A cloud-based operation optimization of building energy systems using a hierarchical multi-agent control

Alexander Kümpel, Thomas Storek, Marc Baranski, Markus Schumacher and Dirk Müller
RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustraße 10, 52072 Aachen, Germany
E-mail: akuempel@eonerc.rwth-aachen.de

Abstract. This work presents an agent-based control concept that we integrate into a cloud framework for controlling building energy systems. The agents are arranged in a hierarchical structure, where a coordinator agent sends optimized set point values to sub-agents. Each sub-agent controls a subsystem in order to reach the given set points and provides the coordinator agent with a cost function for the overall set point optimization. The multi-agent system and the building energy system exchange data via the cloud framework FIWARE. The components of the building energy system (e.g. boiler, air-handling unit) are connected to the cloud framework and send measurements (e.g. temperature values or volume flow) via a scalable IoT-Interface to the corresponding data object in the platform. The sub-agents of the agent control receive the measurements and send the calculated control action back to the corresponding components of the energy system. In a simulation study, we use the framework to control an air-handling unit. The aim of the agent control is to reach a given supply air temperature while reducing the total consumed energy. The implemented cloud-based control is capable of efficiently reaching the given temperature set points while communicating via the IoT interface.

1. Introduction
Around one third of the total worldwide final energy consumption rise in the building sector [1]. Hence, the building sector offers high potentials for reducing global energy consumption and CO₂ emissions by increasing the efficiency of building energy systems. To exploit the potentials, the building energy systems become more complex due to the integration of renewable energy sources and more efficient components. As a consequence, a high amount of different components, especially in non-residential buildings, have to be controlled efficiently in order to exploit the saving potentials. Standard control strategies, such as On/Off control, often lead to non-optimal building operation. A promising control approach for complex energy systems is the model predictive control (MPC) [2]. However, the modeling effort can be high, especially for large energy systems [3]. Further, complex models result in high computation times. Here, distributed model predictive control or agent-based approaches that split the overall system into subsystems lead to promising results [4]. Additionally, using specific agents for each type of subsystem can decrease the overall modeling effort.

Besides the development of an efficient control, the implementation of the communication...
between the controller and energy system can be a challenging task. Especially in large buildings, a high amount of different sensors and actuators have to be integrated. Here, the Internet of Things (IoT) offers the possibility to connect a high amount of devices and use cloud computing for complex control tasks.

In this paper, we present an agent-based control approach that communicates with the energy system via a cloud platform. The agent-based approach is presented in chapter 2.1 and the integration into the cloud framework in chapter 2.2. In order to demonstrate the cloud-based control, an air-handling unit is controlled in a simulation (chapter 3). Chapter 4 concludes this work.

2. Method
The first part of this section presents the agent based control concept that has been developed in [5]. The second part deals with the integration of the control concept into the cloud platform FIWARE.

2.1. Agent based control
The basic idea of the agent based control concept is to divide the overall system into subsystems, which are controlled by agents. In the following, the agents controlling a subsystem are called sub-agents. The connections between the subsystem are called coupling variables, which can be, for instance, mass flow rates, temperatures or pressures. Each sub-agent aims to reach given set points for the coupling variables while reducing a local cost function, e.g. the energy costs. In order to operate the overall system in an efficient way, a coordinator agent calculates optimized set points for the coupling variables. The objective function of the set point optimization is the sum of the local cost of each subsystem. The proposed structure leads to problem formulation given in equation 1, which has the same structure as the primal decomposition formulation [6]. The coordinator minimizes the cost function \( f^*(x) \), where \( x \) denote the coupling variables of the subsystems. The cost function is the sum of the minimal local costs \( f_i^*(x) \) for given coupling variables \( x \) of each subsystem \( i \). The variable \( u_i \) denotes the control action that each sub-agent can perform. The coupling variables \( x \) and the control action \( u \) has to be part of the feasible set \( X \) and \( U \).

\[
\begin{align*}
\min_x f^*(x) &= \sum f_i^*(x), \\
\text{with} \quad f_i^*(x) &= \min_{u_i} f(x, u_i), \\
x &\in X \\
u &\in U
\end{align*}
\] (1)

In contrast to iterative solving methods for equation 1, where the local cost functions are calculated in each iteration until convergence, we propose the generation of the cost functions \( f_i^*(x) \) on the feasible set \( X \) in a single step: In an initialization phase, each sub-agent generates the corresponding cost function and passes it to the coordinator agent. In order to generate the cost function, the sub-agent calculates the operation costs for different sets of the coupling variables and stores the results in a lookup table. A simulation model including the closed-loop control of the corresponding sub-agent is used for this purpose. We apply a simulation of the closed-loop system in order to enable the implementation of advanced algorithms, such as model predictive control (MPC), as well as PID or rule-based control within the sub-agents. The latter are less complex and could lead to near optimal operation for simple subsystems as well, if the controller is well tuned. After the generation of the lookup table, the sub-agents generate the function \( f_i^*(x) \) based on a n-dimensional piece-wise linear interpolation of the costs.
stored in the lookup table. Additionally, the sub-agent approximates the feasible set $X$ using the smallest convex hull of the stored data. In order to assure a feasible solution of the coordinator’s optimization problem in each time step, the feasible set is defined by soft constraints.

2.2. Cloud Framework

The communication between the controller and the building energy system is realized with a cloud framework. Using the internet as communication interface enables the integration of the components of the energy system as IoT devices. Further, cloud computing resources can be utilized for demanding control tasks. The implemented framework is illustrated in Figure 1 and consists of the controller, the open-source framework FIWARE [7] and the building energy system. The controller consists of a coordinator and one sub-agent for each subsystem of the energy system as described in section 2.1. The coordinator calculates the optimal set points, whereas the sub-agents control the subsystems by sending control actions and receiving measurements from FIWARE. The core of FIWARE is the Orion context broker that gathers and manages context information. The context broker enables the registration of entities with different attributes in a structured manner. Groups of entities can be handled separately by defining specific services and service paths. The short-term history of context data is stored in a MongoDB database. In order to communicate with IoT devices, the framework includes an IoT Agent which serves as an interface to the context broker. IoT devices can be registered using the IoT Agent, which informs the context broker to register a new entity. In our set-up of the framework, we use a Mosquitto MQTT broker, which communicates with the IoT Agent and IoT devices via the MQTT protocol (Message Queuing Telemetry Transport). The IoT devices, e.g. sensors and actuators, are part of the energy system and exchange MQTT messages with the FIWARE platform.

In order to send control actions (e.g. a valve opening) to the energy system, a sub-agent posts a command via HTTP to the context broker. The context broker passes the command to the IoT Agent, which publishes the command to the MQTT broker under a specific topic. The MQTT topic consists of the entity name and the attribute name of the registered device and an API key that can be defined for each service and service path. The IoT devices of the energy system subscribe the corresponding topics to receive the control action. Sensor measurements are published by the IoT devices via MQTT with a specific topic that the IoT Agent subscribes. If the IoT Agent receives a new message, it informs the context broker by passing the corresponding message value. Via a HTTP request to the context broker, the sub-agents are able to get the sensor measurements send by the energy system.

![Figure 1. Cloud-based control framework.](image-url)
3. Use-case
We applied the cloud-based control to a simulated air-handling unit (AHU). The AHU consists of the subsystems heat recovery system (HRS), preheater, cooler and reheater (Figure 2). Each subsystem has a pump and a valve that is controlled by the corresponding sub-agent. In order to reduce the electricity consumption of the pumps in the subsystems, we implement the control strategy presented in [8]. The control strategy includes two control modes: in part load, the valve is controlled by a PID controller and the pump operates at minimal frequency. For high energy demand, the valve is fully opened and the pump frequency is varied by a PID controller. The local cost function that each sub-agent generates includes the thermal and the electrical energy consumption. The coordinator agent optimizes the outflow air temperature set points of the subsystems in order to reduce the total energy consumed by the air-handling unit. To reach a given supply air temperature set point, the overall cost function of the coordinator includes a quadratic penalty function, which punishes the deviation from a given set point. In the simulation, we chose 20°C as supply temperature set point. For simplicity, the air volume flow and the return air temperature is constant at 3000 m\(^3\)/h and 22°C. The temperature of the outside air correlates to a typical meteorological year (TMY3).

![Controlled air-handling unit](image_url)

**Figure 2.** Controlled air-handling unit.

The FIWARE framework runs in docker containers on a Linux server. The agent control is implemented in python and runs on a workstation. For the simulation, a model of the AHU is exported as functional mockup unit (FMU) [9] and simulated with the python package PyFMI [10]. The simulation period is one week in summer. Since the dynamic behavior of the system is slow and the time constants are in the range of several minutes, we chose a time step length of 30 seconds: every time step, the control action for the valves and pumps are calculated and passed to the FMU that performs a 30 seconds simulation step after. The coordinator optimizes the overall cost function every 3600 seconds.

Figure 3 illustrates the simulation results. The supply temperature, which is the outflow temperature of the reheater, is close to the set point of 20°C. However, the exact temperature is not reached most of the time. Since the control aims to reduce the consumed energy, the supply temperature deviates from the set point until a trade off between the punishment for deviating from the set point and the energy consumption is reached. If the outside temperatures is below 20°C, the supply temperature is around 19.5°C. In this case, the heat recovery system heats the outside air as much as possible. The preheater heats the air flow to the set point, while
the cooler and reheater are not used. At higher outside temperatures, the supply temperature is increased in order to reduce the energy consumption needed for cooling. The heat recovery system cools the airflow, if the outside air temperature is above the return air temperature of 22 °C. The cooler serves to cool the air for the remaining temperature difference, while the preheater and reheater are not used. However, when the outside air temperature changes quickly, the system starts to oscillate and the AHU heats and cools simultaneously (e.g. at the end of the first day).

![Figure 3. Results of the simulation for one week.](image)

In order to evaluate the effect of the sampling rate on the control quality, we run the simulation for a shorter period with a time step length of 5 seconds. Figure 4 shows the comparison between the simulation with 30 seconds and 5 seconds time step length. As expected, the shorter time interval leads to a higher control quality with less oscillations.

![Figure 4. Comparison between simulations with different time step length.](image)
The presented control leads to an efficient operation of the air-handling unit. At high and low outside air temperatures, the supply temperature is adjusted until a trade-off between energy consumption and control quality is reached. The integration of the control into the cloud framework FIWARE enables the communication with the energy system via the internet. For a real-world application, the execution interval of the closed-loop control could be less than a second to further improve the control quality. However, if the system behavior is slow, larger time intervals, e.g. 5 to 30 seconds, can lead to a sufficiently accurate control, as shown in the use-case. Therefore, IoT devices that have an internet connection with limited data rates, e.g. radio-based devices, or limited battery life time can be used in building automation systems. By increasing the time interval between sending/receiving data, the data rate as well as the energy consumption of the IoT devices can be reduced. Further, IoT enables the use of high computational power of cloud computing in order to solve complex tasks, such as the cost function generation and optimization of the presented agent-based control.

4. Conclusion
This paper presents an agent-based control that communicates via a cloud framework with the building energy system. The agents are arranged in a hierarchical structure, where a coordinator agent calculates optimized set point values. Each sub-agent controls a subsystem in order to reach the given set points and provides the coordinator agent with a cost function for the overall set point optimization. The communication between the sub-agents and the subsystems of the energy system is realized with the cloud platform FIWARE. In a simulation, we implemented the framework in order to control an air-handling unit. The control leads to an efficient operation of the system. However, the time interval for the communication affects the control quality. Nevertheless, a time interval of several seconds can be sufficient for systems with slow dynamics, as shown by the simulation results. Therefore, radio-based communication with limited data rates could be used to control the system. Further research will investigate a real-life application of the presented control framework using radio-based communication and exploiting the performance of cloud computing.

Acknowledgments
We gratefully acknowledge the financial support provided by the BMWi (Federal Ministry for Economic Affairs and Energy), promotional references 03ET1561B.

References
[1] Abergel T, Dean B, Dulac J Towards a zero-emission, efficient, and resilient buildings and construction sector. Global Status Report 2017. UN Environment and International Energy Agency
[2] Afram A, Janabi-Sharifi F 2014 Theory and applications of HVAC control systems—A review of model predictive control (MPC) Build. Environ. 72 343-355
[3] Privara S, Cigler J, Váňa Z, Oldewurtel F, Sagerschnig C and Žáčeková E 2013 Building modeling as a crucial part for building predictive control Energ Buildings 56 8-22
[4] Christofides P D, Scattolini R, de la Pena D M and Liu J 2013 Distributed model predictive control: A tutorial review and future research directions COMPUT CHEM ENG 51 21-41.
[5] Kümpel A, Baranski M, Müller D 2019 Hierarchical multi-agent control of HVAC systems Proc. of ECOS 32
[6] Palomar D P and Chiang M 2006 A tutorial on decomposition methods for network utility maximization IEEE J. Sel. Areas Commun. 1439-1451
[7] FIWARE Foundation 2019 The open source platform for our smart digital future accessed: 2019-05-25 URL https://www.fiware.org/about-us/
[8] Teichmann J, Kümpel A, Mathis P, Matthes P, Oppermann J and Müller D 2018 Reducing energy consumption of air handling units by optimized pump control Proc. Roomvent Ventilation 2018 463
[9] Blochwitz T, et al 2011 The functional mockup interface for tool independent exchange of simulation models Proc. of the 8th International Modelica Conference 105-114.
[10] Andersson C, Åkesson J and Führer C 2016 PyFMI: A Python package for simulation of coupled dynamic models with the functional mock-up interface Centre for Mathematical Sciences, Lund University