State Estimation in Networked Control Systems

Jörg Fischer, Achim Hekler, and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS),
Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT),
Karlsruhe, Germany,
Email: joerg.fischer@kit.edu, achim.hekler@kit.edu, uwe.hanebeck@ieee.org.

***Abstract*—We consider the problem of state estimation in a Networked Control System, where measurements and control inputs are transmitted via a communication network. The network is subject to time-varying delays and stochastic data losses and does not provide acknowledgments of successfully transmitted data packets. A challenge that arises in this configuration is that the estimator has only uncertain information about the actually applied control inputs. In this paper, we derive a multiple-model based estimator that uses the state measurements to estimate the applied control inputs so that the overall state estimation is improved. The efficiency of the proposed approach is demonstrated by means of Monte-Carlo-Simulation runs with an inverted pendulum on a cart.**

I. INTRODUCTION

The increasing spread and availability of data networks like the Internet accentuates a trend to employ non-industrial communication networks, e.g., Ethernet, as low level communication connection within a control-loop. In contrast to traditional point-to-point connections, these so-called *Networked Control Systems* (NCS) not only offer many advantages such as more flexibility in system design and lower costs, but also come with challenges. In particular, the communication network may introduce time-variant transmission delays and stochastic data losses that can critically effect the performance and stability of the overall system [1], [2].

To cope with these network-induced effects, a control method has been discussed in the literature, e.g., in [3], [4], [5], [6], [7], [8], [9], that relies on a predictive approach. Depending on the author, the method is referred to as Networked Predictive Control, Model Predictive Control over Networks, Receding Horizon Networked Control or Packet-Based Control. The main idea of this approach is that the controller sends data packets over the network to the actuator that contain not only the current control input, but also predicted control inputs for future time instants. Correspondingly, the actuator is equipped with a buffer to store the predicted future control inputs so that they could be applied in case a future data packet will be delayed or lost. However, the derived controllers require that the current state of the plant is known perfectly [3], [4], [5], [7] or that there is no process and measurement noise [6], [8]. In practice, these assumptions usually do not hold and, in fact, a state estimator is needed.

In this paper, we are therefore interested in the problem of state estimation of a partially observable plant in the Networked Predictive Control setup as illustrated in Fig. 1. While we assume that there is always a network connection

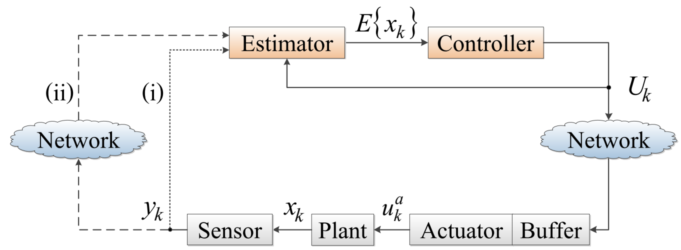


Fig. 1. Networked Predictive Control System: The controller generates control input sequences U_k that are transmitted over the network to the actuator. The actuator holds the most recent control input sequence in a buffer and applies the time-corresponding entry $\boldsymbol{u}_k^a$ of the sequence to the plant. The state $\boldsymbol{x}_k$ of the plant is measured and the measurements $\boldsymbol{y}_k$ are either directly transmitted to the estimator if configuration (**i**) is considered or via a network as illustrated by configuration (**ii**). Based on the received measurements and past control input sequences, the estimator calculates the state estimates $E\{\boldsymbol{x}_k\}$.

between controller and actuator (CA-link), we consider two configurations for the connection between sensor and estimator (SE-link): in configuration (**i**), the sensor data is transmitted directly to the estimator and in configuration (**ii**), the data is sent via a network connection. It turns out that the results derived for configuration (**i**) can easily be extended to configuration (**ii**) and, hence, the main part of the paper will treat the former case. It is important to emphasize that we are not interested to derive a controller or investigate the relationship of control and estimation in this paper. Instead, we assume that a controller is given and discuss the question, how an estimator based on the minimum mean square error criterion (MMSE) can be derived for the described setup.

A. Related Work

The major problem in the derivation of the estimator is that the estimator only has uncertain knowledge about the time when a control packet arrives at the actuator. Therefore, it is uncertain which control inputs have actually been applied to the plant. Most results given in the literature do not consider this kind of uncertainty and only apply to the case with a network present between sensor and estimator, e.g., [10], [11], [12], [13], [14], [15]. The derived results cannot be applied directly if there is also a network present in the CA-link. However, the approaches in [13], [14], [16], that deal with the problem of delayed and out-of-sequence measurements, can be used to extend an estimator derived for system configuration (**i**) so that it is also suitable for configuration (**ii**).

The case when there is only a network between controller and actuator and there are only losses but no delays, was investigated in [17]. The authors suggested choosing the control inputs also under consideration how good the estimator can determine whether a control input has been applied to the plant. The structure of the derived estimator is similar to that of input estimation approaches [18] and unknown input observers [19] which estimate the state of the plant without assuming any prior information about the applied control inputs. However, the estimators in [17], [18], [19] only exist under rather strong rank conditions on the system matrices that, e.g., require that the plant is minimum-phase. The case when there is a network in the CA-link as well as in the SE-link is considered by [9], [20], [21]. In [9], a controller and estimator scheme is derived for a Networked Predictive Control setup where, however, the authors assume that data transmissions are acknowledged immediately by the network. This implies that the acknowledgments sent via the network do not suffer time delays what contradicts the idea of time-variant transmission delays. The estimator stated in [20] considers an unreliable stochastic network but the results only apply if there are no packet delays and only packet losses. In [21], an optimal minimum mean square estimator is presented. The derived filter incorporates delayed and out-of-sequence measurements via a state augmentation approach and considers that the control inputs can be sent via a network. A problem is, however, that the probability distribution of the possibly applied control inputs has to be known a priori. Indeed, the probability distribution is not constant but depends on the measurement history and, therefore, has to be estimated online. This is also the key concept of our approach.

B. Key Idea

In contrast to the former work, we derive an estimator that reduces the uncertainty about the applied control inputs by estimating which control input sequence is buffered in the actuator. In this way, the overall estimation of the state is improved. To the best of our knowledge, an estimator that considers these uncertainties arising in Networked Predictive Control Systems has not been proposed in the literature. It should be mentioned that the concept cannot only be used in the specialized setup of Networked Predictive Control but also in systems, where the inputs are subject to time-varying delays in general. The Networked Predictive Control setup, however, where a predictive controller is used along with an actuator buffer as depicted in Fig. 1, is especially suited for this approach since the buffering mechanism causes a high correlation of the applied control inputs.

C. Outline

The rest of the paper is organized as follows. After an introduction of the used notation, we will give a short description of the Networked Predictive Control scheme and formulate the considered problem in Sec. II. Based on this, the estimator is derived in Sec. III, and in Sec. IV a simulation is presented that shows the efficiency of the estimator. Finally, we give a brief conclusion.

D. Notation

Throughout the paper, random variables $\boldsymbol{a}$ are written in bold face letters, whereas deterministic quantities a are in normal lettering. Furthermore, the notation $\boldsymbol{a} \sim f(a)$ means that the random variable $\boldsymbol{a}$ is characterized by its probability density function $f(a)$. Matrices are always referred to with bold face capital letters, e.g., $\mathbf{A}$. The notation a_k refers to the quantity a at time step k. Furthermore, $a_{k|t}$ denotes the quantity a at time step k based on information up to time t. The terms $\mathbf{0}$ refers to a matrix with all entries equal to 0. Finally, $\mathbf{I}_n$ denotes the identity matrix of dimension $n \times n$.

II. SYSTEM SETUP & PROBLEM FORMULATION

Since we are interested in the problem of state estimation in the context of Networked Predictive Control, we first briefly review the general concept of this control method as, e.g., used in [3], [4], [5], [6], [7], [8], and describe the system setup. Based on this, the problem is formulated in Sec. II-B.

A. System Setup

Consider the Networked Predictive Control System in Fig. 1, where a plant is controlled over a packet-based digital network that induces time-delays and packet losses in the control and measurement channel. The predictive controller not only generates single control inputs for the current control cycle, but rather calculates control inputs for the future N time steps (with $N \in \mathbb{N}$) based on a prediction of the future behavior of the plant. The actual and predicted future control inputs are lumped together in a sequence and sent as one time stamped data packet over the network to the actuator. The actuator has a buffer in which, among all received packets, that packet is hold that carries the most recent information, i.e., the packet with the highest time stamp. If a packet arrives out of order, indicated by a lower time stamp than the currently buffered packet, the new packet is neglected. In every time step, the actuator determines the control input of the buffered sequence that corresponds to the current time step and applies it to the plant.

For the rest of the paper, we use U_k to denote the input sequence generated by the controller at time step k. Entries of that packet are denoted by $u_{k+m|k}$ with $m \in \{0, 1, ..., N\}$, where the first part of the index (here: $k + m$) gives the time step for which the control input is intended to be applied to the plant. The second part of the index (here: k) specifies the time step, when the control input was generated. This gives for a packet of length $N + 1$ generated in time step k

$$U_k = \{u_{k|k}, u_{k+1|k}, \ldots, u_{k+N|k}\} \ . \tag{1}$$

Since we do not assume that the time delays are bounded, it may happen that the buffer runs empty. In the literature, two simple methods exist to cope with this situation: The so-called zero-input strategy and the hold-input strategy. In both cases the actuator applies a default control input u^d, whereas in the first strategy u^d is set to zero while in the latter u^d is equal to the previously applied control input.

B. Problem Formulation

Consider the partially-observable discrete-time linear time-invariant plant

$$\boldsymbol{x}_{k+1} = \mathbf{A}\boldsymbol{x}_k + \mathbf{B}\boldsymbol{u}_k^a + \boldsymbol{w}_k \ , \tag{2}$$

$$\boldsymbol{y}_k = \mathbf{C}\boldsymbol{x}_k + \boldsymbol{v}_k \ , \tag{3}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ denotes the system state at time step k, $\boldsymbol{u}_k^a \in \mathbb{R}^m$ the control input actually applied by the actuator and $\boldsymbol{y}_k \in \mathbb{R}^q$ the measured output. The matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are known and of appropriate dimension. The terms $\boldsymbol{w}_k \in \mathbb{R}^n$ and $\boldsymbol{v}_k \in \mathbb{R}^q$ represent stationary zero-mean discrete-time white noise processes with covariance matrices that are independent of each other and of networked-induced effects.

The data transmission between controller and actuator as well as between sensor and estimator (if configuration (**ii**) is considered) are subject to time-varying and possibly unbounded[1] delays, modeled as discrete random processes[2] $\boldsymbol{\tau}_k^{CA} \in \mathbb{N}$ and $\boldsymbol{\tau}_k^{SE} \in \mathbb{N}$, respectively. The realizations of these processes describe how many time steps a packet will be delayed if it is sent at time step k. It is assumed that $\boldsymbol{\tau}_k^{CA} \sim f(\tau^{CA})$ and $\boldsymbol{\tau}_k^{SE} \sim f(\tau^{CA})$ are white stationary processes, independent of each other and $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$, and that their probability density functions are known. Additionally, it is assumed that the components of the control loop are time-triggered, time-synchronized and have identical cycle times. Furthermore, the employed network is capable of transmitting large time stamped data packets and does not provide acknowledgements whether a packet was successfully transmitted. The latter assumption corresponds to the use of a so-called UDP-like protocol as defined in [20].

Remark 1 *We note that in the literature also so-called TCP-like protocols are considered, where it is assumed that data transmissions are acknowledged immediately without time delays [9]. While networks with TCP protocol resent data packets that got lost, lost packets are not resent in networks with UDP protocol. Therefore, unacknowledged protocols like UDP usually have smaller time-delays but higher loss rates than TCP-like protocols. Which kind of protocol should be used in practice depends on the specific application. It should be noted, however, that even when transmissions are acknowledged, the acknowledgments also suffer from possible time-delays or can get lost. Hence, in a realistic TCP network the applied control inputs would also be subject to uncertainties.*

Finally, we suppose the predictive controller is given and generates to every time step k a control input sequence U_k that consists of $N+1$ control inputs. To process the control sequences, the actuator uses the buffering logic described in Sec. II-A. In the next section we derive a state estimator that uses a finite history of control input sequences and measurements to calculate (suboptimal) estimates based on a minimum mean square error criterion.

[1]By allowing the time-delays to be unbounded, packet losses can be incorporated into the description of the random delay processes since the loss of a packet corresponds to an infinite time-delay

[2]The delay processes are defined as departure processes and, therefore, include the case of burst arrivals.

III. Estimator Design

In this section, the estimator for the Networked Predictive Control System will be derived. To that end, we first discuss the difference between the uncertainties introduced by the CA-link in respect to the SE-link. Then, in Sec. III-A, a stochastic model of the actuator and network is derived and, based on this, the estimator design is discussed in Sec. III-B. Finally, in Sec. III-C, we consider the special case when the time-varying distribution of the uncertain control inputs is approximated by its time-invariant steady state distribution.

We begin with the general observation that the kind of uncertainty introduced by the CA-network is of fundamentally different nature than the uncertainty caused by the SE-network. Packet delays in the CA-link cannot be known by the estimator and lead to an uncertainty about the applied control inputs. Due to the prediction and buffering scheme, the uncertainty is correlated in time and can be reduced by estimating the actually buffered control sequence. For this purpose, we derive a stochastic model of the CA-link that describes this uncertainty in the following section.

In contrast to the CA-link, delays occurring in the SE-link are not related to the state directly and, even more importantly, are perfectly known after a measurement has been received by the estimator. This results from the fact that all packets are time-stamped. From an estimator's point of view, the SE-link can be seen as deterministic and, therefore, deterministic techniques are sufficient.[3] The main problems regarding the SE-link are how to deal with delayed measurements, out-of-sequence measurements and so-called burst arrivals. These cases are discussed in the second part of Sec. III-B.

A. Stochastic Model of Network and Actuator

In order to model network and actuator in state space form, we introduce the vector

$$\eta_k = \begin{pmatrix} [u_{k|k-1}^T \quad u_{k+1|k-1}^T \quad \cdots \quad u_{k+N-1|k-1}^T]^T \\ [u_{k|k-2}^T \quad u_{k+1|k-2}^T \quad \cdots \quad u_{k+N-2|k-2}^T]^T \\ \vdots \\ [u_{k|k-N+1}^T \quad u_{k+1|k-N+1}^T]^T \\ u_{k|k-N} \end{pmatrix} \in \mathbb{R}^d \tag{4}$$

with $d = nN(N+1)/2$ that contains all control inputs of the already sent control input sequences $U_{k-1}, U_{k-2}, \ldots, U_{k-N}$ that could be applied in time step k or later. This is illustrated in Fig. 2, where the relevant control input sequences are shown for the case of $N = 3$. To describe which one of the possible control inputs (given by the first n entries in U_k, the N entries $u_{k|*}$ of η_k and the default input u^d) is actually applied by

[3]For controller design and stability analysis, a stochastic model is needed, however.

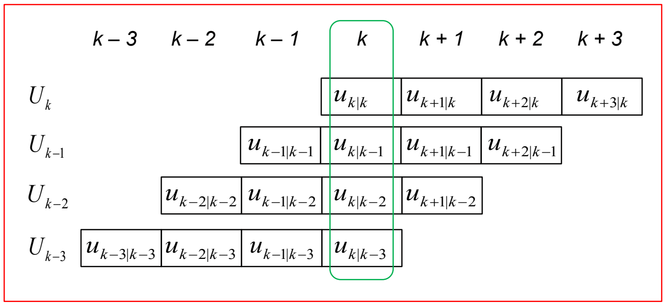


Fig. 2. Representation of control input sequences $U_{k-3}, \ldots, U_k$, whereas control inputs corresponding to the same time step are vertically aligned. The control inputs that could be applied by the actuator at time step k are marked by the rounded rectangle.

the actuator, we introduce a Markov chain with finite state space $\{0, 1, 2, ..., N+1\}$. The state of this Markov chain, θ_k, describes the *age* of the control input sequence buffered in the actuator, i.e., how many time steps ago the buffered sequence was generated by the controller. For example, when there is no delay or loss at time step k then the packet U_k will be taken into the actuator buffer and $\theta_k = 0$. Consequently, if the next packet is delayed, then U_k would still be in the buffer, but θ_k has increased by one. It holds for the transition matrix $\mathbf{P}$ of θ_k

$$\mathbf{P} = \begin{bmatrix} p_{0,0} & p_{0,1} & 0 & 0 & \cdots & 0 \\ p_{1,0} & p_{1,1} & p_{1,2} & 0 & \cdots & 0 \\ p_{2,0} & p_{2,1} & p_{2,2} & p_{2,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & p_{(r-1),r} \\ p_{r,0} & p_{r,1} & p_{r,2} & p_{r,3} & \cdots & p_{r,r} \end{bmatrix} \tag{5}$$

$$\text{with} \quad p_{i,j} = \text{Prob}\left[\theta_{k+1} = j | \theta_k = i\right], \quad r = N+1 \ .$$

The elements of $\mathbf{P}$ in the upper right triangle are zero as θ_k can only increase by one per time step. With the probability distribution $f(\tau^{CA})$ of the time delays given, we can derive the probability q_i of the event that a packet received by the actuator was generated $i \in \mathbb{N}$ time steps ago. Based on q_i the remaining non-zero entries $p_{i,j}$ of $\mathbf{P}$ can be calculated by

$$\begin{aligned} p_{i-1,i} &= 1 - \sum_{j=0}^{i-1} q_i \quad \text{for } i \in [1, 2, \cdots, N+1] \ , \\ p_{i,j} &= q_j \quad \text{for } j \le i \text{ , where } i, j \in [0, 1, \cdots, N+1] \ . \end{aligned}$$

To derive these results, we used that $\boldsymbol{\tau}_k^{CA}$ is a white random process and that a buffered sequence is overwritten if a packet with a smaller time delay than the one of the buffered sequence is received. For instance, a buffered sequence that was generated two time-steps ago ($\theta = 2$) will be replaced in the next time step by a sequence that was generated one time step ago with the probability $p_{21} = (1 - q_0)\frac{q_1}{1-q_0} = q_1$ since the packet generated in the next time step must not be received (Prob = $(1 - q_0)$) and, simultaneously, the packet with delay one has to be received (Prob = $\frac{q_1}{1-q_0}$). Based on η_k and θ_k, a combined state space model of network and actuator can be formulated as

$$\begin{aligned} \eta_{k+1} &= \mathbf{F}\eta_k + \mathbf{G}U_k \ , & (6) \\ \boldsymbol{u}_k^a &= \mathbf{H}_{\theta_k}\eta_k + \mathbf{J}_{\theta_k}U_k \ , & (7) \end{aligned}$$

with

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

$$\mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{nN} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} , \quad \mathbf{J}_{\theta_k} = \begin{bmatrix} \delta_{(\theta_k,0)}\,\mathbf{I}_n & \mathbf{0} \end{bmatrix} ,$$

$$\mathbf{H}_{\theta_k} = \begin{bmatrix} \delta_{(\theta_k,1)}\,\mathbf{I}_n & \mathbf{0} & \delta_{(\theta_k,2)}\,\mathbf{I}_n & \mathbf{0} & \cdots & \delta_{(\theta_k,N)}\,\mathbf{I}_n \end{bmatrix} ,$$

where $\mathbf{0}$ represents matrices of appropriate dimensions with all elements equal to zero and $\delta_{(\theta_k,i)}$ is the Kronecker delta, which is defined as

$$\delta_{(\theta_k,i)} = \begin{cases} 1 & \text{if} \quad \theta_k = i \\ 0 & \text{if} \quad \theta_k \neq i \end{cases} .$$

The combined model in (6) and (7) of the network and actuator can be interpreted as an inhomogeneous Markovian jump linear system (MJLS) with unobserved mode θ_k [22].

B. Estimator Design

Introducing the augmented state $\boldsymbol{\xi}_k = \begin{bmatrix} \boldsymbol{x}_k^T & \eta_k^T \end{bmatrix}^T$ and combining (3), (6), and (7), it holds that

$$\boldsymbol{\xi}_{k+1} = \begin{bmatrix} \mathbf{A} & \mathbf{B}\cdot\mathbf{H}_{\theta(k)} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \boldsymbol{\xi}_k + \begin{bmatrix} \mathbf{B}\cdot\mathbf{J}_{\theta(k)} \\ \mathbf{G} \end{bmatrix} U_k + \begin{pmatrix} \boldsymbol{w}_k \\ \mathbf{0} \end{pmatrix} . \tag{8}$$

First, we consider configuration (**i**), i.e., when there are no delays in the SE-link. For this case, it is well known that the MMSE estimator of the MJLS in (8) is obtained from a bank of Kalman filters, whose memory requirement and computational load increases exponentially with time [23]. Since this is not an applicable solution, suboptimal filters have been proposed in the literature, for instance, [24], [25], [26], [27], whereby two main approximation techniques for hypotheses reduction can be distinguished: merging and pruning [28].

Merging techniques reduce the increasing complexity by combining state estimations that originate from similar hypothesis about the unknown mode θ_k. Commonly used algorithms of this category are the *Generalized Pseudo Gaussian* (GPB) algorithm, originated in [26], and the *Interacting Multiple Model* (IMM) algorithm [27]. Pruning methods on the other hand make hard decisions about the mode history and only keep state estimates that are related to the most likely hypotheses about θ_k. This category includes the *B-best* approach [24] and methods based on the *Viterbi-algorithm* [29].

The question which one of these filters should be used in the considered Networked Predictive Control setup cannot be answered in general and the answer depends on the specific application. However, if the model of the system can differ significantly from the real system, a merging strategy should be preferred [28]. In practical applications, where a mismatch between model and true system is nearly inevitable, it is therefore preferable to use a merging strategy. Thus, we use an IMM-based approach, which offers an efficient trade off between performance and complexity compared to the GPB algorithm [28]. A detailed description of the IMM-algorithm can be found in [28] and the application to (8) and (3) is straightforward and will not be restated here.

Now, we consider system configuration (**ii**), i.e., the case, where packet delays also occur in the SE-link. As mentioned in the beginning of this section, the SE-link does not influence the state directly (it only influences the estimation of the state). Furthermore, the delay of a measurement is known as soon as the measurement is received by the estimator. Therefore, the problems that arise are questions of how the former derived estimator can be extended to deal with lost measurements, out-of-sequence measurements and burst arrivals. In case no measurement has been received in the current time step, this can easily be performed by only performing the prediction step and skipping the filter step. To incorporate delayed and out-of-sequence measurements into the estimation scheme we can use the results of [14], [16]. In [14], the authors proposed to store the measurements and re-compute the filter algorithm over a finite measurement history if a delayed measurement has been received. In [16] the authors derive an approximate solution based on the idea of retrodiction, which avoids storing the measurement history. This approach has been extended in [15] for the IMM algorithm and can directly be applied. The approach in [14] is exact and easy to implement, but if long measurement delays should be considered, the approach is not applicable since the computational load depends on the time delay of the measurement. To consider long time delays, the approximate solution of [15] should be used. Finally, the proposed estimator for configuration (**i**) can be extended to deal with the burst arrival of measurements by the approach formulated in [21]. The idea is to loop the filter algorithm and process one measurement after the other. This can be combined with the re-computation approach for delayed measurements. Then, the measurement history can be updated by all received measurements at once and the re-computation of the measurement history must only be executed once.

C. Steady State Approximation

In this subsection, we briefly discuss the case if the time-varying probability distribution over the possibly buffered control input sequences, indicated by θ_k, is approximated by its time-invariant stationary probability solution π_∞. With transition matrix $\mathbf{P}$ as given in (5), π_∞ can be computed by the equilibrium equation

$$\pi_\infty = \mathbf{P}^T \pi_\infty \ , \tag{9}$$

which always has a unique solution according to Markov chain theory. This approximation decouples the state estimation problem from the delays actually occuring in the CA-link. Therefore, by using π_∞ instead of the real time-variant distribution, the IMM-based estimator derived in the previous section reduces to a simple Kalman filter with the possible inputs weighted according to their probability given by π_∞. This filter corresponds to the Kalman filter described in [21] when π_∞ is used as a priori information about the probabilities of the applied control inputs. In concrete terms, ρ_j in [21] are the entries of π_∞. In the following section, we will compare this Kalman filter approach with the proposed IMM-based estimator.

IV. SIMULATIONS

In this section, we compare the performance of the proposed IMM-based estimator with the Kalman filter approach described in [21]. In the latter case, the probability that a certain control input is applied is approximated by its stationary probability distribution as described in Sec. III-C. As example system we choose an inverted pendulum that is controlled over a network.

Mass of the cart	0.5 kg
Mass of the pendulum	0.5 kg
Friction of the cart	0.1 N/m/s
Length to pendulum center of mass	0.3 m
Inertia of the pendulum	0.006 kg·m^2

TABLE I
PARAMETERS OF THE INVERTED PENDULUM USED IN THE SIMULATIONS.

A. Simulation Setup

We consider the system configuration (**ii**) of Fig. 1, where a network is present in the CA- and SE-link. The plant is an inverted pendulum on a cart as, e.g., described in [30], with state

$$\boldsymbol{x}_k = \left[\boldsymbol{s}_k, \frac{\Delta \boldsymbol{s}_k}{\Delta t}, \boldsymbol{\phi}_k, \frac{\Delta \boldsymbol{\phi}_k}{\Delta t} \right]^T ,$$

where s_k is the position of the cart and ϕ_k the angle of the pendulum. With the parameters of the pendulum chosen according to Table I and a sample time of 0.01 s, the system matrices of the time-discrete system are given by

$$\mathbf{A} = \begin{bmatrix} 1.0000 & 0.0100 & 0.0004 & 0.0000 \\ 0 & 0.9982 & 0.0774 & 0.0004 \\ 0 & -0.0000 & 1.0026 & 0.0100 \\ 0 & -0.0053 & 0.5160 & 1.0026 \end{bmatrix} ,$$

$$\mathbf{B} = \begin{bmatrix} 0.0001 \\ 0.0179 \\ 0.0003 \\ 0.0526 \end{bmatrix} , \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} .$$

For this system, a networked predictive controller is designed according to [6] based on an underlying *Linear*

Quadratic Regulator (LQR) [31]. The weighting matrices of the LQR are chosen with

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

so that the resulting state feedback matrix of the LQR is given by

$$\mathbf{L} = [-6.46, -5.54, 32.28, 5.17] \ .$$

The length of the control input sequence is set to 12. In the simulation, the controller works on the real state of the plant so that the estimators have no effect on the generation of the control inputs.

At every time step k, we add a process noise $\boldsymbol{w}_k$ to $\boldsymbol{s}_k$ and $\boldsymbol{\phi}_k$, that is characterized by a zero-mean Gaussian noise with standard deviation σ_w. For every simulation run, σ_w is randomly and independently chosen from the set

$$\{0.001,\ 0.003,\ 0.006,\ 0.009,\ 0.012,\ 0.02\} \ .$$

Correspondingly, the measurements are perturbed by a zero-mean measurement noise $\boldsymbol{v}_k$ with standard deviation σ_v that is element of the set

$$\{0.1,\ 0.5,\ 1,\ 2,\ 3\} \ .$$

To simulate the network connections, we use two probabilistic models whose probability density functions of the delay distributions can be seen in Fig. 3. The packet loss rate of Network A is 0.2 and 0.005 for Network B. At the beginning of every simulation run, the employed models for the networks are chosen randomly and independently of each other. In case the actuator buffer runs empty, caused by too many packet losses, the actuator applies the default control input $u^d = 0$. For all simulation runs, the plant and estimators are initialized according to

$$\mathrm{E}\{\boldsymbol{x}_0\} = [0, 0.2, 0.2, 0]^T \ ,$$

$$\mathrm{E}\left\{(\boldsymbol{x}_0 - \mathrm{E}\{\boldsymbol{x}_0\})(\boldsymbol{x}_0 - \mathrm{E}\{\boldsymbol{x}_0\})^T\right\} = \begin{bmatrix} l_1 & 0 & 0 & 0 \\ 0 & l_2 & 0 & 0 \\ 0 & 0 & l_3 & 0 \\ 0 & 0 & 0 & l_4 \end{bmatrix} ,$$

where $l_1, \cdots, l_4$ are in the interval $(0, 1)$.

Overall, we conducted 200 Monte-Carlo-Simulation runs. A run consisted of 200 time steps, where the standard deviations of the process and measurement noise and the initial state parameters were randomly chosen from the specified sets. At the start of the simulation, the set point of the controlled pendulum was set to $[2, 0, 0, 0]^T$ and changed after 100 time steps to $[-2, 0, 0, 0]^T$.

B. Results

The Fig. 4 shows the root mean square error (RMSE) of the estimated states for both filters. An expected result is that the RMSE's of the non-directly observable states $\frac{\Delta s_k}{\Delta t}$ and $\frac{\Delta \phi_k}{\Delta t}$ are significantly higher than the RMSE's of the corresponding states s_k and ϕ_k which are measured directly. Furthermore, it can be seen that in time intervals, where the set point is hold constant, the RMSE of both estimators become very similar. This makes sense since the plant enters a stationary state and the control input candidates, which could be applied by the actuator, also become very similar. In other words, the variance of the control inputs becomes very small. Therefore, the proposed estimator cannot infer from the measured outputs which sequence is buffered in the actuator. The estimates of the buffered control input sequence converge to its steady state distribution and, hence, both estimators converge to each other.

However, if a set point change occurs, the RMSE of the proposed estimator are smaller than the one of the Kalman filter with steady state approximation. This is a result of the higher variance of the input uncertainty that leads to a better distinguishability of the possibly applied control input candidates. In this case, the estimated probabilities for the buffered control input sequences differ significantly from the steady state distribution. Thus, we conclude that the proposed estimator is suitable in particular for applications with frequent set point changes, i.e., for systems that are often operated in a transient state.

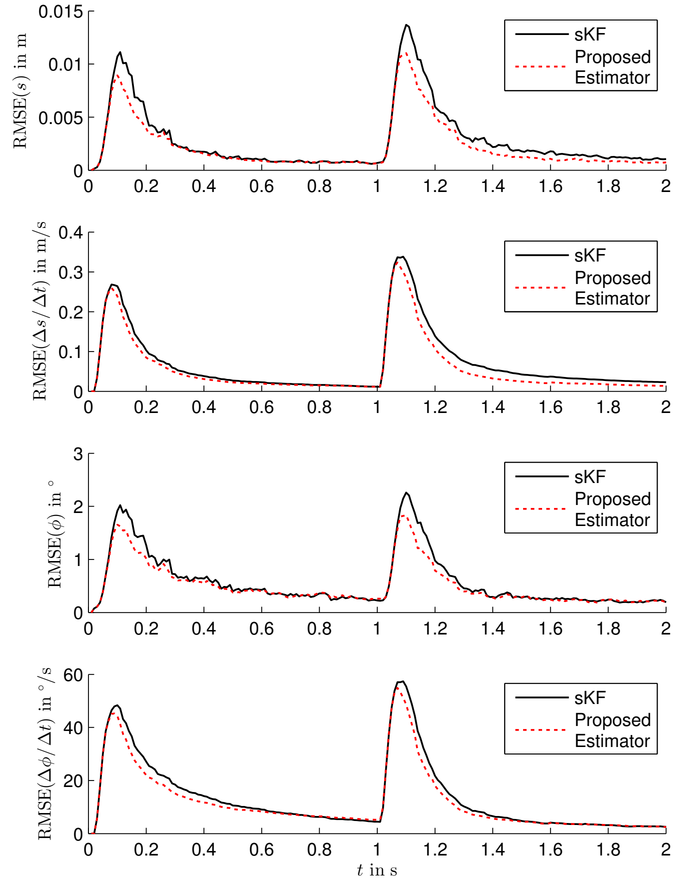
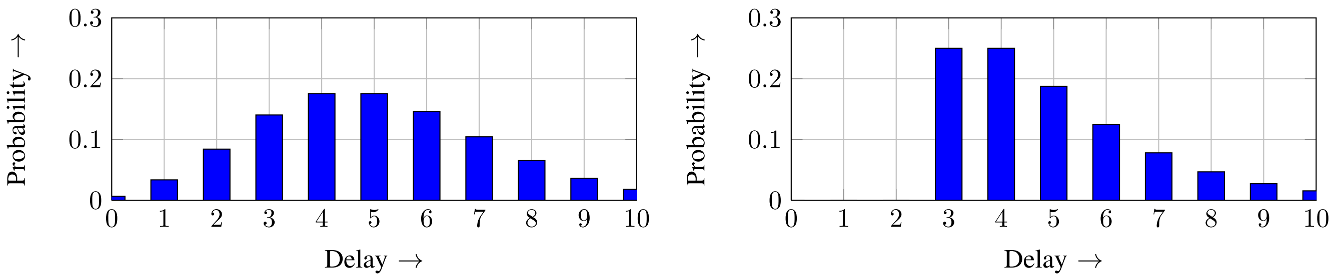


Fig. 4. Root mean square error of the estimated states over time, solid line: Kalman filter using the steady state distribution of the network (sKF); dashed line: proposed estimator.



(a) Network A

(b) Network B

Fig. 3. Probability density functions over the time delays of the two networks considered in the evaluation.

V. CONCLUSIONS

We considered the problem of state estimation in a Networked Predictive Control System, where the control inputs and measurements are sent via a network that is subject to random packet delays and data losses. It was pointed out that the estimator has only uncertain information about the control inputs actually applied by the actuator. This uncertainty, however, is correlated in time and can be reduced by using the proposed multiple-model-based estimator. The efficiency of this approach was demonstrated by means of Monte-Carlo-Simulation runs with an inverted pendulum on a cart. The simulations have shown that the estimator is suitable in particular for applications with frequent set point changes, i.e., for systems which are often operated in a transient state.

Furthermore, we point out that this concept cannot only be used in the specialized setup of Networked Predictive Control Systems but also for systems that are subject to stochastic input delays in general. The Networked Predictive Control setup, i.e., with predictive controller and actuator buffer as depicted in Fig. 1, is, however, perfectly suited for this approach since the buffering mechanism causes that the uncertainty about the applied control input is highly correlated in time.

VI. ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) within the framework of the Priority Programme Control Theory of Digitally Networked Dynamical Systems.

REFERENCES

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

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

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

[4] L. Grüne, J. Pannek, and K. Worthmann, "A Prediction Based Control Scheme for Networked Systems with Delays and Packet Dropouts," in *Proceedings of the 48th IEEE Decision and Control 2009 held jointly with the 28th Chinese Control Conference. CDC/CCC 2009*. IEEE, pp. 537–542.

[5] D. Quevedo and D. Nesic, "Input-to-State Stability of Packetized Predictive Control over Unreliable Networks Affected by Packet-Dropouts," *IEEE Transactions on Automatic Control*, no. 99, pp. 1–1, 2011.

[6] G. Liu, "Predictive Controller Design of Networked Systems with Communication Delays and Data Loss," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 6, pp. 481–485, 2010.

[7] P. Tang and C. De Silva, "Stability Validation of a Constrained Model Predictive Networked Control System with Future Input Buffering," *International Journal of Control*, vol. 80, no. 12, pp. 1954–1970, 2007.

[8] G. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked Predictive Control of Systems with Random Network Delays in both Forward and Feedback Channels," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 3, pp. 1282–1297, 2007.

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

[10] S. Smith and P. Seiler, "Estimation with Lossy Measurements: Jump Estimators for Jump Systems," *IEEE Transactions on Automatic Control*, vol. 48, pp. 2163–2171, 2003.

[11] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[12] L. Schenato, "Optimal Estimation in Networked Control Systems Subject to Random Delay and Packet Drop," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1311–1317, 2008.

[13] X. Lu, H. Zhang, W. Wang, and K. Teo, "Kalman Filtering for Multiple Time-Delay Systems," *Automatica*, vol. 41, no. 8, pp. 1455–1461, 2005.

[14] T. Larsen, N. Andersen, O. Ravn, and N. Poulsen, "Incorporation of Time Delayed Measurements in a Discrete-Time Kalman Filter," in *Proceedings of the 37th IEEE Conference on Decision and Control*, vol. 4. IEEE, 1998, pp. 3972–3977.

[15] Y. Bar-Shalom and H. Chen, "IMM Estimator with Out-of-Sequence Measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 90–98, 2005.

[16] Y. Bar-Shalom, "Update with Out-of-Sequence Measurements in Tracking: Exact Solution," *IEEE Transactions on Aerospace and Electronic Systems*.

[17] M. Epstein, L. Shi, and R. Murray, "An Estimation Algorithm for a Class of Networked Control Systems using UDP-like Communication Schemes," in *45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 5597–5603.

[18] S. Pan, H. Su, H. Wang, and J. Chu, "The Study of Joint Input and State Estimation with Kalman Filtering," *Transactions of the Institute of Measurement and Control*, vol. 33, no. 8, pp. 901–918, 2011.

[19] P. Kitanidis, "Unbiased Minimum-Variance Linear State Estimation," *Automatica*, vol. 23, no. 6, pp. 775–778, 1987.

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

[21] M. Moayedi, Y. Foo, and Y. Soh, "Adaptive Kalman Filtering in Networked Systems with Random Sensor Delays, Multiple Packet Dropouts and Missing Measurements," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1577–1588, 2010.

[22] O. do Valle Costa, M. Fragoso, and R. Marques, *Discrete-Time Markov Jump Linear Systems*. Springer Verlag, 2005.

[23] Y. Bar-Shalom and X. Li, "Estimation and Tracking- Principles, Techniques, and Software," *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[24] J. Tugnait and A. Haddad, "A Detection-Estimation Scheme for State

Estimation in Switching Environments," *Automatica*, vol. 15, no. 4, pp. 477–481, 1979.

[25] O. Costa, "Linear Minimum Mean Square Error Estimation for Discrete-Time Markovian Jump Linear Systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 8, pp. 1685–1689, 1994.

[26] G. Ackerson and K. Fu, "On State Estimation in Switching Environments," *IEEE Transactions on Automatic Control*, vol. 15, no. 1, pp. 10–17, 1970.

[27] H. Blom and Y. Bar-Shalom, "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients," *IEEE Transactions on Automatic Contro*, vol. 33, no. 8, pp. 780–783, 1988.

[28] X. Li and V. Jilkov, "A Survey of Maneuvering Target TrackingPart V: Multiple-Model Methods," in *Conference on Signal and Data Processing of Small Targets*, vol. 4473, 2003, pp. 559–581.

[29] G. Forney Jr, "The Viterbi Algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[30] C. Anderson, "Learning to Control an Inverted Pendulum using Neural Networks," *Control Systems Magazine, IEEE*, vol. 9, no. 3, pp. 31 –37, apr 1989.

[31] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*. Wiley-Interscience New York, 1972, vol. 172.