4.5 Stochastic model-based control with virtual control inputs

In Section 4.1 , a controller scheme is described where a controller that generates sequences of control inputs is used along with an actuator-buffer to compensate time delays and data losses in the network connection between actuator and controller. The method is particularly appealing in the context of MPC since a standard model-predictive controller already calculates control input sequences as a byproduct of the receding-horizon optimization. The optimized control sequence only has to be sent to the actuator instead of being discarded. A problem of this procedure is, however, that in order to generate a control sequence, the controller needs not only to know which control inputs have been applied by the actuator, but also which ones will be applied in the future. Unfortunately, in the presence of time-varying transmission delays and stochastic data losses, in the general case, the controller only has uncertain information about the applied control inputs. One way to solve this problem, in the case of bounded time delays, is presented in Section 4.2 , where different methods are proposed to ensure *prediction consistency* (see Definition 4.1). If this property holds, the controller has perfect knowledge about the content of the actuator-buffer and, thus, about the applied control inputs. However, ensuring prediction consistency comes at the price of additional time delays introduced into the system.

In this section, we choose a different design philosophy and do not try to restore the perfect information case by ensuring prediction consistency, but directly design the controller for the situation that only uncertain information about the applied control inputs is available. This requires a description of the control inputs that addresses this uncertainty. Therefore, we introduce the concept of *virtual control inputs*. Virtual control inputs characterize the potentially applied control inputs by means of probability density functions.

Based on this probabilistic description, two methods for controller design are presented. In the first method, the MPC algorithm described in Algorithm 4.1 is extended by the virtual control inputs for a nonlinear system with discrete-valued inputs. In the second method, we consider linear systems with continuous-valued inputs and extend a nominal feedback-controller, which is designed for the system without consideration of network-induced effects, so that it generates control sequences suitable to compensate time delays and data losses in the network connection.

In the following section, we describe the system setup and its components in more detail. In Section 4.5.2, the concept of virtual control inputs is described. Based on this, Sections 4.5.3 and 4.5.4 present the two control methods.

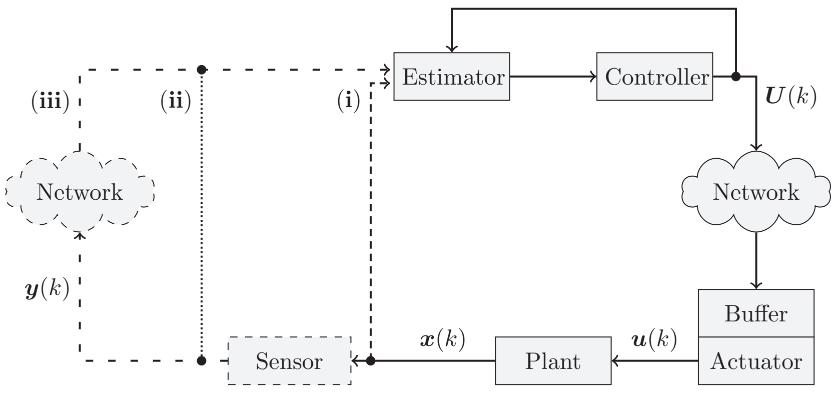

Fig. 4.1. Considered system: The controller generates a control input sequence $\boldsymbol{U}(k)$, which also contains predicted control inputs for future time steps. This sequence is transmitted in form of a data packet over a network to the actuator. The actuator holds the most recent control input sequence in a buffer and applies the time-corresponding entry of this sequence to the plant $\boldsymbol{u}(k)$. In configuration (i), the plant state $\boldsymbol{x}(k)$ is directly accessible by the estimator. In configuration (ii), the state is measured by a sensor and the measurements $\boldsymbol{y}(k)$ are available without time delay. In configuration (iii), the measurements are transmitted over a network.

4.5.1 System setup and controller-actuator scheme

In this section, we describe the considered system setup and introduce the notation needed in the next section. Furthermore, some aspects and assumptions regarding the controller-actuator scheme and the digital network are discussed.

We will consider three different system configurations as depicted in Fig. 4.1. In every configuration, there is a network between controller and actuator (CA-link). While in configuration (i), the state of the plant is known perfectly by the controller, in configuration (ii) the state of the plant is not directly accessible and, thus, measured by a sensor. Additionally, in configuration (iii) there is also a network present between sensor and controller (SE-link).

To introduce the concept of virtual control inputs, we consider the following discrete-time stochastic nonlinear plant described in state-space form via

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{w}(k), k) \ , \tag{4.1}$$

$$\boldsymbol{y}(k) = \boldsymbol{g}\left(\boldsymbol{x}(k), \boldsymbol{v}(k), k\right) \ , \tag{4.2}$$

where $\boldsymbol{x}(k) \in \mathrm{I\!R}^n$ denotes the system state at time step k, $\boldsymbol{u}(k) \in \mathrm{I\!R}^s$ the control input applied by the actuator, and $\boldsymbol{y}(k) \in \mathrm{I\!R}^q$ the measured output (if configuration (ii) or (iii) is considered). The terms $\boldsymbol{w}(k) \in \mathrm{I\!R}^n$ and $\boldsymbol{v}(k) \in \mathrm{I\!R}^q$ represent discrete-time white noise processes with probability density functions $\boldsymbol{p}^w(w(k), k)$ and $\boldsymbol{p}^v(v(k), k)$, respectively, that are independent of network-induced effects. It is assumed that the system model and the probability density functions are known by the controller. For the purpose of controller design, we consider two specializations of this setup: in Section 4.5.3, we assume that the control inputs $\boldsymbol{u}(k) \in \mathcal{U}_k$ take values from a finite discrete-valued set $\mathcal{U}(k)$ and in in Section 4.5.4 we consider a linear plant.

Controller and actuator. As described in the introduction, the overall design idea is to use a controller that not only generates a single control input for the current control cycle, but also control inputs for future N time steps (with $N \in \mathbb{N}$). The whole control input sequence is dispatched in one data packet and sent over the network to the actuator. Attached to the actuator is a buffer, where the actuator stores the control input sequence with the most recent information received. If a data packet is received that contains older information than in the buffer, the new packet is discarded. In every time step, the actuator applies the appropriate control input of the buffered sequence to the plant, i.e., the control input of the sequence that corresponds to the current time step. By doing so, in case of time delays and data losses in the CA-link, the actuator can fall back upon former calculated control inputs. It may, however, happen that the buffer runs out of applicable control inputs. In this case, the actuator applies a default control input denoted by $\boldsymbol{u}_{\mathrm{def}}(k)$ that is known to the controller.

In the following, we will refer to a control input sequence generated by the controller at time k by $\boldsymbol{U}(k)$. Entries of that packet are denoted by $\boldsymbol{u}(k+m|k)$ with $m \in \{0, 1, ..., N\}$, where the first part of the argument $(k+m)$ gives the time step for which the control input is intended to be applied to the plant. The second part of the argument (k) specifies the time step, when the control input was generated. For a packet of length $N+1$ with $N \in \mathbb{N}$, which was generated in time step k, this gives

$$\boldsymbol{U}(k) = \{\boldsymbol{u}(k|k), \boldsymbol{u}(k+1|k), \dots, \boldsymbol{u}(k+N|k)\} \ . \tag{4.3}$$

Example 4.1 *Controller-actuator procedure*

For example, let us assume the actual time step is $k = 5$ and the actuator receives the controller packet $\boldsymbol{U}(2)$. If the actuator has not received any of the packets $\boldsymbol{U}(3)$, $\boldsymbol{U}(4)$, and $\boldsymbol{U}(5)$, then the buffer is overwritten with $\boldsymbol{U}(2)$ and the 4-th entry of $\boldsymbol{U}(2)$, i.e., $\boldsymbol{u}(5|2)$, is applied to the plant. Otherwise, if the actuator has already received, for example, the sequence $\boldsymbol{U}(4)$, the received sequence $\boldsymbol{U}(2)$ is neglected and the 2nd entry of the buffered sequence $\boldsymbol{U}(4)$ is applied, i.e., $\boldsymbol{u}(5|4)$. □

Digital network. The digital network between controller and actuator (CA-link) and (if present) between sensor and estimator (SE-link) are subject to time-varying delays and stochastic data losses. By interpreting lost transmissions as transmissions with infinite time delay, we unify the description of both effects by only considering time-varying but possibly *unbounded* time delays. The time delays are described by random processes $\tau^{CA}(k) \in \mathbb{N}$ and $\tau^{SE}(k) \in \mathbb{N}$ that specify how many time steps a packet will be delayed if sent at time step k. It is assumed that $\tau^{CA}(k)$ and $\tau^{SE}(k)$ are white stationary processes that are mutually independent and independent of $\boldsymbol{v}(k)$ and $\boldsymbol{w}(k)$ and that their probability density functions $\boldsymbol{p}^{CA}(\tau^{CA})$ and $\boldsymbol{p}^{SE}(\tau^{SE})$ are known.

In addition, 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 uses a so called UDP-like protocol, i.e., the network does not provide acknowledgements for successfully transmitted data packets.

In the literature, also so called TCP-like protocols are considered, which assume that successfully transmitted data packets are instantaneously acknowledged at the sender side. In case of the CA-link, using a TCP-like protocol implies that the controller always has perfect information about the applied control inputs. It has been shown [3] that under this condition, the separation principle holds even in the sequence-based setup. Furthermore, the optimal sequence-based controller was derived in [3] and extended in [5] to the setup with multiple sensors and sequence-based information structure in the SE-link. In the following, we focus, however, on the more general case of UDP-like protocols.

4.5.2 Concept of virtual control inputs

In the literature, a widely used approach for modeling time-varying time delays (and data losses) in the CA-link is to describe network and actuator as a Markov Jump Linear System (MJLS) [2]. The MJLS description consists of a set of linear systems, where at every time step, only one system of the set is active. Each of the linear systems relates controller sequences $\boldsymbol{U}(\cdot)$ with an actuator output $\boldsymbol{u}(k)$ for a specific time delay. Which one of the possible systems is active is determined by the realization of a random process that reflects the stochastic characteristics of the time delays. Therefore, the MJLS relates *realizations* of the applied control inputs $\boldsymbol{u}(k)$ with *realizations* of the time delays. This kind of model is called a generative model. It is convenient for stability analysis and for controller synthesis in the case of TCP-like protocols.

However, in the considered case of UDP-like protocols, a probabilistic description of the applied control input $\boldsymbol{u}(k)$, which relates the *probability density function* of the time delays with the *probability density function* of the applied control input $\boldsymbol{u}(k)$, is more useful. This description is advantageous as, on the one hand, the applied control inputs are in fact stochastic and, on the other hand, the state of the plant has to be estimated when it is not directly accessible as in configurations (ii) and (iii). There, the estimation of the state is strongly coupled with the estimation of the applied control inputs so that a probabilistic description is needed for the estimator design. In the following, we introduce a probabilistic description by means of so called virtual control inputs [7].

> **Definition 4.1 (Virtual control input).** *A virtual control input $\hat{\boldsymbol{u}}(k+m|k)$ is a random variable, characterized by a probability density function $\boldsymbol{p}\left(\hat{\boldsymbol{u}}(k+m|k), k\right)$, that describes the control input $\boldsymbol{u}(k+m)$ applied to the plant at time step $k+m$ conditioned on the information available to the controller at time step k.*

The information available to the controller is denoted by the set $\mathcal{I}(k)$ and consists of the measured states $\boldsymbol{x}(0), \ldots, \boldsymbol{x}(k)$ (respectively $\boldsymbol{y}(0), \ldots, \boldsymbol{y}(k)$ in system configurations (ii) and (iii)), the last N packets $\boldsymbol{U}(k-1), \ldots, \boldsymbol{U}(k-N)$ the controller fed into the network, the model of the plant, the characteristics of the process noise and the delay distribution of the network.

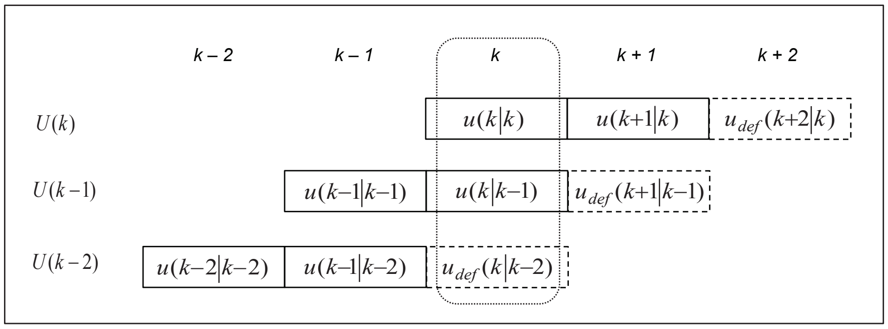

Fig. 4.2. Representation of the control input sequences $\boldsymbol{U}(k-2)$, $\boldsymbol{U}(k-1)$, and $\boldsymbol{U}(k)$, where every sequence consists of two control inputs. For the sake of completeness, the default control input $\boldsymbol{u}_{def}(\cdot)$ is added to the end of each sequence. Control inputs applicable to the same time step are vertically aligned. The rounded rectangle, e.g., marks the control inputs that could be applied at time step k.

At a certain time step, there is only a finite set of deterministic control input candidates that, based on $\mathcal{I}(k)$, could be applied by the actuator. This is illustrated in Fig. 4.2. The virtual control inputs characterize these candidates via a probability density function.

Example 4.2 *Virtual control inputs*

For example, if the controller calculates at time k the current control input $\boldsymbol{u}(k|k)$ and one predicted control input for the next time step $\boldsymbol{u}(k+1|k)$, a control sequence with two elements is sent to the actuator. Let the probabilities be 0.2 that there is no delay and 0.8 that the delay is one time step. Then, the packet has to arrive at the current time step or the next time step.

From the controller's point of view, the applied control input is uncertain, since there are always two control input candidates that are possible to be applied: at time step $k+1$ these would be $\boldsymbol{u}(k+1|k+1)$ (applied with probability 0.2), which the controller is calculating in this time step, and $\boldsymbol{u}(k+1|k)$ (applied with probability 0.8) that was calculated in the last time step. A virtual control input describes this situation by means of a probability density function, which in this case is a Dirac mixture density with components at the positions $\boldsymbol{u}(k+1|k+1)$ and $\boldsymbol{u}(k+1|k)$ with weighting factors 0.2 and 0.8, respectively. □

The probability density of a set of deterministic values that are chosen with a certain probability is given by a Dirac mixture density, so that the virtual control input can be described by

$$
\begin{aligned}
h\left(\hat{\boldsymbol{u}}\left(k+m|k\right)\right) = \xi_{N+1}(k+m|k)\cdot\delta\left[\boldsymbol{u}^r\left(k+m|k\right)-\boldsymbol{u}_{\text{def}}\left(k\right)\right]+ \\
\sum_{i=0}^{N}\xi_i(k+m|k)\cdot\delta\left[\boldsymbol{u}^r(k+m|k)-\boldsymbol{u}(k+m|k-i)\right] \qquad (4.4) \\
\text{with} \;\; m\in\{0,1,\cdots,N\}\;, \qquad \sum_{i=0}^{N+1}\xi_i(k+m|k)=1\;,
\end{aligned}
$$

where $\delta(\cdot)$ is the Dirac delta function, $\boldsymbol{u}^r(k+m|k)$ denotes the domain of the realizations of the virtual control input $\hat{\boldsymbol{u}}(k+m|k)$, and $\boldsymbol{u}_{\text{def}}$ is the known default control input in case the buffer runs empty. The weighting factors $\xi_i(k+m|k)$ express the probability that the corresponding control input $\boldsymbol{u}(k+m|k-i)$ is applied by the actuator and can be calculated by

$$
\xi_i(k+m|k) = \text{Prob}\big(\boldsymbol{u}(k+m)=\boldsymbol{u}(k+m|k-i)|\mathcal{I}(k)\big)\;.
$$

The control input $\boldsymbol{u}(k+m|k-i)$ 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, $\boldsymbol{u}(k+m|k-i)$ 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 $m+i$. In the following, we denote the age of the buffered sequence at time step k by $\theta(k)$. With this notation, it holds that

$$
\xi_i(k+m|k) = \text{Prob}\big(\theta(k+m)=i|\mathcal{I}(k)\big)\;.
$$

Therefore, the weighting factors can be interpreted as estimates of $\theta(k+m)$. It is shown in [8] that $\theta(k)$ can be formulated as the state of a Markov chain (with state space $\{0,1,2,...,N+1\}$) that is governed by the transition matrix $\boldsymbol{P}$ given by

$$\boldsymbol{P} = \begin{pmatrix} p_{0,0} & p_{0,1} & 0 & 0 & \cdots & 0 \\ p_{1,0} & p_{1,1} & p_{1,2} & 0 & \cdots & 0 \\ p_{2,0} & p_{2,1} & p_{2,2} & p_{2,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & p_{N,N+1} \\ p_{N+1,0} & p_{N+1,1} & p_{N+1,2} & p_{N+1,3} & \cdots & p_{N+1,N+1} \end{pmatrix} , \qquad (4.5)$$

$$\text{with} \quad p_{i,j} = \begin{cases} 0 & \text{for } j \geq i+2 \ , \\ 1 - \sum\limits_{r=0}^{i} q_r & \text{for } j = i+1 \ , \\ q_j & \text{for } j \leq i < N+1 \ , \\ 1 - \sum\limits_{r=0}^{N} q_r & \text{for } j = i = N+1 \ . \end{cases}$$

The term q_i (with $i \in \mathbb{N}$) describes the probability of the event that a packet is delayed by i time steps. These probabilities can be derived as the probability density function of the time delays of the network connection are known. Arranging the weighting factors $\xi_i(k+m|k)$ in form of a vector

$$\boldsymbol{\xi}(k+m|k) = \left[\xi_0(k+m|k), \quad \ldots, \quad \xi_{N+1}(k+m|k)\right]^{\mathrm{T}} \ ,$$

it holds that

$$\boldsymbol{\xi}(k+m|k) = \frac{\left[\boldsymbol{I}_{N+2-m} \ \boldsymbol{0}_{(N+2-m)\times m}\right] \left(\boldsymbol{P}^m\right)^{\mathrm{T}} \cdot \boldsymbol{\xi}(k|k)}{\boldsymbol{1}^{\mathrm{T}} \cdot \left[\boldsymbol{I}_{N+2-m} \ \boldsymbol{0}_{(N+2-m)\times m}\right] \left(\boldsymbol{P}^m\right)^{\mathrm{T}} \cdot \boldsymbol{\xi}(k|k)} \ , \qquad (4.6)$$

where $\boldsymbol{1}$ denotes a column vector of appropriate dimension with all entries equal to 1. The term in the numerator represents a m-step future prediction of $\theta(k)$, where only the first $N+2-m$ entries are kept. The denominator of (4.6) normalizes the extracted subset of the predicted vector so that $\boldsymbol{\xi}(k+m|k)$ sums up to one.

The term $\boldsymbol{\xi}(k|k)$ can be interpreted as an estimate of $\theta(k)$ arranged in vector form. If $\boldsymbol{\xi}(k|k)$ is known, the other weighting factors $\boldsymbol{\xi}(k+1|k)$ up to $\boldsymbol{\xi}(k+N|k)$ can be easily derived by means of the transition matrix $\boldsymbol{P}$ according to (4.6). In the following, we briefly survey how the weighting factors $\boldsymbol{\xi}(k|k)$, i.e., the estimate of $\theta(k)$ in vector form, can be obtained. The estimation algorithm differs depending on the considered system configuration shown in Fig. 4.1. It should be noticed that in configuration (ii) and (iii), the estimation of $\theta(k)$ is strongly coupled with the estimation of the state $\boldsymbol{x}(k)$ and both have to be treated simultaneously.

(i): Perfect State Information. If the controller has perfect knowledge about the actual state of the plant, the configuration corresponds to a *hidden Markov model* with continuous-valued outputs and the continuous-valued version of the Wonham filter can be used together with (4.6). It holds

$$\boldsymbol{\xi}(k|k) = \frac{\boldsymbol{H} \cdot \boldsymbol{\xi}(k|k-1)}{\mathbf{1}^{\mathrm{T}} \cdot \boldsymbol{H} \cdot \boldsymbol{\xi}(k|k-1)} \ , \tag{4.7}$$

$$\boldsymbol{H} = \mathrm{diag}\begin{pmatrix} \boldsymbol{p}^w\left(\boldsymbol{x}(k) - \boldsymbol{f}\left(x(k-1), \boldsymbol{u}(k-1|k-1), k\right), k\right) \\ \boldsymbol{p}^w\left(\boldsymbol{x}(k) - \boldsymbol{f}\left(x(k-1), \boldsymbol{u}(k-1|k-2), k\right), k\right) \\ \vdots \\ \boldsymbol{p}^w\left(\boldsymbol{x}(k) - \boldsymbol{f}\left(x(k-1), \boldsymbol{u}(k-1|k-N), k\right), k\right) \\ \boldsymbol{p}^w\left(\boldsymbol{x}(k) - \boldsymbol{f}\left(x(k-1), \boldsymbol{u}_{\mathrm{def}}(k), k\right), k\right) \end{pmatrix} \ , \tag{4.8}$$

where $\mathrm{diag}\left(\boldsymbol{s}\right)$ denotes a matrix with the elements of the vector $\boldsymbol{s}$ on the diagonal and zeros elsewhere.

(ii): Imperfect state information without time delays. For this case the optimal estimator for $\theta(k)$ and $\boldsymbol{x}(k)$ is obtained from a bank of Kalman filters, whose memory requirement and computational load increase exponentially with time [1]. Since this is not an applicable solution, a suboptimal filter has been proposed in [4]. The algorithm is based on the *generalized pseudo Gaussian* (GPB) and *interacting multiple model* (IMM) algorithms, where a merging technique is used to reduce the increasing complexity by combining state estimates that originate from similar hypothesis about the unknown variable θ_k. Also pruning techniques are discussed [4], such as the *B-best* and *Viterbi algorithm*, which make hard decisions about the mode history and only keep the state estimates related to the most likely hypothesis about θ_k. However, it is pointed out that if the model of the system can differ significantly from the real system, a merging strategy should be preferred. A detailed description of the mentioned algorithms can also be found in [9].

(iii): Imperfect state information with time delays. If a network is present in the SE-link, then also measurements can be delayed or lost. In [4], methods are surveyed how delayed measurements, measurements that arrive out of sequence, and measurements that arrive in bursts, can be efficiently incorporated in the estimator design.

In order to deal with packet losses in the SE-link, the sequence-based design philosophy can be extended to the SE-link by transmitting sequences that also contain measurements of former time steps. It has been shown [5] that, even in an extended scenario with multiple sensors, the information contained in a measurement sequence can be recursively compressed by the sensors without loss of information. Therefore, not the whole sequence has to be sent, but only a fraction of that data.

4.5.3 Model-predictive controller design using virtual control inputs

In this section, it is described how the concept of virtual control inputs can be used in model-predictive controller design to compensate time delays and data losses in the CA-link. We consider the system (4.1) and (4.2), where, however, the control inputs $\boldsymbol{u}(k) \in \mathcal{U}_k$ take values from a finite discrete-valued set. As shown in [6], the first step to extend the basic MPC algorithm of Algorithm 4.1 is to replace the deterministic cost function Equation 4.6 by a stochastic version, that is defined on the *expected* value of the cost. As a second step, the virtual control inputs have to be included in the optimization problem. This is done by replacing part 1 of Algorithm 4.1 by the following stochastic open-loop optimization problem with terminal time $T \in \mathbb{N}$:

$$\min_{\boldsymbol{U}(k)} \mathrm{E}\left\{ \sum_{i=k}^{k+T-1} l(\boldsymbol{x}(i), \hat{\boldsymbol{u}}(i|k)) + l(\boldsymbol{x}(k+T)) \middle| \mathcal{I}(k) \right\} \tag{4.9}$$

subject to

$$\begin{aligned} &\boldsymbol{x}(i+1) = \boldsymbol{f}\left(\boldsymbol{x}(i), \hat{\boldsymbol{u}}(i|k), \boldsymbol{w}(i), i\right) && \text{and} \\ &\boldsymbol{u}(i|k) \in \mathcal{U}(i) && \text{with} \\ &i = k, k+1, \ldots, k+T \;. \end{aligned}$$

Note that the cumulative cost is still minimized over *deterministic* input sequences.

4.5.4 Model-based extension of feedback-controllers using virtual control inputs

This section is based on [7] and describes how the concept of virtual control inputs can be used to extend a given feedback-controller, so that it generates control input sequences suitable to compensate network-induced effects in the CA-link.

We consider a discrete-time *linear* plant with *continuous-valued* inputs described by

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{w}(k) \;, \tag{4.10}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{v}(k) \;, \tag{4.11}$$

with $\boldsymbol{u}(k) \in \mathbb{R}^q$ and the noise processes $\boldsymbol{w}(k) \in \mathbb{R}^n$ and $\boldsymbol{v}(k) \in \mathbb{R}^q$ are supposed to be stationary and zero mean.

We assume that a linear feedback-controller of the form

$$\boldsymbol{u}(k) = \boldsymbol{K}\boldsymbol{x}(k) \tag{4.12}$$

is given, where the feedback matrix $\boldsymbol{K}$ was designed for the plant (4.10) without consideration of network effects, e.g., by pole placement, LQR, H_2 or H_∞ methods. In the following, we use the feedback matrix $\boldsymbol{K}$ to generate control input sequences based on the predicted future states of the plant, which are influenced by virtual control inputs.

Based on the estimated state $\mathrm{E}\{\boldsymbol{x}(k)\}$, which corresponds to $\boldsymbol{x}(k)$ in configuration (i), the entries of the control input sequence $\boldsymbol{U}(k)$ are calculated by

$$\boldsymbol{u}(k|k) = \boldsymbol{K} \cdot \mathrm{E}\{\boldsymbol{x}(k)\} \ , \tag{4.13}$$

$$\boldsymbol{u}(k+1|k) = \boldsymbol{K} \cdot \mathrm{E}\{\boldsymbol{x}(k+1|k)\} \ , \tag{4.14}$$

$$\vdots$$

$$\boldsymbol{u}(k+N|k) = \boldsymbol{K} \cdot \mathrm{E}\{\boldsymbol{x}(k+N|k)\} \ , \tag{4.15}$$

where $\boldsymbol{x}(k+m|k)$ denotes the prediction of state $\boldsymbol{x}(k+m)$ based on an estimate of $\boldsymbol{x}(k)$ and the virtual control inputs $\hat{\boldsymbol{u}}(k|k), \ldots, \hat{\boldsymbol{u}}(k+m|k)$. The state predictions are random with respect to the process noise and the virtual control inputs and can be calculated for time step $k+1$ by

$$\begin{aligned}
\mathrm{E}\{\boldsymbol{x}(k+1|k)\} &= \mathop{\mathrm{E}}_{w,u}\{\boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{w}(k)\} \\
&= \boldsymbol{A} \cdot \mathrm{E}\{\boldsymbol{x}(k)\} + \boldsymbol{B} \cdot \mathrm{E}\{\hat{\boldsymbol{u}}(k|k)\} + \mathrm{E}\{\boldsymbol{w}(k)\} \\
&= \boldsymbol{A} \cdot \mathrm{E}\{\boldsymbol{x}(k)\} + \boldsymbol{B} \cdot \left(\sum_{i=0}^{N} \xi_i(k|k) \cdot \boldsymbol{u}(k|k-i) + \xi_{N+1}(k|k) \cdot \boldsymbol{u}_{\mathrm{def}}(k) \right) ,
\end{aligned}$$

and, in general, for time steps $k+m$ by

$$\begin{aligned}
\mathrm{E}\{\boldsymbol{x}(k+m|k)\} &= \mathop{\mathrm{E}}_{w,u}\{\boldsymbol{A}\boldsymbol{x}(k+m-1|k) + \boldsymbol{B}\boldsymbol{u}(k+m-1) + \boldsymbol{w}(k+m-1)\} \\
&= \boldsymbol{A} \cdot \mathop{\mathrm{E}}_{w,u}\{\boldsymbol{x}(k+m-1|k)\} + \boldsymbol{B} \cdot \mathrm{E}\{\hat{\boldsymbol{u}}(k+m-1|k)\} + \mathrm{E}\{\boldsymbol{w}(k)\} \\
&= \boldsymbol{A}^m \cdot \mathrm{E}\{\boldsymbol{x}(k)\} + \sum_{j=0}^{m-1} \sum_{i=0}^{N-j} \boldsymbol{A}^{m-j-1}\boldsymbol{B} \cdot (\xi_i(k+j|k) \cdot \boldsymbol{u}(k+j|k-i) + \\
&\qquad \xi_{N-j+1}(k+j|k) \cdot \boldsymbol{u}_{\mathrm{def}}(k+j|k)) \, .
\end{aligned} \tag{4.16}$$

For taking the expected value, we have used the assumption that $\boldsymbol{w}(k)$ is zero mean and independent of $\boldsymbol{x}(k)$ and $\hat{\boldsymbol{u}}(k|k)$. Furthermore, we have used that given the available information to the controller at time step k, it holds for the expected value of the future control inputs

$$\begin{aligned}
\mathrm{E}\left\{\hat{\boldsymbol{u}}(k+m|k)\right\} &= \int_{-\infty}^{\infty} \boldsymbol{u}^r(k+m|k) h\left(\hat{\boldsymbol{u}}(k+m|k)\right) d\boldsymbol{u}^r(k+m|k) \\
&= \int_{-\infty}^{\infty} \boldsymbol{u}^r(k+m|k) \left(\xi_{N+1}(k+m|k) \cdot \delta\left[\boldsymbol{u}^r\left(k+m|k\right) - \boldsymbol{u}_{\mathrm{def}}\left(k\right)\right] + \right. \\
&\qquad \left. \sum_{i=0}^{N} \xi_i(k+m|k) \cdot \delta\left[\boldsymbol{u}^r(k+m|k) - \boldsymbol{u}(k+m|k-i)\right]\right) d\boldsymbol{u}^r(k+m|k) \\
&= \sum_{i=0}^{N-m} \xi_i(k+m|k) \cdot \boldsymbol{u}(k+m|k-i) + \xi_{N-m+1}(k+m|k) \cdot \boldsymbol{u}_{\mathrm{def}} \ .
\end{aligned} \tag{4.17}$$

The extended controller given by (4.13), (4.14), (4.15), and (4.16) is nonlinear, because the weighting factors $\boldsymbol{\xi}(k|k)$ depend on the measured state as described in Section 4.5.2. To reduce the complexity of this controller, the weighting factors $\boldsymbol{\xi}(k|k)$ can be approximated by its stationary probability solution, that can be computed by the equilibrium equation

$$\boldsymbol{\xi}_\infty = \boldsymbol{P}^{\mathrm{T}} \cdot \boldsymbol{\xi}_\infty \ , \tag{4.18}$$

which always has a unique solution according to Markov chain theory. With this approximation and considering configuration (i), the resulting controller is linear in the measured state $\boldsymbol{x}(k)$ and in the control input sequences $\boldsymbol{U}(k), \ldots, \boldsymbol{U}(k-N-1)$, what follows from (4.15), (4.16), and (4.18). Therefore, the approximated controller can be formulated as a linear feedback controller

$$\boldsymbol{U}(k) = \widetilde{\boldsymbol{K}} \cdot \boldsymbol{\eta}(k) \ ,$$

working on the augmented state

$$\boldsymbol{\eta}(k) = \left[\boldsymbol{x}^{\mathrm{T}}(k), \boldsymbol{U}^{\mathrm{T}}(k), \cdots, \boldsymbol{U}^{\mathrm{T}}(k-N-1)\right]^{\mathrm{T}} .$$

The approximated extended controller is easy to implement and, in particular, suitable for scenarios where calculation capacity is strongly limited. The stability properties of this controller have been investigated in [7], where a criterion for mean square stability of the closed-loop system was derived.

Summary. We pointed out that the controller only has uncertain information about the applied control inputs available, when NCS with time-varying transmission delays and stochastic data losses are considered. While most techniques try to reestablish the perfect information about the applied control inputs by special measures, we proposed a method to directly incorporate the uncertainties into the controller design. This method was derived by formulating the uncertain control inputs as random processes described by sets of discrete probability density functions, called virtual control inputs. Finally, we showed that the concept of the virtual control inputs cannot only be used for controller design, but also for extending a given standard feedback controller to the NCS situation.

References

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

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

[3] J. Fischer, A. Hekler, M. Dolgov, and U. D. Hanebeck. Optimal Sequence-Based LQG Control over TCP-like Networks Subject to Random Transmission Delays and Packet Losses. *Arxiv preprint, arXiv:1211.3020*, 2012.

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

[5] J. Fischer, M. Reinhardt, and U. D. Hanebeck. Optimal Sequence-Based Control and Estimation of Networked Linear Systems. *Arxiv preprint, arXiv:1211.5086*, 2012.

[6] A. Hekler, J. Fischer, and U. D. Hanebeck. Control over unreliable networks based on control input densities. In *Proceedings of the 15th International Conference on Information Fusion*, Singapore, 2012.

[7] A. Hekler, J. Fischer, and U. D. Hanebeck. Sequence-Based Control for Networked Control Systems Based on Virtual Control Inputs. In *Proceedings of the Conference on Decision and Control CDC*, 2012.

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

[9] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target TrackingPart V: Multiple-Model Methods. In *Conference on Signal and Data Processing of Small Targets*, volume 4473, pages 559–581, 2003.