Sequence-Based Receding Horizon Control over Networks With Delays and Data Losses

Florian Rosenthal[1], Maxim Dolgov[2], and Uwe D. Hanebeck[1]

***Abstract*— In this paper, we are concerned with sequence-based receding horizon control over networks. We address the most general case where acknowledgments are provided but are also subject to delays and losses. This is in contrast to the majority of the related work in literature, where they are either delivered instantaneously and without losses or not issued at all. As in the case where acknowledgments are not issued, the separation principle does not hold in the considered setup, rendering the optimal control law generally nonlinear. Based on previous results, we present an iterative algorithm for the computation of the parameters of a linear receding horizon controller that does not assume separation a priori, taking the dual effect into account. The resulting controller is optimal in the sense that it minimizes an upper bound of the underlying quadratic cost function with respect to the control sequences. Its performance is demonstrated in a numerical example.**

I. Introduction

Employing general-purpose networks in control loops does not only simplify installation and maintenance but also provides enhanced flexibility in comparison to dedicated point-to-point connections [1], [2]. Consequently, such Networked Control Systems (NCS) have become increasingly important in a variety of fields and applications [3]. However, using IP-based networks, such as WiFi or Ethernet, introduces additional factors that are known to impact the achievable control performance [4], [5].

In particular, packet delays and losses are critical since these can lead to delayed or missing control inputs at the actuator side. One popular approach to tackle this issue, known as *sequence-based control* in literature [6]–[8], is to transmit a sequence $\underline{U}_k$ of control inputs which also consists of inputs for the next, say N, time steps in addition to the current one $\underline{u}_k$. Such controllers are typically based on receding horizon principles [7], [9], build upon nominal controllers that disregard the network [10], or directly minimize a cost function with respect to control sequences [11], [12].

It is a well-known insight that the transmission of acknowledgment packets by the actuator upon reception of control inputs is essential for the existence of tractable control policies and to avoid a *dual effect* [13] even when only losses are considered [4]. However, this property does no longer hold when the acknowledgments can get lost as well [12],

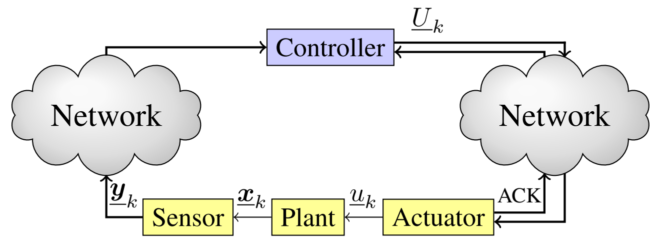

Fig. 1: Schematical overview of the considered setup.

[14]. As a consequence, in such situations one must rely on suitable approximations to obtain tractable control laws. For instance, optimal linear controllers for scenarios where no acknowledgments are provided[1] have been derived in [11], [15], and more recently in [16]. The work [11] also considered packet delays.

Yet, to the best of our knowledge, the most general case, namely the case where acknowledgments are provided but also subject to delays and losses, has not gained much attention in literature. Previously, in [17], we considered state estimation in such setups and introduced an estimator based on results for the UDP-like case. Similarly, in this paper, we will use ideas for sequence-based UDP-like control presented in [11] in order to introduce a receding horizon control approach for NCS similar to the one sketched in Fig. 1, where acknowledgments sent out by the actuator can also be delayed or lost.

More precisely, the contribution of this work is as follows. First, we present a novel holistic model that expresses both the control loop and the network in the described setup, thereby extending results from [12] for TCP-like and from [11] for UDP-like networks. Second, based on this model and assuming a linear controller, we present an iterative algorithm for the computation of the controller parameters. The resulting controller, in a receding horizon manner, minimizes a quadratic cost function at every time step and does not assume separation a priori, thus taking the dual effect into account.

Notation: Throughout this paper, vectors will be indicated by underlined letters ($\underline{x}$), random vectors will be underlined and in bold ($\underline{\boldsymbol{x}}$), and we will employ boldface capital letters to indicate matrices, e.g., $\mathbf{A}$. We use $\mathbf{I}_n$ to denote the n-dimensional identity matrix, $\mathbf{0}$ to denote zero matrices of arbitrary dimension, and a subscript k, e.g., $\underline{x}_k$, to indicate the time step. Transposition of a vector or matrix

[1]Florian Rosenthal and Uwe D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. Email: florian.rosenthal@kit.edu, uwe.hanebeck@ieee.org

[2]Maxim Dolgov is with the Robert Bosch GmbH, Corporate Research. Email: maxim.dolgov@de.bosch.com

*This work is supported by the German Science Foundation (DFG) within the Priority Programme 1914 "Cyber-Physical Networking".

[1]In NCS literature, such networks are called *UDP-like*, while networks that provide instantaneous acknowledgments that do not get lost, are referred to as *TCP-like*.

is indicated by $\underline{x}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}$, and $\mathbf{A} \geq 0$ ($\mathbf{A} > 0$) means that the matrix $\mathbf{A}$ is positive semidefinite (positive definite). The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$, and $\underline{e}_D$ denotes the unit vector (of arbitrary dimension) with one at position D and zero elsewhere. Finally, $\mathbb{1}_{i=j}$ is the indicator function, i.e., $\mathbb{1}_{i=j} = 1$ if $i = j$ and 0 otherwise, and the shortcut $a_{0:n}$ is used to indicate a sequence $a_0, a_1, \ldots, a_n$.

II. Problem Formulation

Consider the NCS illustrated in Fig. 1, where all components are synchronized and time stamps are attached to data packets upon transmission. The linear, discrete-time dynamics of the plant is described by

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}_k \underline{\boldsymbol{x}}_k + \mathbf{B}_k \underline{u}_k + \underline{\boldsymbol{w}}_k \,, \tag{1}$$

$$\underline{\boldsymbol{y}}_k = \mathbf{C}_k \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k \,, \tag{2}$$

with $\underline{\boldsymbol{x}}_k \in \mathbb{R}^d$ the plant state, $\underline{u}_k \in \mathbb{R}^l$ the control input, and $\underline{\boldsymbol{y}}_k \in \mathbb{R}^m$ the measurement. The Gaussian noise processes $\underline{\boldsymbol{w}}_k$ and $\underline{\boldsymbol{v}}_k$ are white and zero mean with covariance matrices $\mathbf{W}_k$ and $\mathbf{V}_k$, respectively, and mutually independent for any two time steps k, k'. The initial plant state $\underline{\boldsymbol{x}}_0$ is Gaussian with given mean $\hat{\underline{x}}_0$ and covariance $\mathbf{X}_0$, and independent of $\underline{\boldsymbol{w}}_i$ and $\underline{\boldsymbol{v}}_j$.

The sensor is collocated with the plant and sends the measurement to the remote controller. In the communication channel between the sensor and the controller, the measurements can be delayed or even get lost. As a consequence, none, one, or multiple measurements can arrive at the controller at a given time step. We will denote the set of received measurements at every time step by $\mathcal{Y}_k$. In order to treat delays and losses in a coherent manner throughout this paper, we interpret losses as infinite delays. This allows us to model the delay of a data packet sent from the sensor to the controller at time k by the random variable $\boldsymbol{\tau}_k^{SC} \in \mathbb{N}_0$. Additionally, we assume that the $\boldsymbol{\tau}_k^{SC}$ are independent and identically distributed (i.i.d.) and that the corresponding probability mass function (PMF) f^{SC} is known.

The control inputs transmitted from the remote controller to the actuator, which is attached to the plant, are also subject to delays and losses due to the network. Again, we assume that both can be modeled by i.i.d. random variables $\boldsymbol{\tau}_k^{CA}$ with PMF f^{CA}. In order to compensate for these effects, the controller transmits predicted control inputs for the next N time steps together with the current one. That is, at each time step the data packet that is sent to the actuator contains a sequence of $N+1$ control inputs

$$\underline{U}_k = \left[\underline{u}_{k|k}^{\mathrm{T}} \; \underline{u}_{k+1|k}^{\mathrm{T}} \; \cdots \; \underline{u}_{k+N|k}^{\mathrm{T}} \right]^{\mathrm{T}} \in \mathbb{R}^{(N+1)l} \,.$$

Here the notation $\underline{u}_{k+i|k}$ is employed to indicate that the control input is computed at time step k for application at time $k+i$. At the plant side, the actuator performs an *active packet dropout strategy* [2]: From the set of received control sequences, only the most recent one, i.e., the sequence with the largest time stamp, is kept. The control inputs from this sequence are then fed into the plant one after another until a newer sequence arrives at the actuator. Each time the buffered control sequence is replaced, an acknowledgment (ACK) is issued and sent back to the controller. It is important to emphasize that not every received data packet is acknowledged, but only that one corresponding to the new control sequence in use, so that these ACKs can be regarded as *application layer acknowledgments*[2]. Hence, from the network's perspective, they are regular data packets that can also be delayed or get lost. As above, we use i.i.d random variables $\boldsymbol{\tau}_k^{AC}$ with PMF f^{AC} to describe both effects. Consequently, the controller can receive multiple ACKs at every time step. It will be shown in Section III-A that the set of received ACKs, denoted by $\mathcal{A}_k$ in the remainder of this paper, enables the controller to infer control inputs that where applied in the past. Clearly, it may happen, due to consecutive packet losses or large delays, that the buffered sequence is not replaced early enough, so that no more control inputs are available. In such cases, the default input $\underline{u}_k^{df} = \underline{0}$ is used.

In this setup, which is visualized in Fig. 1, consider the usual quadratic cost function

$$\begin{aligned} \mathcal{J}_k^K &= \mathrm{E}\Big\{ \underline{\boldsymbol{x}}_{k+K}^{\mathrm{T}} \mathbf{Q}_K \underline{\boldsymbol{x}}_{k+K} \\ &\quad + \textstyle\sum_{n=0}^{K-1} \underline{\boldsymbol{x}}_{k+n}^{\mathrm{T}} \mathbf{Q}_n \underline{\boldsymbol{x}}_{k+n} + \underline{u}_{k+n}^{\mathrm{T}} \mathbf{R}_n \underline{u}_{k+n} \;\Big|\; \mathcal{I}_k \Big\} \,, \end{aligned} \tag{3}$$

where $K \in \mathbb{N}$ is the horizon length, and $\mathbf{Q}_i \geq 0$ and $\mathbf{R}_i > 0$ are the state and input weightings. The information set $\mathcal{I}_k$ available to the controller is given by

$$\mathcal{I}_k = \left\{ \hat{\underline{x}}_0, \mathbf{X}_0, \underline{U}_{0:k-1}, \mathcal{Y}_{0:k}, \mathcal{A}_{0:k} \right\} \,.$$

Our goal is, at every time step, to minimize the cost function (3) with respect to the control sequences $\underline{U}_k, \underline{U}_{k+1}, \ldots, \underline{U}_{k+K-1}$. In a receding horizon manner, the first element $\underline{U}_k^*$ of the minimizer is then transmitted to the plant before the optimization is carried out again at the next time step. Similar to [11], the key to finding a solution to this problem is to perform an appropriate state augmentation.

III. Derivation of the Control Law

In this section, we will first perform a state augmentation that allows us to derive a stochastic model to jointly express the original system, given by (1) and (2), and the underlying network by means of a single Markov jump linear system (MJLS) [19] with *two* jumping parameters.[3] Based on this model and a control law assumption, we will then formulate the cost function (3) in terms of the resulting closed-loop dynamics. Finally, we conclude this section by presenting an iterative algorithm for the computation of the controller parameters, which adapts ideas from [20].

[2]These ACKs are not to be confused with the dedicated acknowledgment packets that are issued upon successful transmissions by certain transport layer protocols in real networks. A common example is TCP, which retransmits data if such acknowledgments are delayed or missing, thus enhancing the reliability of the communication. However, since this trades losses for large delays, TCP is usually not desired in control applications [18].

[3]The jumping parameter is usually called *mode* of the system.

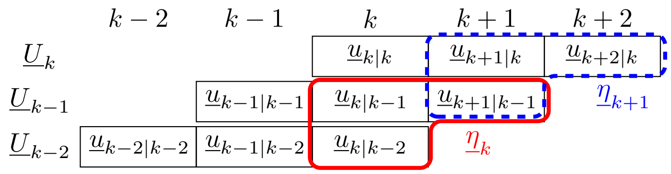

Fig. 2: Visualization of the relationship between $\underline{\eta}_{k+1}$, $\underline{\eta}_k$, and $\underline{U}_k$ according to (4) for $N = 2$.

A. Augmented System Dynamics

In the given setup, the controller does not exactly know which control input is actually applied due to the packet delays and losses introduced by the network. The resulting uncertainty must be considered in the computation of future control sequences. This can be achieved by i) defining a vector $\underline{\eta}_k$ encompassing all control inputs from past control sequences that are still applicable at time k or later, and by ii) introducing a discrete, scalar random variable $\boldsymbol{\theta}_k$, which allows for expressing the actual input in terms of this vector. In the following, we briefly summarize this approach, for more detailed derivations, refer to [17], [21].

Formally, we define $\underline{\eta}_k$ as

$$\underline{\eta}_k = \begin{bmatrix} \begin{bmatrix} \underline{u}_{k|k-1}^{\mathrm{T}} & \underline{u}_{k+1|k-1}^{\mathrm{T}} & \cdots & \underline{u}_{k+N-1|k-1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \begin{bmatrix} \underline{u}_{k|k-2}^{\mathrm{T}} & \underline{u}_{k+1|k-2}^{\mathrm{T}} & \cdots & \underline{u}_{k+N-2|k-2}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \vdots \\ \begin{bmatrix} \underline{u}_{k|k-N+1}^{\mathrm{T}} & \underline{u}_{k+1|k-N+1}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \\ \underline{u}_{k|k-N} \end{bmatrix} \in \mathbb{R}^{\frac{lN(N+1)}{2}} ,$$

with dynamics according to

$$\underline{\eta}_{k+1} = \mathbf{F}\underline{\eta}_k + \mathbf{G}\underline{U}_k \,, \tag{4}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(N-1)l} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{(N-2)l} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_l & \mathbf{0} \end{bmatrix} ,$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{Nl} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} .$$

As illustrated in Fig. 2 for $N = 2$, $\mathbf{F}$ is used in (4) to discard obsolete entries from $\underline{\eta}_k$, while $\mathbf{G}$ adds the relevant entries from $\underline{U}_k$.

Based on the observation that the control input actually applied at time k is either the default input ($\underline{u}_k^{df} = \underline{0}$) or must be part of one of the control sequences $\underline{U}_{k-N}, \underline{U}_{k-N+1}, \ldots, \underline{U}_k$, namely the one currently buffered by the actuator, we introduce

$$\boldsymbol{\theta}_k = \begin{cases} k - \tilde{k} & \text{if } \underline{U}_{\tilde{k}} \text{ is currently buffered} \\ N+1 & \text{else} \end{cases} , \tag{5}$$

with $k - N \leq \tilde{k} \leq k$. Then we can express the actually applied control input by virtue of

$$\underline{\boldsymbol{u}}_k = \mathbf{H}^{(\boldsymbol{\theta}_k)}\underline{\eta}_k + \mathbf{J}^{(\boldsymbol{\theta}_k)}\underline{U}_k \,, \tag{6}$$

with

$$\begin{aligned} \mathbf{H}^{(\boldsymbol{\theta}_k)} &= \begin{bmatrix} \mathbb{1}_{\boldsymbol{\theta}_k=1}\mathbf{I}_l & \mathbf{0} & \mathbb{1}_{\boldsymbol{\theta}_k=2}\mathbf{I}_l & \mathbf{0} & \ldots & \mathbb{1}_{\boldsymbol{\theta}_k=N}\mathbf{I}_l \end{bmatrix} , \\ \mathbf{J}^{(\boldsymbol{\theta}_k)} &= \begin{bmatrix} \mathbb{1}_{\boldsymbol{\theta}_k=0}\mathbf{I}_l & \mathbf{0} \end{bmatrix} . \end{aligned} \tag{7}$$

Note that it follows from (5) that $\boldsymbol{\theta}_k \in \{0, \ldots, N+1\}$, and from (6) and (7) we can conclude that $\boldsymbol{\theta}_k = N+1$ means that the default input $\underline{u}_k^{df} = \underline{0}$ is applied. For all other cases, the value of $\boldsymbol{\theta}_k$ denotes the age of the currently buffered control sequence. Regarding $\boldsymbol{\theta}_k$, we also have the following result, which has been proved in [21].

Theorem 1 (from [21]) *The process $\{\boldsymbol{\theta}_k\}$ is a Markov chain with lower Hessenberg transition matrix*

$$\mathbf{T} = \begin{bmatrix} p_0 & q_0 & 0 & \cdots & \cdots & \cdots & 0 \\ p_0 & p_1 & q_1 & \ddots & & & \vdots \\ p_0 & p_1 & p_2 & q_2 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ p_0 & p_1 & \ddots & \ddots & p_{N-1} & q_{N-1} & 0 \\ p_0 & p_1 & p_2 & \ddots & p_{N-1} & p_N & q_N \\ p_0 & p_1 & p_2 & \cdots & p_{N-1} & p_N & q_N \end{bmatrix} , \tag{8}$$

where $p_l = f^{CA}(l)$, $q_l = 1 - \sum_{m=0}^{l} p_m$, and the entries $t_{ij} = \mathrm{P}[\boldsymbol{\theta}_{k+1} = j \mid \boldsymbol{\theta}_k = i]$ are the transition probabilities.

Equipped with $\underline{\eta}_k$ and $\boldsymbol{\theta}_k$, we are able to formulate the cost function (3) in terms of $\underline{U}_k$. However, the computation of the sequences requires that the controller maintains a state estimate because only the noisy measurements $\mathcal{Y}_{0:k}$ are available. Since in the considered setup measurements can reach the controller with arbitrarily large delays, the following assumption is necessary to implement a feasible controller with finite (and fixed-size) memory.

Assumption 1 *Measurements with a delay larger than M time steps are discarded.*

Discarding measurements is clearly always suboptimal. However, remaining optimal would require infinite memory when unbounded delays are possible [22]. Due to this assumption, it is clear that, at every time step, it suffices to consider the measurements $\underline{\boldsymbol{y}}_{k-M:k}$. If any of these measurements arrives at the current time step, it is part of $\mathcal{Y}_k$ and, hence, available for state estimation. On the other hand, we can identify two cases where a measurement is not part of $\mathcal{Y}_k$. In the first case, the measurement has arrived at an earlier time step, i.e., it was part of $\mathcal{Y}_{\tilde{k}}$ for some $\tilde{k} < k$, and has thus already been processed. In the second case, the measurement has not yet arrived.

When we define the random variable $\gamma_{k|k-i} \in \{0, 1, 2\}$ to encode whether the measurement $\underline{\boldsymbol{y}}_{k-i}$ i) has not yet arrived

($\gamma_{k|k-i} = 0$), ii) arrives at the controller at the current time k ($\gamma_{k|k-i} = 1$), or iii) has already been processed in a previous time step ($\gamma_{k|k-i} = 2$), the following lemma holds.

Lemma 1 *For every $i \in \{0, \dots, M\}$, $\{\gamma_{k|k-i}\}$ is a Markov chain with transition matrix*

$$\mathbf{Z}^{(i)} = \begin{bmatrix} 1-a_i & a_i & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

with

$$a_i = \frac{f^{SC}(i+1)}{1 - \sum_{j=0}^{i} f^{SC}(j)}.$$

Proof: Since measurements must not be accumulated over time, we get that $z_{12}^{(i)} = \mathrm{P}[\gamma_{k+1|k-i} = 2 \mid \gamma_{k|k-i} = 1] = 1$, from which the second row of $\mathbf{Z}^{(i)}$ follows. Likewise, it is obvious that $z_{22}^{(i)} = \mathrm{P}[\gamma_{k+1|k-i} = 2 \mid \gamma_{k|k-i} = 2] = 1$ must hold, which yields the last row. For the first row of $\mathbf{Z}^{(i)}$ we have $z_{02}^{(i)} = \mathrm{P}[\gamma_{k+1|k-i} = 2 \mid \gamma_{k|k-i} = 0] = 0$, because a measurement that has not arrived at time k cannot be processed prior to its arrival, which is at $k+1$ at the earliest. To conclude the proof, it thus remains to compute

$$\begin{aligned} a_i = z_{01}^{(i)} &= \mathrm{P}[\gamma_{k+1|k-i} = 1 \mid \gamma_{k|k-i} = 0] \\ &= \frac{\mathrm{P}[\gamma_{k+1|k-i} = 1, \gamma_{k|k-i} = 0]}{\mathrm{P}[\gamma_{k|k-i} = 0]} \\ &= \frac{\mathrm{P}[\boldsymbol{\tau}_{k-i}^{SC} = i+1, \boldsymbol{\tau}_{k-i}^{SC} > i]}{\mathrm{P}[\boldsymbol{\tau}_{k-1}^{SC} > i]} \\ &= \frac{\mathrm{P}[\boldsymbol{\tau}_{k-i}^{SC} = i+1]}{\mathrm{P}[\boldsymbol{\tau}_{k-i}^{SC} > i]} = \frac{f^{SC}(i+1)}{1 - \sum_{j=0}^{i} f^{SC}(j)}. \end{aligned}$$

■

Note that the structure of $\mathbf{Z}^{(i)}$ implies that $\{\gamma_{k|k-i}\}$ is an absorbing Markov chain[4] with absorbing state $\gamma_{k|k-i} = 2$.

In the same manner, we can introduce a vector-valued random variable according to

$$\underline{\gamma}_k = \begin{bmatrix} \gamma_{k|k} & \gamma_{k|k-1} & \dots & \gamma_{k|k-M} \end{bmatrix}^{\mathrm{T}} \in \{0, 1, 2\}^{M+1},$$

to encode the availability of all measurements of interest, expressed in terms of the stacked vector

$$\underline{\tilde{\boldsymbol{y}}}_k = \begin{bmatrix} \underline{\boldsymbol{y}}_k^{\mathrm{T}} & \underline{\boldsymbol{y}}_{k-1}^{\mathrm{T}} & \underline{\boldsymbol{y}}_{k-2}^{\mathrm{T}} & \cdots & \underline{\boldsymbol{y}}_{k-M}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(M+1)m}.$$

Following ideas from [23], [24] we interpret $\underline{\gamma}_k$ as a ternary number with $M+1$ digits. In combination with Lemma 1, the following result is then readily obtained.

Theorem 2 *The process $\{\underline{\gamma}_k\}$ forms a Markov chain with 3^{M+1} states and transition matrix $\mathbf{Z}$ determined by the transition matrices $\mathbf{Z}^{(i)}$ from Lemma 1 and $f^{SC}(0)$.*

Proof: Consider the mapping

$$\phi(\underline{\gamma}_k) = 3^0 \gamma_{k|k} + 3^1 \gamma_{k|k-1} + \dots + 3^M \gamma_{k|k-M}. \tag{9}$$

[4]To be precise, this requires that $a_i > 0$ and hence $f^{SC}(i+1) > 0$, which we tacitly assume in the remainder, since we consider arbitrarily large delays.

It translates the ternary number given by $\underline{\gamma}_k$ into a nonnegative integer and is thus a bijection on the set $\{0, \dots, 3^{M+1} - 1\}$. This proves first part of the theorem.

For the second part, consider $\underline{\gamma}_{k+1} = \begin{bmatrix} j_0 & j_1 & \dots & j_M \end{bmatrix}^{\mathrm{T}}$ and $\underline{\gamma}_k = \begin{bmatrix} i_0 & i_1 & \dots & i_M \end{bmatrix}^{\mathrm{T}}$ with corresponding integers $\phi(\underline{\gamma}_{k+1}) = j$ and $\phi(\underline{\gamma}_k) = i$. For the entries of the transition matrix we then have

$$\begin{aligned} z_{ij} &= \mathrm{P}[\phi(\underline{\gamma}_{k+1}) = j \mid \phi(\underline{\gamma}_k) = i] \\ &= \mathrm{P}\Big[\gamma_{k+1|k+1} = j_0, \gamma_{k+1|k} = j_1, \dots, \gamma_{k+1|k-M+1} = j_M \\ &\qquad \Big| \gamma_{k|k} = i_0, \gamma_{k|k-1} = i_1, \dots, \gamma_{k|k-M} = i_M\Big] \\ &= \mathrm{P}[\gamma_{k+1|k+1} = j_0] \textstyle\prod_{a=0}^{M-1} \mathrm{P}[\gamma_{k+1|k-a} = j_{a+1} \mid \gamma_{k|k-a} = i_a] \\ &= \mathrm{P}[\gamma_{k+1|k+1} = j_0] \textstyle\prod_{a=0}^{M-1} z_{i_a j_{a+1}}^{(a)}, \end{aligned} \tag{10}$$

because $\gamma_{k+1|k-M}$ is not part of $\underline{\gamma}_{k+1}$ and the measurement delays are independent. The proof is concluded by noticing that

$$\mathrm{P}[\gamma_{k+1|k+1} = j_0] = \begin{cases} 1 - f^{SC}(0) & j_0 = 0 \\ f^{SC}(0) & j_0 = 1 \\ 0 & j_0 = 2 \end{cases}.$$

■

From (10) we can conclude that $\mathbf{Z}$ contains only relatively few nonzero elements and, moreover, exhibits a repetitive block structure. This observation is important, as this enables a storage-saving processing.

Now we introduce the augmented state $\underline{\boldsymbol{\xi}}_k$

$$\underline{\boldsymbol{\xi}}_k = \begin{bmatrix} \underline{\boldsymbol{x}}_k^{\mathrm{T}} & \underline{\boldsymbol{x}}_{k-1}^{\mathrm{T}} & \underline{\boldsymbol{x}}_{k-2}^{\mathrm{T}} & \dots & \underline{\boldsymbol{x}}_{k-M}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{(M+1)d},$$

which brackets the current plant state and all past states that are affected by $\underline{\tilde{\boldsymbol{y}}}_k$ together. Combining (1) and (6) yields the corresponding dynamics

$$\underline{\boldsymbol{\xi}}_{k+1} = \bar{\mathbf{A}}_k \underline{\boldsymbol{\xi}}_k + \bar{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} \underline{\eta}_k + \hat{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} \underline{U}_k + \underline{\bar{\boldsymbol{w}}}_k, \tag{11}$$

where

$$\bar{\mathbf{A}}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{I}_d & \mathbf{0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{I}_d & \mathbf{0} \end{bmatrix}, \quad \bar{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} = \begin{bmatrix} \mathbf{B}_k \mathbf{H}^{(\boldsymbol{\theta}_k)} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix},$$

$$\hat{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} = \begin{bmatrix} \left(\mathbf{B}_k \mathbf{J}^{(\boldsymbol{\theta}_k)}\right)^{\mathrm{T}} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}^{\mathrm{T}}, \quad \underline{\bar{\boldsymbol{w}}}_k = \begin{bmatrix} \underline{\boldsymbol{w}}_k^{\mathrm{T}} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}^{\mathrm{T}}.$$

Then, by means of another state augmentation according to $\underline{\boldsymbol{\psi}}_k = \begin{bmatrix} \underline{\boldsymbol{\xi}}_k^{\mathrm{T}} & \underline{\eta}_k^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$ and by combining (2), (4), and (11) we finally arrive at

$$\begin{aligned} \underline{\boldsymbol{\psi}}_{k+1} &= \tilde{\mathbf{A}}_k^{(\boldsymbol{\theta}_k)} \underline{\boldsymbol{\psi}}_k + \tilde{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} \underline{U}_k + \underline{\tilde{\boldsymbol{w}}}_k, \\ \underline{\tilde{\boldsymbol{y}}}_k &= \mathbf{S}^{(\underline{\gamma}_k)} \tilde{\mathbf{C}}_k \underline{\boldsymbol{\psi}}_k + \mathbf{S}^{(\underline{\gamma}_k)} \underline{\tilde{\boldsymbol{v}}}_k, \end{aligned} \tag{12}$$

with

$$\tilde{\mathbf{A}}_k^{(\boldsymbol{\theta}_k)} = \begin{bmatrix} \bar{\mathbf{A}}_k & \bar{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} \\ \mathbf{0} & \mathbf{F} \end{bmatrix}, \quad \tilde{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} = \begin{bmatrix} \hat{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} \\ \mathbf{G} \end{bmatrix}, \quad \underline{\tilde{\boldsymbol{w}}}_k = \begin{bmatrix} \underline{\bar{\boldsymbol{w}}}_k \\ \underline{0} \end{bmatrix},$$

and

$$\mathbf{S}^{(\underline{\gamma}_k)} = \begin{bmatrix} \mathbb{1}_{\gamma_{k|k}=1} & 0 & \cdots & 0 \\ 0 & \mathbb{1}_{\gamma_{k|k-1}=1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbb{1}_{\gamma_{k|k-M}=1} \end{bmatrix} \otimes \mathbf{I}_m ,$$

$$\tilde{\mathbf{C}}_k = \begin{bmatrix} \mathbf{C}_k & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{k-1} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_{k-M} & \mathbf{0} \end{bmatrix} , \quad \tilde{\underline{\boldsymbol{v}}}_k = \begin{bmatrix} \underline{\boldsymbol{v}}_k \\ \underline{\boldsymbol{v}}_{k-1} \\ \vdots \\ \underline{\boldsymbol{v}}_{k-M} \end{bmatrix} .$$

The covariance matrices of the augmented noise terms $\tilde{\underline{\boldsymbol{w}}}_k$ and $\tilde{\underline{\boldsymbol{v}}}_k$ are given by

$$\tilde{\mathbf{W}}_k = \begin{bmatrix} \mathbf{W}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} , \quad \tilde{\mathbf{V}}_k = \begin{bmatrix} \mathbf{V}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{k-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{V}_{k-M} \end{bmatrix} .$$

The initial state $\underline{\boldsymbol{\psi}}_0$ is Gaussian with mean and covariance

$$\hat{\underline{\psi}}_0 = \left[\hat{\underline{x}}_0^{\mathrm{T}} \; \ldots \; \hat{\underline{x}}_0^{\mathrm{T}} \; \underline{0}^{\mathrm{T}}\right]^{\mathrm{T}} , \quad \mathbf{\Sigma}_0 = \begin{bmatrix} \mathbf{1}_{M+1} \otimes \mathbf{X}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} ,$$

where $\mathbf{1}_{M+1}$ denotes the $(M+1)$-dimensional matrix of ones.

Eq. (12) provides the desired holistic model that describes both the original dynamical system and the underlying network. Since the two parameters $\boldsymbol{\theta}_k$ and $\underline{\boldsymbol{\gamma}}_k$ form independent Markov chains, (12) constitutes an MJLS with two independent modes. The mode $\underline{\boldsymbol{\gamma}}_k$ affects only the measurement equation and can be computed from the received measurements $\mathcal{Y}_k$. Hence, the true value of this mode is available to the controller at every time step.

In contrast, $\boldsymbol{\theta}_k$ only impacts the system dynamics and is *not* completely known to the controller. Instead, only a subset of the mode history is available. Recall from Section II that an ACK is sent back by the actuator once the buffered sequence is replaced by a newer one. Thus, at every time step, the controller can infer mode realizations only from $\mathcal{A}_k$, which, however, generally, will contain only ACKs from previous time steps due to the communication delays.

To illustrate this, suppose that $\mathcal{A}_k$ contains an ACK that was issued by the actuator at time $k-2$ to acknowledge the sequence $\underline{U}_{k-4}$, that is, the sequence that was sent by the controller at time step $k-4$. Then, from (5) it follows that $\boldsymbol{\theta}_{k-2} = 2$. Hence, the mode distribution at time k can be computed by means of

$$\underline{\mu}_k = \left(\mathbf{T}^2\right)^{\mathrm{T}} \underline{e}_3 , \tag{13}$$

with $\mathbf{T}$ given by (8). Further, note that the mode realization $\boldsymbol{\theta}_k = N+1$ will never be available to the controller since in such a case no applicable sequence would have been received by the actuator at time k, and thus no ACK would have been issued.

Hence, the considered problem boils down to finding a control law for an MJLS with one observed mode and one mode that is only partially and belatedly available. To the best of our knowledge, this particular problem has not yet been considered in literature, although there exist plenty of results concerning the control of MJLS where the mode is completely observed, e.g., [25], [26], is completely unobserved, e.g., [20], [27]–[29], or where the mode is available only in parts [30], [31] or with a constant delay [12], [23], [24].

B. Controller Design

For simplicity, in order to obtain a tractable solution, and in line with previous works on receding horizon control of MJLS [28], [30], we seek to find a linear control law

$$\hat{\underline{\boldsymbol{\psi}}}_{k+1} = \mathbf{M}_k \hat{\underline{\boldsymbol{\psi}}}_k + \mathbf{K}_k \tilde{\underline{\boldsymbol{y}}}_k , \tag{14}$$

$$\underline{U}_k = \mathbf{L}_k \hat{\underline{\boldsymbol{\psi}}}_k , \tag{15}$$

for system (12), where the mode-independent gains $\mathbf{M}_k$, $\mathbf{K}_k$, and $\mathbf{L}_k$ are to be determined such that (3) is minimized, and $\hat{\underline{\boldsymbol{\psi}}}_k$ is the controller's state estimate with initial condition $\hat{\underline{\psi}}_0$. Then, combining (12), (14), and (15) yields the closed-loop dynamics

$$\tilde{\underline{\boldsymbol{x}}}_{k+1} = \mathbf{\Gamma}_k^{(\boldsymbol{\theta}_k)} \tilde{\underline{\boldsymbol{x}}}_k + \mathbf{\Omega}_k \underline{\boldsymbol{\nu}}_k , \tag{16}$$

with

$$\tilde{\underline{\boldsymbol{x}}}_k = \left[\underline{\boldsymbol{\psi}}_k^{\mathrm{T}} \; \hat{\underline{\boldsymbol{\psi}}}_k^{\mathrm{T}}\right]^{\mathrm{T}} , \quad \underline{\boldsymbol{\nu}}_k = \left[\tilde{\underline{\boldsymbol{w}}}_k^{\mathrm{T}} \; \tilde{\underline{\boldsymbol{v}}}_k^{\mathrm{T}}\right]^{\mathrm{T}} ,$$

$$\mathbf{\Gamma}_k^{(\boldsymbol{\theta}_k)} = \begin{bmatrix} \tilde{\mathbf{A}}_k^{(\boldsymbol{\theta}_k)} & \tilde{\mathbf{B}}_k^{(\boldsymbol{\theta}_k)} \mathbf{L}_k \\ \mathbf{K}_k \tilde{\mathbf{S}}_k \tilde{\mathbf{C}}_k & \mathbf{M}_k \end{bmatrix} , \quad \mathbf{\Omega}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_k \tilde{\mathbf{S}}_k \end{bmatrix} ,$$

and where $\tilde{\mathbf{S}}_k = \mathrm{E}\left\{\mathbf{S}^{(\underline{\gamma}_k)}\right\}$. Using the expectation $\tilde{\mathbf{S}}_k$ in (16) eliminates the dependency of the closed-loop dynamics on the observed mode $\underline{\boldsymbol{\gamma}}_k$. This ensures that the optimization problem remains tractable. Considering all possible paths of the evolution of $\underline{\boldsymbol{\gamma}}_k$ is impractical because its state space grows exponentially with M. Given $\underline{\boldsymbol{\gamma}}_k$ with $\phi(\underline{\boldsymbol{\gamma}}_k) = j$ according to (9), $\tilde{\mathbf{S}}_{k+t}$ can easily be computed for the whole optimization horizon by means of

$$\tilde{\mathbf{S}}_{k+t} = \textstyle\sum_{i=0}^{3^{M+1}-1} \lambda_{k+t}^{(i)} \mathbf{S}^{(\phi^{-1}(i))} , \tag{17}$$

where the mode probabilities can be predicted according to

$$\underline{\lambda}_{k+t} = \left(\mathbf{Z}^{k+t}\right)^{\mathrm{T}} \underline{e}_{j+1} , \quad t = 0, 1, \ldots, K-1 , \tag{18}$$

and where ϕ^{-1} is the inverse of (9).

As in [20], we construct the second moment of $\tilde{\underline{\boldsymbol{x}}}_k$, conditioned on a particular mode $\boldsymbol{\theta}_k = i$,

$$\tilde{\mathbf{X}}_k^{(i)} = \mathrm{E}\left\{\tilde{\underline{\boldsymbol{x}}}_k \tilde{\underline{\boldsymbol{x}}}_k^{\mathrm{T}} \mathbb{1}_{\boldsymbol{\theta}_k = i}\right\} ,$$

with dynamics given by

$$\tilde{\mathbf{X}}_{k+1}^{(j)} = \textstyle\sum_{i=0}^{N+1} t_{ij} \left[\mathbf{\Gamma}_k^{(i)} \tilde{\mathbf{X}}_k^{(i)} \left(\mathbf{\Gamma}_k^{(i)}\right)^{\mathrm{T}} + \mu_k^{(i)} \mathbf{\Omega}_k \mathbf{N}_k \mathbf{\Omega}_k^{\mathrm{T}}\right] ,$$

where $\mathbf{N}_k = \mathrm{Cov}\{\underline{\boldsymbol{\nu}}_k\}$ and $\mu_k^{(i)} = \mathrm{P}[\boldsymbol{\theta}_k = i]$ the mode probability. By introducing the cost-to-go $\mathcal{V}_k^t$ for $t = K, \ldots, 1, 0$, for which $\mathcal{J}_k^K = \mathcal{V}_k^0$ holds, we reformulate the cost function (3) in terms of the second moment according to

$$\mathcal{V}_k^t = \sum_{n=t}^{K}\sum_{i=0}^{N+1}\mathrm{tr}\left[\tilde{\mathbf{Q}}_{k+n}^{(i)}\tilde{\mathbf{X}}_{k+n}^{(i)}\right] , \tag{19}$$

with

$$\tilde{\mathbf{Q}}_{k+K}^{(i)} = \begin{bmatrix}\bar{\mathbf{Q}}_K^{(i)} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} , \ \bar{\mathbf{Q}}_K^{(i)} = \begin{bmatrix}\mathbf{Q}_K & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix} ,$$

$$\tilde{\mathbf{Q}}_{k+n}^{(i)} = \begin{bmatrix}\bar{\mathbf{Q}}_n^{(i)} & \mathbf{0}\\ \mathbf{0} & \bar{\mathbf{J}}_{k+n}^{(i)}\end{bmatrix} , \ \bar{\mathbf{Q}}_n^{(i)} = \begin{bmatrix}\mathbf{Q}_n & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \bar{\mathbf{H}}_n^{(i)}\end{bmatrix} ,$$

and

$$\bar{\mathbf{J}}_{k+n}^{(i)} = \left(\mathbf{J}^{(i)}\mathbf{L}_{k+n}\right)^{\mathrm{T}}\mathbf{R}_n\mathbf{J}^{(i)}\mathbf{L}_{k+n} ,$$
$$\bar{\mathbf{H}}_n^{(i)} = \left(\mathbf{H}^{(i)}\right)^{\mathrm{T}}\mathbf{R}_n\mathbf{H}^{(i)} .$$

However, finding the optimal controller gains using the cost-to-go is not straightforward, since minimizing (19) subject to the second moment dynamics directly does not necessarily converge to a solution. This was also pointed out in [20], where the authors obtained a similar expression for the cost-to-go in a finite-horizon control problem.

In [20], this issue was circumvented by introducing a convex upper bound $\bar{\mathcal{V}}_k^t$ for the cost-to-go. Based on a variational approach, this upper bound was then minimized with respect to the controller gains by an iterative algorithm. The algorithm is straightforward to implement since neither descent directions nor step sizes must be determined during the iterations. Hence, adapting the algorithm from [20] to our considered scenario is very appealing.

To that end, first introduce $\overline{\mathbf{X}}_0^{(i)} = \underline{\mathbf{X}}_0^{(i)} = \mathbf{0}$ for $i = 0, \ldots, N$, and also $\overline{\mathbf{X}}_0^{(N+1)} = \mathbf{\Sigma}_0$ and $\underline{\mathbf{X}}_0^{(N+1)} = \mathbf{\Sigma}_0 + \underline{\hat{\psi}}_0\underline{\hat{\psi}}_0^{\mathrm{T}}$. Then, we can combine Lemma 1 and Lemma 2 from [20] in order to obtain the following upper bound for (19).

Theorem 3 *For $t = K, \ldots, 1, 0$, an upper bound $\bar{\mathcal{V}}_k^t \geq \mathcal{V}_k^t$ for the cost-to-go is given by*

$$\bar{\mathcal{V}}_k^t = \sum_{i=0}^{N+1}\mathrm{tr}\left[\overline{\mathbf{P}}_{k+t}^{(i)}\left(\overline{\mathbf{X}}_{k+t}^{(i)} + \underline{\mathbf{X}}_{k+t}^{(i)}\right) + \underline{\mathbf{P}}_{k+t}^{(i)}\overline{\mathbf{X}}_{k+t}^{(i)}\right] + \mu_{k+t}^{(i)}\overline{\omega}_{k+t}^{(i)} , \tag{20}$$

with $\overline{\mathbf{X}}_{k+t}^{(i)}$ and $\underline{\mathbf{X}}_{k+t}^{(i)}$ given by the recursions

$$\begin{aligned}\overline{\mathbf{X}}_{k+t+1}^{(j)} &= \sum_{i=0}^{N+1}t_{ij}\Big[\mu_{k+t}^{(i)}\left(\tilde{\mathbf{E}}_{k+t} + \tilde{\mathbf{W}}_{k+t}\right)\\ &+ \left(\tilde{\mathbf{A}}_{k+t}^{(i)} - \mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t}\right)\overline{\mathbf{X}}_{k+t}^{(i)}\\ &\cdot\left(\tilde{\mathbf{A}}_{k+t}^{(i)} - \mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t}\right)^{\mathrm{T}} + \tilde{\mathbf{D}}_{k+t}^{(i)}\underline{\mathbf{X}}_{k+t}^{(i)}\left(\tilde{\mathbf{D}}_{k+t}^{(i)}\right)^{\mathrm{T}}\Big] ,\end{aligned} \tag{21}$$

$$\begin{aligned}\underline{\mathbf{X}}_{k+t+1}^{(j)} &= \sum_{i=0}^{N+1}t_{ij}\Big[\mu_{k+t}^{(i)}\tilde{\mathbf{E}}_{k+t}\\ &+ \left(\mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t}\right)\overline{\mathbf{X}}_{k+t}^{(i)}\left(\mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t}\right)^{\mathrm{T}}\\ &+ \left(\mathbf{M}_{k+t} + \mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t}\right)\underline{\mathbf{X}}_{k+t}^{(i)}\\ &\cdot\left(\mathbf{M}_{k+t} + \mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t}\right)^{\mathrm{T}}\Big] ,\end{aligned} \tag{22}$$

where

$$\tilde{\mathbf{E}}_{k+t} = \mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{V}}_{k+t}\left(\mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\right)^{\mathrm{T}} ,$$
$$\tilde{\mathbf{D}}_{k+t}^{(i)} = \tilde{\mathbf{A}}_{k+t}^{(i)} - \mathbf{K}_{k+t}\tilde{\mathbf{S}}_{k+t}\tilde{\mathbf{C}}_{k+t} - \mathbf{M}_{k+t} + \tilde{\mathbf{B}}_{k+t}^{(i)}\mathbf{L}_{k+t} ,$$

and $\overline{\mathbf{P}}_{k+t}^{(i)}$, $\underline{\mathbf{P}}_{k+t}^{(i)}$, and $\overline{\omega}_{k+t}^{(i)}$ given by the backward recursions

$$\begin{aligned}\overline{\mathbf{P}}_{k+t}^{(i)} &= \bar{\mathbf{Q}}_t^{(i)} + \bar{\mathbf{J}}_{k+t}^{(i)} + \left(\tilde{\mathbf{D}}_{k+t}^{(i)}\right)^{\mathrm{T}}\mathcal{E}^{(i)}(\underline{\mathbf{P}}_{k+t+1})\tilde{\mathbf{D}}_{k+t}^{(i)}\\ &+ \left(\tilde{\mathbf{U}}_{k+t}^{(i)}\right)^{\mathrm{T}}\mathcal{E}^{(i)}(\overline{\mathbf{P}}_{k+t+1})\tilde{\mathbf{U}}_{k+t}^{(i)} ,\end{aligned} \tag{23}$$

$$\begin{aligned}\underline{\mathbf{P}}_{k+t}^{(i)} &= \bar{\mathbf{J}}_{k+t}^{(i)} + \left(\tilde{\mathbf{B}}_{k+t}^{(i)}\mathbf{L}_{k+t}\right)^{\mathrm{T}}\mathcal{E}^{(i)}(\overline{\mathbf{P}}_{k+t+1})\tilde{\mathbf{B}}_{k+t}^{(i)}\mathbf{L}_{k+t}\\ &+ \left(\tilde{\mathbf{O}}_{k+t}^{(i)}\right)^{\mathrm{T}}\mathcal{E}^{(i)}(\underline{\mathbf{P}}_{k+t+1})\tilde{\mathbf{O}}_{k+t}^{(i)} ,\end{aligned} \tag{24}$$

$$\begin{aligned}\overline{\omega}_{k+t}^{(i)} &= \mathrm{tr}\left[\mathcal{E}^{(i)}(\underline{\mathbf{P}}_{k+t+1} + \overline{\mathbf{P}}_{k+t+1})\tilde{\mathbf{W}}_{k+t}\right]\\ &+ \mathrm{tr}\left[\mathcal{E}^{(i)}(\underline{\mathbf{P}}_{k+t+1})\tilde{\mathbf{E}}_{k+t}\right] + \mathcal{E}^{(i)}(\overline{\omega}_{k+t+1}) ,\end{aligned} \tag{25}$$

that are initialized with $\overline{\mathbf{P}}_{k+K}^{(i)} = \bar{\mathbf{Q}}_K^{(i)}$, $\underline{\mathbf{P}}_{k+K}^{(i)} = \mathbf{0}$, and $\overline{\omega}_{k+K}^{(i)} = 0$, and where

$$\tilde{\mathbf{U}}_{k+t}^{(i)} = \tilde{\mathbf{A}}_{k+t}^{(i)} + \tilde{\mathbf{B}}_{k+t}^{(i)}\mathbf{L}_{k+t} ,$$
$$\tilde{\mathbf{O}}_{k+t}^{(i)} = \mathbf{M}_{k+t} - \tilde{\mathbf{B}}_{k+t}^{(i)}\mathbf{L}_{k+t} ,$$
$$\mathcal{E}^{(i)}(\mathbf{\Lambda}) = \sum_{j=0}^{N+1}t_{ij}\mathbf{\Lambda}^{(j)} .$$

The resulting iterative algorithm for the computation of the control sequence $\underline{U}_k$ is summarized in Algorithm 1. A method to evaluate the necessary condition corresponding to (20) to obtain the controller gains in *Step 6*, based on vectorization and minimum norm, is detailed in [20]. There it is also shown that the cost sequence created by the algorithm is monotonically decreasing, i.e., $\bar{\mathcal{V}}_k^{0,[c+1]} \leq \bar{\mathcal{V}}_k^{0,[c]}$, and indeed converges to $\min\bar{\mathcal{V}}_k^0$, the minimum of the bound.

An implementation of the algorithm is available on github as part of CoCPN-Sim [32].

IV. Numerical Example

In this section, we provide a numerical example to demonstrate the proposed algorithm. To that end, consider a double integrator plant given by (1) and (2) with parameters

$$\mathbf{A}_k = \begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix} , \ \mathbf{B}_k = \begin{bmatrix}0\\ 1\end{bmatrix} , \ \mathbf{W}_k = \begin{bmatrix}0.4^2 & 0\\ 0 & 0.4^2\end{bmatrix} ,$$
$$\mathbf{C}_k = \begin{bmatrix}1 & 0\end{bmatrix} , \ \mathbf{V}_k = 0.8^2 ,$$

and initial state

$$\underline{\hat{x}}_0 = \begin{bmatrix}10 & 0\end{bmatrix}^{\mathrm{T}} , \ \mathbf{X}_0 = 0.5^2\mathbf{I}_2 ,$$

that is to be controlled by the proposed controller using the cost function (3) with cost matrices $\mathbf{Q}_k = \mathbf{I}_2$ and $\mathbf{R}_k =$

Algorithm 1 One Cycle of the Proposed Control Algorithm

Input: Received measurements $\mathcal{Y}_k$, received ACKs $\mathcal{A}_k$
Output: Control Sequence $\underline{U}_k$

- *Step 1:* Construct $\tilde{\underline{\boldsymbol{y}}}_k$ from $\mathcal{Y}_k$, determine the corresponding mode $\underline{\gamma}_k$, and compute $\tilde{\mathbf{S}}_k, \ldots, \tilde{\mathbf{S}}_{k+K-1}$ using (17) and (18).
- *Step 2:* Get the most recent mode observation $\boldsymbol{\theta}_{\tilde{k}} = L$ from $\mathcal{A}_k$ and predict the mode distributions $\underline{\mu}_k, \ldots, \underline{\mu}_{k+K-1}$ using $\underline{e}_{L+1}$ in (13).
- *Step 3:* Initialize the iteration counter $c = 0$, set $\bar{\mathcal{V}}_k^{0,[c]} = \infty$ and choose initial controller gains $\mathbf{M}_{k+t}^{[c]}, \mathbf{K}_{k+t}^{[c]}, \mathbf{L}_{k+t}^{[c]}$ for $t = 0, \ldots, K-1$.
- *Step 4:* Set $\overline{\mathbf{X}}_k^{(i,[c])} = \overline{\mathbf{X}}_k^{(i)}$, $\underline{\mathbf{X}}_k^{(i,[c])} = \underline{\mathbf{X}}_k^{(i)}$. Then compute $\overline{\mathbf{X}}_{k+t}^{(i,[c])}$ and $\underline{\mathbf{X}}_{k+t}^{(i,[c])}$ for $t = 1, \ldots, K$ using $\mathbf{M}_{k+t}^{[c]}$, $\mathbf{K}_{k+t}^{[c]}$, and $\mathbf{L}_{k+t}^{[c]}$ in (21) and (22).
- *Step 5:* Initialize $\overline{\mathbf{P}}_{k+K}^{(i,[c+1])} = \bar{\mathbf{Q}}_K$, $\underline{\mathbf{P}}_{k+K}^{(i,[c+1])} = \mathbf{0}$, and $\overline{\omega}_{k+K}^{(i,[c+1])} = 0$, and set $t = K-1$.
- *Step 6:* Evaluate the necessary optimality condition of (20) to obtain the minimizing gains $\mathbf{M}_{k+t}^{[c+1]}, \mathbf{K}_{k+t}^{[c+1]}, \mathbf{L}_{k+t}^{[c+1]}$.
- *Step 7:* Compute $\overline{\mathbf{P}}_{k+t}^{(i,[c+1])}$, $\underline{\mathbf{P}}_{k+t}^{(i,[c+1])}$, and $\overline{\omega}_{k+t}^{(i,[c+1])}$ using the gains from *Step 6* in (23), (24), and (25).
- *Step 8:* If $t = 0$, go to *Step 9*. Otherwise, set $t = t-1$ and go back to *Step 6*.
- *Step 9:* Compute $\bar{\mathcal{V}}_k^{0,[c+1]}$ using (20). If $\bar{\mathcal{V}}_k^{0,[c]} - \bar{\mathcal{V}}_k^{0,[c+1]}$ is small enough, compute $\underline{U}_k$ by using $\mathbf{L}_k^{[c+1]}$ in (15) and update the state estimate using $\mathbf{M}_k^{[c+1]}$, $\mathbf{K}_k^{[c+1]}$, and $\tilde{\underline{\boldsymbol{y}}}_k$ in (14). Also, calculate $\overline{\mathbf{X}}_{k+1}^{(i)}$ and $\underline{\mathbf{X}}_{k+1}^{(i)}$ using $\mathbf{M}_k^{[c+1]}$, $\mathbf{K}_k^{[c+1]}$, and $\mathbf{L}_k^{[c+1]}$ in (21) and (22). Then terminate the algorithm. Otherwise set $c = c+1$ and return to *Step 4*.

1 in three different scenarios. In all scenarios, the delay distributions f^{CA} and f^{SC} are equal and given by the PMF depicted in Fig. 3. It is chosen such that the vast majority of packets is delayed at most two time steps and delays of zero time steps are also extremely unlikely. In the first scenario (referred to as Setup I in the following), we assume a TCP-like setup, that is, the ACKs from the actuator are received instantaneously and without losses. In the second scenario (Setup II), we consider the opposite extreme case, namely a UDP-like setup, where ACKs are not provided at all. Finally, in the third scenario (Setup III) the setup considered in this paper is employed, with delay distribution f^{AC} equal to f^{CA} and f^{SC}.

To evaluate the performance of the proposed algorithm, we compare it with the sequence-based, finite-horizon controller from [12] that was derived for the TCP-like setting. For the computation of the required state estimate, we use the (suboptimal) estimator from [17].

Every scenario is investigated in a Monte Carlo simulation with 500 runs, each of which comprising 200 time steps, and the optimization horizon used by the proposed controller is $K = 10$. Additionally, both controllers use $M = 2$ and the control sequence length is set to $N + 1 = 3$. To compare the

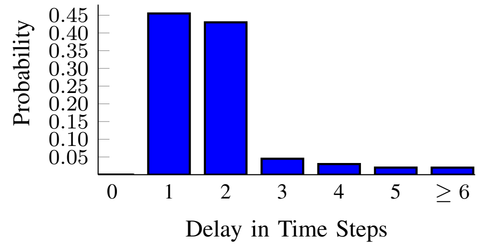

Fig. 3: PMF of the packet delays utilized in the example. The shortcut ≥ 6 is used to denote the probability that a packet is delayed more than 5 time steps or gets lost (infinite delay).

TABLE I: Median of the average costs $\mathcal{J}_{avg}$.

	Setup I	Setup II	Setup III
Controller from [12]	107.148	68.362	66.800
Proposed controller	94.921	95.998	95.938

performance of the controllers, we compute the average true costs $\mathcal{J}_{avg}$ in every run according to

$$\mathcal{J}_{avg} = \frac{1}{200}\left[\sum_{k=0}^{199} \underline{x}_k^{\mathrm{T}} \mathbf{Q}_k \underline{x}_k + \underline{u}_k^{\mathrm{T}} \mathbf{R}_k \underline{u}_k + \underline{x}_{200}^{\mathrm{T}} \mathbf{Q}_{200} \underline{x}_{200}\right] .$$

The resulting medians of $\mathcal{J}_{avg}$ for each controller and scenario are given in Table I.

The numbers indicate that the quality of control of the proposed controller is not influenced by the absence or availability of ACKs, as it performs similarly in all scenarios. An interesting result is that the proposed approach outperforms the controller from [12] in the TCP-like setting (Setup I) although the latter was developed for such setups. One reason for this might be the relatively large simulation time (200 time steps). While the proposed approach operates in a receding horizon fashion where the gains are recomputed every time step, the controller from [12] computes the gains in a finite-horizon fashion, i.e., only once and for the whole simulation time. Additionally, the used estimator may not be optimal in this setup.

However, it might be due to this estimator that the controller from [12] performs better than the proposed one in the scenarios without immediate ACKs (Setup II and Setup III) because it has been tailored to such setups [17]. This somewhat unexpected result could also be caused by the conservatism of the proposed controller resulting from upper-bounding the cost function.

V. Conclusions

In this paper, we addressed sequence-based receding horizon control over networks where acknowledgments, issued by the actuator upon reception of viable controller packets, may also be delayed or get lost. In this setting, which covers the ones usually considered in literature as special cases, namely TCP-like and UDP-like networks, we developed a novel holistic model to express both control loop and network at the same time. Based on this model, and by restricting ourselves to linear control laws to obtain a tractable solution, we then presented an iterative algorithm for the computation of the controller gains. The numerical example indicated that the proposed approach can also be used in TCP-like or UDP-like setups without loss of performance, which, however, has to be investigated more thoroughly in future work. Likewise, future work should assess the benefit of taking delayed acknowledgments into account in greater detail.

The proposed approach by itself also admits some room for improvement. First, the size of the augmented state $\underline{\psi}_k$ is determined by the chosen sequence length $N+1$ and the maximum measurement delay M, and, for instance, grows quadratically with N. The controller parameters scale up accordingly, which could make their computation by means of the proposed algorithm too computationally demanding. Consequently, this issue should be addressed in prospective research, for example by deriving a reduced-order, linear controller for system (12). Second, future work should be concerned with alternative ways for processing delayed and missing measurements because the Markov chain approach presented in this paper does not scale well. In this regard, it is worthwhile to investigate whether the "natural connection" between interacting multiple model (IMM) state estimation and control of MJLS, which has already been highlighted in [33], can be exploited in the given setup using the IMM-based estimator we presented in [17].

REFERENCES

[1] L. Zhang, H. Gao, and O. Kaynak, "Network-Induced Constraints in Networked Control Systems—A Survey," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 403–416, 2013.

[2] X. Zhang, Q. Han, and X. Yu, "Survey on Recent Advances in Networked Control Systems," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1740–1752, Oct 2016.

[3] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[4] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[5] W. M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nesic, "Networked Control Systems with Communication Constraints: Tradeoffs between Transmission Intervals, Delays and Performance," *IEEE Trans. Autom. Control*, vol. 55, no. 8, pp. 1781–1796, 2010.

[6] A. Bemporad, "Predictive Control of Teleoperated Constrained Systems with Unbounded Communication Delays," in *Proceedings of the 37th IEEE Conference on Decision and Control (CDC 1998)*, vol. 2. IEEE, Dec. 1998, pp. 2133–2138.

[7] V. Gupta, B. Sinopoli, S. Adlakha, A. Goldsmith, and R. Murray, "Receding Horizon Networked Control," in *Proc. Allerton Conf. Commun., Control Comput*, 2006.

[8] D. E. Quevedo and I. Jurado, "Stability of Sequence-Based Control with Random Delays and Dropouts," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1296–1302, May 2014.

[9] D. E. Quevedo and D. Nesic, "Input-to-State Stability of Packetized Predictive Control over Unreliable Networks Affected by Packet-Dropouts," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 370–375, 2011.

[10] G. Liu, "Predictive Controller Design of Networked Systems with Communication Delays and Data Loss," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 6, pp. 481–485, Jun. 2010.

[11] M. Dolgov, J. Fischer, and U. D. Hanebeck, "Infinite-Horizon Sequence-based Networked Control without Acknowledgments," in *Proceedings of the 2015 American Control Conference (ACC 2015)*, July 2015, pp. 402–408.

[12] J. Fischer, A. Hekler, M. Dolgov, and U. D. Hanebeck, "Optimal Sequence-Based LQG Control over TCP-like Networks Subject to Random Transmission Delays and Packet Losses," in *Proceedings of the 2013 American Control Conference (ACC 2013)*, Jun. 2013.

[13] Y. Bar-Shalom and E. Tse, "Dual Effect, Certainty Equivalence, and Separation in Stochastic Control," *IEEE Trans. Autom. Control*, vol. 19, no. 5, pp. 494–500, October 1974.

[14] E. Garone, B. Sinopoli, and A. Casavola, "LQG Control Over Lossy TCP-like Networks with Probabilistic Packet Acknowledgements," *International Journal of Systems, Control and Communications*, vol. 2, no. 1-3, pp. 55–81, 2010.

[15] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, "Optimal Linear LQG Control Over Lossy Networks without Packet Acknowledgment," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC 2006)*, Dec 2006, pp. 392–397.

[16] H. Lin, H. Su, P. Shi, Z. Shu, R. Lu, and Z. Wu, "Optimal Estimation and Control for Lossy Network: Stability, Convergence, and Performance," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4564–4579, Sept 2017.

[17] F. Rosenthal, B. Noack, and U. D. Hanebeck, "State Estimation in Networked Control Systems with Delayed and Lossy Acknowledgments," in *Multisensor Fusion and Integration in the Wake of Big Data, Deep Learning and Cyber Physical System*, S. Lee, H. Ko, and S. Oh, Eds. Springer International Publishing, 2018, pp. 22–38.

[18] J. Baillieul and P. J. Antsaklis, "Control and Communication Challenges in Networked Real-Time Systems," *Proc. IEEE*, vol. 95, no. 1, pp. 9–28, 2007.

[19] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-Time Markov Jump Linear Systems*. Springer Science & Business Media, 2006.

[20] M. Dolgov, G. Kurz, and U. D. Hanebeck, "Finite-Horizon Dynamic Compensation of Markov Jump Linear Systems without Mode Observation," in *Proceedings of the 2016 55th IEEE Conference on Decision and Control (CDC 2016)*, Dec 2016, pp. 2757–2762.

[21] J. Fischer, A. Hekler, and U. D. Hanebeck, "State Estimation in Networked Control Systems," in *Proceedings of the 15th International Conference on Information Fusion (Fusion 2012)*, Jul. 2012.

[22] L. Schenato, "Optimal Estimation in Networked Control Systems Subject to Random Delay and Packet Drop," *IEEE Trans. Autom. Control*, vol. 53, no. 5, pp. 1311–1317, Jun. 2008.

[23] J. Xiong and J. Lam, "Stabilization of Discrete-Time Markovian Jump Linear Systems via Time-Delayed Controllers," *Automatica*, vol. 42, no. 5, pp. 747 – 753, 2006.

[24] M. Liu, D. W. Ho, and Y. Niu, "Stabilization of Markovian Jump Linear System over Networks with Random Communication Delay," *Automatica*, vol. 45, no. 2, pp. 416 – 421, 2009.

[25] Y. Ji and H. J. Chizeck, "Controllability, Observability and Discrete-Time Markovian Jump Linear Quadratic Control," *International Journal of Control*, vol. 48, no. 2, pp. 481–498, 1988.

[26] J. Geromel, A. Gonçalves, and A. Fioravanti, "Dynamic Output Feedback Control of Discrete-Time Markov Jump Linear Systems through Linear Matrix Inequalities," *SIAM Journal on Control and Optimization*, vol. 48, no. 2, pp. 573–593, 2009.

[27] B. E. Griffiths and K. A. Loparo, "Optimal Control of Jump-Linear Gaussian Systems," *International Journal of Control*, vol. 42, no. 4, pp. 791–819, 1985.

[28] A. N. Vargas, J. B. R. do Val, and E. F. Costa, "Receding Horizon Control of Markov Jump Linear Systems Subject to Noise and Unobserved State Chain," in *Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC 2004)*, vol. 4, Dec 2004, pp. 4381–4386.

[29] A. N. Vargas, D. C. Bortolin, E. F. Costa, and J. B. Val, "Gradient-Based Optimization Techniques for the Design of Static Controllers for Markov Jump Linear Systems with Unobservable Modes," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 28, no. 3, pp. 239–253, 2014.

[30] J. B. do Val and T. Başar, "Receding Horizon Control of Jump Linear Systems and a Macroeconomic Policy Problem," *Journal of Economic Dynamics and Control*, vol. 23, no. 8, pp. 1099 – 1131, 1999.

[31] J. B. do Val, J. C. Geromel, and A. P. Gonçalves, "The H2-Control for Jump Linear Systems: Cluster Observations of the Markov State," *Automatica*, vol. 38, no. 2, pp. 343 – 349, 2002.

[32] M. Jung and F. Rosenthal, "CoCPN-Sim," 2018. [Online]. Available: https://github.com/spp1914-cocpn/cocpn-sim

[33] L. Campo and Y. Bar-Shalom, "Control of Discrete-Time Hybrid Stochastic Systems," *IEEE Trans. Autom. Control*, vol. 37, no. 10, pp. 1522–1527, Oct 1992.