Equalized Recovery State Estimators for Linear Systems with Delayed and Missing Observations

Syed M. Hassaan, Qiang Shen and Sze Zheng Yong

Abstract—This paper presents a dynamic state observer design for discrete-time linear time-varying systems that robustly achieves *equalized recovery* despite delayed or missing observations, where the set of all temporal patterns for the missing or delayed data is modeled by a finite-length language. By introducing a mapping of the language onto a reduced event-based language, we design a state estimator that adapts based on the history of available data at each step, and satisfies equalized recovery for all patterns in the reduced language. In contrast to existing equalized recovery estimators, the proposed design considers the equalized recovery level as a decision variable, which enables us to directly obtain the global minimum for the intermediate recovery level, resulting in improved estimation performance. Finally, we demonstrate the effectiveness of the proposed observer when compared to existing approaches using several illustrative examples.

Index Terms—Estimation; Delay systems; Observers for Linear systems

I. INTRODUCTION

CYBER-PHYSICAL systems (CPS) typically involve multiple sensors that send data packages to controllers through a shared communication channel, and controllers that compute and transmit control commands to actuators that are connected to the physical system. For the safe and efficient operation of these systems, state estimation plays an essential role. However, time delays and missing data are often inevitable due to sensor failures, package drops or adversaries. Hence, there is a need for designing state estimators that are robust to these delays and missing data. *Literature review:* For the past few decades, active research development has been undertaken in the area of state estimation for systems that are susceptible to packet drops and delayed communication, as highlighted in [1], [2] as typical concerns in networked control systems. Significant amount of research has been done to design state estimators when only intermittent data is available [3], [4], [5], and when observations are arriving as out-of-sequence measurements [6], [7], [8], [9]. The authors in [6] used complete in-sequence information approach to recompute all the estimations from the step when data did not arrive until the point when it finally arrived, while [7] proposed nonlinear filters utilizing a Bayesian filtering framework to correct the previous estimation as soon as the delayed observation arrives. On the other hand, an optimal state estimation approach was proposed for Markovian jump linear systems subject to delays in both the output and mode observations in [8]. However, these works mainly modeled the missing

and delayed observations as stochastic variables with known probability distributions and focused on obtaining the best average/expected estimates as opposed to achieving *best worst-case/robust estimation errors* considered in this paper.

Another relevant area that does consider the worst-case/robust estimation performance is the synthesis of set-valued estimators, which has seen some recent development, e.g., [10], [11], [12]. The authors in [13] introduced the property of equalized performance, which implies that the estimation error always remains equal/invariant. For systems with missing observations, [14] and [15] modeled the feasible missing data patterns with a finite-length language and proposed finite-horizon affine estimators with an extended property called equalized recovery, which implies that within a finite time horizon, especially for times when observations may go missing, the estimation error can have a more relaxed upper bound, but by the end of the horizon should return to the initial upper bound. In more recent work, [16], [17] developed a prefix-based method to predict the possible pattern of missing data to improve the estimation performance. However, this approach does not directly apply for delayed data patterns, and thus, our goal in this paper is to design equalized recovery estimators that can handle them.

Contribution: In this paper, we design a state observer that achieves equalized recovery when the system data is prone to misses and delays (including out-of-sequence observations). Instead of assuming probabilistic missing or delay events, we model them using a fixed-length language that represents the set of all possible temporal patterns of the missing or delayed data and further construct a reduced event-based language with unique event sequences. In contrast to the worst-case language method in [14], [15], our proposed design monitors the history of available data at run time and adapts the estimator gain matrices. Furthermore, we extend existing equalized recovery estimators to allow time-varying intermediate levels and consider the equalized recovery level as a decision variable, enabling us to directly find the global minimum of the intermediate levels. These improvements are shown in simulations to yield better estimation performance.

II. PROBLEM FORMULATION

A. System Dynamics and Delayed Data Language

System Dynamics: We consider a discrete-time linear time-varying system subject to process noise and output noise. The model of the system dynamics is described as follows:

$$
\begin{aligned}
x_{k+1} &= A_k x_k + B_k u_k + W_k w_k, \\
z_k &= C_k x_k + V_k v_k, \\
Y_k &= \{z_{k-\tau(i)} | i + \tau(i) = k, i \le k\},
\end{aligned}
\tag{1}
$$

where $x_k \in \mathbb{R}^n$ is the system state at time k, $u_k \in \mathbb{R}^m$ is the input to the system, $w_k \in \mathbb{R}^n$ is the process noise, $v_k \in \mathbb{R}^p$ is the measurement noise, $z_k \in \mathbb{R}^p$ is the model output, $Y_k \subset \mathbb{R}^p$ is the set of all measurements/outputs that are received at time step k and $\tau(i) \geq 0$ is the unknown time delay of the data at the time step i that satisfies $\tau(i) \leq \bar{\tau}$, where $\bar{\tau}$ is a known upper bound on the number of time steps that a packet can be delayed by. The discrete variable $\tau(i) = 0$ denotes that the measurement from time step i is received/available, while $\tau(i) = \delta$ implies that the data from time step i is delayed by δ steps. We assume that w_k and v_k are bounded with $\|w_k\| \leq \eta_w$ and $\|v_k\| \leq \eta_v$ for each k, where $\| \cdot \|$ denotes the ∞-norm. The system matrices A_k, B_k, C_k, W_k, V_k, η_w and η_v are all known. Without loss of generality, we assume that the initial time is $k = 0$.

Delayed Data Language: Given a fixed length T, we consider a delayed data model in which all delay patterns are restricted to a set expressed by fixed-length language specifications, e.g., 'the i-th observation is delayed by at most m time steps' or 'at most m available measurements in a fixed interval'. Formally, our delayed data model is a fixed-length language $\mathcal{L}$ of length T that specifies the set of allowable delay mode sequences $\tau(0)\tau(1)\tau(2)\ldots\tau(T-1)$ with $\tau(i) \leq \bar{\tau}, \forall i \in \mathbb{N}_0^{T-1}$, where the α-th possible sequence is called a word $\mathcal{W}_\alpha$ with $\alpha \in \mathbb{N}_1^{|\mathcal{L}|}$. Note that $i + \tau(i) \geq T$ means that the i-th data is delayed beyond the horizon T, which is similar to the situation where that data is missing. In other words, the case considered in [14], [15], [16] is a special case of the delayed data language in this paper.

Example 1. *Consider a system where the observation is delayed by at most 2 time steps in a fixed interval of length 2. This means that $\bar{\tau} = 2$ and $T = 2$, and hence we have $\tau(i) \in \{0, 1, 2\}$ for all $i \in \mathbb{N}_0^1$. Therefore, the fixed-length language can be expressed as $\mathcal{L} = \{\mathcal{W}_1, \ldots, \mathcal{W}_9\} = \{00, 01, 02, 10, 11, 12, 20, 21, 22\}$.*

B. Equalized Recovery

The focus of our paper is to design a bounded-error estimator, where the estimation error is guaranteed to return/recover to the same bound that it started with after a fixed number of time steps, as an extension of the notion of *equalized performance* in [13]. In terms of time horizon T, we enforce that the estimation error bound at the end of the horizon is guaranteed to be less than or equal to the bound at the start. Formally, we consider *equalized recovery*, defined as follows, which is a slight modification of the definition in [14] to allow time-varying intermediate levels:

Definition 1 (Equalized Recovery). *An estimator is said to achieve an equalized recovery level μ_1 at time 0 with recovery time T and intermediate levels $\mu_{2,k} \geq \mu_1$ if for any $\|\tilde{x}_0\| \leq \mu_1$, we must have $\|\tilde{x}_k\| \leq \mu_{2,k}$ for all $k \in [0, T]$ and $\|\tilde{x}_T\| \leq \mu_1$, where $\tilde{x}_k \triangleq x_k - \hat{x}_k$ is the estimation error and $\hat{x}_k$ is the state estimate at time k.*

C. Problem Statement

We aim to design a bounded-error estimator that satisfies *equalized recovery*, which can be stated as follows:

Problem 1 (Estimator Design with Delayed Data). *Given the system dynamics (1), a delayed data model specified by a language $\mathcal{L}$ and a recovery time T as a time horizon, design an optimal equalized recovery state estimator with estimate $\hat{x}_k$ and estimation error $\tilde{x}_k = x_k - \hat{x}_k, \forall k \in [0, T]$ that minimizes a cost $J(\mu_1, \{\mu_{2,k}\}_{k=0}^T)$ subject to $\mu_{2,k} \geq \mu_1$, $\|\tilde{x}_k\| \leq \mu_{2,k}, \forall k \in [0, T]$ and $\|\tilde{x}_T\| \leq \mu_1$ for all $\|\tilde{x}_0\| \leq \mu_1$.*

In contrast to [14], [15], [16], [17], the above problem formulation allows time-varying intermediate levels and more importantly, the equalized recovery level μ_1 does not need to be specified *a priori*. As a result, we can directly optimize over $J(\mu_1, \{\mu_{2,k}\}_{k=0}^T)$ and overcome the challenge with the formulation in [14], [15], [16], [17] that the optimal cost is not monotonic in μ_1. Furthermore, if the recovery time T is not given and can be chosen, we can perform a line search over T using the above formulation. Moreover, simple modifications will allow us to consider affine dynamics, similar to [14], [15], [16], [17], and the case when the initial estimation error is greater than μ_1 (cf. Section III-B3).

III. DESIGN APPROACH

In this section, we propose an observer design approach to solve Problem 1, which involves constructing an event-based language $\mathcal{L}^E$ from the fixed-length delayed data language $\mathcal{L}$ and designing an estimator that adapts to the information from the observed data pattern seen so far.

A. Event-Based Language

Given a language set $\mathcal{L}$ of a system, containing all possible words for different allowable delay mode sequences, an event-based language $\mathcal{L}^E$ is constructed to capture the set of indistinguishable event sequences that correspond to the different delay mode sequences in $\mathcal{L}$. To build the event-based language, the following definitions are introduced first:

Definition 2 (Event). *An event $e_{i,j} = d_0 d_1 d_2 \ldots d_i$ at time step $i \in \mathbb{N}_0^{T-1}$ is a finite sequence of binary variables $d_l \in \{0, 1\}$ for all $l \in \mathbb{N}_0^i$, where $j \in \mathbb{N}_0^{2^{i+1}-1}$ is an index denoting the j-th potential event at time step i. The binary variable $d_l = 1$ denotes that the data of time step l is available at current time step i (i.e., all received data up until the current step i), while $d_l = 0$ signifies that the data of time step l is not available at current time step i. Moreover, an event can be defined using $e_{i,j} = \mathtt{binary}(j, i+1)$ at time step i, where the function $\mathtt{binary}$ returns a binary representation of the number $j \in \mathbb{N}_0^{2^{i+1}-1}$ with $i+1$ digits.*

Definition 3 (Event Set). *An event set $e_i = \{e_{i,j}\}_{j=0}^{2^{i+1}-1}$ is a set of all potential events at time step $i \in \mathbb{N}_0^{T-1}$.*

Intuitively, an event at time step i represents the information that is available up until time i. Since any data from previous or current steps only has two possibilities, i.e., measured or not measured at the current time $i \in \mathbb{N}_0^{T-1}$, there is a total of 2^{i+1} different cases. Thus, the index j of $e_{i,j}$ varies from 0 to $2^{i+1} - 1$, as exemplified in the following.

Example 2. *Consider a system with a horizon $T = 2$. Based on above definitions of event and event set, we have*

$e_0 = \{e_{0,j}\}_{j=0}^{2^1-1} = \{e_{0,0}, e_{0,1}\} = \{0, 1\}$ *for time step* $i = 0$, $e_1 = \{e_{1,j}\}_{j=0}^{2^2-1} = \{e_{1,0}, e_{1,1}, e_{1,2}, e_{1,3}\} = \{00, 01, 10, 11\}$ *for time step* $i = 1$. *For instance, the event* $e_{0,0} = 0$ *means that the data of time* 0 *is not available at the time* 0, *and the event* $e_{1,1} = 01$ *means that at the time* 1, *the data of time* 0 *is not available but the data of time* 1 *is available.*

Definition 4 (Event Sequence). *An event sequence* $\mathcal{E}_\alpha = e_{0,j_0} e_{1,j_1} e_{2,j_2} \ldots e_{T-1,j_{T-1}}$ *is a sequence of events corresponding to a word* $\mathcal{W}_\alpha = \{\tau(i)\}_{i=0}^{T-1}$ *from the fixed-length language* $\mathcal{L}$, *where the subscripts* j_i *for all* $i \in \mathbb{N}_0^{T-1}$ *are determined by the word* $\mathcal{W}_\alpha$.

In other words, an event sequence represents the available information at each step. For each delay mode sequence in a language $\mathcal{L}$, we can find its corresponding event sequence, and thus, the language $\mathcal{L} = \{\mathcal{W}_j\}_{j=1}^{|\mathcal{L}|}$ containing all allowable delay mode sequences can be mapped onto an event-based language $\mathcal{L}^E = \{\mathcal{E}_\alpha\}_{\alpha=1}^{|\mathcal{L}|}$ containing all potential event sequences. Specifically, for a word $\mathcal{W}_\alpha = \tau(0)\tau(1)\tau(2)\ldots\tau(T-1)$, the subscript j_k, $k \in \mathbb{N}_0^{T-1}$, in the corresponding event sequence $\mathcal{E}_\alpha = e_{0,j_0} e_{1,j_1} e_{2,j_2} \ldots e_{T-1,j_{T-1}}$ (cf. Definition 4) can be constructed as

$$j_k = \sum_{\ell=0}^{k} 2^\ell \mathbb{1}_{\tau(k-\ell)\le\ell}, \ \forall k \in \mathbb{N}_0^{T-1}, \tag{2}$$

where $\mathbb{1}_{\tau(k-\ell)}$ denotes an indicator defined as

$$\mathbb{1}_{\tau(k-\ell)\le\ell} = \begin{cases} 1, & \tau(k-\ell) \le \ell, \\ 0, & \tau(k-\ell) > \ell. \end{cases} \tag{3}$$

Note that the resulting event-based language $\mathcal{L}^E = \{\mathcal{E}_\alpha\}_{\alpha=1}^{|\mathcal{L}|}$ could have repeated event sequences (i.e., the mapping is surjective). Thus, we will eliminate repeated event sequences in $\mathcal{L}^E$ to obtain a reduced *event-based language* $\mathcal{L}^{E'} = \{\mathcal{E}'_\alpha\}_{\alpha=1}^{|L^{E'}|} \subseteq \mathcal{L}^E$ with unique event sequences $\mathcal{E}'_\alpha$ for $\alpha \in \mathbb{N}_1^{|L^{E'}|}$. The next example demonstrates how to map/transform the fixed-length language $\mathcal{L}$ in the Example 1 to a reduced event-based language $\mathcal{L}^{E'}$.

Example 3. *Consider the delayed data language in Example 1. The word* $\mathcal{W}_2 = 01$ *denoting that the data of time* 0 *has no delay while the data of time* 1 *is delayed by* 1 *time step, can be represented by the event trajectory* $\mathcal{E}_2 = e_{0,1} e_{1,2}$ *using* (2), *where* $e_{0,1} = 1$ *means that the data of time* 0 *is available at the time* 0, $e_{1,2} = 10$ *means that the data of time* 0 *is also available at the time* 1 *(since the data of time* 0 *is previously received at the time* 0*) and the data of time* 1 *is not available at the time* 1 *(since the data of time* 1 *is delayed by* 1 *time step). Using this procedure, we can transform all words in the language* $\mathcal{L} = \{\mathcal{W}_1, \ldots, \mathcal{W}_9\} = \{00, 01, 02, 10, 11, 12, 20, 21, 22\}$ *to an event-based language* $\mathcal{L}^E = \{\mathcal{E}_1, \ldots, \mathcal{E}_9\} = \{e_{0,1}e_{1,3}, e_{0,1}e_{1,2}, e_{0,1}e_{1,2}, e_{0,0}e_{1,3}, e_{0,0}e_{1,2}, e_{0,0}e_{1,2}, e_{0,0}e_{1,1}, e_{0,0}e_{1,0}, e_{0,0}e_{1,0}\}$. *Then, we can eliminate repeated event sequences in* $\mathcal{L}^E$ *and obtain a reduced event-based language* $\mathcal{L}^{E'} = \{\mathcal{E}'_1, \ldots, \mathcal{E}'_6\} = \{e_{0,1}e_{1,3}, e_{0,1}e_{1,2}, e_{0,0}e_{1,3}, e_{0,0}e_{1,2}, e_{0,0}e_{1,1}, e_{0,0}e_{1,0}\}$.

B. Equalized Recovery State Estimator Design

For the estimator design, we will make use of the following definitions and notation, inspired by [16]:

Definition 5 (Principal Block Minor). *The* i-*th leading principal block minor of a matrix* $M \in \mathbb{R}^{an\times bp}$ *is the* $n \times p$ *block matrix,* $\mathcal{BM}_i(M) = M_{1:in,1:ip}$, *for all* $i \in [1, \min(a,b)]$.

Definition 6 (Prefix of an Event Sequence). *For an event sequence* $\mathcal{E}'_\alpha \in \mathcal{L}^{E'}$ *and* $i \le |\mathcal{E}'_\alpha|$, *the length* i *prefix of* $\mathcal{E}'_\alpha$ *is defined as* $\mathcal{E}'^{,[1:i]}_\alpha = e_{0,j_0} e_{1,j_1} e_{2,j_2} \ldots e_{i-1,j_{i-1}}$, *where* $|\mathcal{E}'_\alpha|$ *denotes the number of events in* $\mathcal{E}'_\alpha$. *The set of all non-empty prefixes of* $\mathcal{E}'_\alpha$ *is denoted as* $Pref(\mathcal{E}'_\alpha)$.

Example 4. *Consider event* $\mathcal{E}'_1$ *of the reduced event-based language* $\mathcal{L}^{E'}$ *in Example 3. As* $\mathcal{E}'_1 = e_{0,1}e_{1,3}$, *we have* $|\mathcal{E}'_1| = 2$ *and* $i = \{1, 2\}$. *The length* 1 *prefix of* $\mathcal{E}'_1$ *is* $e_{0,1}$, *while its length* 2 *prefix is* $e_{0,1}e_{1,3}$. *Thus, the set of non-empty prefixes of* $\mathcal{E}'_1$ *is* $Pref(\mathcal{E}'_1) = \{e_{0,1}, e_{0,1}e_{1,3}\}$.

Next, to solve Problem 1, we consider a finite horizon dynamic state estimator, inspired by [15], with augmented states $\bar{x}_k \triangleq [\hat{x}_k^\top \ s_k^\top]^\top$, where $\hat{x}_k \in \mathbb{R}^n$ is the estimate of the system state and $s_k \in \mathbb{R}^n$ an auxiliary state which estimates $\tilde{x}_k = x_k - \hat{x}_k$. The estimator design is as follows:

$$\begin{aligned} \hat{x}_{k+1} &= A_k\hat{x}_k + B_k u_k - u_{e,k}, \\ s_{k+1} &= A_k s_k + u_{e,k} + L_k^{\mathcal{E}'_\alpha} \tilde{z}_k, \end{aligned} \tag{4}$$

with $\tilde{z}_k = \begin{cases} \tilde{y}_k - C_k s_k = z_k - C_k(\hat{x}_k + s_k), & \text{if } z_k \in \bigcup_{j=0}^{k} Y_j, \\ 0, & \text{otherwise,} \end{cases}$

where $\tilde{y}_k \triangleq z_k - C_k\hat{x}_k$, $L_k^{\mathcal{E}'_\alpha} \in \mathbb{R}^{n\times p}$ is the Luenberger gain at step k as a function of the observed prefix $\mathcal{E}'_\alpha$ and $u_{e,k} \in \mathbb{R}^n$ is the *causal* output error injection term given by:

$$u_{e,k} = \nu_k^{\mathcal{E}'_\alpha} + \sum_{i=0}^{k} M_{(k,i)}^{\mathcal{E}'_\alpha} \tilde{z}_i, \tag{5}$$

where $M_{(k,i)}^{\mathcal{E}'_\alpha} \in \mathbb{R}^{n\times p}$ and $\nu_k^{\mathcal{E}'_\alpha} \in \mathbb{R}^n$ are gain matrices at time k as a function of the observed prefix $\mathcal{E}'_\alpha$, which will be designed to satisfy the objectives of Problem 1.

In [15], the estimator design was formulated with essentially one worst-case word in the worst-case language $\mathcal{L}^*$, which was obtained by combining all the words in the given language, resulting in a triplet of stacked (M, L, ν) matrices for the whole time horizon T which would satisfy the conditions in Problem 1 for both the worst-case word in $\mathcal{L}^*$ as well as individual words in $\mathcal{L}$. Since the worst-case language $\mathcal{L}^*$ is used for achieving the equalized recovery, the achievable performance level is conservative. On the other hand, solving Problem 1 for multiple triplets of $(M^\alpha, L^\alpha, \nu^\alpha)$ for each word $\mathcal{W}_\alpha$ in $\mathcal{L}$ may result in implementation conflicts due to causality. This limitation was discussed in detail in [16].

To remedy this, we need to design the individual $(M^\alpha, L^\alpha, \nu^\alpha)$ for each word $\mathcal{W}_\alpha$ of $\mathcal{L}$ such that if two different words are not distinguishable until time $\bar{k}$, then $(M_{(k)}^\alpha, L_{(k)}^\alpha, \nu_{(k)}^\alpha)$ for both words should be constrained to be the same for all $k \in \mathbb{N}_0^{\bar{k}-1}$, where $M_{(k)}^\alpha$ denotes the k-th row of M^α. Instead of associating a triplet $(M^\alpha, L^\alpha, \nu^\alpha)$ to each word $\mathcal{W}_\alpha$ in the language $\mathcal{L}$, we only consider triplets $(M^\alpha, L^\alpha, \nu^\alpha)$ for each event sequence $\mathcal{E}'_\alpha$ of the reduced event-based language $\mathcal{L}^{E'}$. Since all event sequences in $\mathcal{L}^{E'}$ are not repeated and $\mathcal{L}^{E'} \subseteq \mathcal{L}^E$, we can reduce the number of triplets needed and thus the size of the optimization problem.

Using the above, we impose the following constraint due to indistinguishability of unique event trajectories in $\mathcal{L}^{E'}$:

$$C(\mathcal{L}^{E'}) = \left\{ \{(M^\alpha, L^\alpha, \nu^\alpha)\}_{\alpha=1}^{|\mathcal{L}^{E'}|} \,\middle|\, \begin{array}{l} (e \in Pref(\mathcal{E}'_\alpha) \wedge e \in Pref(\mathcal{E}'_\beta)) \\ \implies \forall \mathcal{E}'_\alpha, \mathcal{E}'_\beta \in \mathcal{L}^{E'}: \\ (\mathcal{BM}_{|e|}(M^\alpha) = \mathcal{BM}_{|e|}(M^\beta)) \wedge \\ (\mathcal{BM}_{|e|}(L^\alpha) = \mathcal{BM}_{|e|}(L^\beta)) \wedge \\ ((\nu^\alpha)_{(1:|e|n)} = (\nu^\beta)_{(1:|e|n)}) \end{array} \right\}. \quad (6)$$

Intuitively, if any pair of event sequences share the same prefix of a particular length, then they are indistinguishable at the corresponding time step based on the received information. Since they are indistinguishable (and future information is inaccessible in a causal system), their associated submatrices and subvectors need to be constrained to be the same to avoid conflicts during implementation. Note that while the prefix notation is similar to [16], our estimator uses a different state estimator structure that enables us to consider more general data patterns, including delayed data patterns.

Moreover, for each event sequence $\mathcal{E}'_\alpha \in \mathcal{L}^{E'}$, due to delayed data and causality, all the entries in M^α and L^α corresponding to no available data should also be set to zero. To construct this constraint on M^α and L^α, we first define an event matrix associated with the event sequence $\mathcal{E}'_\alpha \in \mathcal{L}^{E'}$:

$$E_\alpha = \begin{bmatrix} e_{0,j_0}^{(0)} & 0 & 0 & \cdots & 0 \\ e_{1,j_1}^{(0)} & e_{1,j_1}^{(1)} & 0 & \cdots & \vdots \\ e_{2,j_2}^{(0)} & e_{2,j_2}^{(1)} & e_{2,j_2}^{(2)} & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ e_{T-1,j_{T-1}}^{(0)} & e_{T-1,j_{T-1}}^{(1)} & e_{T-1,j_{T-1}}^{(2)} & \cdots & e_{1,j_{T-1}}^{(T-1)} \end{bmatrix},$$

where $e_{i,j_i}^{(l)}$ specifies the $(l+1)$-th digit of event e_{i,j_i}, i.e., d_l (cf. Definition 2). Using this definition, we impose the following constraint due to delayed data:

$$\mathcal{D}(\mathcal{L}^{E'}) = \left\{ \{(M^\alpha, L^\alpha)\}_{\alpha=1}^{|\mathcal{L}^{E'}|} \,\middle|\, \begin{array}{l} \forall i, j \in \mathbb{N}_1^T: \\ M^\alpha_{((i-1)n:(i-E_\alpha(i,j))n-1,} {}_{(j-1)p:(j-E_\alpha(i,j))p-1)} = 0, \\ L^\alpha_{((i-1)n:(i-E_\alpha(i,j))n-1,} {}_{(j-1)p:(j-E_\alpha(i,j))p-1)} = 0 \end{array} \right\}. \quad (7)$$

Next, we provide examples of $\mathcal{C}(\mathcal{L}^{E'})$ and $\mathcal{D}(\mathcal{L}^{E'})$.

Example 5. *Consider two event sequences* $\mathcal{E}'_1 = \{e_{0,1}e_{1,3}\}$ *and* $\mathcal{E}'_2 = \{e_{0,1}e_{1,2}\}$ *of* $\mathcal{L}^{E'}$ *in Example 3. The sets of all non-empty prefixes of* $\mathcal{E}'_1$ *and* $\mathcal{E}'_2$ *are* $Pref(\mathcal{E}'_1) = \{e_{0,1}, e_{0,1}e_{1,3}\}$ *and* $Pref(\mathcal{E}'_2) = \{e_{0,1}, e_{0,1}e_{1,2}\}$, *respectively. It is clear that* $\mathcal{E}'_1$ *and* $\mathcal{E}'_2$ *have the same length 1 prefix, so we need to impose the following constraints in* $\mathcal{C}(\mathcal{L}^{E'})$: $\mathcal{BM}_1(M^1) = \mathcal{BM}_1(M^2)$, $\mathcal{BM}_1(L^1) = \mathcal{BM}_1(L^2)$ *and* $\nu^1_{1:n} = \nu^2_{1:n}$. *Moreover, to formulate the constraints* $\mathcal{D}(\mathcal{L}^{E'})$, *we first construct event matrices associated with* $\mathcal{E}'_1$ *and* $\mathcal{E}'_2$:

$$E_1 = \begin{bmatrix} e_{0,1}^{(1)} & 0 \\ e_{1,3}^{(1)} & e_{1,3}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, E_2 = \begin{bmatrix} e_{0,1}^{(1)} & 0 \\ e_{1,2}^{(1)} & e_{1,2}^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},$$

where a zero element located at the i-th row and j-th column of matrices E_1 (or E_2) indicates that the data of time step j is not available at the time i due to the delay in $\mathcal{E}'_1$ (or $\mathcal{E}'_2$) and causality. This is captured in $\mathcal{D}(\mathcal{L}^{E'})$ by $M^1_{(0:n-1,p:2p-1)} = 0$ for $i = 1$ and $j = 2$ in E_1, $M^2_{(0:n-1,p:2p-1)} = 0$ for $i = 1$ and $j = 2$ in E_2, and $M^2_{(n:2n-1,p:2p-1)} = 0$ for $i = 2$ and $j = 2$ in E_2, which results in M^1 and M^2 with the following

block structures: $M^1 = \begin{bmatrix} * & 0 \\ * & * \end{bmatrix}$, $M^2 = \begin{bmatrix} * & 0 \\ * & 0 \end{bmatrix}$, *where $*$ denotes non-zero submatrices. Similar constraints also need to be imposed on L^1 and L^2 in $\mathcal{D}(\mathcal{L}^{E'})$.*

1) Estimator Gains Design: Next, we present an approach to obtain the estimator gains $(M^\alpha, L^\alpha, \nu^\alpha)$ associated with each unique event sequence $\mathcal{E}'_\alpha$ in $\mathcal{L}^{E'}$ for the estimator in (4). Moreover, we also allow the time-varying intermediate levels to be prefix-dependent, i.e., with $\mu^\alpha_{2,k}$, which can lead to improved estimation error bound when the prefix, i.e., the history of available data, is observed at run time.

Theorem 1 (Equalized Recovery Estimator Design with Delays). *For a system with measurement delays and missing data patterns defined by a fixed-length language $\mathcal{L}$ given in (1), the finite-horizon affine estimator given in (4) can fulfill the objectives in Problem 1 if the following is feasible:*

$$\min_{M^\alpha, \nu^\alpha, \mu^\alpha_2, s_0, L^\alpha, \mu_1} J(\mu_1, \{\mu^\alpha_2\}_{\alpha=1}^{|\mathcal{L}^{E'}|})$$

$$\text{subject to} \quad \begin{array}{l} \forall(\|w\| \le \eta_w, \|v\| \le \eta_v, \|\tilde{x}_0\| \le \mu_1, \alpha \in \mathbb{N}_1^{|\mathcal{L}^{E'}|}: \\ \|\tilde{x}^\alpha\| \le \mu^\alpha_2, \|R_T \tilde{x}^\alpha\| \le \mu_1, \mu^\alpha_2 \ge \mu_1, \mu_1 \ge 0, \\ \tilde{x}^\alpha = \Theta^\alpha w + \Psi^\alpha v + \Xi^\alpha \tilde{x}_0 + \Upsilon^\alpha s_0 + H\nu^\alpha, \\ (M^\alpha, L^\alpha, \nu^\alpha) \in \mathcal{C}(\mathcal{L}^{E'}) \wedge \mathcal{D}(\mathcal{L}^{E'}), \end{array} \quad (8)$$

where

$$\begin{array}{l} R_T = \begin{bmatrix} 0_{n \times nT} & I_n \end{bmatrix}, \ \mu^\alpha_2 = \begin{bmatrix} \mu^\alpha_{2,0}, \mu^\alpha_{2,1}, \ldots, \mu^\alpha_{2,T} \end{bmatrix}^\top, \\ \Theta^\alpha = (I + H(M^\alpha + L^\alpha)C)\Gamma^\alpha W, \\ \Psi^\alpha = (H(M^\alpha + L^\alpha)(I - C\Gamma^\alpha L^\alpha) - \Gamma^\alpha L^\alpha)V, \\ \Xi^\alpha = (I + H(M^\alpha + L^\alpha)C)\Phi^\alpha, \ \Upsilon^\alpha = A - \Xi^\alpha. \end{array} \quad (9)$$

The matrices H, M^α, L^α, C, Γ^α, W, V, Φ^α and A, all of which are stacked matrices for the whole time horizon T, are derived after stacking the system in (1), estimator in (4) and the output error injection term in (5). The definitions of these matrices can be found in the Appendix.

Proof. The estimator design follows similar steps to the design in [15]. It is straightforward to observe that the estimator solves Problem 1 by construction with the additional constraints on the estimator gains in Section III-B. ∎

When compared to our prior work [15], we consider a prefix-based design that enables adaptation of the gain matrices and improved estimation error bounds based on observed prefix, i.e., the history of available data, at run time. It is also noteworthy that in contrast to existing equalized recovery estimators [14], [15], [16], [17], the proposed estimator considers μ_1 as a decision variable, instead of a given parameter. This seemingly small change has an important implication that the difficulty in finding the global minimum for μ_2 with previous designs (due to their non-monotonicity in μ_1) can now be overcome with the new design.

2) Robustification: Next, since the problem in Theorem 1 involves semi-infinite constraints (i.e., *for all* constraints), as in [15], we leverage robust optimization tools, e.g., [18], to obtain a problem with a finite number of constraints:

Proposition 1 (Robustified Equalized Recovery Estimator Design with Delays). *The equalized estimator design that solves Problem 1 via Theorem 1 is equivalent to:*

$$\min_{M^\alpha,\nu^\alpha,\mu_2^\alpha,\mu_1,s_0,L^\alpha,\Pi_1^\alpha,\Pi_2^\alpha} J(\mu_1,\{\mu_2^\alpha\}|_{\alpha=1}^{|\mathcal{L}^{E'}|})$$

$$\text{subject to} \quad \Pi_1^\alpha \geq 0, \Pi_2^\alpha \geq 0, \Pi_3^\alpha \geq 0, \mu_2^\alpha \geq \mu_1, \ \mu_1 \geq 0,$$

$$\begin{bmatrix} \Pi_1^\alpha & \Pi_2^\alpha & \Pi_3^\alpha \end{bmatrix} \begin{bmatrix} \eta_w \mathbb{1} \\ \eta_v \mathbb{1} \\ \mathbb{1} \end{bmatrix} \leq \begin{bmatrix} \mu_2^\alpha \\ \mu_2^\alpha \\ \mu_1 \mathbb{1} \end{bmatrix} - \begin{bmatrix} I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \end{bmatrix} \begin{bmatrix} H\nu^\alpha + \Upsilon^\alpha s_0 \\ R_T(H\nu^\alpha + \Upsilon^\alpha s_0) \end{bmatrix},$$

$$\begin{bmatrix} \Pi_1^\alpha & \Pi_2^\alpha & \Pi_3^\alpha \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ -I & 0 & 0 \\ 0 & I & 0 \\ 0 & -I & 0 \\ 0 & 0 & I \\ 0 & 0 & -I \end{bmatrix} = \begin{bmatrix} I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \end{bmatrix} \begin{bmatrix} G^\alpha \\ R_T G^\alpha \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \mu_1 \mathbb{1} \end{bmatrix},$$

$$(M^\alpha, L^\alpha, \nu^\alpha) \in \mathcal{C}(\mathcal{L}^{E'}) \wedge \mathcal{D}(\mathcal{L}^{E'}), \quad (10)$$

where $G^\alpha \triangleq \begin{bmatrix} \Theta^\alpha & \Psi^\alpha & \Xi^\alpha \end{bmatrix}$ with Θ^α, Ψ^α, Ξ^α, $\forall \alpha \in \mathbb{N}_1^{|\mathcal{L}^{E'}|}$ defined in (9), while Π_1^α, Π_2^α, Π_3^α are dual matrix variables.

Proof. By replacing the semi-infinite constraints in (8) with their robust counterparts based on [18], we obtain a similar problem as in [15, Eq. (11)]. However, since μ_1 is a decision variable in our problem (instead of a parameter as in [15]), we have a bilinear term in the first equality that is a product of dual variables $\tilde{\Pi}_3^\alpha$ with μ_1. To overcome this issue, we post-multiply the second equation on both sides with $\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & \mu_1 \mathbb{1} \end{bmatrix}$, which results in the appearance of the same bilinear term. Then, since the original $\tilde{\Pi}_3^\alpha$ no longer appears independently, a common trick is to replace $\mu_1 \tilde{\Pi}_3^\alpha$ with a new decision variable Π_3^α that is positive since $\mu_1 \geq 0$. ∎

The above optimization problem has bilinear terms but is relatively sparse, so off-the-shelf solvers, e.g., [19], can find optimal solutions quickly. Further, if desired, we can fix L^α and s_0 to perform a line search over μ_1 using a linear program without loss of optimality, as discussed in [15, Section IV-C].

3) Implementation Strategy: The proposed equalized recovery estimator in this paper could be implemented in multiple different ways. First, in the case that the delayed/missing data pattern is periodic with a period of T time steps, we can use the same gains for each period because the estimation error bound at the end of the period is enforced to be the same at the beginning of the period by the proposed estimator. Moreover, if there is no missing/delayed data, we could use an equalized performance estimator, i.e., an equalized recovery estimator with a period of 1 time step (cf. [13]) until a missing/delayed data is encountered, at which point we can switch to an equalized recovery estimator with a T-length language in which the first data is missing/delayed. Then, after the recovery time T, we revert to the equalized performance estimator again until the next time a delayed data is detected. Further, if the initial estimation error does not satisfy the equalized recovery/performance level, the proposed estimator can also be combined with any asymptotic estimator, where the latter is used until the desired equalized level is achieved. Alternatively, we can modify our estimator by replacing μ_1 in the second constraint in (10) with the initial estimation error and repeating the process, as needed, to achieve this.

IV. EXAMPLES AND COMPARISONS

In this section, the performance of the proposed estimator is validated and compared with the approaches in [9] and
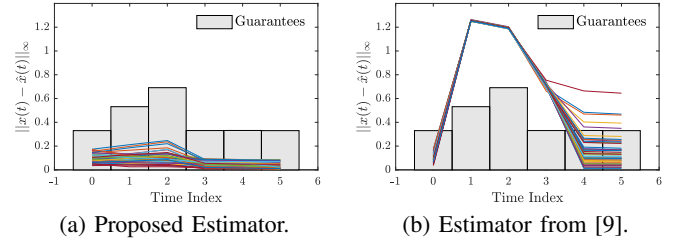

(a) Proposed Estimator. (b) Estimator from [9].

Fig. 1: Estimator comparison for $\mathcal{W}_{sim} = 21210$.

[16]. The examples using our proposed estimator are all run using MATLAB 2017a. As the robustified problem in (10) involves many sparse matrices, the IPOPT solver [19] is used. Moreover, in [15], it was established that the value of s_0 in (10) does not affect the performance of the estimator. So, to simplify the problem of the estimator design, all of the parameters of s_0 are set to zero in all the presented examples.

A. Batch Reactor Process (Comparison with [9])

To demonstrate the capability of the proposed estimator proposed in this paper in comparison with [9] when output delays are involved, we utilize an example of a continuous-time batch reactor process from [20]. This system is first discretized with a sampling time of $T_s = 0.05$ seconds using MATLAB `c2d` command (with zero-order hold) to obtain a discrete-time state-space system with the following matrices:

$$A = \begin{bmatrix} 1.0795 & -0.0045 & 0.2896 & -0.2367 \\ -0.0272 & 0.8101 & -0.0032 & 0.0323 \\ 0.0447 & 0.1886 & 0.7317 & 0.2354 \\ 0.0010 & 0.1888 & 0.0545 & 0.9115 \end{bmatrix}, B = \begin{bmatrix} 0.0006 & -0.0239 \\ 0.2567 & 0.0002 \\ 0.0837 & -0.1346 \\ 0.0837 & -0.0046 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, V = I_p, W = \emptyset.$$

The time horizon is taken to be $T = 5$ with maximum possible delay of 2 steps. This results in the delayed data model that can be expressed as the fixed-length language containing 3^5 words, i.e. $\mathcal{L} = \{\mathcal{W}_1, \ldots, \mathcal{W}_{243}\}$. For the proposed estimator, according to Definitions 2–4, we can find the corresponding event-based language $\mathcal{L}^E$ as well as the reduced language $\mathcal{L}^{E'}$. The measurement noise bound $\eta_v = 0.05$ is chosen to cover 5 standard deviations of $v \sim \mathcal{N}(0, 0.01^2)$, and by solving the robustified problem (10) with the cost function $J(\cdot) = \mu_1 + \sum_{k=0}^T \sum_{\alpha=1}^{|\mathcal{L}^{E'}|} \mu_{2,k}^\alpha$, we obtain recovery levels of $\mu_1 = 0.33$ and $\max_{k,\alpha}(\mu_{2,k}^\alpha) = 0.6912$.

To compare the performance of our proposed design with the Kalman filter based estimator design for systems with delayed data in [9], we let $x(0) = [1, 1, 1, 1]^\top$ and the true delay pattern be $\mathcal{W}_{sim} = 21210$. For the simulation, we randomly generated the initial state error and noise signals from truncated normal distributions with zero means and covariance matrices $P_0 = (\mu_1/5)^2 I_4$, $Q = \emptyset$ and $R = (\eta_v/5)^2 I_2$, where the initial error and noise bounds, μ_1, η_v, represent 5 times their standard deviations. Figure 1 shows the results of 50 runs, where the estimation errors using the proposed estimator stay within the guaranteed bounds, as desired, and are much less than the estimation errors from [9], which are not within the bounds, as one may expect.

B. Adaptive Cruise Control (Comparison with [16])

In the previous example, we showcased the capability of the proposed observer in the case of delay scenarios. Since

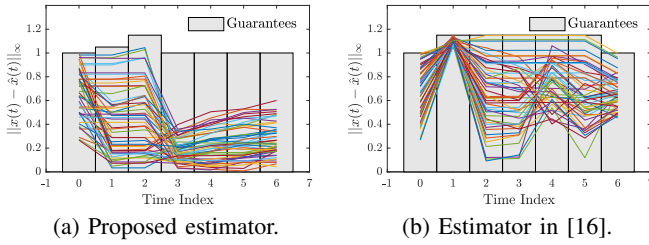(a) Proposed estimator.　　(b) Estimator in [16].

Fig. 2: Estimator comparison for missing data at $k = 1$.

missing data is basically a special case of delays beyond the finite time horizon, we also compare the proposed observer for the missing data scenario with another missing data (only) estimator in [16]. We use the same example presented in [16] of an adaptive cruise control, with the time horizon of $T = 6$ and for the sake of comparison, the equalized recovery level $\mu_1 = 1$ is specified instead of letting it be a decision variable[1]. The equivalent language to the one in [16] is used, i.e., $\mathcal{L} = \{060000, 006000, 000600, 000060\}$. After running the optimization problem, the maximum value of the intermediate upper bound obtained is $\mu_2 \triangleq \max_{k,\alpha} \mu_{2,k}^\alpha = 1.1498$, which is the same value obtained in [16]. Contrary to [16], the value of μ_2, being time-varying in our approach, is not always at its maximum, hence guaranteeing less error even during the intermediate phase. In our simulation, the true missing data pattern is $\mathcal{W}_{sim} = \{060000\}$ that corresponds to missing data at $k = 1$, and we compared the result with those in [16]. A comparison of both the estimators is depicted in Figure 2 that shows that the proposed estimator performs better than the estimator in [16] for the missing data scenario.

V. Conclusions

In this paper, we focused on the problem of synthesizing a dynamic state observer that has the ability to achieve equalized recovery when a discrete-time linear time-varying system is subjected to delayed or missing data in a finite time horizon. To achieve this, we constructed a reduced event-based language capable of capturing the set of indistinguishable event sequences that different delay mode sequences in the delayed data language correspond to, and augmented associated constraints to a novel equalized recovery estimator constructed with time-varying intermediate levels and the recovery level as a decision variable. When compared to existing designs, our proposed estimator can adapt the estimator gains at run time based on observed prefix of the language and can directly optimize the recovery and intermediate levels, leading to improved estimation performance.

References

[1] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.

[2] Z. Jin, C.-K. Ko, and R. M. Murray, "Estimation for nonlinear dynamical systems over packet-dropping networks," in *American Control Conference*. IEEE, 2007, pp. 5037–5042.

[3] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[4] S. C. Smith and P. Seiler, "Estimation with lossy measurements: jump estimators for jump systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, pp. 2163–2171, 2003.

[5] G. Battistelli, L. Benavoli, and L. Chisci, "State estimation with remote sensors and intermittent transmissions," *Systems & Control Letter*, vol. 61, no. 1, pp. 155–164.

[6] S. Zhang and Y. Bar-Shalom, "Optimal update with multiple out-of-sequence measurements with arbitrary arriving order," *IEEE Trans. on Aerospace & Electronic Systems*, vol. 48, no. 4, pp. 3116–3132, 2012.

[7] X. Wang, W. Qin, Y. Bai, and N. Cui, "Cooperative target localization using multiple uavs with out-of-sequence measurements," *Aircraft Engineering and Aerospace Techn.*, vol. 89, no. 1, pp. 112–119, 2017.

[8] I. Matei and J. S. Baras, "Optimal state estimation for discrete-time Markovian jump linear systems, in the presence of delayed output observations," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2235–2240, 2011.

[9] H. Zhang, G. Feng, and C. Han, "Linear estimation for random delay systems," *Systems & Control Letters*, vol. 60, no. 7, pp. 450–459, 2011.

[10] J. Shamma and K. Tu, "Set-valued observers and optimal disturbance rejection," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 253–264, 1999.

[11] M. Milanese and A. Vicino, "Optimal estimation theory for dynamic systems with set membership uncertainty: An overview," *Automatica*, vol. 27, no. 6, pp. 997–1009, 1991.

[12] J. Chen and C. Lagoa, "Observer design for a class of switched systems," in *IEEE CDC/ECC*, 2005, pp. 2945–2950.

[13] F. Blanchini and M. Sznaier, "A convex optimization approach to synthesizing bounded complexity ℓ^∞ filters," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 216–221, 2012.

[14] K. J. Rutledge, S. Z. Yong, and N. Ozay, "Optimization-based design of bounded-error estimators robust to missing data," *IFAC-PapersOnLine*, vol. 51, no. 16, pp. 157–162, 2018.

[15] S. M. Hassaan, Q. Shen, and S. Z. Yong, "Bounded-error estimator design with missing data patterns via state augmentation," in *American Control Conference (ACC)*, 2019, pp. 447–452.

[16] K. J. Rutledge, S. Z. Yong, and N. Ozay, "Prefix-based bounded-error estimation with intermittent observations," in *American Control Conference (ACC)*, 2019, pp. 4320–4325.

[17] K. Rutledge, S. Z. Yong, and N. Ozay, "Finite horizon constrained control and bounded-error estimation in the presence of missing data," *Nonlinear Analysis: Hybrid Systems*, vol. 36, p. 100854, 2020.

[18] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization*. Princeton University Press, 2009.

[19] A. Wächter and L. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Math. Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[20] Y. Shoukry, J. Araujo, P. Tabuada, M. Srivastava, and K. H. Johansson, "Minimax control for cyber-physical systems under network packet scheduling attacks," in *ACM HiCoNS*, 2013, pp. 93–100.

Appendix

Matrices and vectors in Theorem 1 are defined as follows:

$$W = \begin{bmatrix} W_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_{T-1} \end{bmatrix}, V = \begin{bmatrix} V_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & V_{T-1} \end{bmatrix},$$

$$A = \begin{bmatrix} I_n \\ A_0^1 \\ \vdots \\ A_0^T \end{bmatrix}, C = \begin{bmatrix} C_0 & 0 & \cdots & 0 & 0 \\ 0 & C_1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & C_{T-1} & 0 \end{bmatrix}, H = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ A_1^1 & 0 & 0 & \cdots & 0 \\ A_1^2 & A_2^2 & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ A_1^T & A_2^T & A_3^T & \cdots & A_T^T \end{bmatrix},$$

$$M^\alpha = \begin{bmatrix} M_{(0,0)}^\alpha & 0 & \cdots & 0 \\ M_{(1,0)}^\alpha & M_{(1,1)}^\alpha & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ M_{(T-1,0)}^\alpha & M_{(T-1,1)}^\alpha & \cdots & M_{(T-1,T-1)}^\alpha \end{bmatrix}, \Phi^\alpha = \begin{bmatrix} I_n \\ \Phi_0^{\alpha,1} \\ \vdots \\ \Phi_0^{\alpha,T} \end{bmatrix},$$

$$L^\alpha = \begin{bmatrix} L_0^\alpha & 0 & \cdots & 0 \\ 0 & L_1^\alpha & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & L_{T-1}^\alpha \end{bmatrix}, \Gamma^\alpha = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \Phi_1^{\alpha,1} & 0 & 0 & \cdots & 0 \\ \Phi_1^{\alpha,2} & \Phi_2^{\alpha,2} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \Phi_1^{\alpha,T} & \Phi_2^{\alpha,T} & \Phi_3^{\alpha,T} & \cdots & \Phi_T^{\alpha,T} \end{bmatrix},$$

for all $\alpha \in \mathbb{N}_1^{|\mathcal{L}^{E'}|}$, where $A_i^k = A_{k-1}A_{k-2}...A_i$, $\Phi_i^{\alpha,k} = \Phi_{k-1}^\alpha \Phi_{k-2}^\alpha ... \Phi_i^\alpha$ and $\Phi_k^\alpha = A_k - L_k^\alpha C_k$.