $N_{mea} = 1$ and $\gamma = 0$ with $N_{mea} = 2$. We set $N_{init} = 5$, and the historical I/O data are generated

by a 50-step trajectory excited by random inputs. Setting the initial guess to $z_0 = [0\ 0\ \dots 0\ 0]^\top$, the results of $\hat{u}_t(i)$ by R-UIR are plotted in Figure 2(a) and the estimation error $du_t(i) = u_t(i) - \hat{u}_t(i)$ is given in Figure 2(b) and (c), where both proposed design schemes show fast convergence in the estimate even though the underlying dynamics is unstable.

B. Experiment

This experiment is carried out on a whole building, named Polydome, on the EPFL campus, and we estimate the number of occupants by the indoor CO2 level measurement. Although the building dynamic are nonlinear due to the ventilation system, it has a good linear approximation when the ventilation flow rate is constant [25]. Under the assumption that the CO2 generation rate per person doing office work is relatively constant, the proposed schemes in this work are feasible. The offline dataset contains indoor temperature, weather condition, heat pump power, CO2 level, and occupant number recorded by manual headcount (i.e., online measurement is not affordable). The indoor CO2 level is measured as the averaged value from four air quality sensors, whose installation locations are shown in Figure 3. Data from five weekdays are used to build the Hankel matrix, and the proposed data-driven MH-UIR[1] is compared with linear regression (LR) and Gaussian process regression (GR) by another five-weekday data. Note t [redacted] outside the office hour s i.e., between 7 : 00 PM and 7 : 00 AM, we enforce $\hat{u}_t = 0$ within this time interval to improve the estimate. The results are plotted in Figure 4, from which one can see that the proposed MH-UIR scheme is better than LR and slightly worse than GR in terms of mean absolute error (MAE). However, the UIR better tracks the occupancy trajectory while GR shows significant fluctuations in its estimates.

VI. Conclusions

This work proposes two data-driven UIR design schemes based on the Lyapunov condition and the Luenberger-observer-type feedback. The stability of the proposed schemes are discussed, and their efficacy is validated by numerical simulations and a real-world experiment of occupancy estimation.

References

[1] M. Hou and R. J. Patton, "Input observability and input reconstruction," *Automatica*, vol. 34, no. 6, pp. 789–794, 1998.

[2] A. Ansari, "Input and state estimation for discrete-time linear systems with application to target tracking and fault detection," Ph.D. dissertation, 2018.

[3] R. Rajamani, Y. Wang, G. D. Nelson, R. Madson, and A. Zemouche, "Observers with dual spatially separated sensors for enhanced estimation: Industrial, automotive, and biomedical applications," *IEEE Control Systems Magazine*, vol. 37, no. 3, pp. 42–58, 2017.

[4] Z. Chen, C. Jiang, and L. Xie, "Building occupancy estimation and detection: A review," *Energy and Buildings*, vol. 169, pp. 260–270, 2018.

[5] F. Zhu, "State estimation and unknown input reconstruction via both reduced-order and high-order sliding mode observers," *Journal of Process Control*, vol. 22, no. 1, pp. 296–302, 2012.

[1]The R-UIR does not give good performance in this experiment due to the measurement noise within the data and the nonlinearity of the dynamics.

[6] M. E. Valcher, "State observers for discrete-time linear systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 44, no. 2, pp. 397–401, 1999.

[7] S. Sundaram and C. N. Hadjicostis, "Delayed observers for linear systems with unknown inputs," *IEEE Trans. Autom. Control*, vol. 52, no. 2, pp. 334–339, 2007.

[8] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, no. 1, pp. 111–116, 2007.

[9] A. Ansari and D. S. Bernstein, "Deadbeat unknown-input state estimation and input reconstruction for linear discrete-time systems," *Automatica*, vol. 103, pp. 11–19, 2019.

[10] J. C. Willems, P. Rapisarda, I. Markovsky, and B. L. De Moor, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

[11] I. Markovsky and P. Rapisarda, "Data-driven simulation and control," *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.

[12] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the deepc," in *2019 18th Eur. Control Conf. (ECC)*. IEEE, 2019, pp. 307–312.

[13] C. De Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Trans. Autom. Control*, vol. 65, no. 3, pp. 909–924, 2019.

[14] I. Markovsky and P. Rapisarda, "On the linear quadratic data-driven control," in *2007 Eur. Control Conf. (ECC)*. IEEE, 2007, pp. 5313–5318.

[15] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[16] Y. Lian, J. Shi, M. P. Koch, and C. N. Jones, "Adaptive robust data-driven building control via bi-level reformulation: an experimental result," *arXiv preprint arXiv:2106.05740*, 2021.

[17] Y. Lian and C. N. Jones, "Nonlinear data-enabled prediction and control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 523–534.

[18] M. S. Turan and G. Ferrari-Trecate, "Data-driven unknown-input observers and state estimation," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 1424–1429, 2021.

[19] J. Jin, M.-J. Tahk, and C. Park, "Time-delayed state and unknown input observation," *International Journal of Control*, vol. 66, no. 5, pp. 733–746, 1997.

[20] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

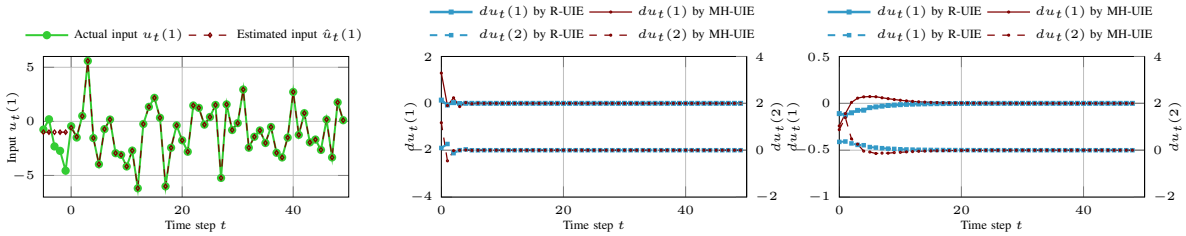
[21] O. Toker and H. Ozbay, "On the np-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback," in *Proceedings of 1995 Amer. Control Conf. -ACC'95*, vol. 4. IEEE, 1995, pp. 2525–2526.

[22] C. A. Crusius and A. Trofino, "Sufficient lmi conditions for output feedback control problems," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 1053–1057, 1999.

[23] D. Luenberger, "An introduction to observers," *IEEE Trans. Autom. Control*, vol. 16, no. 6, pp. 596–602, 1971.

[24] R.-C. Li, "Relative perturbation theory: Ii. eigenspace and singular subspace variations," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 471–492, 1998.

[25] D. Calì, P. Matthes, K. Huchtemann, R. Streblow, and D. Müller, "Co2 based occupancy detection algorithm: Experimental analysis and validation for office and residential buildings," *Building and Environment*, vol. 86, pp. 39–49, 2015.



(a) (b) (c)

Fig. 2: Simulation: input estimation by R-UIE and MH-UIE. (a): R-UIE, $\gamma = 1$, input $u_t(1)$. (b): two UIEs: $\gamma = 1$. (c): two UIEs: $\gamma = 0$.

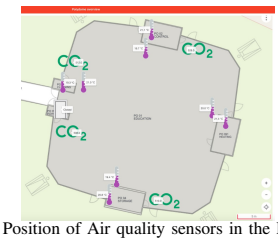


Fig. 3: Position of Air quality sensors in the Polydome

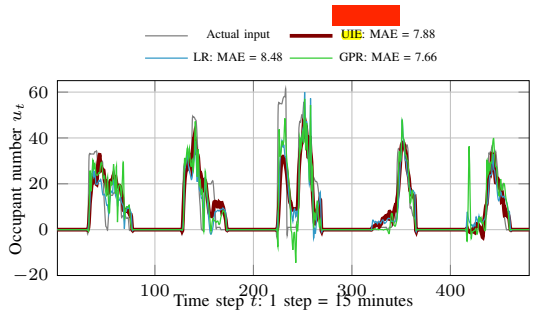


Fig. 4: Comparison of occupant number estimation by the data-driven UIE, LR and GPR. Mean absolute error (MAE) is computed for the data during the office hour.