Control over Unreliable Networks Based on Control Input Densities

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

***Abstract*—Time delays and data losses arising from an unreliable communication between the components of a control loop decrease the quality of control and thus, have to be incorporated explicitly in the control decision. In this paper, a novel concept, the so-called *virtual control inputs*, is presented, which extends the well-established control technique of sending sequences of future inputs by incorporating also the potential effects of previously transmitted sequences on the future system behavior. The key idea of this paper is to model the unknown future inputs as random variables characterized by probability density functions over the finite set of potential future inputs. Subject to this probabilistic description of the future inputs, the controller determines the optimal open-loop sequence over a finite horizon. The high capacity of the proposed approach is demonstrated by simulations, in which a sensor manager schedules sensors for tracking a mobile object.**

I. Introduction

In networked control systems (NCS), the components of the control loop are connected via a shared communication network instead of a transparent connection [1], [2], [3]. Therefore, the data transmission between the individual components of the control loop is in general affected by time-varying delays and randomly occurring data losses. Consequently, if an optimal input is received, it is in general out of date, since its optimality is referred to a previous time step. Therefore, neglecting these time delays by the controller generally decreases the control quality [3], [4], [5].

In this paper, we consider the problem of controlling a system with a finite discrete set of possible control inputs over an unreliable network. For example, such a controller can be applied as a sensor manager in large sensor networks for increasing the operational lifetime, see, e.g., [6], [7]. Instead of switching on all sensors at each time step, a sensor manager selects a subset of sensors, which maximizes a specific objective, e.g., the maximum information gain about the system state. Then, the optimal sensor configuration is typically sent to the sensor nodes by a wireless network, which is usually affected by time-varying delays and packet losses, see Fig. I.

In networked control theory, a well-established control approach for dealing with time delays and packet losses is to employ a model predictive controller (MPC) instead of a classical one. Such a controller does not consider exclusively the current system state, when determining the optimal input, but also the possible future system behavior within a horizon. Since the optimal future inputs based on the current available information are calculated anyway in the optimization process, the MPC does not send only a single value, but a whole sequence of future inputs [8], [9]. This so-called *sequence-based control* is realizable, since in modern communication networks data is typically transmitted in large time-stamped packets. The successfully transmitted sequence with the highest time stamp is stored in a buffer at the side of the receiver and a specific selection logic enables that at every time step, some reasonable inputs are available, see Fig. 2.

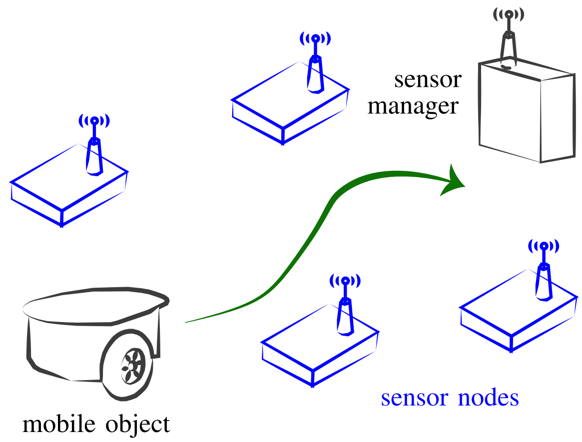

Fig. 1. An exemplary application of the considered problem class is sensor management in wireless networks. In our example, the objective of the sensor manager is to select a subset of sensors in each time step, such that the uncertainty about the position of the mobile object is minimized. The optimal sensor configurations are transmitted over a wireless network, which is affected by time delays and data losses.

The main problem of sequence-based control methods is that the model predictive controller actually needs the knowledge about the applied inputs in the past and in the future, when it determines the current optimal input sequence. However, due to network-induced disturbances, this demand cannot obviously be met.

A. Related Work

In literature, there are several proposals, how to deal with this problem.

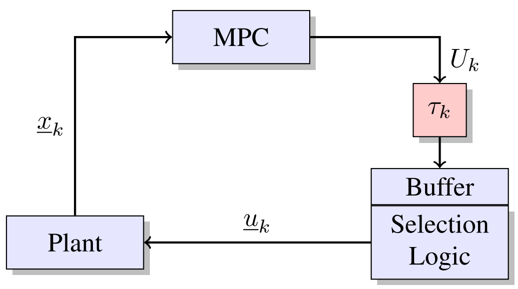

Fig. 2. System architecture of a sequence-based control system. The employed model predictive controller (MPC) does not send only a single input, but a whole sequence U_k of optimal future inputs based on the current available information. The transmission of U_k is affected by time-varying delays and data losses. At the side of the receiver, the successfully transmitted sequence with the highest time stamp is stored in the buffer in each time step. Then, the corresponding input for the current time step is selected by the selection logic in each time step.

In [10] and [11], the consistency of the inputs used by the model predictive controller for propagating the current system state over time and the ones actually applied is guaranteed by a deterministic protocol. However, in the case of long time delays, the controller is often in recovery mode, in which inconsistently predicted sequences are rejected, even when they are based on quite recently taken measurements.

The authors of [12] present a scenario-based controller, which calculates the optimal input sequence for all possible delays of the previously sent packets. Using time stamps, a smart actuator can then select the specific sequence that was calculated based on the assumption of the true delays. It is obvious that the complexity of this method increases enormously if there are long time delays.

Many approaches solve at every time step an open-loop control problem depending exclusively on the current system state, e.g., see [8] or [9]. Hence, these methods ignore the content of the buffer, although these stored entries may obviously influence the future development of the system.

B. Contribution

The key idea of this paper is to model the unknown inputs by random variables, named *virtual control inputs* (VCI). These variables are characterized by probability density functions, which can be derived from the entries of the previously sent packets. With respect to this probabilistic description, the optimal open-loop sequence of inputs within a finite horizon is determined.

C. Outline

The remainder of the paper is structured as follows: In the next Section, the considered optimization problem is defined in detail and the assumptions made are listed. Furthermore, an example application for the considered class of control problem, namely sensor scheduling, is introduced. Section III presents the fundamentals of sequence-based control, on which our approach presented in Section IV is based on. In Section V, simulation results are presented and finally, the paper concludes with a summary and an outlook on future work.

D. General Notation

Throughout this 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 the available information up to time t. For the vector $[x_a, x_{a+1}, \dots, x_b]$, we use the abbreviated notation $x_{a:b}$.

II. Considered Problem

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

$$\underline{\boldsymbol{x}}_{k+1} = \underline{a}_k(\underline{\boldsymbol{x}}_k, \underline{u}_k, \underline{\boldsymbol{w}}_k) \ , \tag{1}$$

where $\underline{\boldsymbol{x}}_k \in \mathbb{R}^n$ denotes the system state at time k and $\underline{\boldsymbol{w}}_k \sim f(\underline{w}_k)$ subsumes the system noise. The (deterministic) input $\underline{u}_k \in \mathcal{U}_k$ takes values from a finite discrete-valued set $\mathcal{U}_k$.

The system (1) is controlled by a time-discrete controller over an unreliable network, which induces disturbances such as time delays and data losses.

Assumption 1. *The transmission of the optimal input $\underline{u}_k^*$ is affected by time-varying delays and randomly occurring data losses, for which probabilistic models are given. The controller does not receive acknowledgements for successfully transmitted data.*

For the sake of clarity, we interpret packet losses as infinitely long time delays from now on and thus, only a single probabilistic description of the network-induced disturbances is needed. Additionally, we assume that the components of the control loop meet following assumption:

Assumption 2. *The components of the control loop are time-triggered, time-synchronized and have identical cycle times.*

Finally, we utilize a property of modern communication networks:

Assumption 3. *The communication network is capable of transmitting large time-stamped data packets.*

In the following, an example application for the considered problem class, namely sensor scheduling, is discussed.

Example System: Sensor Scheduling
As mentioned in Section I, the aim of scheduling in a sensor network is to select a subset of sensors, which maximizes a specific objective, e.g. the information gain.

Here, we consider the localization and tracking of a single mobile object, which dynamic behavior is described by the constant velocity model

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}\underline{\boldsymbol{x}}_k + \underline{\boldsymbol{w}}_k \ , \tag{2}$$

where the system noise $\underline{\boldsymbol{w}}_k$ is assumed to be a zero-mean Gaussian noise. The system state $\underline{\boldsymbol{x}}_k = [\boldsymbol{x}_k, \dot{\boldsymbol{x}}_k, \boldsymbol{y}_k, \dot{\boldsymbol{y}}_k]^T$

comprises the two-dimensional position $[\boldsymbol{x}_k, \boldsymbol{y}_k]^T$ and the two-dimensional velocity $[\dot{\boldsymbol{x}}_k, \dot{\boldsymbol{y}}_k]^T$. The system matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} ,$$

and the covariance matrix of the system noise is

$$\mathbf{C}_k^w = q \cdot \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} ,$$

where T is the sampling interval and q is the scalar diffusion strength [6]. Information about the current system state is provided by S distance sensors. Hence, the nonlinear sensor model of sensor $i \in \{1, \ldots, S\}$ is given by

$$\boldsymbol{y}_k^i = \sqrt{(\boldsymbol{x}_k - x^i)^2 + (\boldsymbol{y}_k - y^i)^2} + \boldsymbol{v}_k^i , \tag{3}$$

where x^i and y^i are the x-coordinate and the y-coordinate of sensor i and the sensor noise $\boldsymbol{v}_k^i$ of sensor i is zero-mean and normally distributed with covariance $\mathbf{C}_k^v$. For simplicity reasons, we linearize (3), which results in

$$\boldsymbol{y}_k^i = \mathbf{H}_k^i \underline{\boldsymbol{x}}_k + \boldsymbol{v}_k^i , \tag{4}$$

where $\mathbf{H}_k^i$ denotes the measurement matrix of sensor i. In order to extend the operational lifetime, a sensor manager is employed, which selects in each time step a single sensor by optimizing a cost function. In this application, the decision variable u_k is scalar and the set $\mathcal{U}_k = \{1, \ldots, S\}$ contains the identifier of the S sensors. For realization of the cost function, a covariance-based function depending on the expected posterior covariance matrix is used, as proposed in [6]. This covariance matrix is defined by

$$\begin{aligned} \mathbf{C}_k^e &:= \mathsf{Cov}\{\underline{\boldsymbol{x}}_k^e | \underline{\boldsymbol{x}}_k^p, \boldsymbol{y}_k, u_k\} \\ &= E\{(\underline{x}_k - \hat{\underline{x}}_k^e)(\underline{x}_k - \hat{\underline{x}}_k^e)^T | u_k\} , \end{aligned} \tag{5}$$

where $\hat{\underline{x}}_k^e$ is the mean vector of the posterior state estimate after the filter step. For the considered linear case, $\mathbf{C}_k^e$ can be calculated according to the Kalman filter step equation

$$\mathbf{C}_k^e = \mathbf{C}_k^p - \mathbf{K}_k^i \mathbf{H}_k^i \mathbf{C}_k^p \tag{6}$$

with the a priori covariance matrix

$$\mathbf{C}_k^p = \mathbf{A}_{k-1} \mathbf{C}_{k-1}^e \mathbf{A}_{k-1}^T + \mathbf{B}_{k-1} \mathbf{C}_{k-1}^w \mathbf{B}_{k-1}^T \tag{7}$$

and the Kalman gain

$$\mathbf{K}_k^i = \mathbf{C}_k^p \left(\mathbf{H}_k^i\right)^T \left(\mathbf{H}_k^i \mathbf{C}_k^p \left(\mathbf{H}_k^i\right)^T + \mathbf{C}_k^v\right)^{-1} . \tag{8}$$

By using a scalar function

$$g(\mathbf{C}_k^e(u_k)) \tag{9}$$

as, e.g., the determinant, the eigenvalues or the trace of $\mathbf{C}_k^e$, a quantification of the expected uncertainty can be realized.

The sensor manager broadcasts the optimal sensor configuration minimizing the cost function (9) over a shared communication network as, e.g., a wireless network, to the S sensors. This transmission is typically affected by time delays and thus, it is obviously likely that the received data is not optimal with respect to the arrival time.

In the following, we present a method that extends the sequence-based control approach by incorporating also the potential effects of previously sent sequences in the control decision.

III. Sequence-Based Control

Since our approach is based on the general concept of sequence-based control as, e.g., presented in [8], [10], [13], [14], [15], we briefly introduce this well-established concept for NCS in this section.

In sequence-based control, an open-loop feedback MPC is employed, which calculates the current optimal input $\underline{u}_k^*$ if no further information are available in future time steps [16]. More precisely, the procedure of an open-loop feedback controller is as follows:

1) Based on all available information, i.e., the received measurements and the previously applied control inputs, the current state $\underline{\boldsymbol{x}}_k$ is estimated.
2) Subsequently, the control sequence $\underline{u}_{k:k+N-1}$ is determined that minimizes the expected cumulative costs

$$\min_{\underline{u}_{k:k+N-1}} E\left\{ g_{k+N}(\underline{\boldsymbol{x}}_{k+N}) + \sum_{i=k}^{k+N-1} g_i(\underline{\boldsymbol{x}}_i, \underline{u}_i) \right\} \tag{10}$$

subject to

$$\begin{aligned} &\underline{\boldsymbol{x}}_{i+1} = \underline{a}_i\left(\underline{\boldsymbol{x}}_i, \underline{u}_i, \underline{\boldsymbol{w}}_i\right) && \text{and} \\ &\underline{u}_i \in \mathcal{U}_i && \text{with} \\ &i = k, k+1, \ldots, k+N-1 . \end{aligned}$$

3) Finally, the first control input $\underline{u}_k^*$ of the optimal sequence is applied to the system.

The identifier $\underline{\boldsymbol{x}}_i \sim f(\underline{x}_i)$ denotes the propagated system states over time up to time i starting with $\underline{\boldsymbol{x}}_k$. These extrapolated system states are calculated by means of the Chapman-Kolmogorov equation [17]

$$f(\underline{x}_{i+1}) = \int f_{\underline{u}_i}^T(\underline{x}_{i+1} | \underline{x}_i) f(\underline{x}_i) d\underline{x}_i , \tag{11}$$

where $f_{\underline{u}_i}^T(\cdot|\cdot)$ denotes the transition density, which is defined by

$$\begin{aligned} &f_{\underline{u}_i}^T(\underline{x}_{i+1} | \underline{x}_i) = \\ &\quad \int \delta(\underline{x}_{i+1} - \underline{a}_i(\underline{x}_i, \underline{u}_i, \underline{w}_i)) \cdot f(\underline{w}_i) \; d\underline{w}_i . \end{aligned} \tag{12}$$

The function $g_i(\cdot)$ in (10) represents the application-specific per-stage cost function that assigns each state sequence $\underline{\boldsymbol{x}}_{k:k+N}$ and its corresponding input sequence $\underline{u}_{k:k+N-1}$ to a real number. In doing so, the desired dynamic behavior of the system can be modelled by assigning low costs to preferred state trajectories and control inputs sequences, whereas undesired states and inputs induce high costs.

Sequence-based control techniques do not only transmit the current optimal input $\underline{u}_k^*$, but the whole input sequence $\underline{u}_{k:k+N-1}$, which is optimal with respect to (10) based on the available information. This sequence is calculated anyway in the optimization process and due to assumption 3, the transmission of more data volume is not more expensive.

The receiver is equipped with a buffer, in which it stores the sequence with the highest time stamp among all received packets. Thus, a received packet is neglected, if its time stamp is lower than the packet already hold in the buffer. Otherwise,

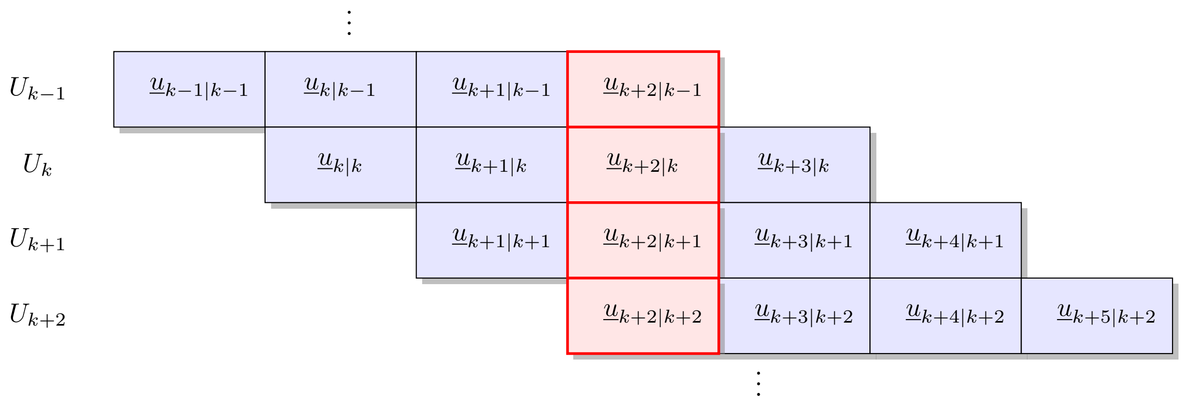

Fig. 3. Schematic illustration of the transmitted packets in a sequence-based control method. Control inputs corresponding to the same time step are vertically aligned. For example, the potential control inputs that could be applied by the actuator at time step $k+2$ are indicated by red rectangles.

the buffer is overwritten with the received packet. Finally, in every time step a selection logic at the receiver side selects the appropriate input of the buffered sequence, i.e., that input of the sequence that corresponds to the current time step.

For the following, we need some further notations. The identifier U_k denotes the input sequence generated by the controller at time k. An entry of that packet is denoted by $\underline{u}_{k+m|k}$ with $m \in \{0, 1, ..., N-1\}$, where the first part of the subindex (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. Thus, the packet U_k contains the entries

$$U_k = [\underline{u}_{k|k}, \underline{u}_{k+1|k}, \dots, \underline{u}_{k+N-1|k}] \ . \tag{13}$$

For the description of the selection logic, let us assume that the packet U_k is successfully transmitted at time step $k+\tau_k$ with $\tau_k \in \mathbb{N}$. If none of the packets

$$U_{k+1}, U_{k+2}, \dots, U_{k+\tau_k} \tag{14}$$

has been successfully transmitted so far, then the buffer is overwritten with the entries of U_k and $\underline{u}_{k+\tau_k|k}$ is applied to the plant. Otherwise, if the actuator has received any packet from (14) until time step $k+\tau_k$, say, e.g., U_{k+i}, for $i = 1, \dots \tau_k$, U_k is neglected and $\underline{u}_{k+\tau_k|k+i}$ of the buffered sequence U_{k+i} is applied.

Example System: Sensor Scheduling
Applying sequence-based control, the sensor manager does not broadcast a single sensor configuration $u_k \in \{1, \dots, S\}$ at time step k, but a whole sequence $u_{k:k+N-1}$ of sensor configurations. This sequence contains the information, which sensor should be switched on based on the current information available for the controller.

The sensors has to be equipped with buffers, in which the successfully transmitted sequences with the highest time stamps are stored. At each time step, each sensor takes the appropriate sensor configuration from its buffer and takes a measurement if and only if this input equals its identifier. Note that due to the probabilistic nature of the delays, it cannot be guaranteed that only a single sensor is activated at each time step.

It is obvious that the actually applied inputs depend on the packet delays 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 Incorporation of Buffer Content

It should be obvious from the preceding section that in sequence-based control, control inputs from packets sent in previous time steps may affect the future development of the plant. Therefore, it seems only reasonable to take all relevant inputs explicitly into account, when calculating new inputs what is also the main idea of the proposed approach. Therefore, the controller is also equipped with a buffer, in which the most recently transmitted control sequences are stored.

In the following, we first derive a stochastic description of the unknown control inputs, named virtual control inputs. Then, utilizing the concept of VCI, we design the optimal open-loop feedback controller.

A. Virtual Control Inputs

Due to the time-varying delays, the controller cannot know in general all inputs actually applied. To overcome this problem, we propose to model these inputs in a stochastic way and name these inputs virtual control inputs.

Defintion 1 (Virtual Control Input)
A virtual control input $\underline{\boldsymbol{u}}^v_{k+m|k} \sim f(\underline{u}^v_{k+m|k})$ is a random variable that models the applied control input at time step $k+m$ based on the available information for the controller up to time k.

The information available for the controller at time k comprises the state estimate $\underline{\boldsymbol{x}}_k$ and the last N packets $U_{k-1}, \dots, U_{k-N}$ the controller fed into the network.

Remark 1. *For the following, it is important to distinguish*

1) *the virtual control input $\underline{\boldsymbol{u}}^v_{k+m|k}$ that summarizes the information given up to time step k about the control input possibly applied at time step $k+m$*

2) $\underline{u}^v_{k|k}$, *which is a specific realization of* $\underline{\boldsymbol{u}}^v_{k|k}$, *and*
3) $\underline{u}_{k+m|k}$, *which denotes the entry of the packet* U_k *for time step* $k+m$.

To derive $f(\underline{u}^v_{k+m|k})$, we note that for a specific time step, there is only a finite set of deterministic control inputs that, based on the available information to the controller, possibly could be applied by the actuator. The stochastic formulation for a finite set of deterministic values that are chosen with a certain probability is given by a Dirac mixture density. It holds

$$f(\underline{u}^v_{k+m|k}) = \sum_{i=0}^{N-m} \alpha_{i,m}\, \delta(\underline{u}^v_{k+m|k} - \underline{u}_{k+m|k-i}),$$

with

$$m \in \{0, 1, \cdots, N-1\}\ , \qquad \sum_{i=0}^{N-m} \alpha_{i,m} = 1\ ,$$

where the weighting factors $\alpha_{i,m}$ represent the probability that the respective control input is applied. The weighting factors can be calculated based on the probability p_i of the event that the packet that was generated $i \in \mathbb{N}$ time steps ago is transmitted successfully in this time step.

In the following, we state the equations for the first two groups $\alpha_{i,0}$ and $\alpha_{i,1}$ where we used $\bar{p}_i = 1 - p_i$ to denote the complementary event of p_i: Then the weighting factors can be calculated by

$$\begin{aligned}
\alpha_{0,0} &= p_0 \\
\alpha_{1,0} &= \bar{p}_0(p_0 + p_1) \\
\alpha_{2,0} &= (\bar{p}_0)^2 \bar{p}_1 (p_0 + p_1 + p_2) \\
&\vdots \\
\alpha_{m,0} &= \left(\sum_{i=0}^{m} p_i\right) \prod_{j=0}^{m-1} (\bar{p}_j)^{m-j} \\
\alpha_{N+1,0} &= 1 - \sum_{i=0}^{N} \alpha_{i,0}
\end{aligned}$$

$$\begin{aligned}
\alpha_{0,1} &= p_0 + p_1 \\
\alpha_{1,1} &= \bar{p}_0 \bar{p}_1 (p_0 + p_1 + p_2) \\
\alpha_{2,1} &= (\bar{p}_0)^2 (\bar{p}_1)^2 \bar{p}_2 (p_0 + p_1 + p_2 + p_3) \\
&\vdots \\
\alpha_{m,1} &= \sum_{i=0}^{m+1} p_i (\bar{p}_0)^m \prod_{j=1}^{m} (\bar{p}_j)^{m+1-j} \text{ for } m \leq N-1 \\
\alpha_{N,1} &= 1 - \sum_{i=0}^{N-1} \alpha_{i,1}\ .
\end{aligned}$$

B. Determination of the Optimal Open-Loop Input Sequence

In this section, we describe, how the optimal open-loop input sequence is determined.

If we consider the example depicted in Fig. 3 and assume that the controller is currently in time step $k+2$, the controller has to determine the optimal sequence $\underline{u}_{k+2:k+5}$ of packet U_{k+2}. But due to the unknown network-induced disturbances and the unavailable acknowledgements, the controller cannot know, which packets are stored in the buffer of the actuator in the current and in the future time steps. Consequently, the inputs actually applied in future time steps are unknown. Therefore, the classical open-loop feedback optimization problem as introduced in (5) seems not be suitable, since the future system behavior is also potentially influenced by the entries of the previously transmitted packets. Thus, a deterministic modelling of the inputs does not reflect adequately the real-world situation at the receiver side.

It is more realistic to employ a stochastic model of the inputs, the VCI, as introduced in the previous section. If the controller is in time step $k+2$, it cannot change the entries of the packets $U_{k-3}, \ldots, U_{k+1}$ of course, since these packets are already sent over the network. But the effects of these inputs on the system development can be taken into account, when calculating the entries of packet U_{k+2}. In doing so, the entries of U_{k+2} should be chosen in such way that the expected cost is minimized.

This can be realized by extending the cumulative cost of (10) to stochastic inputs, i.e.,

$$\min_{\underline{u}_{k:k+N-1|k}} E\left\{ g_{k+N|k}(\underline{\boldsymbol{x}}_{k+N|k}) + \sum_{i=k}^{k+N-1} g_{i|k}(\underline{\boldsymbol{x}}_{i|k}, \underline{\boldsymbol{u}}^v_{i|k}) \right\} \tag{15}$$

subject to

$$\begin{aligned}
&\underline{\boldsymbol{x}}_{i+1|k} = \underline{a}_i\left(\underline{\boldsymbol{x}}_{i|k}, \underline{\boldsymbol{u}}^v_{i|k}, \underline{\boldsymbol{w}}_i\right) && \text{and} \\
&\underline{u}^v_{i|k} \in \mathcal{U}_i && \text{with} \\
&i = k, k+1, \ldots, k+N-1\ .
\end{aligned}$$

Please note that the cumulative costs are still minimized over *deterministic* input sequences. But in contrast to (5), the sequence $\underline{u}_{k:k+N-1|k}$ influences not only the future system states $\underline{\boldsymbol{x}}_{k:k+N|k}$, but also the probability distributions of the VCI $\underline{\boldsymbol{u}}_{k:k+N-1|k}$ by adding further components to the Dirac mixture densities. Furthermore, it is worth mentioning that the expected value is not only taken over $\underline{\boldsymbol{x}}_{k:k+N|k}$, but also over $\underline{\boldsymbol{u}}_{k+N-1|k}$.

Due to the fact that we consider systems with discrete-valued control inputs, the optimization problem (15) can be represented by a decision tree. In particular, there are finitely many different realization of $\underline{u}_{k:k+N-1|k}$ and for each realization, the corresponding future virtual control inputs and the cumulative reward (15) can be calculated. Finally, the sequence that minimizes the cumulative cost function is selected and transmitted over the network. Please note that techniques for complexity reduction such as Branch-and-Bound are still applicable, even if the inputs are modelled by a random variable.

Example System: Sensor Scheduling
Transferring the concept of virtual control inputs to our example application means that the sensor manager now models the applied sensor configuration by random variables $\boldsymbol{u}^v_k$. Thus, the per-stage cost function (9) depends not on a deterministic

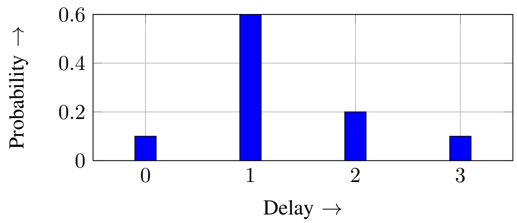

Fig. 4. Probability density function over time delays used in the simulations.

	SB-VCI	SB
Averaged Cumulated Costs	12,12	14,6

TABLE I
CUMULATED TRACES OF $\mathbf{C}_k^e$ FOR EACH RUN AVERAGED OVER THE 20 MONTE CARLO SIMULATION RUNS FOR THE STANDARD SEQUENCE-BASED CONTROL APPROACH (SB), AND THE PROPOSED TECHNIQUE WITH VIRTUAL CONTROL INPUTS (SB-VCI)

sensor configuration $\underline{u}_k$, but on a random quantity. Hence, the covariance of the posterior covariance matrix of each future step is a random matrix $\mathbf{C}_k^e(\underline{\boldsymbol{u}}_k^v)$ and the objective of the optimization is to minimize the expected costs, such as the expected determinant or the expected trace.

V. EVALUATION

A. Setup

In order to attain quantitative statements, we conducted 20 Monte Carlo simulation runs with the example application described in the example sections. The sampling interval is $T = 1$ seconds and the scalar diffusion strength is set to $q = 0.1$. The sensor network consists of six distance sensors, whose positions are drawn randomly from $\{[-15, 15] \times [-15, 15]\}$ with a uniform distribution. We assume that the sensor noise is zero-mean and normally distributed with the individual standard deviations $C_k^{v,1} = C_k^{v,3} = C_k^{v,6} = 0.1$ and $C_k^{v,2} = C_k^{v,4} = C_k^{v,5} = 0.5$.

The per-stage cost function $g(\cdot)$ in (10) is realized by the trace of the expected posterior covariance matrix

$$g_k(\underline{\boldsymbol{x}}_k, u_k) = \text{trace}(\mathbf{C}_k^e(u_k)) \ .$$

By minimizing the trace, the expected squared estimation error [6] is minimized.

In each run, the initial state is

$$x_0 = [1, 1, 1, 1]^T$$

and each Monte Carlo run consists of 16 steps. Furthermore, the open-loop feedback controller uses a horizon of length $N = 5$ and the nonlinear measurement equation is linearized in every step. The employed static probabilistic description of the transmission characteristics mentioned in assumption 1 is depicted in Fig. 4.

B. Evaluation Results

We compare our approach to the standard sequence-based control method described in [8] or [9]. Here, the entries of the packet U_k depends only on the current system state $\underline{\boldsymbol{x}}_k$ and future control inputs are modelled deterministically as in (10).

Exemplary resulting trajectories of the two control methods can be seen in Fig. 5. The estimated trajectory of the proposed methods using the concept of VCI is very similar to the real trajectory of the mobile object, whereas the the one of the standard sequence-based controller differs more.

For comparing the results of the 20 Monte Carlo simulation runs, we consider the average of the cumulated real traces of the covariance matrix over all 20 runs, i.e.,

$$\sum_{i=1}^{20} \frac{1}{20} \cdot \sum_{j=1}^{16} \text{trace}(\mathbf{C}_{i,j}^e) \ ,$$

where $\mathbf{C}_{i,j}^e$ denotes the posterior covariance matrix of step j in run i. The results are shown in Table I.

VI. CONCLUSIONS & FUTURE WORK

In this paper, we presented a technique, how to extend sequence-based control in such a way that entries of previously transmitted packets, which can potentially influence the future system behavior, are also considered in the control decision. The key idea of the paper is to model the unknown control inputs by random variables characterized by probability density functions, which can be derived from previously transmitted packets. With respect to this probabilistic description, the optimal open-loop sequence is derived.

The proposed control technique can be applied in control applications, in which time delays in the transmission of the optimal input are significant in relation to the sampling interval.

Future work will be concerned with incorporation of further information into the control decision. For example, promising aspects might be the incorporation of

1) the likelihoods of the individual control inputs in order to estimate more precisely, which control input might have been applied by the actuator and
2) acknowledgements of successfully transmitted packets allowing to reduce the components of the virtual control inputs.

REFERENCES

[1] T. Yang, "Networked Control System: a Brief Survey," *Control Theory and Applications, IEEE Proceedings -*, 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] A. Bemporad, M. Heemels, and M. Johansson, *Networked Control Systems*. Springer-Verlag New York Inc, 2010, vol. 406.

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

[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," *Automatic Control, IEEE Transactions on*, vol. 55, no. 8, pp. 1781–1796, 2010.

[6] M. Huber, *Probabilistic Framework for Sensor Management.* KIT Scientific Publishing, 2009, vol. 7.

[7] J. Williams, "Information Theoretic Sensor Management," Ph.D. dissertation, Massachusetts Institute of Technology, 2007.

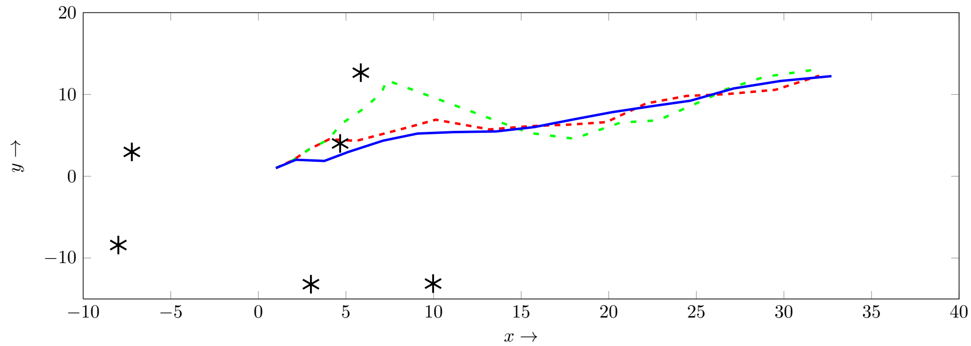

Fig. 5. Example trajectories of a single Monte Carlo simulation run. The real trajectory of the mobile object is depicted with a solid blue line (——), the estimated trajectory of the proposed approach with a dashed red line (- - -) and of the sequence-based approach without consideration of the buffer content with the loosely dashed green line (- -). The positions of the six distance sensors are indicated with the black asteriks.

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

[9] D. Quevedo, E. Silva, and G. Goodwin, "Packetized Predictive Control over Erasure Channels," in *American Control Conference, 2007. ACC'07*. IEEE, 2007, pp. 1003–1008.

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

[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," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 4, pp. 672–685, 2008.

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

[14] 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.

[15] 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.

[16] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, Belmont, Massachusetts, 2000, vol. 1.

[17] A. Papoulis and S. Pillai, *Probability, random variables and stochastic processes with errata sheet*. McGraw Hill Higher Education, 2002.