State Estimation in Networked Control Systems with Delayed and Lossy Acknowledgments

Florian Rosenthal(✉), Benjamin Noack, and Uwe D. Hanebeck

Intelligent Sensor-Actuator-Systems Laboratory (ISAS),
Institute for Anthropomatics and Robotics,
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{florian.rosenthal,noack}@kit.edu, uwe.hanebeck@ieee.org

Abstract. In this article, we are concerned with state estimation in Networked Control Systems where both control inputs and measurements are transmitted over networks which are lossy and introduce random transmission delays. We focus on the case where acknowledgment packets transmitted by the actuator upon reception of applicable control inputs are also subject to delays and losses, as opposed to the common notion of TCP-like communication where successful transmissions are acknowledged instantaneously and without losses. As a consequence, the state estimator in the considered setup has only partial and belated knowledge concerning the actually applied control inputs which results in additional uncertainty. We derive an estimator by extending an existing approach for the special case of UDP-like communication which maintains estimates of the applied control inputs that are incorporated into the estimation of the plant state. The presented estimator is compared to the original approach in terms of Monte Carlo simulations where its increased robustness towards imperfect knowledge of the underlying networks is indicated.

Keywords: State estimation · Networked control systems · Delays
Packet losses · Markov jump linear systems · IMM filter

1 Introduction

Networked Control Systems (NCSs), such as the one sketched in Fig. 1, are a special class of control loops where the individual components, i.e., plant/actuator, sensor, and controller, communicate over packet-based and typically general-purpose networks such as WiFi or Ethernet. In comparison to traditional control loops, where dedicated point-to-point connections are utilized, such systems profit from reduced expenditure for installation and maintenance, and from enhanced flexibility and reliability [1]. On the downside, they have to cope with effects and constraints induced by the networks and constraints like random packet losses, delays and limited bandwidth. Since these factors affect the overall system performance and stability, communication and control should not be addressed independently from each other [2–4]. Consequently, several control methods have been proposed in recent years that explicitly consider the underlying network. Among these, the approach of *sequence-based control* has gained much attention [5–9]. Here the idea is to compute control inputs for the next, say N, time steps in addition to the one for the current time instant. By transmitting this *control sequence* in a single data packet which is buffered at the actuator upon reception, the problem of delayed or missing control inputs can be alleviated. Such controllers, often called predictive controllers, are usually based on model predictive control approaches [6,7], or adapted from nominal controllers which disregard the network [10,11]. Also, controllers which directly minimize a quadratic cost function with respect to the control sequence have been proposed [8,9]. Since most of the derived control algorithms either explicitly demand a state estimate or assume a perfectly known state or noise-free plants and measurements, state estimation is generally required in an NCS. This article is an extended version of our previous paper [12], where we developed an estimator based on the minimum mean squared error (MMSE) criterion for a given predictive controller in an NCS scenario. Due to the presence of the networks, the estimator is confronted with the problem that measurements and control inputs can arrive delayed or get lost so that out-of-sequence and burst arrivals are probable. In particular, the resulting uncertainty about the actual applied control inputs poses a major challenge.

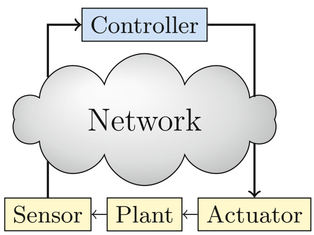

Fig. 1. Schematical overview of a Networked Control System.

Estimators being able to cope with delayed or absent measurements have been proposed, for instance, in [13–15], whereas the problem of estimation subject to missing control inputs has been investigated in [16,17]. Moayedi et al. [18] presented a filter for Networked Control Systems which can handle both delayed and missing control inputs and measurements. Yet, here the probabilities that a particular control input is actually applied are assumed to be time-invariant and have to be known beforehand. As opposed to this, the filter developed in [19], where a setup similar to ours was considered, utilizes an estimate of the currently buffered control sequence for the state estimation. We consider the case where the actuator is able to acknowledge data packets, i.e., control sequences, that were successfully transmitted from the controller, which is in contrast to [19] where such acknowledgments are not provided. Additionally, we take into account that the acknowledgment packets sent from the actuator to the controller also suffer from random delays and losses. Consequently, the setup in [19] can be seen as a special case of the scenario we consider in that acknowledgments are provided by the actuator but always get lost.

Outline. This article is structured as follows. First, in Sect. 2 we give a detailed description of the considered scenario. Then, in Sect. 3 we derive an estimator based on a formal model of the considered problem. The performance of the presented estimator is then assessed in Sect. 4. Finally, Sect. 5 concludes this work.

Notation. Throughout the article, vectors will be indicated by underlined letters ($\underline{x}$) while random vectors will be underlined and in bold ($\underline{\boldsymbol{x}}$). To denote matrices, we will employ boldface capital letters, 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 a matrix is indicated by $\underline{x}^{\mathrm{T}}$ and $\mathbf{A}^{\mathrm{T}}$. Finally, $\delta_{i,j}$ stands for the Kronecker delta, i.e., $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

2 Problem Formulation

Consider an NCS where both plant and sensor are linear and described by

$$\begin{aligned} \underline{\boldsymbol{x}}_{k+1} &= \mathbf{A}_k \underline{\boldsymbol{x}}_k + \mathbf{B}_k \underline{\boldsymbol{u}}_k + \underline{\boldsymbol{w}}_k \,, \\ \underline{\boldsymbol{y}}_k &= \mathbf{C}_k \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k \,, \end{aligned} \tag{1}$$

with $\underline{\boldsymbol{x}}_k \in \mathbb{R}^n$, and $\underline{\boldsymbol{y}}_k \in \mathbb{R}^m$ state and measurement, respectively, and $\underline{\boldsymbol{u}}_k \in \mathbb{R}^l$ the control input provided by a given controller. The zero mean white noise sequences $\underline{\boldsymbol{w}}_k$ and $\underline{\boldsymbol{v}}_k$ are Gaussian and independent of each other with covariance matrices $\mathbf{W}_k$ and $\mathbf{V}_k$. The initial plant state $\underline{\boldsymbol{x}}_0$ is Gaussian with mean $\hat{\underline{x}}_0$ and covariance matrix $\mathbf{\Sigma}_0$ and is independent of $\underline{\boldsymbol{w}}_i$ and $\underline{\boldsymbol{v}}_j$. Furthermore, we assume that all components are synchronized and that the networks assign time stamps to data packets upon transmission.

The actuator is collocated with the plant and connected to the controller via a lossy network (CA-network), which means that each transmitted data packet can experience a (potentially unbounded) delay or get lost. By interpreting losses as infinite delays, we can model the delay of a packet that is sent from the controller to the actuator at time k by the random variable $\boldsymbol{\tau}_k^{CA} \in \mathbb{N}_0$. We additionally assume that the $\boldsymbol{\tau}_k^{CA}$ are independent and identically distributed (i.i.d.) with known probability mass function (PMF) f^{CA}. In order to account for these network-induced effects, the controller does not only transmit the current control input $\underline{u}_k$ at time k but also predicted control inputs for the next N time steps. Consequently, the data packet sent to the actuator consists of the control sequence

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

where $\underline{u}_{k+i|k}$, $i = 0, \ldots, N$ denotes the control input computed at time k and to be applied at time $k + i$. The buffer located at the actuator side employs the so called *past packets rejection logic* [1]: From the set of all received control sequences, the one with the largest time index, that is, only the most recent sequence, is maintained while all others are discarded. The control inputs provided by this sequence are then successively applied at the corresponding time steps until a newer sequence reaches the actuator. However, in case of subsequent packet losses or large delays it may occur that the next control sequence arrives too late so that the control inputs from the buffered sequence are not applicable anymore. In such a case, the default input $\underline{u}_k^{df} = \underline{0}$ is applied.

Remark 1. Applying the default control input $\underline{u}_k^{df} = \underline{0}$ is known as zero-input strategy in literature. Another common alternative is to apply the previous control input, i.e., $\underline{u}_k^{df} = \underline{u}_{k-1}$, which is known as hold-input strategy. While the first one is mathematically more convenient, it has been shown in [20] that even for scalar systems and when only packet dropouts are considered neither strategy is superior.

Each time the stored control sequence is replaced by a more recent one, an acknowledgment (ACK) is sent back by the actuator to indicate a successful transmission of the corresponding sequence. It is important to emphasize that these ACKs are application layer acknowledgments, since not every received data packet is acknowledged by the actuator, but only that one containing the actually utilized control sequence. From the perspective of the underlying CA-network they are just regular payload to be delivered. Consequently, the ACKs are also subject to delays and losses (infinite delays) which are modeled by the i.i.d. random variables $\boldsymbol{\tau}_k^{AC}$ with PMF f^{AC}. Due to this acknowledgment procedure, the controller is able to infer applied control inputs upon the reception of ACKs from the actuator.

Remark 2. Note that, besides the actuator acknowledgment procedure, the transport layer protocol employed by the CA-network might send out dedicated acknowledgment packets upon successful reception of data packets. A common example is the TCP protocol, while UDP is an example for a transport layer protocol that does not acknowledge received packets. TCP implementations enhance the reliability of the communication compared to UDP, since they issue packet retransmissions in case such an acknowledgment packet is delayed or missing. However, it is known that this behavior often poses a severe problem for relatively short transfers [21,22]. Also, data losses are traded for large delays, which is typically not desired in Networked Control Systems [3].

Remark 3. In NCS literature the notion of *UDP-like* networks is used to describe transmissions where received data packets are not acknowledged by the receiver, while the term *TCP-like* refers to idealized transmission schemes that acknowledge successful transmissions instantaneously and without losses [17]. In this regard, the setup we consider could be summarized as UDP-like network with application layer acknowledgments.

Finally, at each time step, a sensor takes a noisy measurement of the state and sends it over another network (SC-network) to an estimator which is attached to the controller. Delays and losses in this network are described by the i.i.d. random variables $\boldsymbol{\tau}_k^{SC}$ with given PMF f^{SC}, so that at each time instant multiple measurements (or none) can arrive at the estimator. Note that in contrast to the CA-network, (i) all delayed packets do provide valuable information about past states and hence should be processed by the estimator and (ii) this network appears deterministic for the estimator since the packet delays are known due to the assigned time stamps. However, as the estimator's buffer is finite, only up to $M \in \mathbb{N}$ measurements can be stored at the same time. As will be discussed in Sect. 3, an appropriate approach to deal with burst and out-of-sequence arrivals of measurements is to maintain a fixed measurement history. The following assumption is thus justified.

Assumption 1. Measurements with a delay larger than $M - 1$ time steps are discarded by the estimator upon reception.

Remark 4. Discarding measurements according to the above assumption always results in a suboptimal estimator. However, for the case that all measurement packets either arrive with a delay of at most $M-1$ time steps or do not arrive at all, that is, for the case that no measurements have to be discarded, the optimal estimator was presented [14].

The complete setup is depicted in Fig. 2. Our goal is to design an estimator which, at each time step k, supplies the controller with an estimate $\hat{\underline{x}}_k^e$ of the plant state based on the MMSE criterion for the given setup. We do so by extending the filter proposed by Fischer et al. [19] for a UDP-like CA-network.

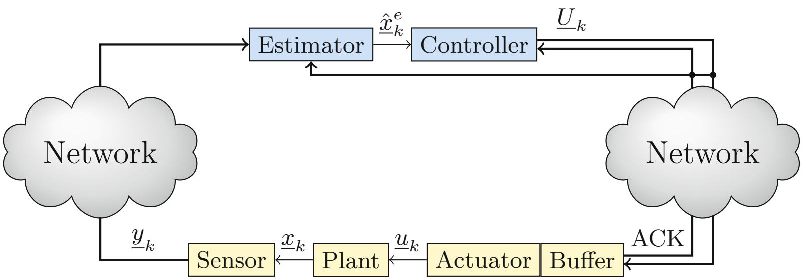

Fig. 2. Considered NCS Setup. A control sequence $\underline{U}_k$ computed by the controller is transmitted to the actuator which buffers the most recent sequence. From this sequence, the control input $\underline{u}_k$ corresponding to the current time step is selected and applied to the plant.

3 Derivation of the Proposed Estimator

The estimator from [19] relies on a stochastic model which jointly describes the CA-network and the actuator as dynamical system. With this model and a suitable state augmentation, the considered NCS is then expressed in terms of a Markov jump linear system (MJLS) [23]. As we build upon this estimator, we provide a condensed summary of the resulting model in Sect. 3.1. A more detailed derivation can be found in [8,19]. We then present the proposed estimator in Sect. 3.2.

3.1 Modeling the NCS as a Markov Jump Linear System

Main ingredients of the network-actuator model are a vector $\underline{\eta}_k$ which encompasses all control inputs from the sequences $\underline{U}_{k-N}, \ldots, \underline{U}_{k-1}$ that are still applicable at time k or later, and an additional discrete, scalar random variable $\boldsymbol{\theta}_k$. Formally, $\underline{\eta}_k$ is given by

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

which is illustrated in Fig. 3 for the case $N = 2$. The dynamics of $\underline{\eta}_k$ can be expressed by

$$\underline{\eta}_{k+1} = \mathbf{F}\underline{\eta}_k + \mathbf{G}\underline{U}_k , \tag{3}$$

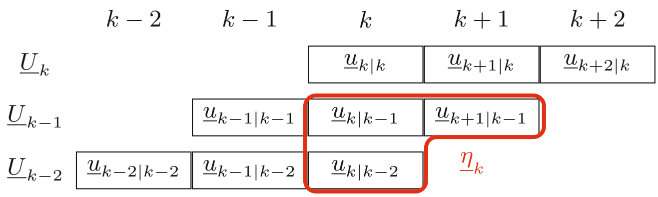

Fig. 3. Visualization of the elements of $\underline{\eta}_k$ (*encircled*) for $N = 2$. Applicable control inputs for the same time step are shown one below another.

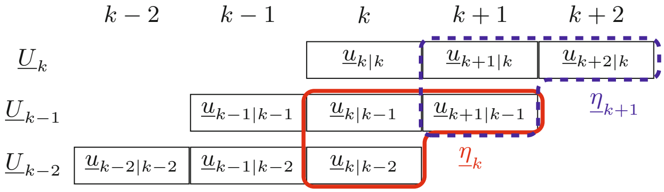

Fig. 4. Illustration of the relationship between $\underline{\eta}_{k+1}$ (*dashed*), $\underline{\eta}_k$ (*solid*), and $\underline{U}_k$ for $N = 2$. Applicable control inputs for the same time step are shown one below another.

with

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(N-1)l} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{(N-2)l} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I}_l & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{\frac{lN(N+1)}{2} \times \frac{lN(N+1)}{2}},$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{Nl} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{\frac{lN(N+1)}{2} \times (N+1)l}.$$

The relationship between $\underline{\eta}_{k+1}$, $\underline{\eta}_k$ and $\underline{U}_k$ is visualized for the case $N = 2$ in Fig. 4. By defining

$$\boldsymbol{\theta}_k = \begin{cases} k - t & \text{if the currently buffered control sequence is } \underline{U}_t \\ N + 1 & \text{else} \end{cases}, \tag{4}$$

where $k - N \leq t \leq k$, we have that $\boldsymbol{\theta}_k \in \{0, \dots, N+1\}$ and that $\boldsymbol{\theta}_k = N + 1$ corresponds to the case when the buffer ran empty and the default input $\underline{u}_k^{df} = \underline{0}$ is applied. In [19] it was shown that $\boldsymbol{\theta}_k$ is a Markov chain with transition matrix $\mathbf{T}$ given by

$$\mathbf{T} = \begin{bmatrix} p_0 & q_0 & 0 & 0 & 0 & 0 & \dots & 0 \\ p_0 & p_1 & q_1 & 0 & 0 & 0 & \dots & 0 \\ p_0 & p_1 & p_2 & q_2 & 0 & 0 & \dots & \vdots \\ p_0 & p_1 & p_2 & p_3 & q_3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ p_0 & p_1 & p_2 & p_3 & \dots & p_{N-1} & q_{N-1} & 0 \\ p_0 & p_1 & p_2 & p_3 & \dots & p_{N-1} & p_N & q_N \\ p_0 & p_1 & p_2 & p_3 & \dots & p_{N-1} & p_N & q_N \end{bmatrix} \in \mathbb{R}^{N+2 \times N+2}, \tag{5}$$

where $p_j = f^{CA}(j)$ denotes the probability that a control sequence arrives with a delay of j time steps, and $q_j = 1 - \sum_{i=0}^{j} p_i$. By means of $\underline{\eta}_k$ and $\boldsymbol{\theta}_k$, we can write the actual applied control input according to

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

with

$$\mathbf{H}_k = \begin{bmatrix} \delta_{\boldsymbol{\theta}_k,1}\mathbf{I}_l \ \mathbf{0} \ \delta_{\boldsymbol{\theta}_k,2}\mathbf{I}_l \ \mathbf{0} \ \ldots \ \delta_{\boldsymbol{\theta}_k,N}\mathbf{I}_l \end{bmatrix} \in \mathbb{R}^{l \times \frac{lN(N+1)}{2}} ,$$
$$\mathbf{J}_k = \begin{bmatrix} \delta_{\boldsymbol{\theta}_k,0}\mathbf{I}_l \ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{l \times (N+1)l} .$$

Finally, defining the augmented state $\underline{\boldsymbol{\xi}}_k = [\underline{\boldsymbol{x}}_k^{\mathrm{T}} \ \underline{\eta}_k^{\mathrm{T}}]^{\mathrm{T}}$ and combining (1), (3) and (6) yields

$$\begin{aligned} \underline{\boldsymbol{\xi}}_{k+1} &= \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k\mathbf{H}_k \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \underline{\boldsymbol{\xi}}_k + \begin{bmatrix} \mathbf{B}_k\mathbf{J}_k \\ \mathbf{G} \end{bmatrix} \underline{U}_k + \begin{bmatrix} \underline{\boldsymbol{w}}_k \\ \underline{0} \end{bmatrix} , \\ \underline{\boldsymbol{y}}_k &= \begin{bmatrix} \mathbf{C}_k & \mathbf{0} \end{bmatrix} \underline{\boldsymbol{\xi}}_k + \begin{bmatrix} \underline{\boldsymbol{v}}_k \\ \underline{0} \end{bmatrix} , \end{aligned} \tag{7}$$

which is a MJLS with parameter $\boldsymbol{\theta}_k$, usually referred to as the *mode* of the system. Recall from (4) that $\boldsymbol{\theta}_k \in \{0, \ldots, N+1\}$, the augmented system hence possesses $N+2$ modes.

3.2 Estimator Design

A major challenge for the design of an estimator for (7) is that in the given setup only a subset $\mathcal{I}_k$ of the mode history $\mathcal{S}_k = \{\theta_0, \theta_1, \ldots, \theta_k\}$ is available to the estimator. More precisely, $\mathcal{I}_k$ contains those modes that could be inferred from ACKs that have been received by the controller up to time k. It is well-known that a time-varying Kalman filter is the optimal MMSE estimator for MJLS if $\mathcal{I}_k = \mathcal{S}_k$, i.e., if the complete mode history is available. However, it is also well-known that for the other extreme case $\mathcal{I}_k = \varnothing$, that is, the case of a completely unknown mode history, the optimal estimator is not only nonlinear but also intractable [24, 25]. This is mainly due to the fact that an exponentially increasing number of hypotheses about the true mode trajectory must be tracked. The computational complexity thus grows exponentially in time. Consequently, a variety of approximations to the optimal solution have been proposed, ranging from LMMSE estimators [26–28] to approaches which at each time instant maintain only a fixed number of hypotheses about the mode history by applying some hypothesis reduction strategy [29]. Among the latter, the *Interacting Multiple Model (IMM) filter* [24] has gained much attraction as it exhibits a good trade-off between estimation quality and complexity. A variant of the IMM filter for a UPD-like NCS scenario, which corresponds to this extreme case $\mathcal{I}_k = \varnothing$, has been introduced in [19]. As already mentioned, we will build upon this estimator and generalize it to the case $\mathcal{I}_k \subset \mathcal{S}_k$.

Another merit of the IMM is that it is widely employed in (multi-)target tracking applications, where, as in Networked Control Systems, delayed and out-of-sequence arrivals of sensor data are common. Hence, a lot of work has been conducted in this community to handle these issues, yielding IMM filters where retrodiction techniques are used to incorporate both arbitrarily delayed measurements and out-of-sequence arrivals [30]. Note that this is in contrast to other estimators for MJLS that have been developed to cope with lost or delayed measurements. For instance, the estimators proposed in [31,32] only consider packet losses, and the MMSE estimator from [33] assumes fixed measurement delays. On the downside, applying retrodiction in our setup necessitates that the system matrix $\mathbf{A}_k$ in (1) is invertible which is not always given. Fischer et al. [19] thus proposed to adapt the approach from [34] and to instead maintain a history of past estimates which is updated upon the reception of a delayed measurement. Besides being simple, this approach has the advantage that it is inherently suited for dealing with burst arrivals of measurements which can be, for instance, processed one by one. Moreover, it can be easily extended to deal with delayed mode observations. We introduce this extension in the following.

In essence, the IMM filter is composed of a bank of Kalman filters, one for each mode, which are individually reinitialized at each time step by mixing all mode-conditioned estimates from the previous time step [29]. For the given system (7) with $N + 2$ different modes the IMM filter thus requires $N + 2$ individual filters, so that the state estimate is maintained as a Gaussian mixture distribution with $N + 2$ components. Each component is weighted according to the estimated mode probability distribution $\underline{\pi}_k$. At the end of each measurement update, the mode distribution is updated according to the mode-conditioned measurement likelihoods. The point estimate $\hat{\underline{x}}_k^e$ for the controller is then simply the mean of the mixture. Suppose now, that at time k, the estimator can infer a mode realization $\boldsymbol{\theta}_t = L$, $t \leq k$, $L \in \{0, \ldots, N\}$, due to a received ACK. The proposed extension exploits that the distribution of $\boldsymbol{\theta}_t$ then reduces to

$$\underline{\pi}_t = \underline{e}_{L+1} \,, \tag{8}$$

where $\underline{e}_{L+1}$ is the $(N + 2)$-dimensional unit vector with one at position $L + 1$ and zero elsewhere.[1] Please note that the mode realization $\boldsymbol{\theta}_t = N + 1$ will never be available to the filter since this indicates that at time t the default input was applied. In such a case, no applicable control sequence would have been received in time by the actuator, and hence no ACK would have been sent back. Note also that, since the measurement equation in (7) is independent of the mode, $\boldsymbol{\theta}_t$ only affects the prediction step at $t + 1$. Combined with Assumption 1 this means that it is reasonable to discard all ACKs with a delay larger than M time steps. Integrating a delayed mode observation at time k finally consists of updating $\underline{\pi}_t$ according to (8) and then recomputing the estimates from $t+1$ to k. This procedure is also well-suited to handle burst arrivals of ACKs, which means that multiple modes can be inferred at once. Starting with the oldest one, they are simply integrated into the recomputation of the state estimates one after another. One cycle of the proposed estimator is summarized in Algorithm 1. For a detailed description of the IMM-specific steps in lines 5, 9, 13, 14 and 16 refer to, for instance, [24,29]. We can clearly get from the algorithm that both computational complexity and required memory increase with the buffer length. In particular, it necessary to store at least the mode observations $\theta_{k-M}, \ldots, \theta_{k-1}$, the control sequences $\underline{U}_{k-M}, \ldots, \underline{U}_{k-1}$, the measurements $\underline{y}_{k-M+1}, \ldots, \underline{y}_k$, and the Gaussian mixture denoting the estimate from time $k - M$. To represent the

[1] The mixture then essentially degenerates into a single Gaussian.

latter, $N+2$ mode-conditioned means and covariance matrices and the estimated mode distribution $\underline{\pi}_{k-M}$ must be stored.

A reference implementation of the algorithm is available on github as part of the CoCPN-Sim simulation framework [35].

Algorithm 1. One Cycle of the Proposed IMM-based Estimator

Input: Estimate from time $k-M$, i.e., Gaussian mixture
Output: Point estimate $\underline{\hat{x}}_k^e$

for $i = M-1$ **to** 0 **do**
  if mode θ_{k-i-1} is available **then**
    Update $\underline{\pi}_{k-i-1}$ according to (8)
  end if
  Reinitialize the mode-conditioned Kalman filters
  Create $\underline{\eta}_{k-i-1}$ according to (2)
  `// Prediction Step`
  for all mode-conditioned filters **do**
    Compute mode-conditioned input by using $\underline{\eta}_{k-i-1}$, $\underline{U}_{k-i-1}$ in (6)
    Perform prediction using the mode-conditioned input
  end for
  `// Measurement Update`
  if measurement $\underline{y}_{k-i}$ is available **then**
    for all mode-conditioned filters **do**
      Perform measurement update using $\underline{y}_{k-i}$
      Evaluate measurement likelihood
    end for
    Update $\underline{\pi}_{k-i}$ using to the individual measurement likelihoods
  end if
end for
Compute mixture mean $\underline{\hat{x}}_k^e$
return $\underline{\hat{x}}_k^e$

Remark 5. We conclude this section with the remark that $\boldsymbol{\theta}_k$ describes an ergodic Markov chain and hence possesses a stationary distribution [9]. As an alternative to the proposed algorithm, the Kalman filter approach from [18], equipped with the stationary distribution, could also be used in the given setup. However, using this stationary distribution is clearly an approximation and in [19] it was shown that this approach is inferior to an IMM-based filter.

4 Evaluation

In this section we assess the performance of the proposed estimator in a scenario similar to the one used in [19], namely controlling an inverted pendulum on a cart which operates in a transient state. We compare the proposed estimator with the original approach from [19] which does not have access to the partial mode

Table 1. Parameters of the inverted pendulum used in the evaluation.

Mass of the cart	0.5 kg
Mass of the pendulum	0.5 kg
Coefficient of friction for the cart	0.1 N s/m
Length to pendulum center of mass	0.3 m
Moment of inertia of the pendulum	0.006 kg m^2
Gravitational acceleration	9.81 m/s^2

history $\mathcal{I}_k$. Our aim is to evaluate to what extent the information advantage of the proposed filter results in improved estimates.

To that end, consider the state of the pendulum given by $\underline{\boldsymbol{x}}_k = \left[\boldsymbol{s}_k \; \dot{\boldsymbol{s}}_k \; \boldsymbol{\phi}_k \; \dot{\boldsymbol{\phi}}_k\right]^{\mathrm{T}}$. Here, $\boldsymbol{s}_k$ denotes the position of the cart (in m) and $\boldsymbol{\phi}_k$ is the deviation (in rad) of the pendulum from the upward equilibrium. Linearizing the nonlinear pendulum dynamics (cf., for instance, [36]) with parameters given in Table 1 around the upward equilibrium and performing a subsequent discretization with sampling time t_A =0.01 s results in the linear model (1) with

$$\mathbf{A}_k = \begin{bmatrix} 1 & 0.0099911 & 0.0003871 & 0.0000013 \\ 0 & 0.9982114 & 0.0774447 & 0.0003872 \\ 0 & -0.0000263 & 1.0025820 & 0.0100086 \\ 0 & -0.0052630 & 0.5165563 & 1.0025820 \end{bmatrix}, \quad \mathbf{B}_k = \begin{bmatrix} 0.0000894 \\ 0.0178855 \\ 0.0002631 \\ 0.0526298 \end{bmatrix},$$

$$\mathbf{C}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

The state variables $\boldsymbol{s}_k$ and $\boldsymbol{\phi}_k$ are thus directly accessible, i.e., the are measured, while $\dot{\boldsymbol{s}}_k$ and $\dot{\boldsymbol{\phi}}_k$ are non-accessible state variables. The covariances of $\underline{\boldsymbol{w}}_k$ and $\underline{\boldsymbol{v}}_k$ are chosen to be $\mathbf{W}_k = 0.01\mathbf{I}_4$ and $\mathbf{V}_k = 0.2\mathbf{I}_2$. As in [19], we employ a nominal predictive state feedback linear quadratic regulator [36] and compute the control sequences $\underline{U}_k$ based on the true state of the plant. State and input weighting matrix for the calculation of the regulator gain $\mathbf{L}$ are given by

$$\mathbf{Q} = \begin{bmatrix} 5000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{R} = 100\,.$$

Overall, we carried out two Monte Carlo simulations with 2000 runs each where each run comprised 250 time steps. In each run, the initial plant state was randomly drawn from a Gaussian distribution with mean and covariance matrix

$$\hat{\underline{x}}_0 = \left[0 \; 0.2 \; 0.2 \; 0\right]^{\mathrm{T}}, \quad \mathbf{\Sigma}_0 = 0.5\mathbf{I}_4\,.$$

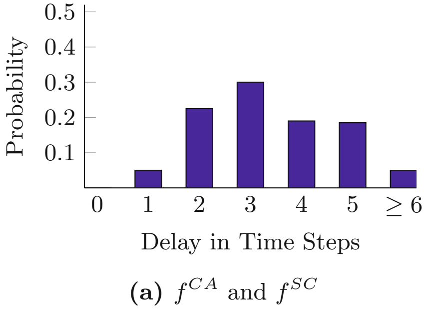

Fig. 5. PMFs of the packet delays. Delays larger than five time steps in the SC-link are treated as packet losses (infinite delay) by the estimators.

The probability mass functions f^{CA}, f^{AC} and f^{SC} used to model the networks employed in the simulations are depicted in Fig. 5. In each simulation run the actual delay of each packet was independently drawn according to the corresponding PMF.

We chose to set the length of the measurement history to $M = 6$. Accordingly, measurements with a delay larger than five time steps and ACKs with a delay larger than six time steps were discarded. For the SC-network described by the PMFshown in Fig. 5a the measurement loss rate was thus 4. For the ACKs sent from the actuator to the controller, we decided to utilize a distribution (cf. Fig. 5b) according to which delays larger than three time steps were very unlikely. As mentioned above, the plant operates in a transient state, i.e., set point changes occur. Thus, in each simulation run, the initial set point of the pendulum was $[2\ 0\ 0\ 0]^{\mathrm{T}}$ which changed to $[-2\ 0\ 0\ 0]^{\mathrm{T}}$ after 100 time steps and then changed back after another 100 time steps. The length of the control sequences computed by the controller was $N + 1 = 7$, resulting in an MJLS with 8 modes. In each run, the mode-conditioned Kalman filters of both estimators were initialized with a Gaussian with mean $\underline{\hat{x}}_0$ and covariance matrix $\mathbf{\Sigma}_0$ and the initial mode distribution was $\underline{\pi}_0 = \underline{e}_8 \in \mathbb{R}^8$. Note that neither estimator had impact on the computation of the control sequences because the control inputs were computed based on the true state.

In the first simulation, the true PMFf^{CA} from Fig. 5a was used in (5) to compute the transition matrix $\mathbf{T}$ of the Markov chain $\boldsymbol{\theta}_k$, while in the second simulation we assumed that the filters were completely unaware of the behavior of the CA-link. Hence, we employed a uniform PMFinstead in (5) to obtain $\mathbf{T}$. This decision was motivated by the time-varying nature of real networks, due to which model mismatches are likely in practical applications.

The simulation results in terms of the root mean squared error (RMSE) are shown in Fig. 6 for the directly accessible states $\boldsymbol{s}_k$ and $\boldsymbol{\phi}_k$ and in Fig. 7 for the non-accessible states $\dot{\boldsymbol{s}}_k$ and $\dot{\boldsymbol{\phi}}_k$. We can immediately see from the results that the performance of both filters does not differ much most of the time. In particular with respect to $\boldsymbol{s}_k$ and $\boldsymbol{\phi}_k$, the information advantage of the proposed filter only

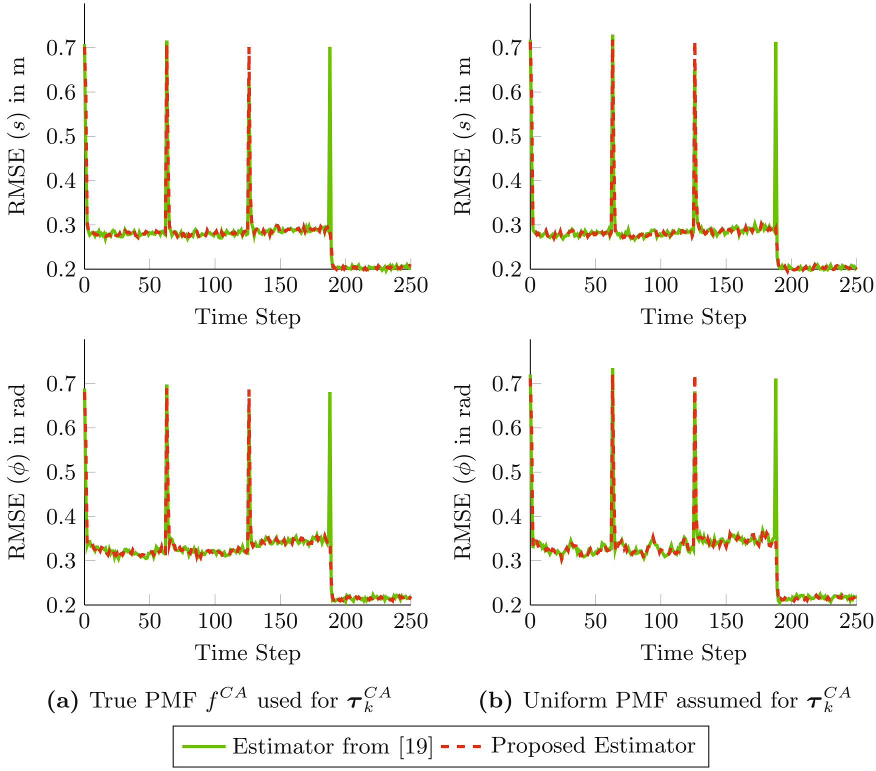

Fig. 6. Results of the proposed estimator and the original approach from [19]: Comparison of the RMSE for the directly accessible states $\boldsymbol{s}_k$ and $\boldsymbol{\phi}_k$.

pays off at a single time step ($k = 190$) in both simulation scenarios. At that time step, the estimation error of the filter from [19] increases drastically while it remains at the same level for the proposed filter. An additional interesting observation is that the estimation quality of both filters is not affected by the model mismatch introduced in the second simulation.

As opposed to this, the RMSE curves of the non-accessible states $\dot{\boldsymbol{s}}_k$ and $\dot{\boldsymbol{\phi}}_k$ exhibit that the overall estimation quality of both filters is corrupted by the wrong PMFassumed for $\boldsymbol{\tau}_k^{CA}$. However, although both filters achieve almost equal performance most of the time, the proposed approach can yield significantly lower estimation errors compared to the approach from [19] even in case of a model mismatch. In particular for the angular velocity component of the state $\dot{\boldsymbol{\phi}}_k$, the original approach is not able to improve its estimates at times where the proposed approach achieves considerable improvements.

Overall, we can conclude that the additional available information in terms of the partial mode history $\mathcal{I}_k$ can result in an enhanced estimation quality and help increase the filter's robustness towards modeling errors with respect to the

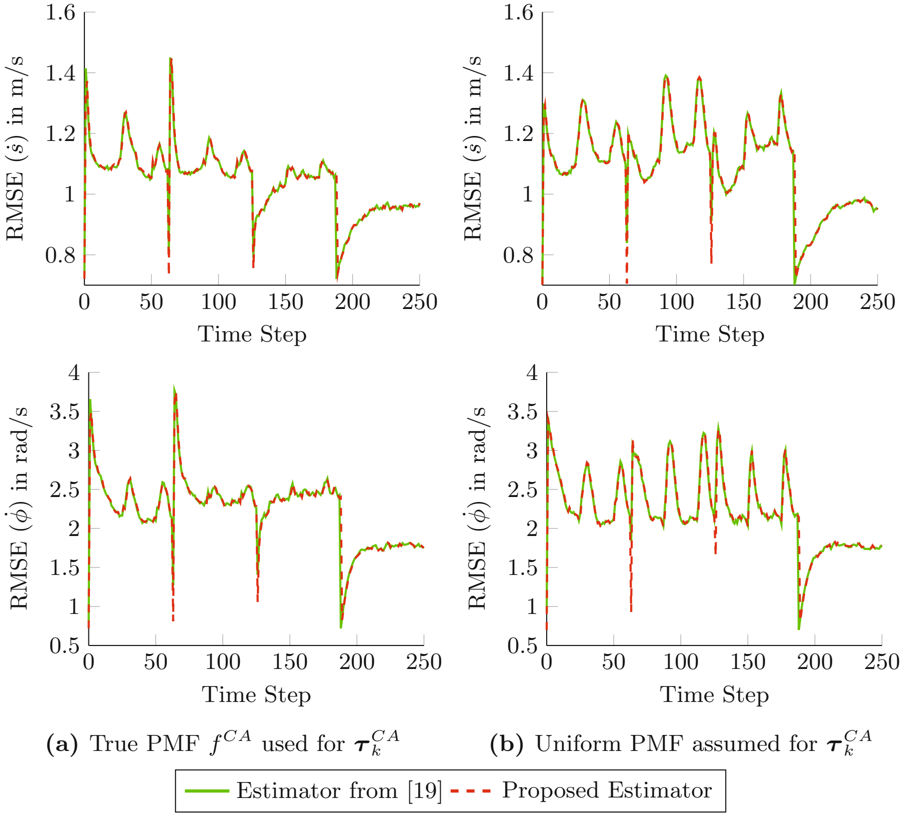

Fig. 7. Results of the proposed estimator and the original approach from [19]: Comparison of the RMSE for the non-accessible states $\dot{\boldsymbol{s}}_k$ and $\dot{\boldsymbol{\phi}}_k$.

nature of the networks. Finally, it is worth to remark that the set point changes do not result in an increased estimation error, in contrast to what was reported in [19].

5 Conclusions

In this article, we considered state estimation in Networked Control Systems. In particular, we focused on networks where acknowledgments are also subject to random delays and losses, which is in contrast to most of the related work in literature where TCP-like or UDP-like communication is considered. We highlighted that, as opposed to TCP-like communication schemes, the estimator has only belated and partial knowledge on the actually applied control inputs. We also showed that scenarios with UDP-like communication are subsumed by the one considered in this article. Based on an existing IMM approach for such networks, we derived a state estimator which is able to incorporate the information on applied past control inputs retroactively into the estimate of the current state.

The evaluation results indicated that the integration of this belated information can be an appropriate means to make the filter more robust towards imperfect knowledge of the network characteristics, in particular for components of the state that are not directly accessible.

Prospective research in this context will address scenarios with time-varying packet delays and losses. This is especially of interest with regards to the increasing number of cyber-physical systems where usually multiple control loops share a network. We will focus on the derivation of an estimator which does not require a priori knowledge of the underlying network delay distributions or relies on assumptions. Here, the joint estimation of the plant state and the properties of $\boldsymbol{\theta}_k$ seems promising. Likewise, approaches from robust estimation, such as $\mathcal{H}_\infty$ estimation, which exploit the structure of the transition matrix, could be employed. Future work should also be concerned with the incorporation of the hold-input strategy into the presented estimator. Finally, it is worth to examine whether an additional, similar state augmentation which explicitly takes the network between sensor and estimator into account, can serve as a starting point for the derivation of an estimator for the given setup.

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

References

1. Zhang, L., Gao, H., Kaynak, O.: Network-induced constraints in networked control systems a survey. IEEE Trans. Ind. Inform. **9**(1), 403–416 (2013). https://doi.org/10.1109/TII.2012.2219540
2. Hespanha, J.P., Naghshtabrizi, P., Xu, Y.: A survey of recent results in networked control systems. Proc. IEEE **95**(1), 138–162 (2007). https://doi.org/10.1109/JPROC.2006.887288
3. Baillieul, J., Antsaklis, P.J.: Control and communication challenges in networked real-time systems. Proc. IEEE **95**(1), 9–28 (2007). https://doi.org/10.1109/JPROC.2006.887290
4. Heemels, W.M.H., Teel, A.R., Van de Wouw, N., Nesic, D.: Networked control systems with communication constraints: tradeoffs between transmission intervals, delays and performance. IEEE Trans. Autom. Control **55**(8), 1781–1796 (2010). https://doi.org/10.1109/TAC.2010.2042352
5. Bemporad, A.: Predictive control of teleoperated constrained systems with unbounded communication delays. In: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 2, pp. 2133–2138. IEEE (1998). https://doi.org/10.1109/CDC.1998.758651
6. Gupta, V., Sinopoli, B., Adlakha, S., Goldsmith, A., Murray, R.: Receding horizon networked control. In: Proceedings of the Allerton Conference on Communication Control, and Computing (2006)

7. Quevedo, D.E., Nesic, D.: Input-to-state stability of packetized predictive control over unreliable networks affected by packet-dropouts. IEEE Trans. Autom. Control **56**(2), 370–375 (2011). https://doi.org/10.1109/TAC.2010.2095950
8. Fischer, J., Hekler, A., Dolgov, M., Hanebeck, U.D.: Optimal sequence-based LQG control over TCP-like networks subject to random transmission delays and packet losses. In: 2013 American Control Conference, Washington, D.C., USA, pp. 1543–1549 (2013). https://doi.org/10.1109/ACC.2013.6580055
9. Dolgov, M., Fischer, J., Hanebeck, U.D.: Infinite-horizon sequence-based networked control without acknowledgments. In: 2015 American Control Conference (ACC), Chicago, Illinois, USA, pp. 402–408 (2015). https://doi.org/10.1109/ACC.2015.7170769
10. Liu, G.P., Xia, Y., Chen, J., Rees, D., Hu, W.: Networked predictive control of systems with random network delays in both forward and feedback channels. IEEE Trans. Ind. Electron. **54**(3), 1282–1297 (2007). https://doi.org/10.1109/TIE.2007.893073
11. Liu, G.: Predictive controller design of networked systems with communication delays and data loss. IEEE Trans. Circuits Syst. II Express Briefs **57**(6), 481–485 (2010). https://doi.org/10.1109/TCSII.2010.2048377
12. Rosenthal, F., Noack, B., Hanebeck, U.D.: State estimation in networked control systems with delayed and lossy acknowledgments. In: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea, pp. 435–441 (2017). https://doi.org/10.1109/MFI.2017.8170359
13. Sinopoli, B., Schenato, L., Franceschetti, M., Poolla, K., Jordan, M.I., Sastry, S.S.: Kalman filtering with intermittent observations. IEEE Trans. Autom. Control **49**(9), 1453–1464 (2004). https://doi.org/10.1109/TAC.2004.834121
14. Schenato, L.: Optimal estimation in networked control systems subject to random delay and packet drop. IEEE Trans. Autom. Control **53**(5), 1311–1317 (2008). https://doi.org/10.1109/TAC.2008.921012
15. Thapliyal, O., Nandiganahalli, J.S., Hwang, I.: Optimal state estimation for LTI systems with imperfect observations. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC), pp. 2795–2800. IEEE (2017). https://doi.org/10.1109/CDC.2017.8264065
16. Epstein, M., Shi, L., Murray, R.M.: An estimation algorithm for a class of networked control systems using UDP-like communication schemes. In: Proceedings of the 45th IEEE Conference on Decision and Control, pp. 5597–5603. IEEE (2006). https://doi.org/10.1109/CDC.2006.377481
17. Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., Sastry, S.S.: Foundations of control and estimation over lossy networks. Proc. IEEE **95**(1), 163–187 (2007). https://doi.org/10.1109/JPROC.2006.887306
18. Moayedi, M., Foo, Y.K., Soh, Y.C.: Filtering for networked control systems with single/multiple measurement packets subject to multiple-step measurement delays and multiple packet dropouts. Int. J. Syst. Sci. **42**(3), 335–348 (2011). https://doi.org/10.1080/00207720903513335
19. Fischer, J., Hekler, A., Hanebeck, U.D.: State estimation in networked control systems. In: 2012 15th International Conference on Information Fusion, Singapore, pp. 1947–1954 (2012)
20. Schenato, L.: To zero or to hold control inputs with lossy links? IEEE Trans. Autom. Control **54**(5), 1093–1099 (2009). https://doi.org/10.1109/TAC.2008.2010999

21. Kim, D., Yoo, H.: TCP performance improvement considering ACK loss in ad hoc networks. J. Commun. Netw. **10**(1), 98–107 (2008). https://doi.org/10.1109/JCN.2008.6388333
22. Cardwell, N., Savage, S., Anderson, T.: Modeling TCP latency. In: Proceedings of IEEE INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1742–1751. IEEE (2000). https://doi.org/10.1109/INFCOM.2000.832574
23. Costa, O.L.V., Fragoso, M.D., Marques, R.P.: Discrete-Time Markov Jump Linear Systems. Springer Science & Business Media, New York (2006)
24. Blom, H., Bar-Shalom, Y.: The interacting multiple model algorithm for systems with markovian switching coefficients. IEEE Trans. Autom. Control **33**(8), 780–783 (1988). https://doi.org/10.1109/9.1299
25. Ackerson, G., Fu, K.: On state estimation in switching environments. IEEE Trans. Autom. Control **15**(1), 10–17 (1970). https://doi.org/10.1109/TAC.1970.1099359
26. Costa, O.: Linear minimum mean square error estimation for discrete-time markovian jump linear systems. IEEE Trans. Autom. Control **39**(8), 1685–1689 (1994). https://doi.org/10.1109/9.310052
27. Costa, O.L.V., Guerra, S.: Stationary filter for linear minimum mean square error estimator of discrete-time markovian jump systems. IEEE Trans. Autom. Control **47**(8), 1351–1356 (2002). https://doi.org/10.1109/TAC.2002.800745
28. Terra, M.H., Ishihara, J.Y., Jesus, G.: Information filtering and array algorithms for discrete-time markovian jump linear systems. IEEE Trans. Autom. Control **54**(1), 158–162 (2009). https://doi.org/10.1109/TAC.2008.2007181
29. Li, X.R., Jilkov, V.P.: Survey of maneuvering target tracking. Part V: multiple-model methods. IEEE Trans. Aerosp. Electron. Syst. **41**(4), 1255–1321 (2005). https://doi.org/10.1109/TAES.2005.1561886
30. Bar-Shalom, Y., Chen, H.: IMM estimator with out-of-sequence measurements. IEEE Trans. Aerosp. Electron. Syst. **41**(1), 90–98 (2005). https://doi.org/10.1109/TAES.2005.1413749
31. Fioravanti, A.R., Gonçalves, A.P., Geromel, J.C.: Filtering of discrete-time markov jump linear systems with cluster observation: an approach to Gilbert-Elliot's network channel. In: Control Conference (ECC), 2009 European, pp. 2283–2288. IEEE (2009)
32. Gonçalves, A.P., Fioravanti, A.R., Geromel, J.C.: Markov jump linear systems and filtering through network transmitted measurements. Signal Process. **90**(10), 2842–2850 (2010). https://doi.org/10.1016/j.sigpro.2010.04.007
33. Matei, I., Baras, J.S.: Optimal state estimation for discrete-time markovian jump linear systems, in the presence of delayed output observations. IEEE Trans. Autom. Control **56**(9), 2235–2240 (2011). https://doi.org/10.1109/TAC.2011.2160027
34. Larsen, T.D., Andersen, N.A., Ravn, O., Poulsen, N.K.: Incorporation of time delayed measurements in a discrete-time kalman filter. In: Proceedings of the 37th IEEE Conference on Decision and Control, vol. 4, pp. 3972–3977. IEEE (1998). https://doi.org/10.1109/CDC.1998.761918
35. Jung, M., Rosenthal, F.: CoCPN-Sim (2018). https://github.com/spp1914-cocpn/cocpn-sim
36. Kwakernaak, H., Sivan, R.: Linear Optimal Control Systems, vol. 1. Wiley-Interscience, New York (1972)