Wireless Control Communications Co-Design via Application-Adaptive Resource Management

Lucas Scheuvens[1], Tom Hößler[1,2], André Noll Barreto[2], and Gerhard P. Fettweis[1,2]
[1]Vodafone Chair Mobile Communications Systems, Technische Universität Dresden, Germany
[2]Barkhausen Institut, Dresden, Germany
Email: {lucas.scheuvens, tom.hoessler, gerhard.fettweis}@tu-dresden.de, andre.nollbarreto@barkhauseninstitut.org

***Abstract*—Ultra reliable low latency communications (URLLC) systems are usually designed to achieve availability values larger than 99.999 % at latencies in the millisecond range. However, these requirements can often be significantly relaxed by a control communications co-design approach, in which both control and communications domains adapt to each other. We present a framework to determine the quality of control (QoC) requirements of a given application and introduce the concept of *application reliability*, based on the example of an automated guided vehicle (AGV). It is observed that stability alone is not sufficient to cover QoC, and the required control precision has to be considered as well. These requirements are then mapped to possible network design strategies. Additionally, we propose an application-adaptive resource management that is able to reduce the average amount of links provisioned to the the application by approximately 1.7 while ensuring a maximum deviation from the desired path.**

***Index Terms*—Control Communications Co-Design, Wireless Control, URLLC, Adaptive Resource Management**

I. Introduction

Employing wireless communications in control loop systems is a main focus of ultra reliable low latency communications (URLLC) research. Thereby, in the often referenced 3rd Generation Partnership Project (3GPP) definition, URLLC makes up one of three fifth generation (5G) pillars, targeting use cases requiring availability values larger than $99.999\,\%$ and latency below $1\,\mathrm{ms}$ [1], [2]. Similarly, the guideline by the German Federal Ministry of Education and Research for developing wireless communications systems as replacement for fieldbus solutions for industrial use cases aims at error probabilities of less than 10^{-9} with round-trip latency values below $1\,\mathrm{ms}$ [3].

However, it is crucial to understand that these latency and availability values are not always necessary for wireless control applications. It was shown in [4], for an automated guided vehicle (AGV) application, that the actual communications requirements are sometimes orders of magnitude more relaxed than the values given above, even for closed-loop control applications. This is because the above values are the ones achieved with current implementations (mostly based on fieldbus systems), which motivated a 1:1 replacement strategy. This seems undesirable, as this approach poses overly rigorous requirements on the wireless network and wastes wireless system resources on a large scale. Instead, a co-design approach should be taken, that jointly designs the application control parameters together with the communications network, which must be considered as one non-ideal component. With this in mind, the application should apply coping strategies in order to limit the effect of the impairments in the communications network. In this work, we will present a framework on how to co-design such a system demonstrated for the example of an AGV.

Naturally, the definition of the key performance indicators (KPIs) plays a vital role in system analysis. For instance, it is important to clearly state if the availability constitutes an average value or a lower bound for each transmitted packet. It is also critical to define whether latency and availability are interconnected, i.e., if every received packet above the latency limit is considered unsuccessful or not. In our prior work [5], we cleared up some of these questions by specifying well-defined KPI metrics for wireless communications based on reliability theory. These metrics are also adopted in this article.

It still remains to be answered when it makes sense to deploy wireless instead of wired communications, even though the former introduces significant impairments. This is also a question that every application engineer needs to carefully assess before turning their application into wireless. Most common reasons include (a) the reduction of Capital Expenditure (CapEx) due to lower installation cost (fewer cables) [6], (b) the reduction of Operational Expenditure (OpEx) due to less maintenance [6], (c) the emergence of applications that are not possible with wired components, e.g., due to rotating or moving parts, (d) a significant benefit from a central coordination entity in a mobile scenario [4], (e) a networked system of control entities[1] [7]. In the AGV use case analyzed in this article, cables would severely limit the mobility, and the application benefits from a central coordination entity that is able to coordinate a large number of AGVs on a factory floor through a centralized planner/controller in a nearby Multi-Access Edge Cloud (MEC).

This work contributes to a better understanding of the fundamental interdependence between the control application and the wireless communication system. We derive recommendations for a co-design in order to define and improve the correct operation of an application. We present possible tools for this *holistic approach* based on the exemplary AGV use case. Thereby, we focus on the impact of packet loss on

[1] For instance, Cooperative Adaptive Cruise Control (CACC) is an example of a physically tightly-coupled control system.

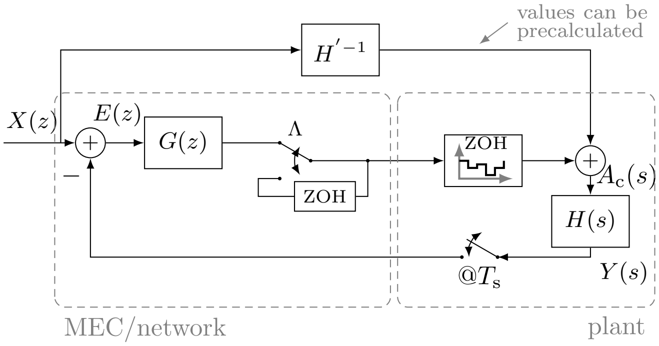

Fig. 1: System Model

the system. We show that with appropriate system design, the consequences arising from packet loss are not as severe as commonly assumed (and thus relax the communications requirements) and we sketch how to describe this property with well-known metrics from control theory. Thereafter, we introduce new communication-related metrics that we call *application availability* and *application reliability*, respectively, that capture the desired application behavior from a communications point of view and indicate how these metrics can be improved by multi-connectivity (MC). We thereby highlight the limits of classical metrics, e.g., packet loss rate (PLR), for communications system design for control applications.

II. System Model

In this article, a wireless control system with a single AGV is considered. The AGV is limited to a 1-D trajectory, i.e., it is able to accelerate and decelerate without the capability of turning. The extension to the omnidirectional 2-D case is straightforward and omitted for simplicity. The system model is depicted in Fig. 1. It comprises a standard control loop with a feed-forward component. $X(z)$ denotes the reference signal, i.e., the desired position of the vehicle. $Y(s)$ is the plant output, i.e., the *actual* position of the vehicle. $H(s)$ describes the plant dynamics and thereby characterizes how controller commands $A_c(s)$ impact the vehicle. $E(z)$ denotes the error between reference and actual position, and $G(z)$ describes the controller transfer function.

In Fig. 1, some of the quantities are described using the Laplace transform variable s and others through its discrete-time counterpart z. This is because the controller is implemented digitally and is assumed to operate periodically at the rate of the sampling period T_s. This allows to describe the system in the z-domain. Having an s-domain as well as a z-domain requires $z \leftrightarrow s$ conversion at the interfaces.

In the following, first all components of the control system are introduced. Thereafter, the wireless system model and its integration into the control system is presented and discussed.

A. Control System Components

The plant's dynamics $H(s)$ are schematically depicted in Fig. 2. The core component for the present AGV application is a DC motor model, specifying how the motor reacts to an input voltage signal v_{DC}. The output of the model describes the rotational velocity of the motor shaft, denoted as $\dot{\theta}$. Their relationship can be accurately described by [8]

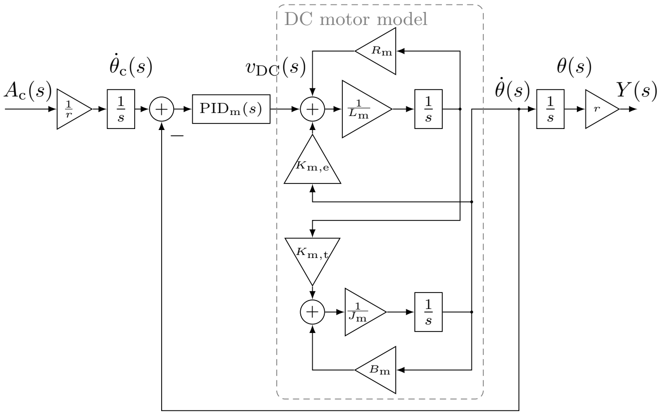

Fig. 2: Vehicle transfer function $H(s)$

TABLE I: Summary of DC motor physical properties.

Param.	Value	Unit	Description
J_m	0.03	$\mathrm{kg\,m^2}$	rotor moment of inertia
B_m	0.1	$\mathrm{N\,m\,s}$	motor viscous friction
$K_{m,e}$	0.01	V/rad/s	electromotive force const.
$K_{m,t}$	0.01	Nm/A	motor torque const.
R_m	1	Ω	electr. resistance
L_m	0.5	H	electr. industance

$$\frac{\dot{\theta}(s)}{v_{DC}(s)} = \frac{K_{m,t}}{(L_m s + R_m)(J_m s + B_m) + K_{m,t}K_{m,e}} \quad . \qquad (1)$$

All parameters are described and summarized in Tab. I.

Since the motor is modeled by a transfer function $v_{DC} \rightarrow \dot{\theta}$ but the vehicle is supposed to translate between controller command $A_c(s)$ (acceleration) and position $Y(s)$ instead, additional input and output circuitry is required, see Fig. 2. $Y(s)$ constitutes a position, which requires that $\dot{\theta}$ – the output of the motor model – be integrated and multiplied by the diameter of the wheel r to yield the appropriate actual position.

For the input, in order to convert between a controller command $A_c(s)$ (which physically represents a desired acceleration value) and the input voltage v_{DC} of the motor, an additional DC motor control system is required, as is well-known from standard control system design knowledge. This additional DC motor control system is presumed to run on the host device itself, hence, Laplace domain design suffices [9]. Practicable controller gains for the motor control system were obtained in an iterative design process (not focus of this paper) and are summarized in Tab. II.

The feed-forward component $H^{'-1}$ effectively turns the position control cycle into a slave controller, ensuring that the

TABLE II: Summary of DC motor control parameters.

$K_{d,m}$	$K_{p,m}$	$K_{i,m}$	N_m
20	170	330	83

for

$$\mathrm{PID_m}(s) = K_{p,m} + K_{i,m}/s + K_{d,m} \cdot \frac{N_m s}{s+N_m}$$

effect of $X(z)$ on $Y(s)$ is immediate and that inside the control system only the *error in position* is controlled as opposed to the position itself, improving performance dramatically. The extra dash denotes that the inverse of $H(s)$ might be difficult to derive and therefore, $H^{'-1}$ might constitute an approximation only, yielding a deviation that the control systems needs to adjust. In order to efficiently optimize a communications system for this use case, it is crucial to understand that this addendum to the control system relaxes its requirements tremendously by separating the non-time-critical information (path information) from the mission-critical information. That is, when a motion control task is performed, first the path is planned and then, during operation, a control cycle ensures the precise execution of the path. The path information can be transmitted well before operation and is therefore not time-critical. In control theory, this is a well-established solution to improve control performance.

The proportional, derivative (PD) controller[2] is a powerful compromise between performance and simplicity, hence, it is chosen here for $G(z)$. It consists of a proportional component $P(z)$ and a derivative component $D(z)$, such that $G(z) = P(z) + D(z)$. The proportional component yields a correctional signal that is *proportional* to the measured position error, i.e., $P(z) = K_{\mathrm{p}}$. The further away the vehicle is from the desired position $X(z)$, the higher the proportional correction signal will be, i.e., the more the vehicle will accelerate or decelerate in order to reduce the error. The derivative component is designed to yield an output that is proportional to the *derivative* of the error signal $E(z)$. For the present discrete-time case, the difference between successive samples is calculated, the result is weighted by $\frac{K_{\mathrm{d}}}{T_{\mathrm{s}}}$ and directly output (backward Euler), i.e., $D(z) = \frac{K_{\mathrm{d}}}{T_{\mathrm{s}}}\frac{z-1}{z}$. The derivative component linearly predicts the trend of the error curve and counteracts the determined relation. That is, if the error curve increases, the derivative component will work against the increase and if it decreases, $D(z)$ will work against the decrease. $D(z)$ takes a crucial role in the design of the controller because it significantly affects its performance in multiple ways. Firstly, correct dimensioning of $D(z)$ greatly improves the response time of the controller since sudden errors can be counteracted very quickly. Secondly, $D(z)$ – also reasonable dimensioning assumed – greatly reduces overshoot errors. Thirdly, $D(z)$ is a powerful tool to stabilize the control cycle.

There are in fact controller designs that outperform PD but in order to illustrate the control communications co-design, a simple design is preferable. The methodology presented in this paper, however, is also applicable to other controller designs.

B. Impact of Sampling

The choice of the sampling period is crucial for the performance of any discrete-time control system. On the one hand, a large sampling period requires only few wireless resources. On the other hand, a large sampling period inherently destabilizes the control system and leaves less room for communication errors.

In the pursuit of finding an optimal sampling rate T_{s}, it is helpful to understand the impact that sampling has on the application. It acts similarly to a delay of $T_{\mathrm{s}}/2$ on the communication link, since through ideal sampling the sampled value lags on average half a sampling period behind the actual value. Therefore, sampling inherently introduces a stability degradation to the system, which is described by the phase margin reduction of $\delta\phi = -\omega T_{\mathrm{s}}/2$ [9]. For conventional discrete-time system design that is intended to resemble Laplace-domain behaviour as close as possible, a sampling rate of at least 30 times the system bandwidth is presented as a guideline [9]. For wireless systems, the optimal sampling period T_{s} might differ, depending on the optimization problem formulation. For instance, a larger number of vehicles can be served if each vehicle is sampled less often. This in return destabilizes each vehicle, therefore, one must compromise.

C. Packet Loss

Returning to Fig. 1, in order to outsource computation to a nearby MEC instance, wireless links need to be introduced between the edge cloud and the plant. These links typically are subject to transmission delay and packet loss. The effects of transmission delay and coping strategies for the present use case were covered in our prior work [4] and shall henceforth be left aside. The effects of packet loss are the focus of this paper.

Only packet loss in the downlink is assumed for simplicity. Hence, in the system model in Fig. 1, the uplink (the feedback) constitutes a perfect connection[3]. Every packet loss in the downlink is assumed to be fatal, i.e., no packet recovery can be performed and the information it carried is unrecoverable. From an information theoretic point of view, this resembles an *erasure channel* that erases the packet during transmission with probability q. In a clocked system, immediate action needs to be applied to replace the missing value because the next block expects a new value. This replacement is performed by extrapolation and a whole variety of extrapolation methods is applicable, which should not be detailed here. In this article, the intention is not to optimize extrapolation but to investigate the impact of packet errors on a control system. Hence, we simply feed back the last-known value, which is equivalent to a Zeroth Order Hold (ZOH) block.

D. Small-Scale Fading as Cause of Failure

In wireless communications, packet loss occurs when the received signal power is too weak compared to noise or an interfering signal. Assuming that the unwanted signal cannot be attenuated, coarsely spoken, three main factors contribute to the overall received power level: (1) path loss, (2) large-scale fading and (3) small-scale fading. Since an automatic

[2] The integral component of a possible proportional, integral, derivative (PID) controller is omitted because the plant itself already has a double integrating behaviour

[3] In future works, the fading downlink connection must be merged with a fading uplink connection to yield even more thorough insights about the effect of dual packet loss in both, uplink and downlink.

gain control (AGC) can accommodate for the first two, only small-scale fading is analyzed as a cause of failure in this article.

Small-scale fading characterizes rapid fluctuations of a radio signal in space, time, and frequency, caused by multipath propagation and the Doppler effect. Small-scale fading is a relevant cause of failure because it can deteriorate the received signal power by orders of magnitude, which potentially disrupts connectivity.

Two metrics mainly determine the statistical properties of the channel outage: (1) The maximum Doppler frequency f_D of the channel and (2) the fading margin F of the receiver. Herein, the maximum Doppler frequency – linearly related to the maximum relative speed of objects – is inversely related to the coherence time T_coh of the channel. However, even for environments that exhibit maximum relative speeds of only $v_\mathrm{max} = 2\,\mathrm{m\,s^{-1}}$ (low mobility scenario), the coherence time can be approximated as

$$T_\mathrm{coh} \approx \frac{9}{16\pi f_\mathrm{D}} = \frac{9c_0}{16\pi v_\mathrm{max} f_\mathrm{c}} \approx 7\,\mathrm{ms} \tag{2}$$

for c_0 as speed of light and $f_\mathrm{c} = 3.75\,\mathrm{GHz}$[4] as carrier frequency.

Since the smallest sampling period we consider in this article is $T_\mathrm{s} = 10\,\mathrm{ms}$, all transmissions can be assumed to fade independently. For smaller sampling periods, the temporal correlation of link outage needs to be taken into account.

Furthermore, the fading margin F is a receiver metric that describes how much a signal can be attenuated until the receiver is not able to decode the information. In this article, a rather low fading margin of $F = 10\,\mathrm{dB}$ is assumed to emphasize that the presented concept is also suitable in harsh conditions.

III. Application-aware KPIs

Considering small-scale fading as a cause of failure, it is straightforward to construct the ratio of up-time / total time in order to evaluate the system's availability *on average*. From a packetized view, this relation is described by the PLR, which is key for a reliability assessment of communications systems. For traditional applications, the PLR is a perfectly valid KPI. This is because the occurrence time of a packet error only plays a subordinate role for the performance, because the value of the information a packet carries only decreases slowly with time. However, in control systems engineering, the value of information decreases much more rapidly. For short sampling periods, acknowledgments and retransmissions become increasingly unfeasible, since by the time the transmitter receives the knowledge about a lost packet at time instant k, the next packet at $k+1$ was sent already and the information sent at k is outdated and may be dropped.

Therefore, for control applications, the PLR alone does not suffice and the time information needs to be factored in, i.e., *when* errors occur and how they are correlated to previous errors.

The conventional approach in URLLC to enable wireless closed-loop applications is the development of a wireless communications system that transmits packets at an extremely high reliability, preferably at packet loss probabilities $< 10^{-6}$. In [10], fundamental availability analysis has shown that in order to realize outage durations larger than $10\,\mathrm{ms}$ at probabilities $< 10^{-4}$ for $F = 10\,\mathrm{dB}$, at least four independently fading links need to be deployed simultaneously (with selection combining under Rayleigh-fading conditions). The number of links has to be increased even further to realize packet loss probabilities $< 10^{-6}$.

In this article, however, we show that an alternative approach is much more efficient in terms of resource usage, while maintaining the application functionality. It is fundamentally different from the conventional URLLC approach in the sense that it aims at ensuring high *application reliability* instead of high *communication reliability*.

In [5], [11], we presented alternative metrics for wireless communications systems that – rather than averaging over the whole measurement period – include the time factor in dependability metrics and thereby complement the classical PLR. It is important to distinguish *availability* (the probability that a transmission at a time instant will be successful) from *reliability* (the probability that all transmissions during an interval continue to be successful). In [12], the application of these new metrics to a WLAN 802.11ax system were shown, based on the industrial indoor channel model developed in [13]. Therein, it was attempted to link these temporal communication dependability metrics to application dependability metrics by introducing the notions of *application availability* and *application reliability*, that allow for packet errors as long as not more than p consecutive packet errors occur. The assumption behind the introduction of these new metrics is that most real-time applications tolerate packet errors as long as not *too many* occur consecutively. This will be demonstrated later on. These metrics play a vital role in a true Control Communication Codesign (CoCoCo) since they are able to map application quality of control (QoC) metrics into communication Quality of Service (QoS) metrics and thereby provide an interface with which engineers from both domains are able to work.

IV. Adaptive Resource Allocation

We introduce a new application-adaptive resource allocation scheme that builds on the new application dependability metrics presented in the last section. Here, we assume that between samples an adjustment of the success probability for the next transmission is possible by dynamically adding and removing one or multiple links. This is reasonable to assume, since the sampling period is chosen as $T_\mathrm{s} \geq 10\,\mathrm{ms}$ and a dynamic adjustment is reasonable in this time frame. We assume that every erroneous transmission triggers adding two links while every successful transmission resets the number of links to one. Obviously, for other use cases the number of links can

[4] The German Federal Network Agency has reserved the band $3.7\,\mathrm{GHz}$ to $3.8\,\mathrm{GHz}$ for local *on-campus usage*. Hence, this band will soon become available for wireless industrial use cases in actual factory environments.

be freely chosen. In the present application, if S_k denotes the event of successfully decoding a transmitted packet in time slot k, all of the following relations hold true:

$$P_1 \equiv P(S_k|S_{k-1}) = 1 - q \tag{3}$$

$$P_2 \equiv P(S_k|S_{k-2} \cap \bar{S}_{k-1}) = 1 - q^3 \tag{4}$$

$$P_3 \equiv P(S_k|S_{k-3} \cap \bar{S}_{k-2} \cap \bar{S}_{k-1}) = 1 - q^5 \tag{5}$$

$$\vdots$$

This adaptive algorithm reduces the probability of long burst errors tremendously while simultaneously not over-provisioning the system with wireless resources when they are not necessary, i.e., when the last packet was received successfully.

V. Simulation Scenario and Results

Because of possible packet loss, the system depicted in Fig. 1 is a linear, time-variant (LTV) system. LTV systems cannot be described in the frequency domain by means of well-known transforms like Laplace (continuous-time) or z (discrete-time). Hence, our analysis consists of four parts.

First, we assume no packet loss and thereby create a linear, time-invariant (LTI) system that can be described in the frequency domain using the z-transform. This enables us to find basic stability and QoC boundaries, depending on the sample period T_s and the controller gains $(K_\mathrm{d}, K_\mathrm{p})$, implemented in $G(z)$. This approach is valid for all LTI systems, independent of the chosen (linear) controller architecture. The resulting phase margins are depicted in Fig. 3 on the left side. From standard control engineering lectures it is known that a feedback phase of 360° at a feedback gain > 1 causes system instability. The phase margin describes how far the system is clear of that point in terms of phase and consequently, the system becomes unstable at $\phi_\mathrm{PM} = 0°$. For traditional control cycle designs of DC motors, phase margins of 30° to 60° are targeted. [14, p. 64] For the present case, the controller parameters $(K_\mathrm{d}, K_\mathrm{p}) < (4.5, 4.5)$ seem promising as they feature high phase margins for low sampling periods.

Second, we simulate the time-domain response of the system to a given input path. The path consists of the locations $\mathcal{X} = \{0, 35, 70, 35, 0\}\mathrm{m}$. At $0\,\mathrm{m}$ and $35\,\mathrm{m}$ in both directions, the vehicle is supposed to wait for $3\,\mathrm{s}$, simulating loading/unloading of goods, while at $70\,\mathrm{m}$ it turns around. In order to evaluate the influence of higher dynamics, two driving styles with different maximum accelerations are considered. For the dynamic vehicle at $|a_\mathrm{max}| = 10\,\mathrm{m\,s^{-2}}$, this leads to a total mission duration of $30\,\mathrm{s}$, while for the slow dynamics vehicle at $|a_\mathrm{max}| = 1\,\mathrm{m\,s^{-2}}$, the duration is approx. $60\,\mathrm{s}$. The maximum deviation from the desired path is denoted as e_max and plotted for varying controller gains and sampling periods in Fig. 3 at the right-hand side. We emphasize that even for zero packet loss the deviation from the desired path is not zero as the modeled vehicle is not perfectly capable of realizing a desired acceleration because $H(s)$ is not ideal. However, lower dynamics lead to less deviation from the desired path, which is intuitive. Also, the maximum deviation increases for lower controller gains as the correctional signals are not so dominant. Consequently, low sampling periods in combination with large controller gains enable small deviations. In combination with the phase margin diagram on the left side, a desired working point can be chosen. E.g., for the setting of $(K_\mathrm{d}, K_\mathrm{p}) = (4.5, 4.5)$, a sampling period of $T_\mathrm{s} = 30\,\mathrm{ms}$ results in a phase margin of $\phi_\mathrm{PM} = 55°$, while still providing $\phi_\mathrm{PM} = 29°$ at a sampling period of $T_\mathrm{s} = 120\,\mathrm{ms}$, which translates to only receiving every fourth packet. As it can be seen, this application is already quite tolerant against packet loss *as long as not too many occur consecutively* while at the same time leading to path deviations $< 15\,\mathrm{cm}$. At the same time, for the chosen set of parameters, we note that falling below a certain sampling period will not yield better control performance. The maximum deviation remains rather constant for varying sampling periods T_s while also the stability only increases marginally for values $T_\mathrm{s} < 30\,\mathrm{ms}$. Hence, we identify that sampling periods chosen too low unnecessarily congest the network with superfluous information.

Third, we introduce rather high packet loss ($q = 10\,\%$) to the system and evaluate the system's reaction to driving the same path as above, by recording the maximum deviation from the desired location during the whole simulation. For easier reading, only the high dynamics case (fast acceleration) is considered. As mentioned before, the time of an error matters for closed-loop applications, hence, 1000 realizations with different seeds of a Bernoulli sequence with the above value for q are considered for each configuration. The results are depicted in Fig. 4. As expected, the introduction of packet loss has a negative impact on the maximum deviation and the different seeds also cause different maximum error values. Compared to no packet loss, an overall increase of the maximum deviation e_max can be observed, also leading to instability earlier than the LTI stability boundary.

Fourth, we analyze the benefit of reducing burst errors through our presented application-adaptive resource allocation scheme. The value of $q = 10\,\%$ per link remains unchanged. The results are depicted in Fig. 4 that shows the maximum deviation e_max for the four schemes (a) no packet loss, (b) fixed error probability of $q = 10\,\%$ (1 link), (c) fixed error-probability of $q = 1\,\%$ (2 links) and (d) our newly proposed application-adaptive resource allocation scheme (on average 1.16 links). Comparing these different schemes, however, is not straight forward because ensuring a fair comparison is difficult due to the dynamic behaviour of the adaptive approach. With that in mind, it can be shown that for 1000 simulations the adaptive scheme outperforms the single-link setting clearly, providing a lower upper bound that is a more tangible and reliable. However, one must keep in mind that the adaptive resource management also consumes approx. $16\,\%$ more resources, as indicated above. Also compared to scheme (c) that uses approximately 1.7 as many resources, the adaptive approach performs better in terms of the maximum e_max, reshaping the probability density function (PDF) towards more determinism by reducing the probability of high outlier values.

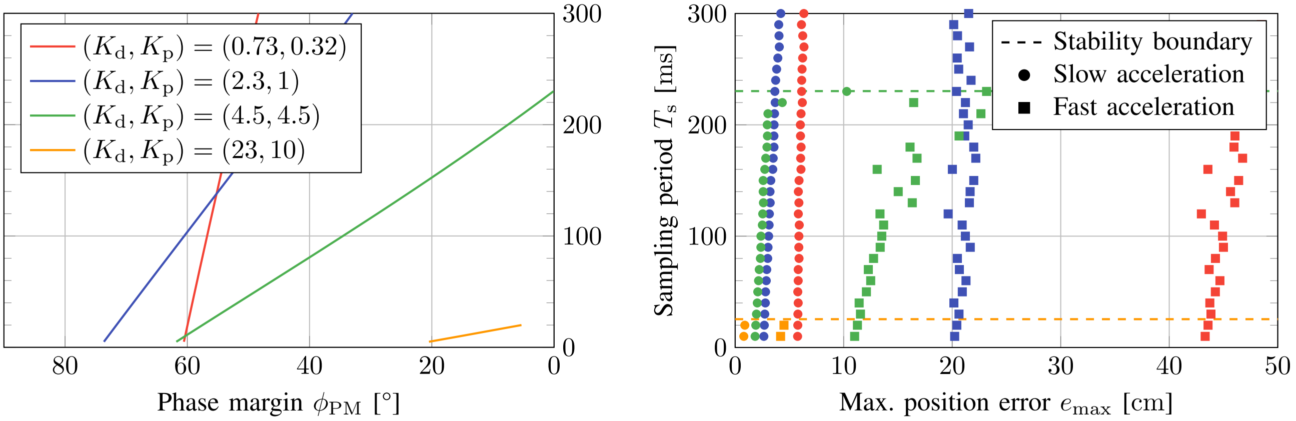

Fig. 3: Phase margin ϕ_PM and maximum deviation e_max with ideal communications system. The colors of the legend on the left also apply to the diagram on the right.

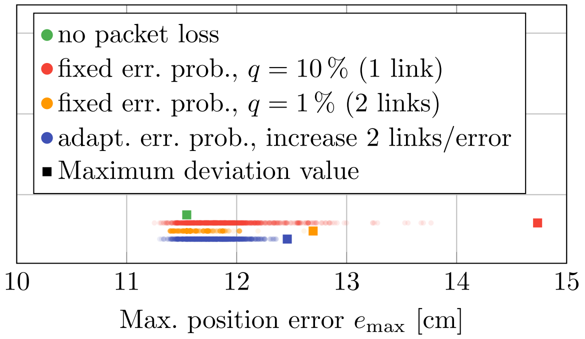

Fig. 4: Maximum deviation e_max with lossy communications system for $(K_\mathrm{d}, K_\mathrm{p}) = (4.5, 4.5)$ at $T_\mathrm{s} = 30\,\mathrm{ms}$

Due to the limited number of simulations performed and the small absolute difference in performance, more investigation is necessary to quantitatively describe the benefit of the proposed solution.

VI. Conclusion

For wireless control applications, packet loss often may not be as severe as commonly assumed, depending on the application. However, many consecutive packet errors have a high implication on the application's behaviour. Hence, it must be ensured that not *too many* packet errors occur at once. Through the presented application-adaptive resource allocation scheme, which adapts the number of links according to the history of lost packets, the application reliability remains high while the application is not over-provisioned with wireless resources when they are not needed.

Acknowledgments

This work was in part funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden. This work was also in part sponsored by the Federal Ministry of Education and Research within the program ”Twenty20 - Partnership for Innovation” under contract 03ZZ0528E - ”fast robotics”. This research was co-financed by public funding of the state of Saxony/Germany.

References

[1] G. Brown, “Ultra-Reliable Low-Latency 5G for Industrial Automation,” 2017. [Online]. Available: https://www.qualcomm.com/media/documents/files/read-the-white-paper-by-heavy-reading.pdf

[2] 5G Americas, “New Services & Applications with 5G Ultra-Reliable Low Latency Communications,” 2018. [Online]. Available: http://www.5gamericas.org/files/5115/4169/8314/5G_Americas_URLLLC_White_Paper_Final_11.8.pdf

[3] German Federal Ministry of Education and Research, “Announcement: Reliable Wireless Communications in Industry, BMBF's funding programme ”ICT 2020 - Research for Innovations”,” 2013. [Online]. Available: http://www.bmbf.de/foerderungen/22967.php

[4] L. Scheuvens, M. Simsek, A. Noll-Barreto, N. Franchi, and G. P. Fettweis, “Framework for Adaptive Controller Design Over Wireless Delay-Prone Communication Channels,” *IEEE Access*, vol. 7, no. 1, pp. 49 726–49 737, 2019.

[5] T. Hößler, L. Scheuvens, N. Franchi, M. Simsek, and G. P. Fettweis, “Applying reliability theory for future wireless communication networks,” in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–7.

[6] German Federal Ministry of Education and Research, “Industrie 4.0,” 2017. [Online]. Available: https://www.bmbf.de/pub/Industrie_4.0.pdf

[7] A. Gonzalez, N. Franchi, and G. P. Fettweis, “A Feasibility Study of LTE-V2X Semi-Persistent Scheduling for String-Stable CACC,” in *IEEE Wireless Communication and Networking Conference (WCNC)*, Marrakesh, Marroco, 2019.

[8] R. Krishnan, *Electric Motor Drives: Modeling, Analysis, and Control*, 1st ed. Pearson, 2001.

[9] G. F. Franklin, M. L. Workman, and D. Powell, *Digital Control of Dynamic Systems*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1997.

[10] D. Öhmann and G. P. Fettweis, “Minimum duration outage of wireless Rayleigh-fading links using selection combining,” in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2015.

[11] T. Hößler, M. Simsek, and G. P. Fettweis, “Mission Reliability for URLLC in Wireless Networks,” *IEEE Communications Letters*, vol. 22, no. 11, pp. 2350–2353, 11 2018.

[12] A. Traßl, T. Hößler, L. Scheuvens, N. Franchi, and G. P. Fettweis, “On Dependability Metrics for Wireless Industrial Communications - Applied to IEEE 802.11ax,” in *IEEE 5G World Forum*, 2019, (submitted for publication).

[13] A. Traßl, T. Hößler, L. Scheuvens, N. Franchi, and G. Fettweis, “Deriving an Empirical Channel Model for Wireless Industrial Indoor Communications,” in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Istanbul, Turkey, 2019, (submitted for publication).

[14] S.-H. Kim, *Electric Motor Control*, 1st ed. Elsevier, 2017.