Sequence-Based Control for Networked Control Systems Based on Virtual Control Inputs

Achim Hekler, Jörg Fischer, and Uwe D. Hanebeck

***Abstract*— In this paper, we address the problem of controlling a system over an unreliable UDP-like network that is affected by time-varying delays and randomly occurring packet losses. A major challenge of this setup is that the controller just has uncertain information about the control inputs actually applied by the actuator. The key idea of this work is to model the uncertain control inputs by random variables, the so-called *virtual control inputs*, which are characterized by discrete probability density functions. Subject to this probabilistic description, a novel, easy to implement sequence-based control approach is proposed that extends any given state feedback controller designed without consideration of the network-induced disturbances. The high performance of the proposed controller is demonstrated by means of Monte Carlo simulation runs with an inverted pendulum on a cart.**

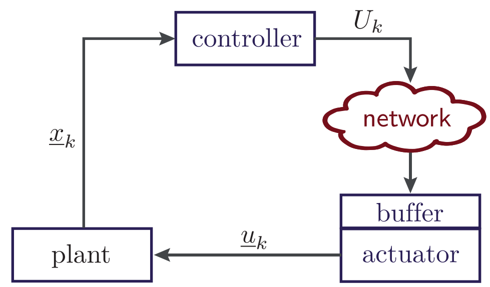

Fig. 1. Considered NCS architecture. Controller and actuator are connected through a communication network, whereas the link between plant and controller is transparent. For compensation of time delays and packet losses, a sequence-based controller is employed, which transmits an entire sequence U_k of adequate future control inputs instead of a single one.

I. INTRODUCTION

In networked control systems (NCS), the communication between components of the control loop can be realized via a communication network instead of a transparent connection [1], [2]. This system architecture offers many advantages, such as simple installation and maintenance, as well as a high flexibility in the system structure.

However, it is well known that compared to a transparent connection, the presence of a communication network in the control loop decreases the quality of control or even destabilizes the system [3], [4], [5]. This is mainly caused by time-varying transmission delays, randomly occurring packet losses, limited bandwidth of the communication channel, or quantization errors. Consequently, control methods for NCS have to consider both communication and control aspects.

Our approach is based on the well-known control technique for NCSs to transmit not just a single control input, but a whole sequence of adequate inputs for the future time steps, e.g., see [6], [7], [8], [9], [10]. These *sequence-based* control methods take advantage of the property of modern communication networks, in which data is transmitted in large time-stamped packets. The successfully transmitted sequences are stored in a buffer at the actuator and a specific selection logic enables that an adequate input can be passed on to the plant at every time step.

The main challenge of sequence-based control over an unreliable network that does not provide acknowledgements of successfully transmitted packets is that the controller only has uncertain information about the control inputs actually applied by the actuator. In the following, various sequence-based approaches are presented, which can deal with this uncertainty.

A. Related Work

In [7] and [11], a deterministic protocol is proposed to guarantee that the sequence of control inputs used by the controller for state predictions coincides with the sequence actually applied by the actuator. By forcing this property, the so-called prediction consistency, there are some significant drawbacks. Especially in the case of long time delays, the controller is frequently in recovery mode, in which the actuator rejects inconsistently predicted sequences, even when they are based on recent measurements.

In [12], a scenario-based NCS controller is proposed calculating the optimal control inputs for each possible delay of the previously transmitted sequences. Then, the set of control sequences is transmitted to a smart actuator, which selects the correct sequence. Obviously, this approach is impracticable if longer time delays should be considered.

Many sequence-based controller send at every time step a sequence of inputs resulting from a deterministic open-loop control problem exclusively depending on the current system state, e.g. see [8] or [13]. Consequently, these approaches do not incorporate into the control decision that control inputs for future time steps have already been sent by the controller and stored in the buffer of the actuator. However, these inputs have obviously a significant effect on the future development of the system.

In [14], a sequence-based controller is proposed that is based on a standard feedback controller, which has been

A. Hekler, J. Fischer, and U. D. Hanebeck are with the Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Institute for Anthropomatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, achim.hekler@kit.edu, joerg.fischer@kit.edu, uwe.hanebeck@ieee.org.

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

designed without consideration of network-induced disturbances. Exclusively based on the current system state, the controller predicts the future system development by means of a deterministic system model and the given feedback controller.

B. Key Idea

In this paper, we propose a sequence-based control method for NCS that models the unknown future inputs by random variables, named *virtual control inputs*. These random variables are characterized by probability density functions over potentially applied control inputs, which are derived from the data transmitted by the controller in the past. Based on this probabilistic description representing the best knowledge of the controller about the situation at the actuator, a sequence of control inputs for the future time steps is determined.

C. 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 $f(a)$. A vector-valued quantity $\underline{a}$ is indicated by underlining the corresponding identifier and 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. For the vector $[x_a, x_{a+1}, \ldots, x_b]$, we use the abbreviated notation $x_{a:b}$.

D. Outline

The remainder of the paper is organized as follows: In the next section, the considered problem is defined and the assumptions made are listed. Then, the proposed controller scheme for NCSs is described in detail and stability properties are examined. Section VI presents simulation results with an inverted pendulum and compares the proposed approach to a standard NCS technique. A summary and an outlook to future work concludes the paper.

II. CONSIDERED PROBLEM

Throughout the paper, we consider a discrete-time linear dynamic plant described in state-space form via

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

where $\underline{\boldsymbol{x}}_k \in \mathbb{R}^s$ denotes the system state at time step k and $\underline{u}_k \in \mathbb{R}^n$ the control input actually applied by the actuator. Note that due to time delays and packet losses in the network, $\underline{u}_k$ may differ from the control input the controller intended to apply. The system noise is subsumed by $\underline{\boldsymbol{w}}_k \sim f^w(\underline{w}_k)$ and is assumed to be a zero-mean Gaussian noise process. Furthermore, the system matrices $\mathbf{A} \in \mathbb{R}^{s\times s}$ and $\mathbf{B} \in \mathbb{R}^{s\times n}$ are known. The components of the control loop are time-triggered, synchronized and have identical cycle times.

In this paper, we restrict our considerations to the case, where the communication network is solely present in the controller-to-actuator link. For simplicity, we further assume that the controller has perfect information about the current system state $\underline{x}_k$ of the plant, i.e., the system state is completely measurable by the sensor and the connection between sensor and controller is perfect.

The employed network is capable of transmitting large time-stamped data packets and does not provide acknowledgements for successfully transmitted data as in UDP-like protocols. The data transmission might be subject to time-varying delays and randomly occurring packet losses, for which static probabilistic descriptions in form of probability density functions are given. For the rest of the paper, we subsume both the distribution over the time delays, as well as the occurrence probability of packet losses into a single probability density $f^\tau(\tau_k)$ by interpreting packet losses as infinitely long time delays.

Finally, we assume that a controller with a linear state feedback control law

$$\underline{u}_k = \mathbf{L} \cdot \underline{x}_k \tag{2}$$

is given that is designed without consideration of the network-induced disturbances.

In the following, we propose a scheme that extends this given non-networked controller (2) in such a way that it can deal with time delays and packet losses.

III. SEQUENCE-BASED CONTROL

In this section, we briefly review the general concept of sequence-based control as, e.g., used in [6], [7], [8], [9], [10], [14], because our control approach presented in the next section is based on this fundamental control concept.

In sequence-based control, the controller generates not only a single control input for the current control cycle, but also control inputs for future N time steps (with $N \in \mathbb{N}$). The entire control input sequence is lumped into one data packet and sent over the network to the actuator. The actuator is equipped with a buffer in which the most recent control input sequence is held, i.e., that sequence that has the latest time stamp among all received packets. Therefore, when a new packet is received by the actuator, it is taken into the buffer if its time stamp is later than the one of the packet already held in the buffer, else it is neglected. Finally, in every time step, the actuator applies the appropriate control input of the buffered sequence to the plant, i.e., that control input of the sequence that corresponds to the current time step.

In the following, we will denote the control input sequence generated by the controller at time k by U_k. An entry of that packet is denoted by $\underline{u}_{k+m|k}$ 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. For a packet of length $N+1$ generated in time step k, this gives

$$U_k = \begin{bmatrix} \underline{u}_{k|k}^{\mathrm{T}} & \underline{u}_{k+1|k}^{\mathrm{T}} & \cdots & \underline{u}_{k+N|k}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \ . \tag{3}$$

For example, let us assume the controller packet U_k is received by the actuator at time step $k+\tau_k$ with $\tau_k \in \mathbb{N}$. If none of the packets

$$U_{k+1}, U_{k+2}, \ldots, U_{k+\tau_k} \tag{4}$$

has been received by the actuator so far, then the buffer is overwritten with the entries of U_k and the entry $\underline{u}_{k+\tau_k|k}$ of U_k is applied to the plant. Otherwise, if the actuator has received any packet from (4) until time step $k+\tau_k$, then U_k is neglected and the entry $\underline{u}_{k+\tau_k|k+i}$ of the buffered sequence U_{k+i} is applied.

Since we do not assume that the time delays are bounded, it may happen that the buffer runs empty. In this case, the controller applies a default control input $\underline{u}^d$.

It is obvious that the control inputs applied by the actuator depend on the packet delays as well as losses and, therefore, inherit the stochastic nature of the network. This gives rise to the stochastic control approach discussed in the next section.

IV. SEQUENCE-BASED CONTROL WITH VIRTUAL CONTROL INPUTS

A. Virtual Control Inputs

In this section, we introduce the novel concept of virtual control inputs. To that end, we first define the information set $\mathcal{I}_k$ that summarizes the information the controller can use at time step k to calculate U_k. Considering causal controllers, the information set includes all measurements and all control packets that were received and sent until time step k. Furthermore, the information set contains the information about the given feedback matrix $\mathbf{L}$, the dynamics of the system $\mathcal{D}$ given by (1), the buffering logic $\mathcal{B}$ of the actuator described in Sec. III, and the stochastic characteristics of the process noise and the delay distribution of the nework according to

$$\mathcal{I}_k{}^1 = \{\underline{x}_{0:k}, U_{0:k-1}; \mathbf{L}, \mathcal{D}, \mathcal{B}, f^w(\underline{w}_k), f^\tau(\tau_k)\} \ .$$

Based on $\mathcal{I}_k$, we define the virtual control inputs (VCI) as follows.

Definition 1 (Virtual Control Inputs) *A virtual control input $\underline{\boldsymbol{u}}^v_{k+m|k} \sim f(\underline{u}^v_{k+m|k})$ is a prediction of the control input $\underline{\boldsymbol{u}}_{k+m}$ conditioned on the information set $\mathcal{I}_k$ (with $k, m \in \mathbb{N}$), so that*

$$f(\underline{u}^v_{k+m|k}) = f(\underline{u}_{k+m}|\mathcal{I}_k) \ .$$

To derive the probability density function $f(\underline{u}^v_{k+m|k})$ of the virtual control inputs $\underline{\boldsymbol{u}}^v_{k+m|k}$, we note that based on the information set $\mathcal{I}_k$, there is only a finite set of discrete values of control inputs that could be applied by the actuator. This is illustrated in Fig. 2 for the case of $N=2$, where the control inputs possibly applied at time step k are marked by the red rectangle. It should be noted that although this finite set of control inputs is discrete, the control inputs itself are over a continuous domain. The structure of the uncertainty can formally be described by a Dirac mixture density

$$f(\underline{u}^v_{k+m|k}) = \left[\sum_{i=0}^{N-m} \boldsymbol{\alpha}^{(i)}_{k+m|k} \cdot \delta(\underline{u}^v_{k+m|k} - \underline{u}_{k+m|k-i})\right] + \boldsymbol{\alpha}^{(N-m+1)}_{k+m|k} \cdot \delta(\underline{u}^v_{k+m|k} - \underline{u}^d) \ ,$$

where $m \in \{0, 1, \cdots, N\}$, the terms $\boldsymbol{\alpha}^{(i)}_{k+m|k}$ are scalar weighting factors with $\sum_{i=0}^{N-m+1} \boldsymbol{\alpha}^{(i)}_{k+m|k} = 1$, and $\delta()$ is the Dirac delta function. The weighting factors express the probability that the corresponding control input $\underline{u}_{k+m|k-i}$ is applied by the actuator, i.e.,

$$\boldsymbol{\alpha}^{(i)}_{k+m|k} = \mathrm{Prob}(\underline{\boldsymbol{u}}_{k+m} = \underline{u}_{k+m|k-i}|\mathcal{I}_k) \ .$$

The control input $\underline{u}_{k+m|k-i}$ is applied by the actuator if the sequence buffered in the actuator at time step $k+m$ has been generated by the controller $k+m-(k-i) = m+i$ time steps ago. In other words, $\underline{u}_{k+m|k-i}$ is applied by the actuator if the *age* of the buffered sequence, i.e., the difference between time step of generation and actual time step, at time step $k+m$ is equal to $m+i$. Denoting the age of the buffered sequence at time step k by the random variable $\boldsymbol{\theta}_k$, it holds that

$$\boldsymbol{\alpha}^{(i)}_{k+m|k} = \mathrm{Prob}(\boldsymbol{\theta}_{k+m} = i|\mathcal{I}_k) \ .$$

It has been shown in [15] and [16] that $\boldsymbol{\theta}_k$ can be described as state of a Markov chain with transition matrix $\mathbf{P}$, which depends on $f_\tau(\tau_k)$. Furthermore it has been shown that $\boldsymbol{\theta}_{k+m}$ (and therefore $\boldsymbol{\alpha}^{(i)}_{k+m|k}$) can be optimally estimated and predicted using the Wonham filter. However, this estimation leads to time-varying weighting factors. For the remainder of this paper, we approximate the weighting factors $\boldsymbol{\alpha}^{(i)}_{k+m|k}$ by its stationary probability solution $\lim_{\mathrm{k}\to\infty} \boldsymbol{\alpha}^{(i)}_{k|0} = \alpha^{(i)}_\infty$, which is also described in [15] and [16]. This stationary solution has the advantage that all weighting factors are time-invariant and the control inputs can be calculated easier. Furthermore, results on the stability of the closed-loop system can be obtained (see Sec. V). It should be noted, however, that the controller described in the next chapter can also be implemented using the exact, time-varying weighting factors.

Finally, we calculate the expected values of the virtual control inputs, since these will be needed in the derivation of the controller in the next chapter. For the expected, it holds

$$\begin{aligned}
E\left\{\underline{\boldsymbol{u}}^v_{k+m|k}\right\} &= \int_{-\infty}^{\infty} \underline{u}^v_{k+m|k} \ f(\underline{u}^v_{k+m|k}) \, \mathrm{d}\underline{u}^v_{k+m|k} \\
&= \int_{-\infty}^{\infty} \underline{u}^v_{k+m|k} \left(\sum_{i=0}^{N-m} \boldsymbol{\alpha}^{(i)}_{k+m|k} \ \delta(\underline{u}^v_{k+m|k} - \underline{u}_{k+m|k-i}) \right. \\
&\quad \left. + \boldsymbol{\alpha}^{(N-m+1)}_{k+m|k} \ \delta(\underline{u}^v_{k+m|k} - \underline{u}^d)\right) \mathrm{d}\underline{u}^v_{k+m|k} \\
&= \sum_{i=0}^{N-m} \boldsymbol{\alpha}^{(i)}_{k+m|k} \ \underline{u}_{k+m|k-i} + \boldsymbol{\alpha}^{(N-m+1)}_{k+m|k} \ \underline{u}^d \ ,
\end{aligned} \tag{5}$$

[1]The information set does not include the information that the controller will generate and sent control input sequences to the actuator in the future.

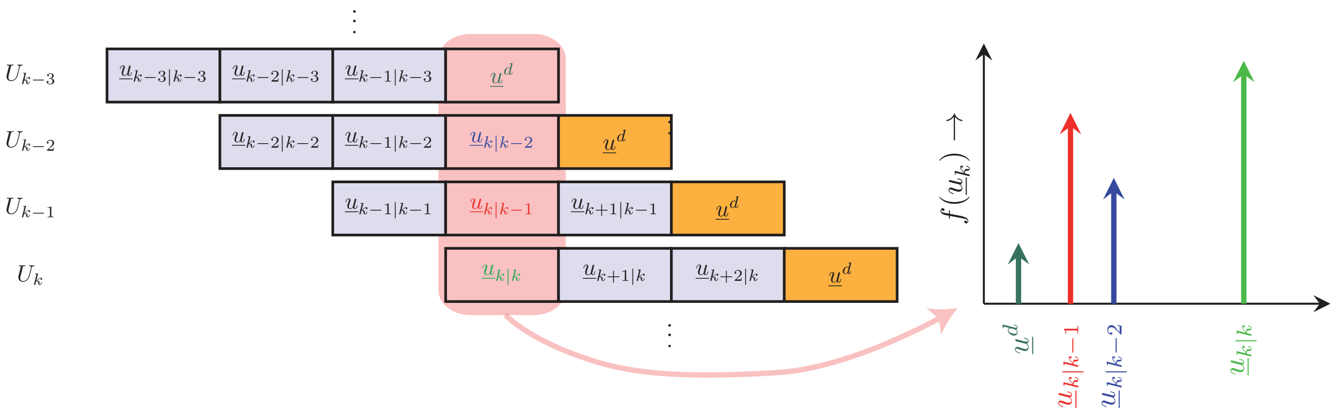

Fig. 2. Schematic illustration of the transmitted packets. Control inputs corresponding to the same time step are vertically aligned. For example, the control inputs that could be potentially applied by the actuator at time step k are indicated by the red rectangle. The yellow entries denote the default control inputs that would be employed if the buffer runs empty.

which becomes with the steady state approximation

$$E\left\{\underline{\boldsymbol{u}}^v_{k+m|k}\right\} \approx \sum_{i=0}^{N-m} \alpha_\infty^{(i)} \, \underline{u}_{k+m|k-i} + \alpha_\infty^{(N-m+1)} \, \underline{u}^d \ . \quad (6)$$

B. Controller Design

This subsection describes how to design the sequence-based controller if a linear feedback controller $\underline{u}_k = \mathbf{L} \cdot \underline{x}_k$ is given, where the feedback matrix $\mathbf{L}$ has been designed for the plant (1) without consideration of network effects by, e.g., pole placement or another control method, such as LQR, H_2, or H_∞. In the following, we use the feedback matrix $\mathbf{L}$ to generate control input sequences based on the predicted future states of the plant.

Based on the measured state $\underline{x}_k$ at time step k, the entries of the control input sequence U_k are calculated by

$$\underline{u}_{k|k} = \mathbf{L} \cdot \underline{x}_k \ , \quad (7)$$

$$\underline{u}_{k+1|k} = \mathbf{L} \cdot E\{\underline{\boldsymbol{x}}_{k+1}|\mathcal{I}_k\} \ , \quad (8)$$

$$\vdots$$

$$\underline{u}_{k+N|k} = \mathbf{L} \cdot E\{\underline{\boldsymbol{x}}_{k+N}|\mathcal{I}_k\} \ . \quad (9)$$

The future states $\underline{\boldsymbol{x}}_{k+m}$ are random with respect to the process noise and the virtual control inputs. The conditional expectation of the future states, $E\{\underline{\boldsymbol{x}}_{k+m}|\mathcal{I}_k\}$, can be calculated by

$$\begin{aligned} &E\{\underline{\boldsymbol{x}}_{k+m}|\mathcal{I}_k\} \\ &= E\{\mathbf{A}\underline{\boldsymbol{x}}_{k+m-1} + \mathbf{B}\underline{\boldsymbol{u}}_{k+m-1} + \underline{\boldsymbol{w}}_{k+m-1}|\mathcal{I}_k\} \\ &= \mathbf{A} \cdot E\{\underline{\boldsymbol{x}}_{k+m-1}|\mathcal{I}_k\} + \mathbf{B} \cdot E\{\underline{\boldsymbol{u}}^v_{k+m-1|k}\} \\ &\approx \mathbf{A} \cdot E\{\underline{\boldsymbol{x}}_{k+m-1}|\mathcal{I}_k\} \\ &\quad + \mathbf{B} \cdot \left(\sum_{i=0}^{N-m+1} \boldsymbol{\alpha}_\infty^{(i)} \, \underline{u}_{k+m-1|k-i} + \boldsymbol{\alpha}_\infty^{(N-m+2)} \, \underline{u}^d \right) \\ &= \left(\mathbf{A} + \alpha_\infty^{(0)} \mathbf{BL}\right) E\{\underline{\boldsymbol{x}}_{k+m-1|k}\} \quad (10) \\ &\quad + \mathbf{B} \cdot \left(\sum_{i=1}^{N-m+1} \alpha_\infty^{(i)} \, \underline{u}_{k+m-1|k-i} + \alpha_\infty^{(N-m+2)} \, \underline{u}^d \right) \ . \end{aligned}$$

For taking the expected value, we use (5) and the assumption that $\underline{\boldsymbol{w}}_{k+m-1}$ is zero-mean and independent of $\underline{\boldsymbol{x}}_{k+m-1}$ and $\underline{\boldsymbol{u}}_{k+m-1}$. Equation (10) shows that the expected value of the predicted state can be calculated *recursively*, which basically results from the fact that the state equation of the plant and the expectation operation are linear. The recursive formulation of the control algorithm given by (9) and (10) is efficient for practical implementation.

To analyze the stability of the closed-loop system in the next section, we additionally derive an *explicit* formulation of the controller. To that end, we introduce the augmented state

$$\underline{\psi}_k = \left[\underline{x}_k^{\mathrm{T}} \quad \underline{\eta}_k^{\mathrm{T}}\right]^{\mathrm{T}} \ , \quad (11)$$

with

$$\underline{\eta}_k = \begin{bmatrix} [\underline{u}^{\mathrm{T}}_{k|k-1} \ \ \underline{u}^{\mathrm{T}}_{k+1|k-1} \ \ \cdots \ \ \underline{u}^{\mathrm{T}}_{k+N-1|k-1}]^{\mathrm{T}} \\ [\underline{u}^{\mathrm{T}}_{k|k-2} \ \ \underline{u}^{\mathrm{T}}_{k+1|k-2} \ \ \cdots \ \ \underline{u}^{\mathrm{T}}_{k+N-2|k-2}]^{\mathrm{T}} \\ \vdots \\ [\underline{u}^{\mathrm{T}}_{k|k-N+1} \ \ \underline{u}^{\mathrm{T}}_{k+1|k-N+1}]^{\mathrm{T}} \\ \underline{u}_{k|k-N} \\ \underline{u}^d \end{bmatrix} \ .$$

The vector $\underline{\eta}_k$ contains the default control input and all control inputs of the already sent control input sequences U_{k-1}, U_{k-2}, …, U_{k-N} that still could be applied in time step k or later. Using (3), (9), (10), and (11), the controller can be formulated as a linear state feedback controller given by

$$U_k = \begin{bmatrix} \underline{u}_{k|k} \\ \underline{u}_{k+1|k} \\ \vdots \\ \underline{u}_{k+N|k} \end{bmatrix} = \begin{bmatrix} \mathbf{L} \cdot \underline{x}_k \\ \mathbf{L} \cdot E\{\underline{\boldsymbol{x}}_{k+1}|\mathcal{I}_k\} \\ \vdots \\ \mathbf{L} \cdot E\{\underline{\boldsymbol{x}}_{k+N}|\mathcal{I}_k\}\} \end{bmatrix} = \tilde{\mathbf{L}} \cdot \underline{\psi}_k \ . \quad (12)$$

Due to space constraints, we refer to [16] for a detailed description and derivation of the matrix $\tilde{\mathbf{L}}$.

V. STABILITY OF THE PROPOSED APPROACH

In this section, we will derive a criterion for closed-loop stability of the proposed controller. Therefore, a model of network and actuator is derived, that, in a second step, will be combined with the model of the plant (1) and the controller (12). Based on $\underline{\eta}_k$ and $\boldsymbol{\theta}_k$ as introduced in section IV-A, the combined state space model of network and actuator can be formulated similar as in [17] as

$$\begin{aligned} \underline{\eta}_{k+1} &= \mathbf{F}\underline{\eta}_k + \mathbf{G}U_k \ , \quad (13)\\ \underline{\boldsymbol{u}}_k &= \mathbf{H}_{\theta_k}\underline{\eta}_k + \mathbf{J}_{\theta_k}U_k \ , \quad (14) \end{aligned}$$

with

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{J}_{\theta_k} = \begin{bmatrix} \delta_{(\theta_k,0)}\mathbf{I} & \mathbf{0} \end{bmatrix},$$

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

where the $\mathbf{0}$'s are matrices with all elements equal to zero and the $\mathbf{I}$'s are identity matrices, each of appropriate dimension. The term $\delta_{(\theta_k,i)}$ is the Kronecker delta, which is defined as

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

By using the augmented state $\underline{\psi}_k$ from (11) and combining (1), (13), and (14), it holds

$$\underline{\boldsymbol{\psi}}_{k+1} = \begin{bmatrix} \mathbf{A} & \mathbf{B}\cdot\mathbf{H}_{\theta_k} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \underline{\psi}_k + \begin{bmatrix} \mathbf{B}\cdot\mathbf{J}_{\theta_k} \\ \mathbf{G} \end{bmatrix} U_k + \begin{bmatrix} \underline{\boldsymbol{w}}_k \\ \mathbf{0} \end{bmatrix} \ .$$

Using (12) results in

$$\begin{aligned} \underline{\boldsymbol{\psi}}_{k+1} &= \left(\begin{bmatrix} \mathbf{A} & \mathbf{B}\cdot\mathbf{H}_{\theta_k} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} - \begin{bmatrix} \mathbf{B}\cdot\mathbf{J}_{\theta_k} \\ \mathbf{G} \end{bmatrix} \cdot \tilde{\mathbf{L}} \right) \underline{\psi}_k + \begin{bmatrix} \underline{\boldsymbol{w}}_k \\ \mathbf{0} \end{bmatrix} \\ &= \tilde{\mathbf{A}}_{\theta_k}\underline{\psi}_k + \underline{\tilde{\boldsymbol{w}}}_k \ . \quad (15) \end{aligned}$$

The closed-loop system described by (15) can be interpreted as an inhomogeneous Markovian jump linear system (MJLS). For this kind of system, several results on mean square stability are available in the literature, e.g., [18] and [19]. In the following, we adopt the concept of mean square stability and proof from [18].

Definition 2 *The system (15) with Markovian jump parameter $\boldsymbol{\theta}_k$ is mean square stable (MSS), if for any initial condition $\theta_0 \in \{0, 1, \cdots, N\}$ and $\underline{\psi}_0 \in \mathbb{R}^{d+s}$ there exist a bounded $\underline{\mu} \in \mathbb{R}^{d+s}$ and a symmetric positive-semidefinite matrix $\mathbf{M}$ (independent of $\underline{\psi}_0$ and θ_0) such that*

$$\lim_{k\to\infty} E\left\{\underline{\boldsymbol{\psi}}_k\right\} = \underline{\mu} \ , \quad (16)$$

$$\lim_{k\to\infty} E\left\{\underline{\boldsymbol{\psi}}_k\underline{\boldsymbol{\psi}}_k^T\right\} = \mathbf{M} \ . \quad (17)$$

Mass of the cart	1.096 kg
Mass of the pendulum	0.109 kg
Friction of the cart	0.1 N/m/s
Length to pendulum center of mass	0.25 m
Inertia of the pendulum	$0.0034\text{kg}\cdot\text{m}^2$

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

Theorem 1 *The system (15) with Markovian jump parameter $\boldsymbol{\theta}_k$ and transition matrix $\mathbf{P}$ is stable in the mean square sense, if and only if*

$$r_\sigma((\mathbf{P}^T \otimes \mathbf{I}_{n^2}) \cdot \text{diag}[\tilde{\mathbf{A}}_i \otimes \tilde{\mathbf{A}}_i]) < 1 \ , \quad (18)$$

where $r_\sigma(\mathbf{M})$ is the spectral radius of $\mathbf{M}$ and $\text{diag}[\mathbf{S}_i]$ is the block diagonal matrix built by $\mathbf{S}_i$ in the diagonal with $i = \{0, 1, ..., N\}$ and zero everywhere else, i.e.,

$$\text{diag}[\mathbf{S}_i] = \begin{bmatrix} \mathbf{S}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S}_N \end{bmatrix} \ . \quad (19)$$

Proof: The result follows from theorem 3.9 and 3.33 in [18]. ■

VI. SIMULATION RESULTS

In this section, we evaluate the presented method by means of simulations with an inverted pendulum on a cart, which is a classical benchmark for illustrating the performance of control techniques. A basic description of this experimental setup can be found, e.g., in [20].

A. Simulation Setup

Table I shows the plant parameters of the inverted pendulum used in the simulations.

The continuous differential equation is linearized and discretized with sampling time 0.01 s, which results in the discrete-time linear system model (1) with

$$\mathbf{A} = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1.0015 & 0.01 \\ 0 & 0 & 0.2941 & 1.0015 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -0.0001 \\ -0.01 \\ 0.0002 \\ 0.03 \end{bmatrix} \ .$$

For realization of (2), a classical LQR controller is deployed [21]. The LQR minimizes the cumulated costs given by

$$\sum_{k=0}^{\infty} \underline{x}_k^{\mathrm{T}}\mathbf{Q}\underline{x}_k + \underline{u}_k^{\mathrm{T}}\mathbf{R}\underline{u}_k \ . \quad (20)$$

Choosing the weighting matrices of the cumulated costs with

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{R} = 0.1 \ ,$$

results in a feedback matrix $\mathbf{L}$ of the LQR with

$$\mathbf{L} = \begin{bmatrix} 6.57 & 5.95 & 36.21 & 6.66 \end{bmatrix} \ .$$

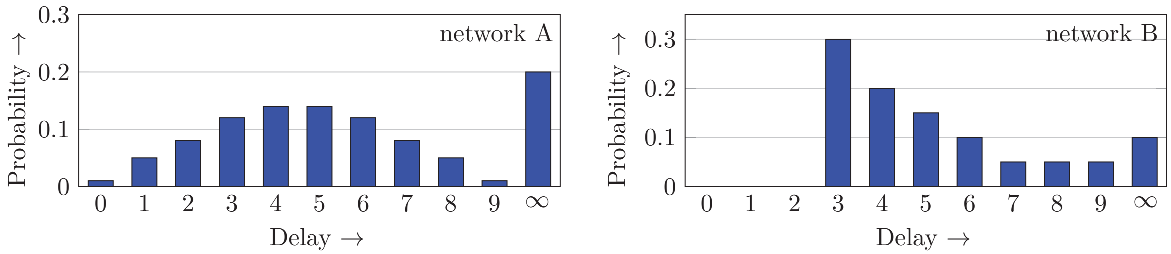

Fig. 3. Probability density functions over the time delays of the two networks considered in the evaluation.

	NNC	VCI-NCS	Det-NCS
$\sigma_w = 0.001$ / network A	14.0	14.3	14.4
$\sigma_w = 0.005$ / network A	28.4	37.5	38.2
$\sigma_w = 0.01$ / network A	76.3	113.3	116.7
$\sigma_w = 0.001$ / network B	14.0	14.3	14.4
$\sigma_w = 0.005$ / network B	27.7	36.5	37.6
$\sigma_w = 0.01$ / network B	75.3	111.4	117.1

TABLE II
CUMULATED COSTS AVERAGED OVER THE 100 MONTE CARLO RUNS.

At every time step k, we add a process noise $\underline{\boldsymbol{w}}_k$ to the angle ϕ_k of the pendulum, which is characterized by a zero-mean Gaussian noise with varying standard deviation σ_w. The length of the transmitted packets is $N = 9$ and the default input $\underline{u}_k^d$ is set to 0. The initial state vector $\underline{x}_0$ is

$$\underline{x}_0 = \begin{bmatrix} x_0 & \dot{x}_0 & \phi & \dot{\phi} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 0.1 & 0 \end{bmatrix}^{\mathrm{T}} .$$

In order to simulate the transmission characteristics of the network, the two probabilistic models depicted in Fig. 3 are employed.

Overall, we conduct 100 Monte Carlo simulation runs with varying standard deviations σ_w of the process noise, where each run takes 5 seconds.

We compare the proposed technique for NCS with virtual control inputs (VCI-NCS) to two other approaches. For better analyzing the quality of the compensation technique for time delays and packet losses, we first consider a classical non-networked LQR (abbreviated by NNC) with a transparent connection between controller and actuator. In this case, all calculated control inputs $\underline{u}_k = L \cdot \underline{x}_k$ are received by the actuator without any time delay. Consequently, the control quality of NNC can be seen as a ground truth for the NCS control methods.

Furthermore, we compare VCI-NCS to a widely used sequence-based controller that sends at every time step a sequence of control inputs resulting from a *deterministic* state prediction (Det-NCS) [8], [13]. In more detail, the packet U_k sent in time step k contains the entries

$$\begin{aligned} \underline{u}_{k|k} &= \mathbf{L} \cdot \underline{x}_{k|k} \ , \\ \underline{u}_{k+1|k} &= \mathbf{L} \cdot \underline{x}_{k+1|k} \ , \\ &\vdots \\ \underline{u}_{k+N|k} &= \mathbf{L} \cdot \underline{x}_{k+N|k} \ , \end{aligned}$$

where $\underline{x}_{k+i|k}$ for $1 \leq i \leq N$ is determined according to

$$\underline{x}_{k+i|k} = \mathbf{A}\underline{x}_{k+i-1|k} + \mathbf{B}\underline{u}_{k+i-1|k} \ .$$

Consequently, when using this NCS control technique, the packet entries exclusively depend on the current system state $\underline{x}_k$. In particular, the potential influence of the actuator buffer content on the future system behavior is not considered in the packet design.

B. Results

In Fig. 4, an example state trajectory of a simulation run with network A and a standard deviation $\sigma_w = 0.01$ is depicted. Compared to Det-NCS, the state trajectory of the VCI-NCS shows a lower deviation from the setpoint. Furthermore, the VCI-NCS trajectory is closer to the trajectory of the non-networked controller NNC. Thus, for this example, the proposed method seems to be an adequate technique for compensating time delays and packet losses in NCSs.

In order to make more general, quantitative statements, we conducted 100 Monte Carlo simulation runs with different parameter settings. For each setting, the resulting were calculated according to (20) and averaged over all runs. The results are shown in Table II. For small process noise, the results of Det-NCS and VCI-NCS are very similar. This results from the fact, that due to the low pertubation all control inputs possibly applied at a certain time step, i.e., all buffered control inputs and the entries of the calculated control sequence, differ only marginally. With increasing process noise, however, the variance of these control inputs gets higher and they may differ more significantly. This explains why the propopsed VCI-NCS strategy, which explicitly considers the variance of the control inputs, results in a significantly lower cost compared to the Det-NCS strategy.

VII. CONCLUSIONS

We presented a novel sequence-based control scheme for NCS, which explicitly incorporates communication aspects during the design of the packets. The key idea of our approach is that the controller subsumes its knowledge about the control inputs potentially applied by the actuator in form of a discrete probability density, the so-called *virtual control inputs*. Based on this probabilistic description and a given state feedback control law, the controller calculates a high-quality sequence of future control inputs. In contrast to existing sequence-based control approaches, the content of the actuator buffer, which has obviously a significant effect

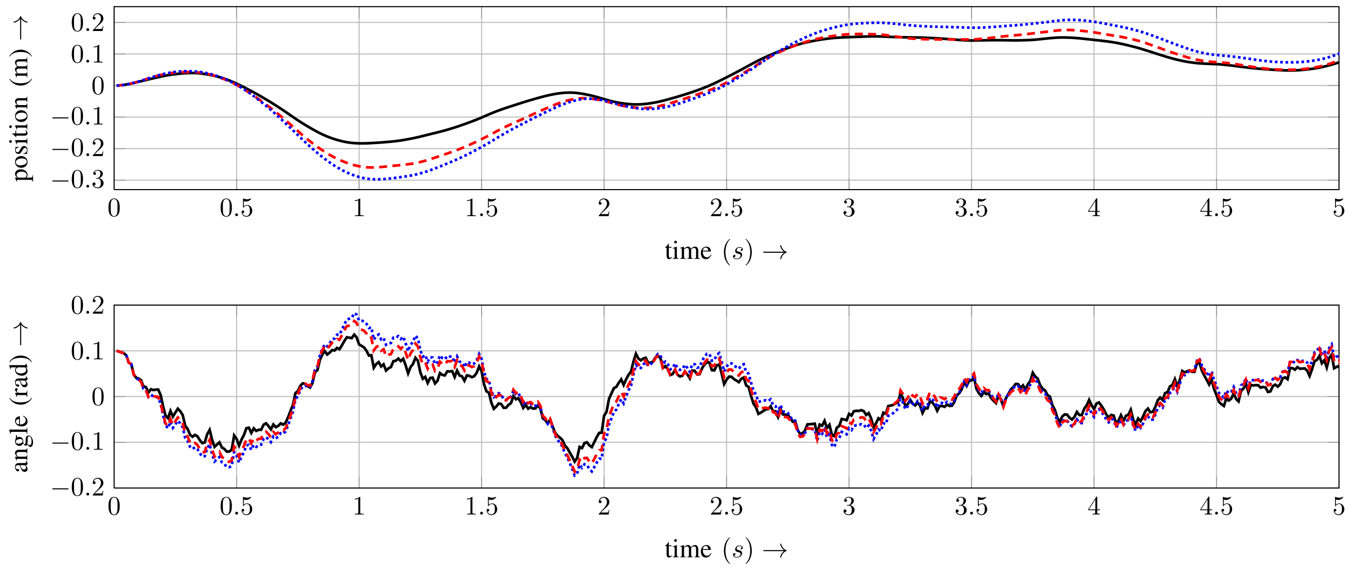

Fig. 4. Example state trajectory for a single simulation run with network A and a standard deviation $\sigma_w = 0.01$ of the process noise. The result of the controller without a network (NNC) is depicted with a solid black line (——), the proposed approach VCI-NCS with a dashed red (- - - -), and the Det-NCS with a dotted blue (·········) line.

on the future system behavior, is explicitly considered in the calculation of the packet entries.

To the best of our knowledge, the concept of virtual control inputs is innovative and promising, especially since simulation results with an inverted pendulum shows an excellent performance of the proposed approach, even in comparison to standard NCS methods.

Future work will be concerned with the TCP case, in which the controller receives time-delayed acknowledgements of successfully transmitted packets. This additional information allows component reduction of the involved probability density functions and hence, a more precise prediction of the future system behavior can be achieved.

References

[1] T. Yang, "Networked Control System: a Brief Survey," *IEEE Proceedings, Control Theory and Applications*, vol. 153, no. 4, pp. 403–412, July 2006.

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

[4] A. Bemporad, M. Heemels, and M. Johansson, *Networked Control Systems*. Springer-Verlag New York Inc, 2010, vol. 406.

[5] W. Heemels, A. Teel, N. van de Wouw, and D. Nesic, "Networked Control Systems With Communication Constraints: Tradeoffs Between Transmission Intervals, Delays and Performance," *IEEE Transactions on Automatic 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 1998*, vol. 2. IEEE, 1998, pp. 2133–2138.

[7] 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 Conference on Decision and Control held jointly with the 28th Chinese Control Conference*. IEEE, 2009, pp. 537–542.

[8] 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.

[9] G. Liu, J. Mu, D. Rees, and S. Chai, "Design and Stability Analysis of Networked Control Systems with Random Communication Time Delay Using the Modified mpc," *International Journal of Control*, vol. 79, no. 4, pp. 288–297, 2006.

[10] 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.

[11] I. Polushin, P. Liu, and C. Lung, "On the Model-Based Approach to Nonlinear Networked Control Systems," *Automatica*, vol. 44, no. 9, pp. 2409–2414, 2008.

[12] D. Quevedo, E. Silva, and G. Goodwin, "Control over Unreliable Networks Affected by Packet Erasures and Variable Transmission Delays," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 672–685, 2008.

[13] ——, "Packetized Predictive Control over Erasure Channels," in *Proceedings of the American Control Conference 2009*. IEEE, 2007, pp. 1003–1008.

[14] 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.

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

[16] A. Hekler, J. Fischer, and U. D. Hanebeck, "Sequence-Based Control for Networked Control Systems Based on Virtual Control Inputs," *Arxiv preprint, arXiv:1206.0549v2*, 2012.

[17] L. Xiao, A. Hassibi, and J. How, "Control with Random Communication Delays via a Discrete-Time Jump System Approach," in *Proceedings of the American Control Conference 2000*, vol. 3. IEEE, 2000, pp. 2199–2204.

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

[19] X. Feng, K. Loparo, Y. Ji, and H. Chizeck, "Stochastic Stability Properties of Jump Linear Systems," *IEEE Transactions on Automatic Control*, vol. 37, no. 1, pp. 38–53, 1992.

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

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