

Delay-aware Model Predictive Control for Fast Frequency Control

Tobias Heins^{*}, René Glebke[§], Mirko Stoffers[§], Sriram Gurumurthy^{*}, Jan Heesemann[§],
Martina Josevski^{*}, Antonello Monti^{*‡}, Klaus Wehrle[§]

^{*} Institute for Automation of Complex Power Systems, RWTH Aachen University, Aachen, Germany

[‡] Fraunhofer FIT, Aachen, Germany

Email: {tobias.heins, sgurumurthy, martina.josevski, amonti}@eonerc.rwth-aachen.de

[§] Chair of Communication and Distributed Systems, RWTH Aachen University, Aachen, Germany

Email: {glebke, stoffers, heesemann, wehrle}@comsys.rwth-aachen.de

Abstract—In secondary frequency control of power systems, measurement data, state information, and control signals must be transmitted over communication networks, where network delays and packet loss may occur. This paper presents a delay- and loss-aware model predictive controller (MPC) for fast frequency control in future power systems. The networked control strategy consists of a time-varying Kalman Filter with a buffer, state augmentation of the prediction model, and an actuator function, hence making the controller delay- and loss-aware compared to traditional delay-tolerant or robust control designs. We present a co-simulation setup of a power system and a communication network simulator, where we demonstrate the networked control strategy in a realistic environment. The performance of the proposed control strategy is compared against a Smith Predictor-based PI controller in a microgrid and a transmission system, each supplied by grid-forming inverters, for varying communication network conditions, including different communication technologies and network traffic.

Index Terms—Delay-aware, loss-aware, Model Predictive Control, Secondary Frequency Control, Co-simulation, Kalman Filter

I. INTRODUCTION

FREQUENCY dynamics in AC power systems have been fully governed by synchronous machines in the past, however, the introduction of fast-actuating inverter-interfaced renewable energy sources now accelerate the frequency dynamics, requiring control systems with higher bandwidths than in the past.

The secondary level of the hierarchical frequency control scheme receives measurement signals and controls the active power set-points of generation units placed at different locations in the power system [1]. It is therefore relying on a communication infrastructure. In such networked control systems, time delays are critical for the performance and could lead to instability of the control loop [2]. Additionally, communication is subject to interrupts and packet loss in overloaded networks. For high bandwidth controllers, the impact of communication delays is more pronounced, hence it is important that control systems are aware of or at least robust against these effects.

Research on secondary frequency control is traditionally focused on PI-based solutions, however model predictive control

(MPC) approaches have gained attention in recent years, due to the MPC predicting and optimizing future system behavior, the ability to handle system constraints and being inherently designed for multi-input multi-output (MIMO) systems [3]. In previous work [4], we propose a centralized MPC for frequency control in inverter-based power systems. However, that work assumes a delay-free communication between the central controller and the local inverters and declares the investigation of time delays to be future work. The design of an MPC for frequency control in future power systems that is aware of the communication network and the demonstration in a co-simulation is thus the matter of this paper.

Many MPC schemes for secondary frequency control have been proposed in literature that neglect communication delay. Yet, the control loop could become unstable in a realistic environment. In [5], [6], the aggregated communication delay is modelled only as a simple transfer function on the input path. A delay-aware MPC was proposed in [5], which was designed for a low bandwidth to decouple from the primary dynamics and only the dynamic behavior of the measuring phase-locked-loop (PLL) was modelled in the MPC. A linear matrix inequality-based robust MPC was designed in [6]. For each area of the power system, the synchronous generation is lumped into an equivalent and only synchronous generation was considered. In [7], communication channels are modelled, which transmit distributed measurements. The MPC is augmented by a constant disturbance model to estimate the disturbance through communication delay and model errors using an observer. The robustness to measurement noise, constant time delays and communication loss was assessed. Yet, only a single inverter-based resource (IBR) was modelled and the control of multiple IBRs considered future work.

The issue of time delays is further studied for damping wide-area oscillations in power systems and the importance of detailed modelling of delays is shown in [8]. A wide-area power system stabilizer that includes time delays is designed in [9] as an LQR using a first-order Padé approximation and an augmented state-space model of the system based on a fixed delay value. The robustness of the controller to nonuniform and varying delays, loss of communication, and topology changes in the communication system was tested.

The effects of communication networks are often modelled as constant and uniform delays across all communication channels and the controllers have a delay-robust design rather than considering delays explicitly in the design. The simulation models often include only constant input delays and communication topologies are not modelled. However, the integration of information and communication technology (ICT) for monitoring and control of power systems with a large number of distributed energy resources (DERs) is considered as a key enabler for the modern smart grid [10]. The performance of the power system thus depends on the performance of the communication system and the interdependence and mutual impact is critical to be included into numerical simulations.

In this paper we address these shortcomings and design a delay- and loss-aware MPC for secondary frequency control in inverter-based power systems. The control design addresses both measurement and actuator delay and considers time-varying delays, individually for each communication path. The delay-aware control scheme enhances the previously proposed MPC in [4] with a time-varying Kalman Filter with a buffer, state augmentation of the prediction model, and an actuator function. The proposed controller is evaluated in a co-simulation of a power system and a communication network simulator to achieve a detailed representation of both specialised systems. We assess the control performance for a microgrid and a transmission system test case. For the communication network, we model three different scenarios, fiber, DSL, and wireless, with increasing the amount of background traffic or wireless communication distance.

II. POWER SYSTEM MODELLING AND MPC SCHEME

The power systems in this paper are supplied purely by inverter-based resources (IBRs), thus representing an extreme scenario of low-inertia systems with primary frequency dynamics below 1 s. The model of a VSC-based IBR including an output LCL filter and the local control structure is shown in Fig. 1. The primary control is implemented as a grid-forming droop control with a first order filter, which provides voltage and frequency references to the inner voltage and current loops. The inner loops use PI-controllers and include feed-forward cross-coupling terms in the dq -domain.

In this work, we adopt our centralized MPC for inverter-based power systems from [4]. The MPC adjusts the active power set-point P_i^* of each IBR $i \in \mathcal{N}_{\text{IBR}} = \{1, \dots, n_{\text{IBR}}\}$ with $u_i = \Delta P_i^*$. We represent each IBR i by its frequency and active power dynamics, with the states $x_i = [\omega_i \ P_i]^T$ in the prediction model. The detailed state equations and their derivation can be found in [4].

The MPC solves a convex optimization problem, with the cost function

$$J(\mathcal{U}_k) = Q \sum_{j=1}^{H_p} (y_{k+j|k} - y_{ref,k+j|k})^2 + R \sum_{j=0}^{H_u-1} \Delta u_{k+j|k}^2 + S \sum_{j=0}^{H_p-1} u_{k+j|k}^2, \quad (1)$$

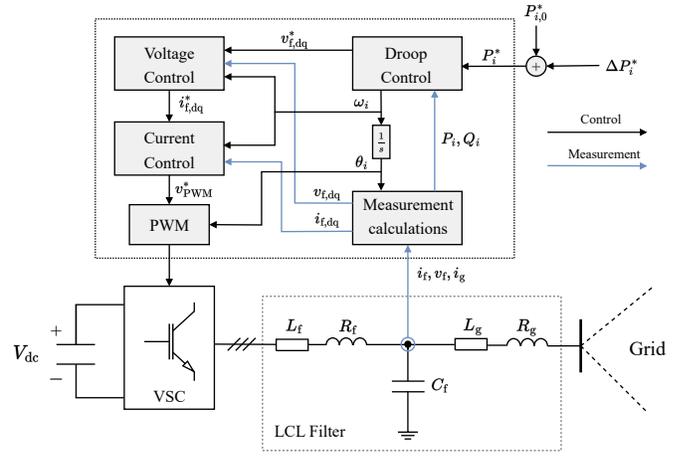


Fig. 1: Model of an IBR consisting of a voltage-sourced converter (VSC), LCL filter, and cascaded primary droop and inner control loops. The IBR receives set-points P_i^* from system level control.

where $\{\cdot\}_{k+j|k}$ denotes the prediction of a variable $\{\cdot\}$ at the future time step $k+j$ given the available information at time k , and Q , R , and S are positive scalar weights. We define the decision vector as $\mathcal{U}_k := [u_{k|k}^T \ \dots \ u_{k+H_u-1|k}^T]^T$.

The following optimization problem is solved by the MPC in a receding horizon fashion:

$$\begin{aligned} & \text{minimize} && J(\mathcal{U}_k) \\ & \text{subject to} && x_{k+1} = A_d x_k + B_d u_k + E_d z_k, \\ & && y_k = C_d x_k, \\ & && x_{k|k} = x_k, \\ & && \Omega \mathcal{U}_k \leq b(k) \end{aligned} \quad (2)$$

Here, let Ω be a matrix and $b(k)$ be a vector of appropriate size to accommodate constraints on the systems inputs, states, and outputs. In particular, we constrain the IBRs' output frequencies and rate of change of frequency (RoCoF). Additionally, the control input ΔP_i^* and second state of each IBR P_i are constrained according to the physical limitations of the IBR.

III. DELAY-AWARE CONTROL STRATEGY

MPC is an optimization-based controller that predicts future system behavior to determine the optimal control input. Its performance heavily depends on the quality of the prediction model and the availability of parameter and current state information. Time delays and loss of information can critically affect the performance of the controller. The most significant time delays occur in the communication network. Depending on the physical distance and the infrastructure, the communication to each IBR will have a different delay and loss probability. In this section, we present a strategy for a delay- and loss-aware MPC that minimizes these negative effects.

A. Measurement Delay

Each IBR sends its current state x_i as a time-stamped measurement to the central controller with a fixed sampling time. For the MPC execution, the full initial state vector $x_0 = [x_{0,1}, \dots, x_{0,n_{\text{IBR}}}]^T$ is required to carry out the prediction. As such, the current state has to be estimated from previous measurements, whereas the values with small delay have to be fused with measurements from IBRs with larger delay.

Different Kalman Filter-based approaches for state estimation with random sensor delays and missing measurements or packet loss have been proposed in literature (e.g., [11], [12]). We adopt the approach presented in [12], where a Kalman Filter is implemented with a memory buffer, storing the received measurement packets, and modify it according to our control scenario.

The buffer has $M \times N$ fields, where $M = n_{\text{IBR}}$ and

$$N = \frac{\tau_{\text{max}}}{T_s^{\text{K}}} \cdot r \in \mathbb{N}. \quad (3)$$

Measurements that are older than $t - \tau_{\text{max}}$ are discarded from the estimation, i.e., τ_{max} is the maximum admissible delay. The parameter $r \in \mathbb{N}$ describes the ratio of measurement samples per Kalman Filter execution, denoted by the sampling time T_s^{K} . Each buffer entry consists of the measurement $y_{i,k} = x_{i,k}$ at time k and an arrival variable $v_{i,k}$ defined as

$$v_{i,k} = \begin{cases} 1 & \text{if } y_{i,k} \text{ arrived before or at time } t, t \geq k \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

If measurement $y_{i,k}$ has not yet arrived at the buffer, zeros are stored in the respective slot. The Kalman Filter uses $v_{i,k}$ to distinguish between real zeros and missing information. The buffer has a moving horizon such that entries from $t - \tau_{\text{max}}$ to present time t are stored. We assume that all packets are time-stamped so that measurements are correctly ordered and entered in the buffer. Accuracy below a millisecond suffices to this end, which can be achieved by GPS and other sources [13]. The packets are sent at time t_{tx} and received at t_{rx} such that the packet delay is $\tau_{i,k} = t_{\text{rx}} - t_{\text{tx}}$. The Kalman Filter provides an estimate of the channel delay $\hat{\tau}_i$ of each IBR i , by averaging $\tau_{i,k}$ over the horizon N .

In addition to buffering the received measurements, the control inputs $u_{i,k}$ from the MPC need to be stored and aligned according to their arrival time. To determine when packets with control inputs $u_{i,k}$ have arrived, every packet has an *id* and the IBRs provide time-stamped acknowledgements and append the acknowledgement of received control signals to the measurement packet.

The Kalman Filter employs Algorithm 1. Matrices of appropriate size represent the buffered measurements (\mathcal{Y}_k), arrival variables (Υ_k), and control inputs (\mathcal{U}_k). The covariances of the Kalman Filter are denoted by $(P, Q_{\text{KF}}, R_{\text{KF}})$. In each execution, Algorithm 1 iterates through the buffer horizon N to determine the estimated state $\hat{x}_{t|k}$. For measurements that have not arrived yet, Υ_k disables the correction step. We point out that for one iteration step, measurements of only a subset of IBRs could have arrived. In this case, $\Upsilon_{j|k}$ consists of both

Algorithm 1 Kalman Filter with a buffer. The Kalman Filter iterates through the buffer horizon N , applying the prediction step. The correction step is only executed for measurements that have arrived, determined by arrival variables Υ_k . See [12].

Require: $\mathcal{Y}_k, \Upsilon_k, \mathcal{U}_k, \hat{x}_{r|k-1}, P_{r|k-1}, A_d, B_d, C_d$

```

 $\hat{x}_{0|k} \leftarrow \hat{x}_{r|k-1}$ 
 $P_{0|k} \leftarrow P_{r|k-1}$ 
for  $j = 1, 2, \dots, N$  do
   $\hat{x}_{j|k}^- = A_d \hat{x}_{j-1|k} + B_d \mathcal{U}_{j|k}$ 
   $P_{j|k}^- = A_d P_{j-1|k} A_d^T + Q_{\text{KF}}$ 
   $K_j = P_{j|k}^- C_d^T (C_d P_{j|k}^- C_d^T + R_{\text{KF}})^{-1}$ 
   $\hat{x}_{j|k} = \hat{x}_{j|k}^- + \Upsilon_{j|k}^T K_j (\mathcal{Y}_{j|k} - C_d \hat{x}_{j|k}^-)$ 
   $P_{j|k} = (I - \Upsilon_{j|k}^T K_j C_d) P_{j|k}^-$ 
  if  $j = r$  then
     $\hat{x}_{r|k} \leftarrow \hat{x}_{j|k}$ 
     $P_{r|k} \leftarrow P_{j|k}$ 
  end if
end for
return  $\hat{x}_{N|k}$ 

```

ones and zeros so that all available measurement data is used for the correction of the estimate.

B. Actuator Delay

While the Kalman Filter with a buffer compensates for both delayed and lost packets on the measurement path, two separate steps are implemented for the actuator path of the networked control system.

Augmentation of the Prediction Model: The effect of time delays can be modelled in the discretized system representation given by

$$x((k+1)T_s) = \Phi(T_s) x(kT_s) + \Gamma(T_s) u(kT_s), \quad (5)$$

for the undelayed case, with $\Phi(t) = e^{At}$ and $\Gamma(t) = \int_0^t e^{As} B ds$. We adopt the definitions in [14] and describe the actuator delay as

$$\tau_a = (d-1)T_s + \tau_a', \quad (6)$$

where d is an integer multiple of T_s and τ_a' is the remaining fractional of the total actuator delay τ_a . If the actuating signal is delayed, the control input switches during the sampling time and the system can be described by [14]

$$x((k+1)T_s) = \Phi(T_s) x(kT_s) + \Phi(T_s - \tau_a') \Gamma(\tau_a') u((k-d)T_s) + \Gamma(T_s - \tau_a') u((k-d+1)T_s). \quad (7)$$

For $\tau_a \leq T_s$ (and therefore $d = 1$) this results in the following state-space representation

$$\begin{bmatrix} x(k+1) \\ u(k) \end{bmatrix} = \begin{bmatrix} \Phi(T_s) & \Phi(T_s - \tau_a') \Gamma(\tau_a') \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} \Gamma(T_s - \tau_a') \\ I \end{bmatrix} u(k). \quad (8)$$

For $\tau_a > T_s$ the system is described by (9). In (8) and (9), we omit T_s from the argument of $x(\cdot)$ and $u(\cdot)$ for readability.

$$\begin{bmatrix} x(k+1) \\ u(k-(d-1)) \\ \vdots \\ u(k-1) \\ u(k) \end{bmatrix} = \begin{bmatrix} \Phi(T_s) & \Phi(T_s - \tau'_a)\Gamma(\tau'_a) & \Gamma(T_s - \tau'_a) & \dots & 0 \\ 0 & 0 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & I \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-d) \\ \vdots \\ u(k-2) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{bmatrix} u(k) \quad (9)$$

Algorithm 2 MPC-Actuator Algorithm

Require: $u_{i,k}, u_{i,k-1}, id_k, id_{k-1}, t_{next}, t$

if $id_k > id_{k-1}$ **then**
 $u_i \leftarrow u_{i,k}(1)$
 $t_{next} = t + T_s$

else if $t \geq t_{next}$ **then**
 $it = \lfloor 1 + \frac{t-t_{next}}{T_s} \rfloor$
 $u_i \leftarrow u_{i,k-1}(it)$
 $u_{i,k} \leftarrow u_{i,k-1}$
 $id_k \leftarrow id_{k-1}$

else if $t < t_{next}$ **then**
 $u_i \leftarrow u_{i,k}(1)$

end if
return u_i

To include actuator delays in the prediction model, the discretized system has to be augmented with the delayed input values and the system matrices expanded as shown in (8), (9). The MPC gets an estimate of the delay $\tau_a = \hat{\tau}_i$ from the Kalman Filter and applies the augmentation for each IBR i , by selecting the respective rows and columns of $\Phi(t)$ and $\Gamma(t)$. As such, individual delays of different IBRs are explicitly modelled. Note that the expected computational delay can be added to $\hat{\tau}_i$.

MPC-Actuator Algorithm: Traditionally, MPC applies the first value of the control vector $u_{i,k} = [u(k|k), \dots, u(k+H_u-1|k)]^T$. We propose that the MPC sends the full control input trajectory over the control horizon H_u to the IBRs and that an MPC-actuator algorithm evaluates, which control input u_i is applied, using Algorithm 2. When a new packet arrives, the control input is applied immediately. Algorithm 2 expects that the next control input arrives at $t + T_s$, and, if it doesn't, it iterates through the trajectory of the control horizon H_u .

This method is uniquely applicable for MPC, since the prediction of the controller and the explicit output of the internal optimization allows to communicate the predicted control sequences instead of only a single control input. If, during the operation, the communication delay increases or packets are lost, it allows the IBR to follow the latest optimized trajectory for the horizon H_u .

C. Summary

The complete delay-aware control strategy is shown in Fig. 2. The delay-aware MPC sends control signals to each IBR i in time-stamped packets including the full trajectory $u_{i,k}$. The communication network subjects the control signal to delay and packet loss. At the IBR site, the packets are

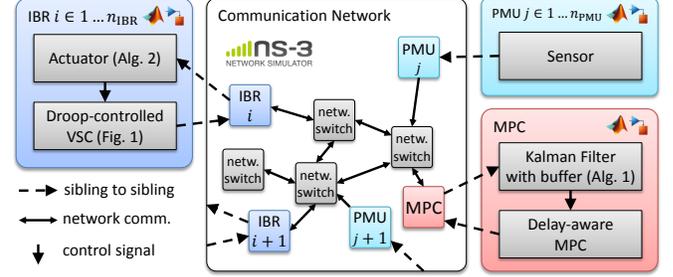


Fig. 2: Overview of the networked control system and the co-simulation design.

received and Algorithm 2 is applied. Each IBR i sends time-stamped measurement packets with the current state $y_i = x_i$ and an acknowledgement of the previous received packet. Additionally, PMUs provide measurements of bus voltages that are required to calculate the prediction model. The centralized control stores all measurements and the control input packets in the moving horizon buffer described in Section III-A. The time-stamp and acknowledgment are used to correctly order and align the buffer. The Kalman Filter employs Algorithm 1 to provide the estimate \hat{x} and additionally $\hat{\tau}$ for the augmentation of the prediction model described in Section III-B.

IV. CO-SIMULATION

To accurately evaluate our delay-aware MPC, we need a framework reflecting two orthogonal properties: (i) the effects of the control strategy on the power system, and (ii) the effects of delays and losses experienced during communication of the required data on the control strategy. We argue that the best fitting candidate is a simulation setup that accurately reflects both electric power systems and communication networks. The behavior and physical effects of power systems as well as advanced control algorithms are commonly modelled in Matlab/Simulink [15], while effects of communication are best modelled by a network simulator such as ns-3 [16]. We hence decided to use these two simulators, applying the common practice of aligning them in a co-simulation framework.

Design Challenges: The co-simulation framework connects two dissimilar simulators, namely the time-continuous power system simulator in Matlab/Simulink and the time-discrete network simulator ns-3. This requires to (i) align the notions of time, (ii) synchronize the time-advancements in both simulators, and (iii) transfer data between the simulators. The latter requires to bridge the gap between continuous signals in Simulink and discrete packets in ns-3. Finally, we need to ensure that both simulators simulate the same scenario

and have a clear understanding of how devices in one simulator correspond to the devices in the other simulator.

Ensuring aligned simulation scenarios: Our realization of the co-simulation design builds on a common configuration script. This script defines the devices that exist in the physical world and are relevant for either or both of the simulators (e.g., IBRs, loads, controllers, network routers). The topologies of the power grid and the communication network may align, but can also be different. We implemented a Matlab procedure that parses the configuration and creates the defined Simulink model. On the ns-3 side, similar functionality sets up and configures the elements of the network simulation. This ensures that both models align and both simulators have a common understanding of which devices exist and how they correspond to devices in the other simulator.

Enabling exchange of continuous signals and discrete packets: In both simulations, we create a *connector* module that enables exchanging data with the respective other simulator. Simulink signals that have to be communicated between devices are discretized with a defined measurement frequency. The signals are sampled and handed over to the Simulink connector to facilitate the simulation of their transmission in the communication network on the ns-3 side. To this end, every device in the power system that needs to communicate must have a sibling in both simulations (see Fig. 2). The Simulink connector assigns the identity of the sender block and transfers the signal’s value to the ns-3 connector, which will create and deliver an ns-3 packet to the sender’s sibling. Ns-3 then simulates the transmission to the receiver’s sibling, so that the information arrives with accurate delay, or does not arrive in case of packet loss. Handing over data from ns-3 back to Simulink works in an analog procedure via the connectors. Note that the handovers from Simulink to ns-3 and back do not add simulated delays to the exchanged data.

Synchronizing time advancements: To synchronize the advancements of time in the two simulators, a simulator that is faster than the other one needs to be paused. When ns-3 advances its local time (from the time of one event to the time of the next one), it communicates that time-stamp to Simulink via the connector. This allows Simulink to advance until that time, requiring it to wait for ns-3 before advancing any further. Likewise, Simulink communicates its local time to ns-3 and ns-3 adapts its pace to Simulink. This ensures that the two simulators run at equal pace and prevents inconsistencies, as data transmitted from simulator *A* to simulator *B* is guaranteed to arrive before *B* has advanced its local time further.

The co-simulation framework enables us to conveniently specify complex simulation scenarios and evaluate our approach with both realistic communication delays and losses in combination with realistic power system simulations.

V. EVALUATION AND DISCUSSION

A. Co-Simulation Scenarios

We evaluate our proposed delay- and loss-aware MPC in co-simulations of a low-voltage microgrid and a high-voltage transmission system with different networking scenarios.

Power system model: To simulate a microgrid we apply a modified low-voltage version of the WSCC 9-bus model; for the high-voltage transmission system we apply the IEEE 39-bus system. In both systems, the synchronous generators are exchanged for IBRs. The system specifications are obtained from [4]. We assume that the steady-state active power and voltage set-points of the IBRs are determined by a higher-level control instance, such that the delay-aware MPC is responsible for stabilizing and recovering the frequency during transients. The proposed controller is benchmarked against a central PI-controller enhanced with a Smith Predictor. The PI-controller is tuned using a first order approximation of the open loop step response of the primary controlled IBRs, as described in [4]. The Smith Predictor uses the same transfer function for the plant model. The delay transfer function of the predictor $G_d(z) = z^{-d}$ is adaptive and the discrete delay d is calculated as the rounded up average of the real-time round trip delay determined from the time-stamps of the measurement packets.

Network topologies: We consider two different communication network configurations in our study. For the microgrid case, we argue that the physical expanse of such systems is rather limited, e.g., in suburban redevelopment areas or industrial environments. Therefore, we assume a simple communication network layout, in which the node hosting the controller (controller node, CN) as well as all relevant electrical grid components (EGCs), i.e., IBRs and PMUs, are connected to separate edge router elements, which in turn are connected to a central neighborhood router. For the 39-bus system case, “New England system” [17], we create a regional communication scenario based on the Ion network from the Internet Topology Zoo [18], which describes a 125-node US fiber carrier network spanning a few hundred kilometers around Boston and New York. The CN is connected to a central node within the Ion topology, the remaining EGCs are distributed over the area and connected to different nodes.

Network details: We configure the networks such that the packets generated by the CN and the EGCs are subjected to realistic conditions regarding delays, bandwidths, and possible background traffic. The core nodes in the Ion network are thus configured to exhibit the high throughput and low delay typical in backbone networks. In turn, the links of the edge routers act as bottlenecks by mimicking typical residential connections in the US in terms of delay, queue sizes, and bandwidths (higher download than upload speeds). The connections of the CN/EGCs to the edge routers either mimic wired local Ethernet connections (high bandwidth, low delay, low error rates) or employ the ns-3 802.11n WiFi model (medium bandwidths and delays, losses dependent on distance to the edge routers). The packets created per round/sampling interval are rather small (in the 3-to 4-digit byte range each) and are sent via UDP. For such packets to experience noticeable variations in delay and loss, we employ ns-3 OnOffApplications to generate additional UDP background traffic to intermittently fill the queues of the routers at the bottleneck links. These applications always use wired connections to only affect queuing but not a potential WiFi connection of the CN/EGCs.

Fiber & DSL Background traffic		Wireless	
		Distance	[Exp. avg. loss rate]
A	No traffic	A 40 m	[6.7 %]
B	Low traffic	B 60 m	[18.3 %]
C	Moderate traffic	C 80 m	[35.3 %]
D	Heavy traffic	D 100 m	[54.3 %]

TABLE I: Overview of the simulation scenarios and variants. We vary the amount of background traffic in the fiber & DSL scenarios. In the wireless cases, we have no background traffic but vary the distance of the nodes and hence the expected loss.

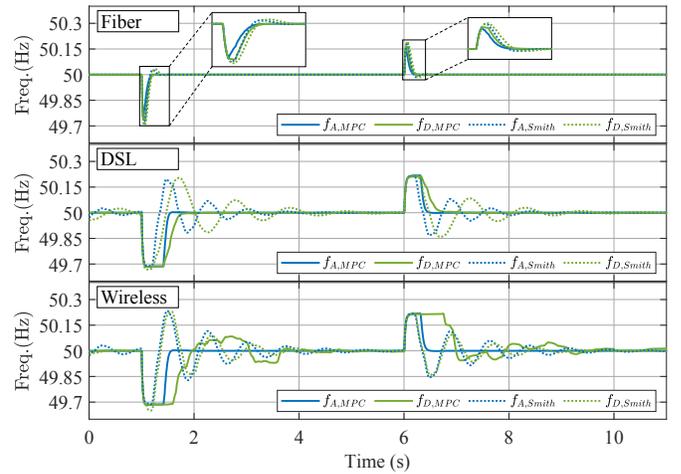
We apply our proposed delay- and loss-aware MPC in three different network scenarios: fiber, DSL, and wireless. The fiber scenario features close-to-optimal conditions with very low delay and negligible packet loss. The DSL scenario has less bandwidth compared to fiber, increasing the communication delay. In the wireless scenario, the connections of the EGCs (IBRs and PMUs) experience non-negligible loss rates. For each scenario, we construct four variants. In the fiber and DSL scenarios, we increase the amount of background traffic in each variant. In the wireless scenarios, we have no background traffic but instead increase the distances between the EGCs and the edge routers, thus increasing the expected packet loss rate. We provide an overview of our settings in Table I.

B. Co-Simulation Results

Fig. 3a presents the average system frequency in the *microgrid case* for the three scenarios fiber, DSL, and wireless for the variants A and D. A quantitative evaluation is provided by calculating the integral absolute error (IAE, Fig. 3b) and the time to recover the frequency within a 0.01 Hz band (Fig. 3c), both averaged over all IBR frequencies.

In the *fiber scenario*, the proposed MPC has an overall improved performance of approx. 30-45 % compared to the Smith Predictor across all metrics. However, as it can be seen in the frequency plot, both controllers show a good response to the load steps. We note that, although variant C has a higher amount of background traffic, a better response is shown compared to variant B for both controllers. We observe that for this variant, only destinations and no sources of traffic were co-located with the IBRs, which fill the router queues and cause varying delay and loss of information. We conjecture that a loss of measurement signals has larger impact on the control system than a loss of control signals.

A significant difference in the controllers is observed in the *DSL and wireless scenarios*. The response of the Smith Predictor is clearly underdamped, but remains stable with a significantly extended frequency recovery time. The delay-aware MPC strongly outperforms the Smith Predictor for the scenarios DSL A-D and wireless A-C, with a well damped response. The IAE is improved by approx. 28-50 %, while the frequency recovery time is approx. 60-86 % shorter. However, for the scenario wireless D, which has an expected loss rate of 54.3 %, the Smith Predictor shows better performance. It is notable that the Smith Predictor response is in general not



(a) Average Frequency for scenarios A and D.

	Fiber		DSL		Wireless	
	MPC	Smith	MPC	Smith	MPC	Smith
A	3.2	5.0	21.2	32.0	21.5	44.8
B	3.8	5.3	22.2	34.6	21.4	42.8
C	3.2	5.0	22.9	36.4	26.9	37.5
D	4.5	6.4	27.8	48.0	61.5	40.2

(b) Average Integral Absolute Error (IAE) [$\times 100$].

	Fiber		DSL		Wireless	
	MPC	Smith	MPC	Smith	MPC	Smith
A	0.148	0.277	0.480	2.302	0.487	3.574
B	0.162	0.288	0.529	2.537	0.563	3.358
C	0.148	0.277	0.741	2.693	1.162	2.826
D	0.191	0.341	0.820	3.403	4.533	2.961

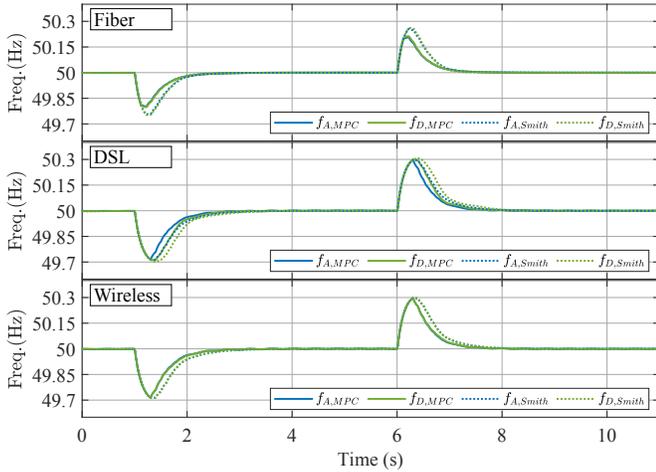
(c) Average recovery time [s].

Fig. 3: Simulation results for the microgrid case.

affected by packet loss. All simulations in the wireless scenario show a comparable performance to the DSL scenarios. Note, however, that an expected loss rate of 54.3 % (i.e., only every second control signal arrives at the actuator) is not common for these kind of applications and poses an extreme scenario, which would not be intended permanently.

The results for the *transmission system case* are provided in Fig. 4. The proposed delay-aware MPC has a approx. 18-35 % reduced IAE compared to the Smith Predictor, with comparable recovery times. Overall, both controllers show a similar and good performance in the transmission system case.

Contrary to the microgrid case, the different communication network scenarios have a lesser impact on the control performance in the transmission system case. We explain this result by comparing the control bandwidth of the secondary control systems in both study cases. In the microgrid case, the proposed MPC and the Smith Predictor have a sampling time of 20 ms, whereas in the transmission system case the sampling time is 100 ms. The increased sampling time results from slower control systems of the primary IBRs on the transmission level. The control bandwidth is limited by the power electronics' switching frequency of the VSC. In low voltage/microgrid applications, the VSC switching frequency can be more than one order of magnitude faster than in high



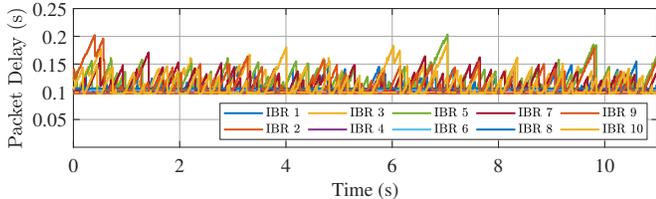
(a) Average Frequency for scenarios A and D.

	Fiber		DSL		Wireless	
	MPC	Smith	MPC	Smith	MPC	Smith
A	20.9	28.2	34.3	41.9	34.3	41.9
B	20.9	28.2	34.6	41.8	34.2	41.9
C	21.0	28.1	36.0	42.7	34.6	41.9
D	21.5	27.9	39.9	48.8	34.8	41.8

(b) Average Integral Absolute Error (IAE) $\times 100$.

	Fiber		DSL		Wireless	
	MPC	Smith	MPC	Smith	MPC	Smith
A	1.220	1.249	1.779	1.739	1.778	1.745
B	1.222	1.241	1.786	1.722	1.802	1.744
C	1.243	1.161	1.752	1.531	1.818	1.744
D	1.241	1.119	1.812	1.958	1.785	1.739

(c) Average recovery time [s].



(d) Packet delay of received IBR measurements, scenario DSL-B.

Fig. 4: Simulation results for the transmission system case.

voltage/transmission systems. In contrast, the communication delays are introduced mostly on the connection of the edge routers and to a lesser extent by the core network. As such, although the absolute delay of the transmission case is larger than in the microgrid case, the delay is smaller in relation to the controller sampling time. Furthermore, the increase of background traffic from variants A to D did not significantly affect the control performance of both the IAE and the recovery time. To illustrate the effect of the background traffic on the communication system, a plot of the measured packet delay for the scenario DSL-B (low traffic) is provided in Fig. 4d. The base communication delay and the variance introduced by the background traffic are visible. Routers not co-located with traffic applications have a constant delay of approx. 100 ms. For variant C (moderate traffic), router queues start to fill up, whilst in variant D (heavy traffic), all affected queues become completely filled. In the latter case, the packet delay becomes

mostly constant at 210 – 220 ms. Note that packets arriving at a full queue are lost.

We conclude that delays up to approx. 2.5 times the controller sampling time are well handled by both our proposed delay-aware MPC and the Smith Predictor, whereas for in relation larger delays our MPC performs significantly better.

VI. CONCLUSION

This paper proposes a delay- and loss-aware MPC for fast frequency control in power systems. The performance of the proposed controller is studied in a co-simulation of a power system and a communication network simulator for realistic network conditions of a microgrid and a transmission system case. In comparison to a Smith Predictor-based PI controller, our delay-aware MPC achieves significantly better results in the microgrid case and a small improvement for the transmission system case. It is shown that both controllers effectively handle packet loss in the control loop.

REFERENCES

- [1] P. Kundur, N. Balu, and M. Lauby, *Power System Stability and Control*. McGraw-Hill, 1994.
- [2] H. Wu, H. Ni, and G. Heydt, “The impact of time delay on robust control design in power systems,” in *Proc. of IEEE Power Eng. Soc. Winter Meeting*, 2004.
- [3] F. Garcia-Torres, A. Zafra-Cabeza, C. Silva, S. Grieu, T. Darure, and A. Estanqueiro, “Model Predictive Control for Microgrid Functionalities: Review and Future Challenges,” *Energies*, vol. 14, 2021.
- [4] T. Heins, M. Joševski, S. Gurumurthy, and A. Monti, “Centralized Model Predictive Control for Transient Frequency Control in Isolated Inverter-Based Microgrids,” *IEEE Trans. Power Sys.*, vol. 38, no. 3, 2023.
- [5] C. Ahumada, R. Cárdenas, D. Sáez, and J. Guerrero, “Second. Control Strategies for Freq. Restoration in Isolated Microgrids With Consideration of Commun. Delays,” *IEEE Trans. Smart Grid*, vol. 7, 2016.
- [6] P. Ojaghi and M. Rahmani, “LMI-Based Robust Predictive Load Frequency Control for Power Systems With Communication Delays,” *IEEE Trans. Power Systems*, vol. 32, 2017.
- [7] A. Ademola-Idowu and B. Zhang, “Frequency Stability Using MPC-Based Inverter Power Control in Low-Inertia Power Systems,” *IEEE Trans. Power Systems*, vol. 36, 2019.
- [8] F. Milano and M. Anghel, “Impact of Time Delays on Power System Stability,” *IEEE Trans. Circuits and Systems*, vol. 59, 2012.
- [9] D. Dotta, A. Silva, and I. Decker, “Wide-Area Measurements-Based Two-Level Control Design Considering Signal Transmission Delay,” *IEEE Trans. Power Systems*, vol. 24, 2009.
- [10] W. Li, M. Ferdowsi, M. Stevic, A. Monti, and F. Ponci, “Cosimulation for Smart Grid Communications,” *IEEE Trans. Ind. Inf.*, vol. 10, 2014.
- [11] M. Moayed, Y. Foo, and Y. Soh, “Adaptive Kalman Filtering in Networked Systems With Random Sensor Delays, Multiple Packet Dropouts and Missing Measurements,” *IEEE Trans. Signal Proc.*, vol. 58, 2010.
- [12] L. Schenato, “Optimal Estimation in Networked Control Systems Subject to Random Delay & Packet Drop,” *IEEE Trans. Aut. Cnt.*, vol. 53, 2008.
- [13] P. H. Dana and B. M. Penrod, “The Role of GPS in Precise Time and Frequency Dissemination,” *GPS World*, 1990.
- [14] B. Wittenmark, “Sampling of a system with a time delay,” *IEEE Trans. Automatic Control*, vol. 30, no. 5, 1985.
- [15] MathWorks, “Simulink Toolbox: Specialized Power Systems, version 7.7 (R2022a),” Natick, MA, US, 2022. [Online]. Available: mathworks.com
- [16] G. Riley and T. Henderson, “The ns-3 Network Simulator,” in *Modeling and Tools for Network Simulation*, Wehrle, Güneş, and Gross, Eds. Berlin, Heidelberg: Springer, 2010.
- [17] T. Athay, R. Podmore, and S. Virmani, “A Practical Method for the Direct Analysis of Transient Stability,” *IEEE Tran. Pow. App. & Sys.*, 1979.
- [18] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The Internet Topology Zoo,” *IEEE J on Sel. Areas in Comm.*, vol. 29, 2011.