2019 IEEE International Conference on Edge Computing (EDGE)

Cloud-Assisted Model Predictive Control

Per Skarin*†, Johan Eker*†, Maria Kihl‡, Karl-Erik Årzén*
*Department of Automatic Control, Lund University, Sweden
†Ericsson Research, Lund, Sweden
‡Department of Electrical and Information Technology, Lund University, Sweden

***Abstract*—In this paper we present a computational offloading strategy with graceful degradation for executing Model Predictive Control using the cloud. We show a method which allows for seamless control assistance and design of flexible controllers using the edge cloud. We examplify using a cyber-physical-system at high frequency and illustrate how the system can be improved while keeping the computational cost down.**

***Index Terms*—Cloud, Edge, Time-sensitive, Mission-critical, Model Predictive Control, Control theory, Cyber-physical**

I. Introduction

Developments in edge computing are paving the way for control of critical systems over the cloud [1], [2] Telecommunication companies are working on edge networks integrated in the 5G Radio Access Networks (RANs), acting alongside their Network Function Virtualisation (NFV) infrastructures [3], [4]. Through wireless ultra reliable low latency communication (URLLC) [5], the telecommunication industry is providing an alternative to wired communication for time sensitive systems.

The potential of these developments invevitably leads to the consideration of networked control systems operated over the cloud [6], [7], [8]. This domain also deals with distributed methods of control [9], [10] which can make use of the abundance of resources and efficient interconnects if applied inside of data centers. A limitiation to approaches of applying control theory in and over the cloud is that traditional control design tends to be used. Attempting to control complex systems with small amounts of data and limited degrees of freedom or replacing Programmable Logic Controllers (PLCs) with virtualized counterparts does not take full advantage of the cloud as an infrastructure and software platform.

The illusion of infinite compute and storage resources that the cloud and the edge/fog provides opens up a number of interesting possibilities for control applications. The resources can be used for executing more advanced control strategies, e.g., based on online optimization and learning using massive data sets, than what is possible on the local device. The cloud can scale resources with the problem and implement efficient strategies for each computation. This allows the controller to evaluate complex problems which are too computationally demanding to perform locally. The drawback of moving online computations into the cloud is the additional delays and uncertainty that it creates. Using edge clouds is one way to manage and mitigate these problems. In parallel and complementary, domain specific application design must make its case to build trust and efficiency into the systems.

In this work we introduce an assisted mode controller which operates on the device, at the edge and in the cloud. Such an assisted mode controller is considered from the perspective of low-cost, resource constrained equipment in a cyber-physical-system. The presented approach uses the cloud to execute a number of model-based optimizations, each with different optimization parameters. The result is twofold. First, the limited local system is extended with an advanced controller through the cloud. Second, the advanced controller can be made powerful and flexible through its implementation in the cloud. The local controller facilitates stability and graceful degradation in case of connectivity loss. The proposed method shows how edge clouds can be used effectively and efficiently for control applications.

II. Method

A. Targeted system

The targeted system is illustrated in Figure 1. The system is composed of a local plant controlled by our proposed assisted mode controller (placed in a so called device, for example a local CPU). The assisted mode controller can use one data center cluster and one edge cluster for processing.

The assisted mode controller is constructed using a Model Predictive Control, MPC [11], based controller in combination with a Linear Quadratic Regulator, LQR [12]. The MPC is computationally heavy while the LQR is light weight. The device is assumed uncapable of executing the optimizations that are necessary for the MPC, which is therefore offloaded to the cloud. The data center can run several variations of optimizations, providing elastisity of the controller. The edge is costly and allowed only one optimization. In addition the optimization at the edge should use as little computational time as possible.

The operation of the MPC depends on the reponses from the network, which are affected by communication delays and execution times. In this work, communication with the data center suffers variable delay while the edge is simplified to a static delay.

B. Control design

The LQR is a special case of the MPC and it is straightforward to generate a LQR from the MPC specification. The same calculations in combination with a *terminal set* can also be used to fullfil the stabilization and feasibility criteria for the MPC. For the interested, details on these control theoretical

978-1-7281-2708-8/19/$31.00 ©2019 IEEE
DOI 10.1109/EDGE.2019.00033

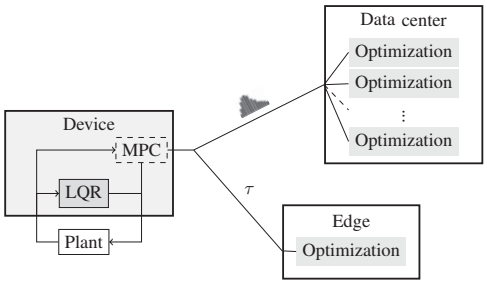

Fig. 1: System setup. The device is capable of operating an LQR but uses the cloud for the heavy computations of the MPC.

aspects are presented in [13] and are also available in common literature on the subject of MPC.

Equation (1) provides a mathematical representation of a discrete-time linear MPC. This numerical optimization is executed with each sample. The MPC uses a quadratic cost function $l(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$, a cost P applied to the final state (refered to as the *terminal cost*), a system model defined by matrices A and B, and inequality and equality constraints set by the matrix vector pairs G, g and H, h respectively. The cost matrix Q penalizes moving away from the desired state while R penalizes the control signal. $\mathcal{T}$ is called the *terminal set* and forces the final state such as to ensure the stability of the controller. An MPC uses the model to predict the future evolution of the system. The number of anticipates steps is referred to as the horizon and denoted N.

$$\begin{aligned} \underset{\mathbf{u}}{\text{minimize}}\ J = & \sum_{i=k}^{k+N-1} x_i^T Q x_i + u_i^T R u_i + x_{k+N}^T P x_{k+N} \\ \text{subject to} \quad & x_{i+1} = A x_i + B u_i \\ & G \begin{bmatrix} x_i \\ u_i \end{bmatrix} \leq g,\ H \begin{bmatrix} x_i \\ u_i \end{bmatrix} = h,\ x_{n+k} \in \mathcal{T} \end{aligned} \tag{1}$$

It is important to note that this problem may have no solution if the horizon N is too small, that computations are heavier with a larger N and that the necessary size of N is unknown.

Through exclusion of the constraints, the specification can be used to create an LQR. The LQR generates a control signal through the matrix operation in Equation (2). Here K is the gain vector obtained through solving the textbook LQR equations.

$$u_k = -K x_k \tag{2}$$

The LQR is executed locally while several instantiations of the MPC are evaluated in the cloud.

C. Evaluation

The approach is evaluated using a simulated ball and beam process

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{-5g}{7} \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ k \end{bmatrix}, C = \begin{bmatrix} \frac{10}{l} & 0 \\ 0 & 0 \\ 0 & \frac{40}{\pi} \end{bmatrix}^T \tag{3}$$

where $k = 0.44$ is a motor constant, $l = 0.55$ is the length of the beam and $g = 9.80665$ is the gravitational constant. The measured variables are the beam angle and the ball position. The control signal is the voltage to the motor that tilts the beam. The sampling frequency is 20 Hz, i.e., the MPC calculations must be returned within 50 ms. The simulation is implemented in Matlab using Simulink and TrueTime, a Simulink toolbox for simulating distributed real-time systems with real-time kernels and networks [14]. The MPC is implemented using the Matlab *quadprog* command.

The infrastructure has two compute locations, the backend data center $\mathcal{C}_{DC}$ and an edge node $\mathcal{C}_E$. The data center can carry any amount of calculations while $\mathcal{C}_E$ is limited to one. It is further assumed that execution time on the edge comes at a significantly higher cost. Response selection is then performed by selecting N. A small N keeps the load down at the edge but a large N must be recovered on set-point changes. The data center keeps running the heavier computations.

The performance of the controller is evaluated using a nominal system without external disturbances and noise. Random delays are applied in the connection to $\mathcal{C}_{DC}$ (Figure 3). The connection to $\mathcal{C}_E$ is simplified to a constant delay of $40\,\mathrm{ms}$. Execution time of the optimizations vary from a single millisecond to $20\,\mathrm{ms}$. .

III. Results

In this section results are presented for optimal control assisted from the edge and the cloud. The examples show simulations of local control, control over a network with disturbances and control over the network with experienced connection loss. The plots in Figure 2 all show the position of the ball where ± 0.55 is the beam endpoints which in all cases will cause operational failure if violated. Clearly, the standard LQR in Figure 2-I fails the simulation in this respect. The dashed trajectory in Figure 2-I is using the same controller gains (K in Equation (2)) but stays within constraints through a conservative error limited by $abs(x_{sp} - x_k) \leq 0.4$. This controller safely operate the reference changes but is limited in its performance. This limited LQR is used as local control and assisted mode is introduced.

In Figure 2-II the trajectory has been replaced by the assisted controller with a dashed LQR trajectory for reference. This controller operates as shown in Figure 1 with relatively large random delays towards the data center and passing only the least expensive request to the edge. The assisted controller outperforms the non-assisted over the full range although on occasion the local LQR has a better response. Figure 2-III shows the effect of a 1.8 seconds connection loss, indicated by the time interval in gray. In this case, the edge is not used and the trajectory is also slightly different with connectivity. When connection is temporarily lost towards the data center the assisted controller enters the gradual switching mode and eventually runs in pure LQR mode until connectivity is restored.

IV. Conclusion

We have proposed and demonstrated a strategy for control systems operated using the cloud. In difference to the example

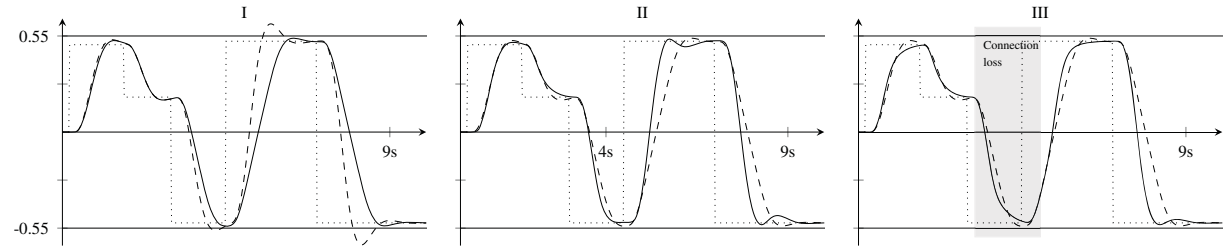

Fig. 2: I: Conservative LQR within constraints and standard LQR violating constraints (dashed). II: Control using on an edge cloud and local LQR only (dashed). III: Connection loss and local LQR only (dashed).

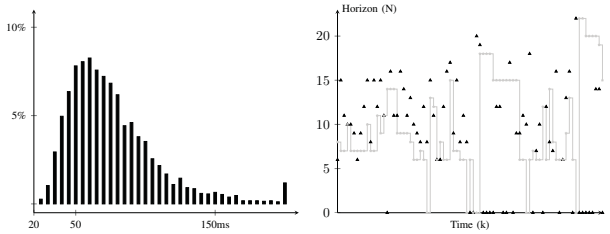

Fig. 3: Left: Latency distribution (log-normal with $\mu = 4$, $\rho = 0.5$ and an offset of $14\,\mathrm{ms}$). Right: Example of selected responses. With edge (gray stars) and without edge (black triangels).

in our previous work [15] this method aims to extend a local controller whenever cloud connectivity is present. We show how this can be achieved using a combination of unconstrained and constrained control, in the form of LQR, and MPC.

We evaluated the approach on a simulated ball and beam process to show that an assisted controller can improve performance while being robust to connectivity issues. The results validate the transition to and from assisted mode, and that operation is smooth even though the controller switches between various MPC horizons. In both cases of assisted control the performance is improved and the trajectory stays away from the constraints. This was combined with an edge strategy to reduce the impact of long delays.

Edge cloud approaches opens up for interesting control designs when elasticity is introduced and trade offs are mede between computation and transport delays. Interesting further work exists also in the specific case, e.g., empirical studies of extended evaluations and implementations, improvement of the assisted approach in terms of a larger range of disturbances, tracking methods, and request/response selection methods. There is also future work that relates to the presented approach, e.g., the variability in the definition of the MPC requests, incorporating larger problem spaces, advanced implementations in the cloud back-end, and scheduling heuristics in the cloud.

See [13] for further control theoretical discussions, more details in the design and additional results from the work presented in this paper.

Acknowledgements

This work is partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, SEC4FACTORY project SSF RIT17-0032 funded by the Swedish Foundation for Strategic Research (SSF), and the HI2OT University Network on Industrial IoT funded by Nordforsk. The authors are part of the Excellence Center at Linköping-Lund in Information Technology (ELLIIT).

References

[1] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, February 2019.

[2] G. Klas, "Edge computing and the role of cellular networks," Computer, vol. 50, no. 10, pp. 40–49, 2017.

[3] C. Boberg, M. Svensson, and B. Kovács, "Distributed cloud: A key enabler of automotive and industry 4.0 use cases," Ericsson Technology Review, Tech. Rep. 11, November 2018. [Online]. Available: https://www.ericsson.com/assets/local/publications/white-papers/wp-5g-systems.pdf

[4] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing - a key technology towards 5g," ETSI White Paper No. 11, 9 2015.

[5] Ericsson, "5G radio access - capabilities and technologies," Ericsson, Tech. Rep., 2016. [Online]. Available: https://www.ericsson.com/assets/local/publications/white-papers/wp-5g.pdf

[6] O. Givehchi, J. Imtiaz, H. Trsek, and J. Jasperneite, "Control-as-a-service from the cloud: A case study for using virtualized PLCs," in 2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014), May 2014, pp. 1–4.

[7] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam, "Delay mitigation in offloaded cloud controllers in industrial IoT," IEEE Access, vol. 5, pp. 4418–4430, 2017.

[8] T. Hegazy and M. Hefeeda, "Industrial automation as a cloud service," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 10, pp. 2750–2763, 2015.

[9] A. Richards and J. P. How, "Robust distributed model predictive control," International Journal of control, vol. 80, no. 9, pp. 1517–1531, 2007.

[10] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," Computers & Chemical Engineering, vol. 51, pp. 21–41, 2013.

[11] J. Rawlings and D. Mayne, Model Predictive Control: Theory and Design. Nob Hill Pub., 2009.

[12] R. E. Kalman, "Contributions to the theory of optimal control," Boletin de la Sociedad Matematica Mexicana, vol. 5, no. 2, pp. 102–119, 1960.

[13] P. Skarin, J. Eker, M. Kihl, and K.-E. Årzén, "An assisting Model Predictive Controller approach to Control over the Cloud," arXiv e-prints, p. arXiv:1905.06305, May 2019.

[14] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Årzen, "How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime," IEEE control systems magazine, vol. 23, no. 3, pp. 16–30, 2003.

[15] P. Skarin, W. Tärneberg, K.-E. Årzen, and M. Kihl, "Towards mission-critical control at the edge and over 5G," in 2018 IEEE International Conference on Edge Computing (EDGE), July 2018, pp. 50–57.