A Control Approach for Cooperative Sharing of Network Resources in Cyber-Physical Systems

Florian Rosenthal and **Uwe D. Hanebeck**
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
florian.rosenthal@kit.edu, uwe.hanebeck@ieee.org

***Abstract*—This paper introduces an approach to cooperatively control the utilization of network resources, which are shared by several control loops, in cyber-physical systems without fixed scheduling schemes. Main goal of the proposed approach is to achieve a fair balancing with regards to the performance of each control loop such that the overall performance of the CPS is maximized. To maintain a loose coupling between the underlying communication system and the control loops on top, the approach relies on a data exchange between the control and the communication domain. This data exchange is carried out by a translator component embedded between each control loop and the communication system. Motivated by previous results, this data exchange is backed up by the principle of event-based control and used to restate the problem of balancing the network resources in terms of a control problem that resembles the popular consensus problem in multi-agent systems. We illustrate the applicability of the proposed approach by means of the task of simultaneously stabilizing two inverted pendulums over a shared medium.**

I. INTRODUCTION

Cyber-physical systems (CPS) are widely considered fundamental building blocks of the largely digitalized world of the future, notable examples ranging from intelligent production and manufacturing systems to Industry 4.0, smart homes and buildings, and the smart grid [1]–[4].

Although such systems may vary in size and operate on different spatial and temporal scales, they usually consist of control loops, that is, sensors, actuators and controllers, that share a common communication system, typically composed of general-purpose network infrastructure, for exchanging the information required to monitor and control the physical plants [5]. Compared to traditional point-to-point connections between the individual devices, employing such networks enables CPS to benefit from reduced costs for installation and maintenance, and also from enhanced flexibility.

However, sharing network resources without fixed scheduling or a priori reservation results in fluctuating capacity that is available for each control loop, in particular in the presence of additional unrelated traffic, thus impacting the achievable control performance. Even worse, if the shared resources have only limited capacity, any overutilization will cause additional queuing delays and packet losses, which in turn can render the control loops instable [6], [7]. Moreover, even if the control loops operate independently of each other or have different requirements, they are implicitly coupled due to the shared network resources. As a consequence, albeit being desirable, a fully independent design of both communication system and control loops is not possible in practice, and the need for multidisciplinary approaches for understanding and developing CPS has been pointed out by researchers of either realm [5], [8]–[10].

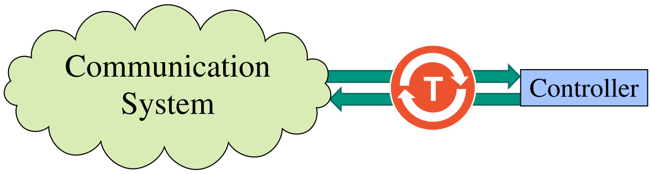


Fig. 1: The translator component is embedded on the data path between control and communication and mediates between the two domains.

In this regard, recent research has lead to the development of a variety of co-simulation and evaluation frameworks [11]–[15], the identification of key performance indicators [16], and the derivation of architectures and design guidelines [17], [18] for CPS.

In our previous work [19], we introduced an architecture for cyber-physical systems that strives to meet the aforementioned challenges by a cooperative usage of the resources provided by standard networking equipment. In particular, we proposed to implement cooperation by means of a data exchange i) between the control loops and the underlying communication system and its internal mechanisms for, e.g., congestion control, and ii) between the different control loops themselves.

Main goals of the proposed architecture are to utilize this data exchange to i) reach a high overall control performance across the whole CPS and ii) to avoid performance imbalances between the involved control loops, such that overutilized network resources are avoided. Key component of the architecture and responsible for the data exchange is a translator component that is embedded in the data path between each control loop and the communication system. As illustrated in Fig. 1, this translator component acts as a mediator, so that a tight coupling between the two domains is avoided.

Based on observations in [19], we present in this paper an approach that combines the data exchange carried out by the translator components and event-based control methods in order to reach a *consensus* among the control loops. This consensus is first computed by the communication system and, in a second step, then used to provide a fair sharing of resources that balances the utilization among the control loops with regards to control performance. Consequently, the aforementioned goals can be accomplished and a high degree of modularity and flexibility within the CPS is maintained.

Outline: The remainder of this paper is organized as follows. In Section II, we introduce the setup that we will consider throughout this work. Then, in Section III, we illustrate the considered problem. After that, in Section IV, an approach to solve this problem is proposed. Subsequently, we evaluate the proposed approach in Section V, before we conclude our work in Section VI.

II. BACKGROUND

Throughout this paper, we will use the task of simultaneously stabilizing two inverted pendulums over a shared communication system. As sketched in Fig. 2, we will utilize two pendulums with different parameter configurations in order to represent a CPS with two independent control loops with different requirements.

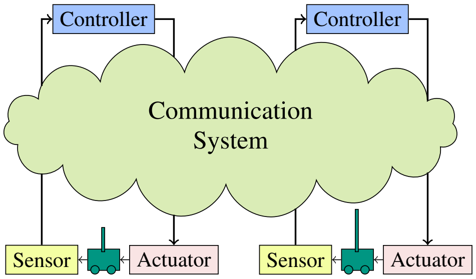


Fig. 2: Sketch of the CPS considered in this paper.

In the first configuration, we consider a pendulum with a uniform rod of length $l_1 = 0.3\,\mathrm{m}$ and of mass $m_1 = 0.2\,\mathrm{kg}$, while the second pendulum has a longer and heavier rod ($l_2 = 1\,\mathrm{m}$, $m_2 = 0.8\,\mathrm{kg}$). As a consequence, the requirements for a successful stabilization of the short pendulum with regards to timely and failure-free delivery of transmitted information are stricter because it is expected to fall over faster than the second, longer one.

The state of each pendulum at time step k is given by

$$\underline{x}_k = \begin{bmatrix} s_k & \dot{s}_k & \phi_k & \dot{\phi}_k \end{bmatrix}^{\mathrm{T}} ,$$

where s_k is the horizontal position (in m) of the cart and ϕ_k denotes the angle of the pendulum (in rad), chosen such that $\phi_k = \pi$ corresponds to the unstable upward equilibrium. The nonlinear, discrete-time dynamics of the pendulums are obtained by discretizing the corresponding continuous-time dynamics with sampling rate $t_{\mathrm{a}} = 0.01\,\mathrm{s}$. In both configurations, we employ the infinite-horizon, sequence-based controller from [20] to stabilize the pendulum over the network. The gain matrix is computed based on a linearization of the respective continuous-time dynamics around the upward equilibrium and a subsequent discretization with t_{a}.

To calculate the control input sequence, each controller reconstructs the corresponding pendulum state $\underline{x}_k$ using the estimator presented in [21] based on noisy measurements of the horizontal position and the pendulum angle. The measurements are taken at every time step by a sensor device attached to each pendulum and then transmitted to the respective controller.

Since we focus on the control-related part of the CPS and not on the communication system, and to facilitate the exposition, we restrict ourselves to a simple network. More precisely, we assume the communication system to be a realization of a discrete and stationary stochastic process with given probability mass function f, where $f(\tau)$ denotes the probability that a packet will experience a delay of τ time steps. That is, the actual delay of a packet to be transmitted is obtained by drawing a random number according to f. Furthermore, we assume that the delays of any two packets are independent of each other and interpret packet losses as infinite delays, i.e., the packet loss probability is given by $f(\infty)$.

III. PROBLEM FORMULATION

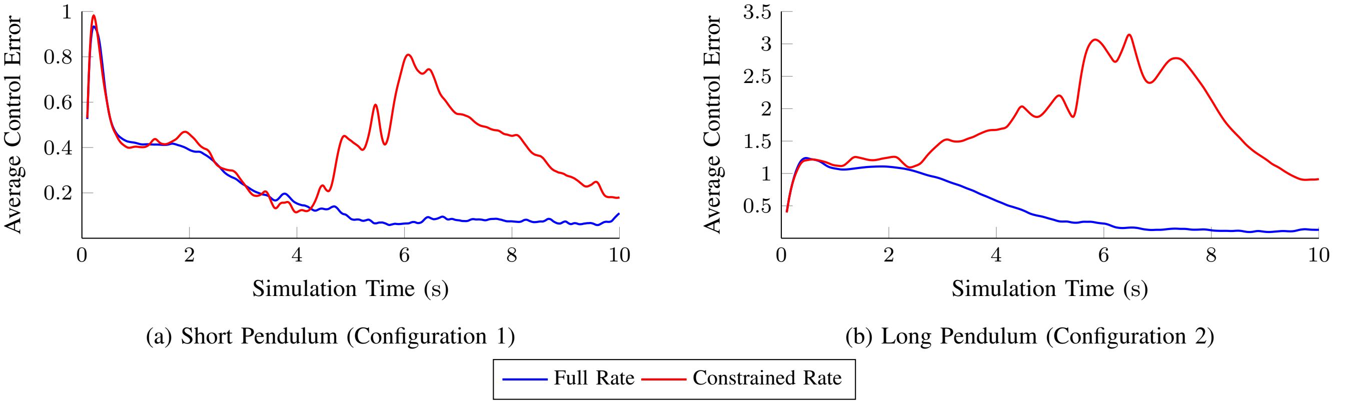
As mentioned in the introduction, the available bandwidth for control-related communication in a CPS is usually fluctuating. Since the control loops cannot know in advance what capacity will be available for their communication or what loss rates or delays are to be expected at what point in time during operation, they are not able to timely adapt their communication behavior, so that bottlenecks, i.e., overutilized network resources, will occur.

For the example CPS introduced in the previous section the impact of a bottleneck link, obtained by means of a Monte Carlo simulation with 200 runs, is shown in Fig. 3 for both configurations. In each run, the CPS was simulated over 1000 time steps, which corresponds to an operation time of $10\,\mathrm{s}$. To simulate a link with fluctuating capacity, we varied the rate r_k^{avail} available for packets sent out from both controllers according to

$$r_k^{\mathrm{avail}} = \begin{cases} 200 \text{ packets/s} & k \leq 100 \\ 140 \text{ packets/s} & 101 \leq k \leq 300 \\ 120 \text{ packets/s} & 301 \leq k \leq 700 \\ 180 \text{ packets/s} & 701 \leq k \leq 800 \\ 200 \text{ packets/s} & k > 800 \end{cases} . \tag{1}$$

That is, a bottleneck occurs after $1\,\mathrm{s}$ of operation that is completely removed only after $8\,\mathrm{s}$. For configuration 1 (short pendulum), the resulting control performance, indicated in terms of the control error, compared to the base case, where the full capacity of 200 controller packets per second is available, is visualized in Fig. 3a. Fig. 3b shows the control performance for configuration 2 (long pendulum). The control error shown in the figure is based on a *sum of the norm of the error* measure and obtained as follows. Using a fixed window size of $K = 10$ time steps, in each run we first compute the average cumulated norm of the control error according to

$$S_k = \frac{1}{K} \sum_{j=k-K+1}^{k} \|\underline{e}_j\| , \quad 10 \leq k \leq 1000 , \tag{2}$$



(a) Short Pendulum (Configuration 1)

(b) Long Pendulum (Configuration 2)

Fig. 3: Impact of fluctuating available communication bandwitdh r_k^{avail} on the control performance in both configurations compared to the unrestricted case.

where $\underline{e}_j$ is the control error at time step j, and then average over all runs.

Although having a decreased control performance in common, both control loops are affected differently by the bottleneck. While we observe that the bottleneck impairs the control performance in the second configuration almost directly from the beginning, the control performance in the first configuration remains roughly the same compared to the base case during the first four seconds of operation. The reason for this is that, due to the lack of a scheduling mechanism or predefined resource reservation, the controller for the short pendulum was still able to acquire a large enough portion of the available rate r_k^{avail}, while the second one was not, yielding an increased number of lost packets and hence an immediate performance decrease.

This nicely illustrates how the control loops in a CPS, even when operating independently, are implicitly coupled due to the shared network resources. No countermeasures can be taken because the control loops are not aware of shared network resources and the communication system has no a priori knowledge about the communication requirements of the different control loops. Thus, imbalances regarding the achievable control performance come about.

It is clear from the above discussions that the proper functioning of a CPS in the presence of shared network resources and potential bottlenecks requires a consensus be implemented among the control loops to maintain a high overall performance and, at the same time, avoid any performance imbalances. In the next section, we present a control approach to implement a consensus that evenly balances the utilization of network resources among the involved control loops with regards to their control performance.

The proposed approach will be based on a data exchange between the control loops and the communication system, which is carried out by the translator component shown in Fig. 1, and, motivated by the results in [19], will also heavily rely on event-based communication. As event-based information exchange is a natural way to reduce the transmitted amount of information, it also impacts the achievable control performance. Hence, it is an obvious aspect to be exploited for cooperation, i.e., to adapt the communication behavior of each control loop in such a way that the resource utilization is evenly balanced.

IV. PROPOSED APPROACH

As mentioned above, the proposed approach relies on event-based communication strategies. In this paper, we will utilize a simple threshold criterion, similar to the one proposed in [22], but our approach also works with other, more sophisticated event triggers such as the one presented in [23]. Picking an appropriate event trigger is of course a task that has to be carried out carefully by the system designer, as it affects the communication demand of the controller, i.e., the number of transmitted data packets, and, hence, as already indicated in the previous sections, the achievable control performance. Leveraging this relationship is the fundamental idea of our approach: Varying the event trigger in such a way that a certain, prescribed control performance is achieved results in a changed data rate, which in turn can resolve an existing bottleneck.

For the setup considered here, the relationship between the communication demand of the controller and the achievable control performance is illustrated in Fig. 4, where the control error as a function of the average packet rate employed by the controller, expressed as the average number of transmitted data packets per second, is depicted for both configurations.

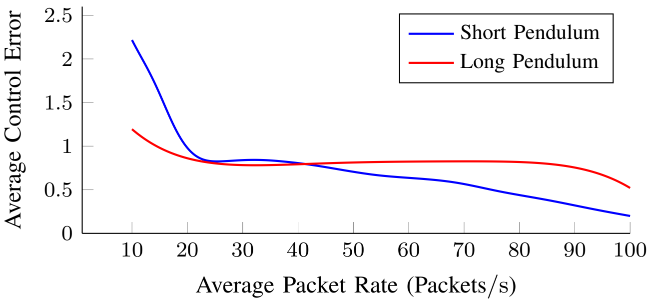


Fig. 4: Relationship between average packet rate employed by the controller and the average cumulated norm of the control error for both pendulum configurations.

For both configuration, the results were obtained by means of two Monte Carlo simulations, one for each configuration to exclude potential disturbances from unrelated traffic. In both simulations, different threshold values were employed by the event trigger and 50 runs were conducted per value. To measure the control error, we first again compute S_k according to (2) using a fixed simulation horizon of $N = 1000$ time steps and a window size of $K = 10$ time steps and then average over all runs. Finally, we average over all time steps. To get a continuous and smooth model, we additionally fit the data using cubic smoothing splines (short pendulum) and a fifth order polynomial (long pendulum). Note that the maximum rate shown in the figure (100 packets per second) corresponds to the base case, where the control inputs are transmitted periodically, that is, at every time step.

As expected, the curves exhibit that the control performance generally decreases with decreasing packet rate. However, they also show that the decrease is not uniform for the whole range of data rates under consideration. For the first pendulum, no significant performance decrease is observed for packet rates between 20 packets per second and 40 packets per second, while for the second one the achieved performance with 20 packets per second does not differ much from the one achieved with 80 packets per second. This indicates that a significant portion of the communication can be saved with only slight impact on the control performance.

For a similar setup, this observation was also made in [19]. Regarding small packet rates, i.e., packet rates less than 20 packets per second, we can also see from the curves that the control error in the first configuration increases significantly and attains much larger values at those than in the second configuration. This is a reasonable result since the rod of the first pendulum is smaller and lighter, resulting in a faster dynamics.

While the relationships are useful to determine what packet rate is required on average to achieve a certain control performance, they are not yet suitable for balancing the utilization of a shared resource. The reason is that control performance is typically a task and plant specific measure and hence not comparable among different control loops. Moreover, even if the same performance measure is used, the range of values will in general differ for different plant as can be seen in Fig. 4 for the two pendulums.

As a consequence, a normalization must be performed. This is done by the translator component. To that end, it is equipped with a function, which is provided by the system designer, that maps the application specific performance measure onto a value q in the unit interval $[0, 1]$. It is the task of the system designer to choose this function appropriately, such that q encodes both the control performance and the criticality of the control loop within the CPS.

If we denote the control errors in dependence of the average packet rate r from Fig. 4 by $S^{(1)}_{\text{avg}}(r)$ (short pendulum) and $S^{(2)}_{\text{avg}}(r)$ (long pendulum), a simple way to obtain a normalized measure of the control performance $q^{(i)}(r)$, referred to as *quality metric* (QM) in the following, is to rescale them according to the mapping

$$q^{(i)}(r) = \left| -1 + \frac{S^{(i)}_{\text{avg}}(r) - S^{(i)}_{\text{min}}}{S^{(i)}_{\text{max}} - S^{(i)}_{\text{min}}} \right| , \quad i = 1, 2 , \tag{3}$$

with

$$S^{(i)}_{\text{min}} = \min_{\tilde{r}} S^{(i)}_{\text{avg}}(\tilde{r}) ,$$
$$S^{(i)}_{\text{max}} = \max_{\tilde{r}} S^{(i)}_{\text{avg}}(\tilde{r}) ,$$

the minimum and maximum control error, respectively. The mapping (3) is shown in Fig. 5 for both pendulums. Note that the different communication requirements are reflected in the different slopes and x-intercepts of the curves.

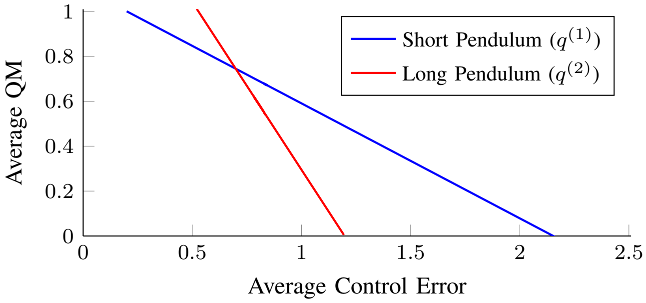


Fig. 5: Mappings employed to translate the control error into the normalized quality metric (QM) shown for both pendulum configurations.

Applying (3) to the curves from Fig. 4 yields the relationships depicted in Fig. 6.

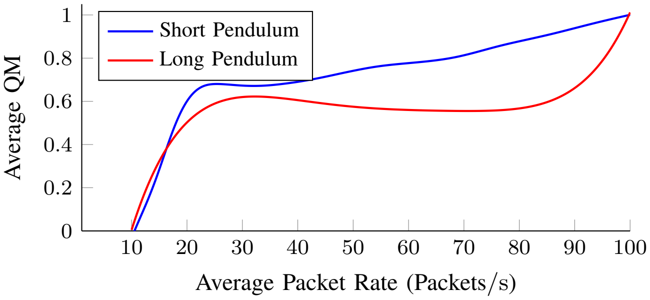


Fig. 6: Relationship between average packet rate and the average QM resulting after (3) is applied.

Each translator component now forwards the current, actual QM value and the relationship between control performance and packet rate i.e., the relationship shown in Fig. 6, to the underlying communication system. By using both pieces of information, the communication system in turn attempts to find target QM values $q^{(1)}_{\text{opt}}, q^{(2)}_{\text{opt}}$ that meet the desired requirements, namely a high overall control performance without imbalances among the control loops.

To that end, it maximizes the objective function $\mathcal{C} \colon [0, 1] \times [0, 1] \to [0, 1]$ given by

$$\mathcal{C}(q^{(1)}, q^{(2)}) = q^{(1)} q^{(2)} \exp\left(-\frac{1}{2} \frac{(q^{(1)} - q^{(2)})^2}{c^2} \right) , \tag{4}$$

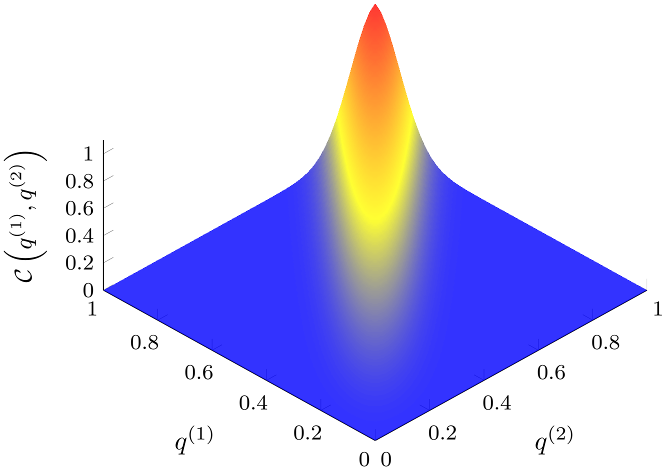


Fig. 7: Visualization of the function $\mathcal{C}$, with $c = 0.1$, to be maximized for the desired consensus.

that is visualized for $c = 0.1$ in Fig. 7. As can be seen from the peak in the figure, this function attains its largest values when $q^{(1)} \approx 1$ and $q^{(2)} \approx 1$, and also $q^{(1)} \approx q^{(2)}$ holds. Hence, it perfectly covers our requirements. Note that the parameter c can be used to control the broadness of the peak, or, in other words, to prescribe of how much importance the requirement of alike control performance is. With this function, the desired consensus can then found by the communication system by solving the optimization problem

$$\begin{aligned} & \max_{r^{(1)}, r^{(2)}} \mathcal{C}\left(q^{(1)}(r^{(1)}), q^{(2)}(r^{(2)})\right) \\ \text{s.t.} \quad & q^{(i)}\left(r^{(i)}\right) \in [0, 1]\,, \quad i = 1, 2\,, \\ & r^{(1)} + r^{(2)} = r^{\text{avail}}\,, \end{aligned} \tag{5}$$

where r^{avail} is the packet rate that is available for both controllers. To implement the consensus, the optimal values $q^{(1)}_{\text{opt}}$ and $q^{(2)}_{\text{opt}}$ that correspond to the maximizer of (5) are handed back to the translator components of the control loops. Each translator component then maps this value back onto the application specific performance or error measure using the inverse of (3) and forwards it to the controller, which in turn changes its event trigger accordingly, that is, such that the given control performance can be reached.

When this procedure is carried out each time the communication system detects a highly utilized resource or a significant change in the available bandwidth, overutilization and bottlenecks, and consequently, their impact on the control loops can be avoided.

V. EVALUATION

In this section, we demonstrate the proposed approach by means of the setup used in the foregoing sections. That is, we consider a CPS with the two inverted pendulums that shall be stabilized over a network whose capacity for the transmission of control inputs is varying according to (1).

To evaluate the performance of the proposed approach, we again carry out a Monte Carlo simulation with 200 runs, each of which comprising 1000 time steps. In each run, the controllers are initially configured such that the event trigger is disabled. Hence, they transmit the computed control inputs at every time step. Each time the available capacity of the network changes as per (1), the procedure introduced in the previous section is used to adapt the packet rates of the controllers to the new condition. To numerically solve the optimization problem (5), the parameter c in (4) is set to 0.1.

For comparison, we conduct a second experiment with the same number of runs and time steps per run, where, in each run, a simple scheduling strategy is employed. More precisely, each controller is allowed to transmit its computed control inputs periodically at every second time step, so that the available capacity is shared evenly while overutilization is avoided. As a consequence, better control performance can be expected, however, at the cost of reduced flexibility of the whole CPS.

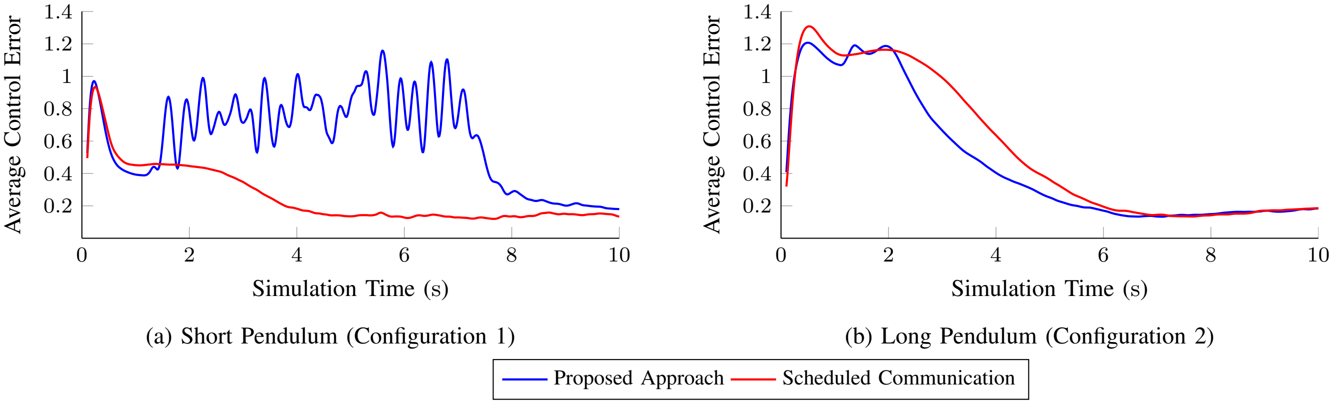
The results of the simulations are given in Fig. 8 and Fig. 9, where the average control errors, computed according to (2), and the corresponding average QM values, computed according to (3), over time are plotted.

While both approaches are able to significantly decrease the control error of the second control loop (long pendulum) compared to the case where no action was taken to avoid the bottleneck (cf. Fig. 3b), the proposed approach performs better than the one with predefined scheduling. However, for the first control loop (short pendulum), the opposite is true. Here, the scheduling scheme performs better than the proposed approach. Since our approach targets a cooperative resource utilization, this is comprehensible. To increase the performance of a control loop in case of a bottleneck, the performance of the other must be decreased in order to balance the control performance among the control loops. This outcome of the experiments indicates that the proposed approach is indeed promising to implement collaboration between the individual control loops.

From the results in Fig. 9 we can see that the proposed approach enables both control loops to achieve a high quality of control, say ≥ 0.8, already within less than $8\,\mathrm{s}$, i.e., at a time when the available bandwidth is lowest. While this also holds true for the scheduled communication scheme, this fact also underlines the usefulness of our approach since, in contrast to the former, adaptations cannot become necessary if the available bandwidth changes differently.

VI. CONCLUSIONS

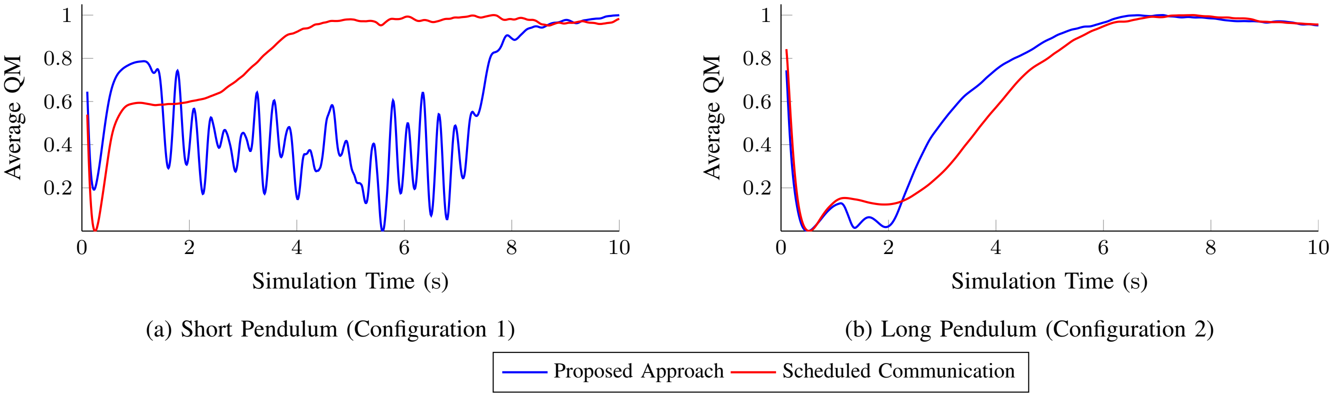
In this paper, we presented a cooperative approach to balance the utilization of shared network resource in cyber-physical systems where neither a fixed scheduling nor an a priori resource reservation is implemented. Being based on a data exchange between two domains, i.e., control and communication, that is carried out by a translator component placed in the middle, the proposed approach avoids a tight coupling between the two domains. By combining the mentioned data exchange with the flexibility offered event-based control, we formulated



(a) Short Pendulum (Configuration 1)

(b) Long Pendulum (Configuration 2)

Fig. 8: Average control error for the proposed approach and the scheduled communication scheme with fluctuating bandwidth r_k^{avail}.



(a) Short Pendulum (Configuration 1)

(b) Long Pendulum (Configuration 2)

Fig. 9: Average normalized control performance (QM) for the proposed approach and the scheduled communication scheme with fluctuating bandwidth r_k^{avail}.

the considered problem in terms of a consensus problem that is solved by the communication system. We demonstrated the usefulness of the proposed approach using a very simple communication system.

Future work will be concerned with investigating the presented approach in the presence of more realistic network configurations. To that end, we plan to fully incorporate it into our open source simulation and evaluation framework CoCPN-Sim [14], [24].

However, in order to take the dynamics of the network into account, the proposed consensus must be computed in a distributed fashion by the intermediate nodes of the network, e.g., by the routers. Consequently, this will result in an distributed optimization problem that is different from (5).

Moreover, a more sophisticated implementation should also take the actual operating point of the controllers into account when the relationships between control performance and communication behavior are derived and not only consider average or long term behavior. Obtaining such relationships for the two pendulums and more involved event-triggers will constitute another important line of research. In this regard, more complex regression and learning approaches will be considered, so that, as can be expected, the resulting functions are both smooth and at least non-increasing.

Finally, future research should be concerned with finding conditions for stability of the proposed approach.

ACKNOWLEDGMENT

This work is supported by the German Science Foundation (DFG) within the Priority Programme 1914 “Cyber-Physical Networking”.

REFERENCES

[1] S. Karnouskos, “Cyber-Physical Systems in the SmartGrid,” in *Proceedings of the 2011 9th IEEE International Conference on Industrial Informatics*, July 2011, pp. 20–23.
[2] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, “Intelligent Manufacturing in the Context of Industry 4.0: A Review,” *Engineering*, vol. 3, no. 5, pp. 616 – 630, 2017.
[3] P. J. Mosterman and J. Zander, “Industry 4.0 as a Cyber-Physical System Study,” *Software & Systems Modeling*, vol. 15, no. 1, pp. 17–29, Feb 2016.
[4] S. K. Khaitan and J. D. McCalley, “Design Techniques and Applications of Cyberphysical Systems: A Survey,” *IEEE Systems Journal*, vol. 9, no. 2, pp. 350–365, June 2015.
[5] P. Hehenberger, B. Vogel-Heuser, D. Bradley, B. Eynard, T. Tomiyama, and S. Achiche, “Design, Modelling, Simulation and Integration of Cyber Physical Systems: Methods and Applications,” *Computers in Industry*, vol. 82, pp. 273 – 289, 2016.
[6] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, “A Survey of Recent Results in Networked Control Systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
[7] X. M. Zhang, Q. L. Han, and X. Yu, “Survey on Recent Advances in Networked Control Systems,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1740–1752, Oct. 2016.

[8] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, “Foundations of Control and Estimation over Lossy Networks,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[9] W. M. H. Heemels, A. R. Teel, N. Van de Wouw, and D. Nesic, “Networked Control Systems with Communication Constraints: Tradeoffs between Transmission Intervals, Delays and Performance,” *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1781–1796, 2010.

[10] R. Poovendran, “Cyber–Physical Systems: Close Encounters Between Two Parallel Worlds [Point of View],” *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1363–1366, Aug 2010.

[11] A. T. Al-Hammouri, “A Comprehensive Co-Simulation Platform for Cyber-Physical Systems,” *Computer Communications*, vol. 36, no. 1, pp. 8 – 19, 2012.

[12] D. Quaglia, R. Muradore, R. Bragantini, and P. Fiorini, “A SystemC/Matlab Co-Simulation Tool for Networked Control Systems,” *Simulation Modelling Practice and Theory*, vol. 23, pp. 71 – 86, 2012.

[13] W. Li, X. Zhang, and H. Li, “Co-Simulation Platforms for Co-Design of Networked Control Systems: An Overview,” *Control Engineering Practice*, vol. 23, pp. 44 – 56, 2014.

[14] M. Jung, F. Rosenthal, and M. Zitterbart, “Poster Abstract: CoCPN-Sim: An Integrated Simulation Environment for Cyber-Physical Systems,” in *Proceedings of the 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2018, pp. 281–282.

[15] D. Baumann, F. Mager, H. Singh, M. Zimmerling, and S. Trimpe, “Evaluating Low-Power Wireless Cyber-Physical Systems,” in *Proceedings of the 2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, April 2018, pp. 13–18.

[16] S. Gallenmüller, S. Günther, M. Leclaire, S. Zoppi, F. Molinari, R. Schöffauer, W. Kellerer, and G. Carle, “Benchmarking Networked Control Systems,” in *Proceedings of the 2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*, April 2018, pp. 7–12.

[17] J. Lee, B. Bagheri, and H.-A. Kao, “A Cyber-Physical Systems Architecture for Industry 4.0-based Manufacturing Systems,” *Manufacturing Letters*, vol. 3, pp. 18 – 23, 2015.

[18] L. Ribeiro, “Cyber-physical Production Systems’ Design Challenges,” in *Proceedings of the 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, June 2017, pp. 1189–1194.

[19] F. Rosenthal, M. Jung, M. Zitterbart, and U. D. Hanebeck, “CoCPN – Towards Flexible and Adaptive Cyber-Physical Systems Through Cooperation,” in *Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC) (CCNC 2019)*, Las Vegas, USA, Jan. 2019.

[20] J. Fischer, M. Dolgov, and U. D. Hanebeck, “On Stability of Sequence-Based LQG Control,” in *Proceedings of the 52st IEEE Conference on Decision and Control (CDC 2013)*, Florence, Italy, Dec. 2013.

[21] F. Rosenthal, B. Noack, and U. D. Hanebeck, “State Estimation in Networked Control Systems with Delayed and Lossy Acknowledgments,” in *Multisensor Fusion and Integration in the Wake of Big Data, Deep Learning and Cyber Physical System*, S. Lee, H. Ko, and S. Oh, Eds. Cham: Springer International Publishing, 2018, pp. 22–38.

[22] Y. Zhao, G. Liu, and D. Rees, “Packet-Based Deadband Control for Internet-Based Networked Control Systems,” *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1057–1067, Sept 2010.

[23] M. Dolgov, J. Fischer, and U. D. Hanebeck, “Event-based LQG Control over Networks Subject to Random Transmission Delays and Packet Losses,” in *Proceedings of the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys 2013)*, Koblenz, Germany, Sep. 2013.

[24] M. Jung and F. Rosenthal, “CoCPN-Sim,” 2018. [Online]. Available: https://github.com/spp1914-cocpn/cocpn-sim