# Self-triggered Control with Energy Harvesting Sensor Nodes

NAOMI STRICKER, ETH Zurich, Switzerland
YINGZHAO LIAN, YUNING JIANG, and COLIN N. JONES, EPFL, Switzerland
LOTHAR THIELE, ETH Zurich, Switzerland

Distributed embedded systems are pervasive components jointly operating in a wide range of applications. Moving toward energy harvesting powered systems enables their long-term, sustainable, scalable, and maintenance-free operation. When these systems are used as components of an automatic control system to sense a control plant, energy availability limits when and how often sensed data are obtainable and therefore when and how often control updates can be performed. The time-varying and non-deterministic availability of harvested energy and the necessity to plan the energy usage of the energy harvesting sensor nodes ahead of time, on the one hand, have to be balanced with the dynamically changing and complex demand for control updates from the automatic control plant and thus energy usage, on the other hand. We propose a hierarchical approach with which the resources of the energy harvesting sensor nodes are managed on a long time horizon and on a faster timescale, self-triggered model predictive control controls the plant. The controller of the harvesting-based nodes' resources schedules the future energy usage ahead of time and the self-triggered model predictive control incorporates these time-varying energy constraints. For this novel combination of energy harvesting and automatic control systems, we derive provable properties in terms of correctness, feasibility, and performance. We evaluate the approach on a double integrator and demonstrate its usability and performance in a room temperature and air quality control case study.

CCS Concepts: • **Computer systems organization → Sensor networks**; **Sensors and actuators**; • **Hardware → Renewable energy;**

Additional Key Words and Phrases: Distributed embedded systems, energy harvesting, self-triggered model predictive control, sensor networks, sustainable automatic control

**ACM Reference format:**
Naomi Stricker, Yingzhao Lian, Yuning Jiang, Colin N. Jones, and Lothar Thiele. 2023. Self-triggered Control with Energy Harvesting Sensor Nodes. *ACM Trans. Cyber-Phys. Syst.* 7, 3, Article 20 (July 2023), 31 pages.
https://doi.org/10.1145/3597311

20

## 1 INTRODUCTION

Distributed embedded systems have become ubiquitous—they are part of Wireless Sensor Networks, cyber-physical systems, and the Internet of Things. A prominent application domain for this class of systems is automatic control, see, for example, Reference [37] for an overview. Their

This work was supported by the Swiss National Science Foundation under the RISK project, grant number 200021 175627, and under the NCCR Automation project, grant agreement 51NF40_180545.
Authors' addresses: N. Stricker and L. Thiele, Computer Engineering and Networks Laboratory, ETH Zurich, Zurich, Switzerland; emails: nstricker@ethz.ch, thiele@ethz.ch; Y. Lian, Y. Jiang, and C. N. Jones, EPFL, Lausanne, Switzerland; emails: yingzhao.lian@epfl.ch, yuning.jiang@epfl.ch, colin.jones@epfl.ch.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2023 Copyright held by the owner/author(s).
2378-962X/2023/07-ART20 $15.00
https://doi.org/10.1145/3597311

use in industrial automation and smart manufacturing systems is described in Reference [5]. Distributed embedded systems are also employed in buildings to enable efficient heating, ventilation, and air conditioning [2, 25]. Embedded systems measure temperature, humidity, human presence, or the state of windows and doors throughout the building. These sensed data provide state information for the automatic control.

Powering these pervasive systems is a fundamental challenge that has to be addressed. Employing batteries has various drawbacks including poor scalability, a negative environmental impact, and high maintenance costs. Energy harvesting is seen as a viable design paradigm to mitigate these drawbacks. A system's lifetime is prolonged, maintenance efforts and costs are reduced, and harvesting-based systems are environmentally friendlier. In automatic control, for example, embedded systems measuring the state of the control plant could be harvesting based. As a result, such sensor nodes can be placed such that they provide high-quality sensing data without being restricted by the availability of wired power and wired communication or by ease of access.

Despite these advantages, it is challenging to employ energy harvesting sensor nodes in control systems to measure the state of a plant. Since nodes are exposed to temporally and spatially varying primary energy, their energy availability is limited and variable. Yet, this energy availability dictates when and how often the plant can be sensed and thus control updates performed. On the contrary, for stable and performant control, control updates need to occur in accordance with the plant's varying control demand. As a result, the timing of control updates can be neither exclusively tied to available energy nor control demand but rather needs to take both into account. Balancing energy availability and control demand is further complicated by the different timescales on which they operate. Energy availability varies on a slower timescale and longer time horizon than the faster dynamics of the plant.

Another major challenge arises from the distributed nature of the energy harvesting sensor nodes. While the distributed nodes jointly sense the state of the control plant, each node is exposed to a unique environment and thus nodes have distinct energy availabilities. Because the controller requires sensed data from all nodes, when and how often the control plant can be sensed and control updates performed is dictated by the joint energy availability of all nodes, i.e., the node with the least energy dictates the frequency of control updates. Therefore, the flow of energy for each harvesting-based node needs to be managed not only in accordance with its local resources and environment but rather by jointly considering all nodes.

While problems related to these challenges have been studied, large gaps exist that inhibit the use of energy harvesting sensor nodes in automatic control applications. Event-triggered and self-triggered control paradigms [23] have been investigated for controllers with resource constraints. Self-triggered control has been employed in the context of wireless networks [47] as well as for energy constraints [49], yet there are no results available that investigate energy-constrained self-triggered control in the context of energy harvesting. However, energy harvesting introduces unique challenges, i.e., the limited resources depend on a variable non-deterministic process and also need to be controlled, however, on a different timescale. Various resource controllers also referred to as schedulers or energy managers have been proposed to manage the energy flow of an energy harvesting system [40]. Typically, these methods are designed for an individual energy harvesting node and do not jointly consider multiple distributed nodes as is necessary to efficiently employ distributed harvesting-based sensor nodes in automatic control systems. Additionally, state-of-the-art energy controllers typically do not specifically incorporate that future energy usage can differ from the optimized and determined behavior due to non-deterministic system behavior. However, supporting that the used energy differs from the provisioned energy, while maintaining long-term correct operation is key for balancing the available resources and the complex varying control demand.

We close these gaps by proposing a novel generally applicable hierarchical system concept for automatic control systems where harvesting-based embedded components sense the control plant. The system concept makes efficient use of the limited resources of the energy harvesting nodes while providing good control quality of the control plant. The concept uniquely incorporates self-triggered **model predictive control (MPC)** in a hierarchical control approach where the self-triggered MPC operates on a faster and a finite-horizon energy controller on a slower timescale. The energy controller jointly manages and optimizes the common energy availability of the distributed harvesting-based nodes, whereas the self-triggered MPC incorporates this time-varying limited energy availability as well as a provided flexibility in energy usage to balance control demand and resource availability. The two controllers interact through a well-defined interface that enables us to prove essential properties of the system concept such as feasibility, convergence, and optimality. In summary, our work presents the following major contributions:

- We propose a novel concept for automatic control systems that employ energy harvesting sensor nodes. The hierarchical approach combines self-triggered MPC with a finite-horizon energy controller. As a result, the harvesting-based systems' energy usages are planned ahead of time, energy is efficiently used when available and when needed, and the control quality is optimized.
- We prove properties of the system concept that are essential for long-term correct and reliable operation, such as feasibility, convergence, and optimality irrespective of the uncertainty of the environment.
- We simulate and compare the proposed approach to two baselines for a double integrator plant and a room climate control case study where temperature and air quality are regulated. For the former, we provide additional insights into the concept in unpredictable harvesting environments, and in the latter, we also consider the self-triggered MPCs behavior under model uncertainties and measurement noise.

In the remainder of this work, we first provide an overview of the system concept and its context in Section 2. Subsequently, we described a detailed system model and the hierarchical control approach in Section 3. We provide provable properties of the proposed method in Section 4. The concept is evaluated for a double integrator in Section 5 and a room climate control case study, Section 6. Last, we present related work in Section 7 and thereafter conclude this work.

## 2 SYSTEM CONCEPT

In this section, we present a general scenario in which the novel hierarchical control can be employed. Subsequently, we describe a high-level overview of the proposed joint management of self-triggered MPC and energy harvesting subsystems.

In the general scenario, depicted in Figure 1, distributed embedded sensor nodes measure the state of a control plant, possibly pre-process these data, and wirelessly communicate the sensed data to a central unit. The central unit as well as the actuators of the control plant do not have any relevant energy constraints, whereas the sensor nodes are powered by energy harvesting. Each sensor node is assumed to communicate point-to-point with the central unit without relying on any other nodes or infrastructure. LoRaWAN [1] is an example of a widely adopted technology that satisfies these assumptions. It enables a simple star topology while bridging distances up to kilometers with sufficient data rates for common sensor readings [1]. The communicated data provide a current sample of the plant state with which the central unit performs self-triggered MPC. The self-triggered MPC optimizes its objective, sets the control input of the actuators, and determines when the next control update will be and thus when the nodes have to sense again. By determining ahead of time when the next control update will occur, nodes can substantially reduce their power

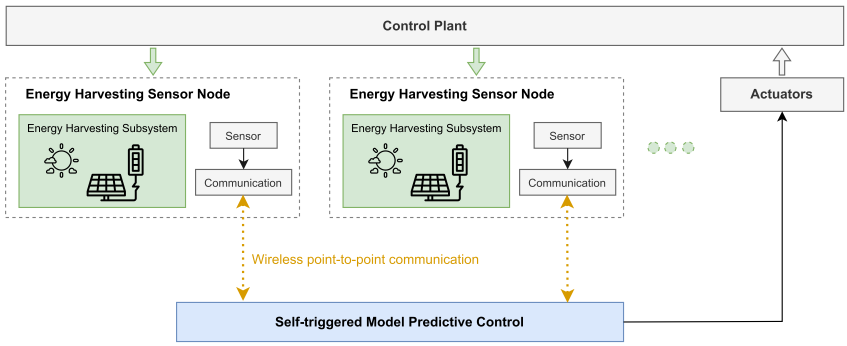



Fig. 1. In the considered scenario multiple distributed energy harvesting nodes sense the control plant and communicate these data to the MPC. The MPC determines the actuation for controlling the plant.

consumption until the next sampling time as they know they are not required to sense, process, or communicate until then. During this time their power consumption is drastically reduced by shutting off peripherals and entering a deep sleep state. Thus the energy usage of the harvesting-based nodes is tightly tied to the timing of the control updates determined by the self-triggered MPC. Moreover, the management of the resources of the energy harvesting nodes depends on the current state of the nodes' energy storages that are influenced by their past energy usage. This results in two intertwined control problems that we address with a hierarchical approach. On the one hand, the energy of each harvesting-based node is managed by a finite-horizon energy controller, and, on the other hand, the inter-sampling times are optimized as part of the self-triggered MPC.

In the remainder of this section, we informally explain the system concept composed of the finite-horizon energy controller, the self-triggered MPC, and the interface between the two. Figure 2 shows these components and their relative timing.

*Energy Harvesting Sensor Nodes and Energy Controller.* Each harvesting-based sensor node has an energy harvesting subsystem consisting of an energy harvester, e.g., a solar cell, and a rechargeable energy storage, e.g., a supercapacitor, with which nodes bridge periods where little energy is harvested. This enables the nodes to efficiently harvest and utilize the non-deterministic, temporally, and spatially varying primary energy source. The finite-horizon energy controller jointly optimizes their long-term behavior and performance in an energy-neutral manner by exploiting predictions of each node's future harvestable energy that are generated by energy predictors.

At periodic intervals on the slow timescale, denoted as epochs, the finite-horizon energy controller is executed. As shown in Figure 2, such an epoch has duration $\Delta_{\text{epoch}}$ and epoch $m$ covers the time interval $[T_m, T_{m+1})$ with $T_m = m \cdot \Delta_{\text{epoch}}$. At times $T_m$ when an epoch begins, the energy controller receives the current energy storage state of charge of all nodes $1 \leq s \leq S$, $E_{\text{stor,s}}(T_m)$, and information about the energy used in the previous epoch, $E_{\text{used}}(T_{m-1})$. The latter is dictated by the number of control updates that are determined by the MPC and thus may come from the MPC, or it can be provided directly by the nodes. Based on the used energy, the state of charge of the energy storages, and energy predictions, the energy controller determines the energy that all nodes can provide and should use in the next epoch to optimize their long-term performance. This energy dictates how often and when all nodes can sense the control plant and thus provide the MPC with the required sensed data. The jointly optimized energy that all nodes can provide, $E_{\text{prov}}(T_m)$, is made available to the self-triggered MPC. Because the self-triggered MPC does not have to precisely adhere to the provided energy but balances the provided energy and control demand, the energy used by nodes can differ from the provided energy. Nonetheless, the interface

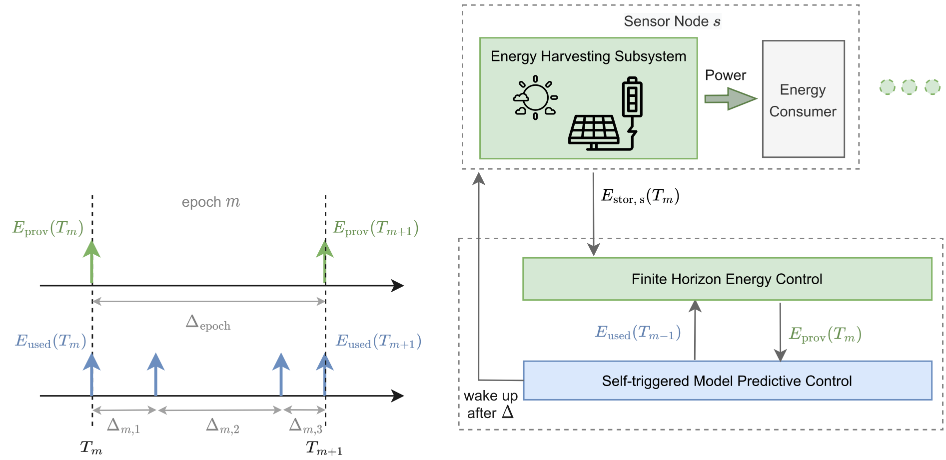



Fig. 2. Energy-related information flows between the main components: energy harvesting subsystem, energy consumer, finite-horizon energy control, and self-triggered MPC. The finite-horizon energy control updates with period $\Delta_{\text{epoch}}$ and determines the energy $E_{\text{prov}}(T_m)$ that nodes can provide in epoch $m$. Sensor nodes used $E_{\text{used}}(T_{m-1})$ in the previous epoch $m-1$ and node $s$ has initial energy storage state $E_{\text{stor, s}}(T_m)$. The self-triggered MPC performs unevenly distributed control updates in epoch $m$ with distance $\Delta_{m,i}$ in accordance with the provided energy and control demand.

guarantees that the used energy in each epoch, $E_{\text{used}}(T_m)$, has a bounded difference to the provided energy, $E_{\text{prov}}(T_m)$. It further ensures proper operation of the energy harvesting sensor nodes despite the variability in energy usage.

*Model Predictive Control.* The self-triggered MPC receives the sensed control plant state and optimizes a cost function for the plant. It determines the input to the actuators and, following the self-triggering control paradigm, the time interval to the next control update. Since the self-triggered MPC operates on a faster timescale than the energy controller, possibly multiple control updates are scheduled within each epoch. Thus, the $k$th execution of the MPC in epoch $m$ at time $\Delta_{m,k}$ after the previous control update also determines the time to the next, $k+1$, control update $\Delta_{m,k+1}$. This time interval is communicated to other system components. Since they thus know that they are not required to operate for a time interval of length $\Delta_{m,k+1}$, they can save energy by turning off or going into a deep sleep state, e.g., turning off their radios for wireless communication and peripherals.

To ensure the provided energy, $E_{\text{prov}}(T_m)$, is respected during an epoch, the MPC's optimization incorporates an energy model. By accounting for a node's energy cost per control update, the MPC thus determines the timing of control updates according to the provided energy. The energy model also grants the MPC a bounded flexibility to diverge from the provided energy that enables increased update rates in times of large control demand and decreased rates when not needed in comparison to the provided energy.

*Summary.* The self-triggered MPC optimizes the control cost and adjusts the control update frequency according to the energy availability at the harvesting-based sensor nodes. It accommodates varying control demand without risking the sensor nodes running into energy scarcity in the future. To support this, an interface couples the energy model of the MPC with the finite-horizon energy controller. The MPC has some flexibility in when and how often control updates are performed and thus how the limited energy of the harvesting-based nodes is used. It requires

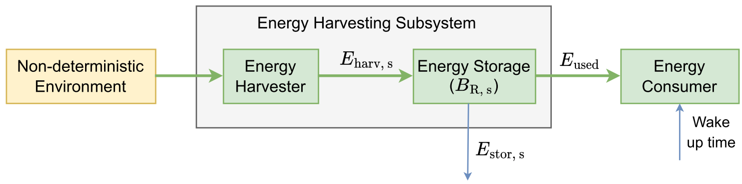



Fig. 3. The model of the energy harvesting subsystem of each node $1 \leq s \leq S$.

consistency with the provided energy only within a specified bound and thus allows the nodes to over- or under-consume compared to the provided energy. We now formalize the above, describe the control algorithms in detail, and provide proofs for optimality, feasibility, and convergence.

## 3 SYSTEM MODEL

In this section, a model for the energy harvesting subsystems of the harvesting-based nodes, the finite-horizon energy controller, and the self-triggered MPC are described, including their mathematical formulation and interface. This is the basis for proving important properties as done in Section 4.

### 3.1 Energy Harvesting Sensor Nodes and Finite-Horizon Energy Controller

There are $S \in \mathbb{Z}_{\geq 1}$ distributed energy harvesting sensor nodes each with their own energy harvesting subsystem. Each energy harvesting subsystem is assumed to have the same architecture and includes an energy harvester, a rechargeable energy storage element, and the energy consumer, as depicted in Figure 3. The finite-horizon energy controller manages the resources of all harvesting-based nodes. The energy controller first optimizes the resources of each node separately and subsequently considers all nodes jointly, as proposed in Reference [43]. Both optimization steps are performed at the central unit that does not have relevant resource constraints. We first describe the model for the energy harvesting subsystems and then the two steps of the finite-horizon energy controller.

We use a discrete-time model for the energy harvesting subsystems where time is discretized into epochs, $T_m = m \cdot \Delta_{\text{epoch}}$ with epoch $m \in \mathbb{Z}_{\geq 0}$ and duration $\Delta_{\text{epoch}}$. The energy harvester of node $1 \leq s \leq S$ converts primary energy into electrical energy thus producing $E_{\text{harv, s}}(T_m)$ during the time $[T_m, T_{m+1})$. The node's rechargeable energy storage element has a finite capacity of $B_{\text{R,s}}$ whose state of charge at time $T_m$ is denoted by $E_{\text{stor,s}}(T_m)$. The state of charge evolves linearly according to

$$E_{\text{stor,s}}(T_{m+1}) = (E_{\text{stor,s}}(T_m) + E_{\text{harv,s}}(T_m) - E_{\text{used}}(T_m))]_0^{B_{\text{R,s}}} \tag{1}$$

where $x]_a^b$ limits $x$ to $[a, b]$, i.e., $x]_a^b = \min\{\max\{x, a\}, b\}$ and $E_{\text{used}}(T_m)$ is the energy used by the energy consumer of node $s$ during $[T_m, T_{m+1})$. The energy is used to sense the plant state, possibly preprocess sensed data depending on the specific application, and communicate the data to the central unit where the MPC is executed.

At times $T_m$, the energy controller is executed to determine how much energy can be provided to and thus used by the energy consumers of all nodes during the next epoch $m$. The energy controller exploits predictions of the future harvestable energy to optimize the long-term performance of the energy harvesting nodes. Thus for each node, energy predictions $E_{\text{pred, m, s}}$, discussed in more detail below, for the subsequent $M$ epochs are required where $M$ is the energy controller's optimization horizon. The resulting optimized energy $E_{\text{prov}}(T_m)$ is made available to the self-triggered MPC.

In the first step at $T_m$, the resources of each harvesting-based node are optimized separately as given in the Optimization Problem (2a)–(2e),

$$
\begin{aligned}
\min_{a_s} \quad & \sum_{0 \le i < M} a_s(i)^2 && \text{(2a)}\\
\text{s.t.} \quad & \forall 0 \le i < M \ : \ b_s(i+1) = b_s(i) + E_{\text{pred, m, s}}(i) - a_s(i) && \text{(2b)}\\
& b_s(M) = b_s(0) && \text{(2c)}\\
& \forall 0 \le i < M \ : \ 0 \le a_s(i)\ , \ 0 \le b_s(i) \le B_{R,s} - 2R && \text{(2d)}\\
& b_s(0) = E_{\text{stor, s}}(T_m) - R - \sum_{0 \le j < m} (E_{\text{prov}}(T_j) - E_{\text{used}}(T_j)) && \text{(2e)}
\end{aligned}
$$

where $a_s$ represents the energy that can be made available to the energy consumer of node $1 \le s \le S$ and $b_s$ its energy storage fill level during the optimization. $R$ is the energy flexibility that the self-triggered MPC has in deciding how the limited energy is used by determining when control updates occur. A formal definition of $R$ is given below with Equation (6). The optimization objective, Equation (2a) is equivalent to maximizing the minimal energy available to the consumer during the subsequent $M$ epochs, a property we prove for Theorem 4.1 in Section 4. Equation (2b) is the energy storage state of charge evolution, Equation (2c) bounds the state of charge, and Equation (2e) is its initialization. Since the used energy can differ from the provided energy, it must be ensured that this is supported by the energy harvesting node. A node needs to, for example, support a bounded increase in energy usage beyond the provided energy. Therefore, the initial energy storage state of charge, Equation (2e), takes into account to what extent the energy flexibility has been exploited so far and the energy storage capacity is adjusted to leave room, respectively, to save energy in the energy storage for this flexibility. A node should nonetheless not consistently over- or under-use harvested energy over prolonged periods of time, since this leads to wasted energy, respectively, energy scarcity. This is ensured with Equation (2c). Last, all energies have to be positive, Equation (2d). The optimization problem (2a)–(2e) is solved for each harvesting-based node and the respective solutions of the available energy are denoted as $a_s^*(i)$ for all nodes $1 \le s \le S$ over the optimization time horizon $0 \le i < M$. Since sensed data from all nodes are required, the amount of energy that is available at all nodes, the common energy, is relevant for the MPC. This common energy for each epoch is the minimum of the available energies of each node determined by the separate optimizations, namely

$$
\forall 0 \le i < M \ : \quad u_{\text{sep}}^*(i) = \min_{1 \le s \le S} a_s^*(i) \tag{3}
$$

The minimum of $u_{\text{sep}}^*(i)$ cannot be improved, see Appendix A. Nevertheless, the energy at epochs other than where the minimum occurs can be increased by considering all nodes together. Based on the results from the separate optimizations, the energy controller subsequently jointly optimizes the resources of all nodes with the Optimization Problem (4a)–(4g),

$$
\begin{aligned}
\max_{\hat{v}, \hat{a}_s} \quad & \sum_{0 \le i < M} \hat{v}(i) && \text{(4a)}\\
\text{s.t.} \quad & \forall 1 \le s \le S,\ 0 \le i < M \ : \ \hat{b}_s(i+1) = \hat{b}_s(i) + E_{\text{pred, m, s}}(i) - \hat{a}_s(i) && \text{(4b)}\\
& \forall 1 \le s \le S,\ 0 \le i < M \ : \ u_{\text{sep}}^*(i) \le \hat{a}_s(i) && \text{(4c)}\\
& \forall 1 \le s \le S \ : \ \hat{b}_s(M) = \hat{b}_s(0) && \text{(4d)}\\
& \forall 1 \le s \le S,\ 0 \le i < M \ : \ 0 \le \hat{a}_s(i)\ , \ 0 \le \hat{b}_s(i) \le B_{R,s} - 2R && \text{(4e)}\\
& \forall 1 \le s \le S \ : \ \hat{b}_s(0) = E_{\text{stor, s}}(T_m) - R - \sum_{0 \le j < m} (E_{\text{prov}}(T_j) - E_{\text{used}}(T_j)) && \text{(4f)}\\
& \forall 1 \le s \le S,\ 0 \le i < M \ : \ \hat{v}(i) \le \hat{a}_s(i) && \text{(4g)}
\end{aligned}
$$

where $\hat{a}_s$ and $\hat{b}_s$ correspond to $a_s$, respectively, $b_s$, in the optimization problem (2a)–(2e) and $\hat{v}$ represents the common energy of all nodes. Equations (4b), (4d), (4e), and (4f) are equivalent to the constraints in the optimization (2a)–(2e). Equation (4c) ensures that the joint optimization only improves on the common energy availability determined from the separate optimizations. Furthermore, the objective, Equation (4a) together with Equation (4g) maximize the common energy. Solutions of this optimization are also denoted with a star, $^*$, and the resulting planned common energy usage is

$$\forall 0 \leq i < M \ : \quad u^*_{\text{joint}}(i) = \hat{v}^*_s(i) \tag{5}$$

In Section 4, we prove that the minimum of this common energy throughout the time horizon, $\min_{0\leq i<M} u^*_{\text{joint}}(i)$, is optimal and thus cannot be increased. The provided energy that is communicated to the self-triggered MPC for the upcoming epoch is $E_{\text{prov}}(T_m) = u^*_{\text{joint}}(0)$.

The energy controller must take into account the flexibility that the self-triggered MPC has in its energy usage, i.e., the possibility to over- or under-use energy within a specified bound from $E_{\text{prov}}(T_m)$. Conversely, the self-triggered MPC must adhere to the provided energy within the energy flexibility it has, which allows it to determine control updates to occur more frequently or less frequently than dictated by the provided energy. The allowed and supported energy flexibility is denoted as $R$ and defined as

$$\left| \sum_{0\leq j<m} (E_{\text{prov}}(T_j) - E_{\text{used}}(T_j)) \right| \leq R \tag{6}$$

where $E_{\text{used}}(T_j)$ is the common used energy during $[T_j, T_{j+1})$. The sum ensures the accumulated difference remains bounded. This bound is guaranteed by design by the MPC and energy consumers. Under this condition, we prove important properties of the system in Section 4.

*Energy Predictions.* The proposed energy controller requires energy predictions $E_{\text{pred, m, s}}$ for the subsequent $M$ epochs for each energy harvesting sensor node. In the considered scenario, the harvesting-based nodes share energy information with the central unit, and subsequently, the predictions for each node are generated at the central unit. Typically, nodes sharing information about the energy storage state of charge, $E_{\text{stor, s}}(T_m)$ and used energy $E_{\text{used}}(T_m)$ suffices to generate the predictions. Although this incurs a communication overhead, energy information only needs to be sent once per epoch and thus the overhead is limited. Various energy prediction algorithms for energy harvesting embedded systems have been developed. They may rely on different approaches such as statistical models, physical models, and machine learning or hybrid methods [40]. While predictors can have small prediction errors in some scenarios, they are typically not accurate. The resulting prediction errors can have a strong influence on the resource management strategy [44]. We prove in Section 4 that for the proposed energy controller this influence is bounded and provide an analytical bound. We also explore the effect of prediction errors on the MPC's control quality in Section 5.

### 3.2 Self-triggered Model Predictive Control

A general dynamic process in continuous time is assumed for the plant

$$\dot{x}(t) = f(x(t), u(t)) \tag{7}$$

where $t \in [0, \infty)$ denotes the time, $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ the process state, and $u(t) \in \mathcal{U} \subseteq \mathbb{R}^k$ the input. There is a cost function $l(x(t), u(t)) : \mathcal{X} \times \mathcal{U} \to [0, \infty)$ associated with the process. The input set, process, and cost are assumed to fulfill Assumption 1 described in Reference [49]. These assumptions state that the set $\mathcal{U}$ is compact and contains the origin. The process specified in

Equation (7) is continuous and locally Lipschitz for $x \in \mathcal{X}$ and $u \in \mathcal{U}$ and is zero at the origin, $f(0,0) = 0$. Furthermore, the process is absolutely continuous for any piece-wise continuous input function in $\mathcal{U}$. Last, the cost function is assumed to be continuous and positive definite. These assumptions enable us to provide provable properties in Section 4.

The self-triggered MPC is executed aperiodically at times $t_{m,k}$ with $m, k \in \mathbb{Z}_{\geq 0}$ where $m$ denotes the current epoch and $k$ the update within epoch $m$. The time between two consecutive MPC executions and thus control updates at $t_{m,k}$ and $t_{m,k+1}$ is called $\Delta_{m,k+1}$, i.e., $t_{m,k+1} - t_{m,k} = \Delta_{m,k+1}$, see Figure 2.

The self-triggered MPC has two decision variables for its optimization: the plant inputs $u$, which are held constant between control updates, and the time intervals between control updates, i.e., when it self-triggers. The Optimization Problem (8a)–(8g) is executed by the self-triggered MPC at time $t_{m,k}$. It determines the optimal interval $\Delta_{m,k+1}$ to the next control update and the plant input $u_{m,k}$ within this time interval, such that all boundary conditions are fulfilled and the cost is minimized. It optimizes over a finite horizon of $N$ control updates and the first step of the optimal solution is applied for $(t_{m,k}, t_{m,k+1}]$,

$$\min_{\hat{u}_j,\, \hat{\Delta}_j} \quad \int_{\hat{t}_0}^{\hat{t}_N} l(\hat{x}(t), \hat{u}(t))\, \mathrm{d}t \tag{8a}$$

$$\text{s.t.} \quad \hat{t}_0 = t_{m,k}\ ; \quad \forall 0 \leq j < N\ :\ \hat{t}_{j+1} = \hat{t}_j + \hat{\Delta}_{j+1} \tag{8b}$$

$$\forall 0 \leq j < N\,,\ \hat{t}_j < t \leq \hat{t}_{j+1}\ :\ \hat{u}(t) = \hat{u}_j \tag{8c}$$

$$\hat{x}(\hat{t}_0) = x(t_{m,k})\ ; \quad \forall \hat{t}_0 \leq t \leq \hat{t}_N\ :\ \frac{\mathrm{d}\hat{x}(t)}{\mathrm{d}t} = f(\hat{x}(t), \hat{u}(t)) \tag{8d}$$

$$\hat{r}_0 = r(t_{m,k})\ ; \quad \forall 0 \leq j < N\ :\ \hat{r}_{j+1} = \hat{r}_j + E_{\text{prov}}(T_m) \frac{\hat{\Delta}_{j+1}}{\Delta_{\text{epoch}}} - \mu \tag{8e}$$

$$\forall 0 \leq j < N\ :\ \hat{r}_{j+1} \in [-R, R]\,,\ \hat{\Delta}_{j+1} \in [\Delta_{\min}, \Delta_{\max}]\,,\ \hat{u}_j \in \mathcal{U}\,,\ \hat{x}(\hat{t}_{j+1}) \in \mathcal{X} \tag{8f}$$

$$t_{m,k+1} = \hat{t}_1\ ;\ x(t_{m,k+1}) = \hat{x}(\hat{t}_1)\ ;\ r(t_{m,k+1}) = \hat{r}_1\ ;\ u_{m,k} = \hat{u}_0 \tag{8g}$$

Here Equation (8a) defines the minimization objective as the integral of the cost throughout the control horizon. Equation (8b) ties the time between control updates to when those updates occur. The plant input stays constant between control updates, which is considered in Equation (8c). Equation (8d) ensures the state evolves according to the plant process specified in Equation (7). State constraints can be enforced at the triggering times, Equation (8f), but not in continuous times as is outlined in Reference [49]. Equation (8f) also specifies bounds on the time between control updates.

Furthermore, the self-triggered MPC optimization incorporates a resource $r$ whose dynamics and characteristics are tied to the finite-horizon energy controller. The resource $r$ is the difference between the energy the harvesting-based nodes use dictated by the timing of control updates and the provided energy. Therefore, the resource $r$ represents the extent to which the energy flexibility is exploited and whether the provided energy is under- or over-used. Its evolution, Equation (8e), ties together the provided energy $E_{\text{prov}}(T_m)$ and the used energy in terms of the energy $\mu$ that is required per control update by each harvesting-based sensor node. In particular, each energy harvesting node senses the control plant, possibly preprocess measurements, and communicates these data to the central unit. $\mu$ summarizes the energy required to perform these tasks. The proposed system concept assumes this energy requirement to be constant and equal for all nodes. Although this is a simplifying assumption, the concept can be extended to allow different energy requirements for each node. The constraints on $r$ in Equation (8f) ensure that the MPC coheres

**ALGORITHM 1:** Combined Self-triggered MPC and Finite-Horizon Energy Control

$m \leftarrow 0$, $T_0 \leftarrow 0$, $r(0) \leftarrow 0$
**while** true **do**
  Measure $E_{\text{stor, s}}(T_m)$ for each harvesting-based node $1 \leq s \leq S$ or determine following Equation (1)
  Generate energy predictions
  Solve optimization problem (2a)–(2e) for each node $1 \leq s \leq S$ and determine $u^*_{\text{sep}}$ according to Equation (3)
  Solve optimization problem (4a)–(4g) and derive $u^*_{\text{joint}}$ by Equation (5)
  Determine $E_{\text{prov}}(T_m)$ while respecting Equation (9)
  $k \leftarrow 0$, $t_{m,0} \leftarrow T_m$
  ▹ Start self-triggered MPC executions in current epoch
  **while** $t_{m,k} < T_m + \Delta_{\text{epoch}}$ **do**
    Measure $x(t_{m,k})$ with energy harvesting nodes or simulate according to Equation (7)
    Solve optimization problem (8a)–(8g) and determine $\Delta_{m,k+1}$, $t_{m,k+1}$, $r(t_{m,k+1})$, and $u_{m,k}$
    Set actuation input to $u_{m,k}$
    $k \leftarrow k + 1$
  **end while**
  ▹ Next control update is at the synchronized time
  Pass resource state to the next epoch, $r(t_{m+1,0}) \leftarrow r(t_{m,k})$
  $T_{m+1} \leftarrow T_m + \Delta_{\text{epoch}}$
  $m \leftarrow m + 1$
**end while**

to the provided energy within the allowed energy flexibility $R$, where $R \geq 0$. The initial resource $\hat{r}_0 = r(t_{m,k})$ in Equation (8e) represents the thus far accumulated difference between the energy the self-triggered MPC determined to be used and the provided energy up to time $t_{m,k}$.

Results of the optimization are defined in Equation (8g), where $u_{m,k}$ denotes the plant input in $t \in (t_{m,k}, t_{m,k+1}]$.

### 3.3 Combined Self-triggered MPC and Energy Controller

Finally, we present our hierarchical approach of combining self-triggered MPC and energy harvesting sensor nodes, i.e., the interface between the energy controller with optimization problems (2a)–(2e) and (4a)–(4g), and the MPC with optimization problem (8a)–(8g). Both the energy controller and self-triggered MPC are applied in a receding horizon control fashion, where only the first step of the optimization solutions is executed and subsequently the respective optimization problems are solved again. At synchronized control updates, at times $T_m$ with $m \in \mathbb{Z}_{\geq 0}$, first the energy controller is executed and the optimized provided energy is communicated to the self-triggered MPC. Within an epoch, the self-triggered MPC is called several times to determine the plant inputs and time intervals to the next control update, see also Figure 2. This combined self-triggered MPC and finite-horizon energy controller scheme is summarized in Algorithm 1.

### 3.4 Model Parameter Design

*System Parameter Design Considerations.* The proposed system concept has several parameters that need to be specified. First, the number of energy harvesting nodes $S$ is driven by the application. For the energy controller that manages their resources, the epoch length and optimization horizon need to be set. The optimization horizon is typically selected such that it correlates with a periodicity of the energy source, e.g., a day [14], a week, [44] or a year [11] for photovoltaic energy harvesting nodes. The length of an epoch can be chosen in conjunction with the dynamics of the control process. Since in the proposed concept, the self-triggered MPC performs control updates at the beginning of each epoch and non-uniformly distributed during the epoch, the epoch length should be chosen large enough such that multiple control updates are necessary within epochs. Furthermore, the energy storage capacity and solar panel size of the harvesting-based nodes need to be dimensioned appropriately. Knowledge of the energy availability at the deployment locations, inefficiencies of the energy harvesting subsystem components, and energy consumer can be leveraged to design an energy harvesting subsystem that can power the energy consumer long-term [13]. Subsequently, the capacity should be augmented by $2R$ to incorporate the flexibility in energy usage. This also ensures that the energy controller optimization problems (2a)–(2e) and (4a)–(4g) are properly defined, i.e., $B_{R,s} - 2R > 0$. The energy flexibility $R$ is directly related to how many control updates more or fewer than the provided energy can be used. For example, in the extreme case where $R = 0$, there is no energy flexibility and the MPC performs uniformly spaced control updates in each epoch. The energy flexibility $R$ should therefore be designed considering the plant's control demand. Finally, the number of control updates $N$ over which the MPC optimizes is limited by the complexity of finding a solution for the optimization problem for large $N$.

*Minimal Provided Energy for the Model Predictive Control.* For the MPC's optimization problem (8a)–(8g) to be feasible, the provided energy needs to satisfy the constraint

$$\mu \cdot \left\lceil \frac{\Delta_{\text{epoch}}}{\Delta_{\max}} \right\rceil \leq E_{\text{prov}}(T_m) \leq \mu \cdot \left\lfloor \frac{\Delta_{\text{epoch}}}{\Delta_{\min}} \right\rfloor \tag{9}$$

where $\mu$ denotes the energy cost per control update and $\Delta_{\min}$ and $\Delta_{\max}$ specify the allowed range of time differences between two control updates.

If the provided energy is too large, then the excess can be retained in the energy storages, as additional energy can only improve future energy provisions or the surplus can simply be wasted. A smaller $E_{\text{prov}}(T_m)$ that satisfies the upper bound is communicated to the self-triggered MPC. Satisfying the lower bound is more challenging. By making assumptions on the environment, the harvesting-based nodes can be dimensioned such that the provided energy is expected to satisfy the lower bound of Equation (9) for all epochs. Even when the local environments in which each node will be deployed can be modeled sufficiently accurately, the environment is non-deterministic and might lead to a violation of the lower bound in Equation (9) for some epochs. During these epochs, the automatic control system may for example revert to a simpler control algorithm that requires fewer or even no sensor data but still guarantees safety or even provides a minimal service. In addition, the energy harvesting nodes can incorporate a small backup battery into their energy harvesting subsystems to bridge such periods as, for example, proposed in Reference [17]. In Reference [17], the use of the backup battery is minimized, while nonetheless providing a specified minimal energy to the energy consumer. In our case, this energy could be specified as the minimal energy to satisfy Equation (9). We will not discuss any of the above strategies further as they are orthogonal to the main results of the present article.

*Realistic Energy Harvesting Model.* Although the model of the energy harvesting subsystems is simple and abstract, it is realistic and accurately represents systems as considered in the scenario described in Section 2. Non-ideal behavior, such as charging and discharging inefficiencies of the energy storage or leakage currents, can be incorporated into the provided system model by adapting its parameters as described in Reference [17]. A suitable choice of parameters can be determined by characterizing an energy harvesting node with a few experiments [17].

## 4 PROVABLE PROPERTIES

In this section, we present provable properties in particular related to the correctness, feasibility, and performance of the proposed method. All proofs are provided in Appendix A.

### 4.1 Finite-Horizon Energy Controller

An important aspect of the finite-horizon energy controller is the objective it pursues while managing the limited resources. The following property relates to the energy controller's optimization objective and states that it manages the energies of harvesting-based nodes such that the minimal common available energy is maximized. Thus it ensures that the minimal energy provided to the self-triggered MPC in any epoch is as high as possible.

THEOREM 4.1. *The optimal solution to the energy controller, $u^*_{joint}(i)$, determined by solving optimization problem (2a)–(2e) for each node $1 \leq s \leq S$, evaluating Equation (3), and subsequently solving optimization problem (4a)–(4g) and determining $u^*_{joint}$ according to Equation (5) maximizes the minimal common planned energy. In other words, there is no common planned energy $\hat{u} = \min_{s,i} \hat{a}_s(i)$ that all nodes $1 \leq s \leq S$ support during the time horizon $0 \leq i < M$ while satisfying Equations (4b), (4d), (4e), and (4f) with $\hat{u} > \min_i u^*_{joint}(i)$.*

This theorem does not consider the errors in the energy predictions that the energy controller relies on. These, however, can have an effect on the energy provided by the energy controller [44]. The presented energy controller's formulation enables us to show that this effect is well behaved. Moreover, we provide an analytical bound on the impact of prediction errors on the optimal solution.

THEOREM 4.2. *Suppose the optimal solution of Equation (2a)–(2e) at time $T_m$ is denoted as a vector $a^*_{s,\,pred}$ where the ith element is $a^*_s(i)$ and that the optimal solution vector of optimization problem (2a)–(2e) at time $T_m$ for replacing $E_{pred,m,s}$ by the actually harvested energy $E_{harv,s}$ is $a^*_{s,harv}$. Then the difference in the two optimal available energies is bounded by the difference between the predicted and harvested energy,*
$\|a^*_{s,pred} - a^*_{s,harv}\|_2 \leq 2(M+1)^{3/2}\|E_{pred,m,s} - E_{harv,s}\|_2$.

### 4.2 Model Predictive Control

Aside from uncertainties in the future harvested energy, there may also be unpredictable control demand. This impacts the self-triggered MPC and results in variability in energy usage at the harvesting-based nodes. Nonetheless, the interface ensures that the proposed method has safe and optimal energy control even in the face of uncertainties of the self-triggered MPC's determined energy usage, as stated below.

THEOREM 4.3. *Suppose the initial energy storage state of charge of each energy harvesting sensor node $1 \leq s \leq S$ lies within the interval $E_{stor,s}(0) \in [R, B_{R,s}]$ and the energy predictions are accurate, i.e., $E_{pred,m,s}(T_m) = E_{harv,s}(T_m)\ \forall m \in \mathbb{Z}_{\geq 0}$ and for all $1 \leq s \leq S$. With the application of Algorithm 1, none of the energy storages become negative, i.e., $E_{stor,s}(t) \geq 0\ \forall t \geq 0$ and for all $1 \leq s \leq S$.*

To further demonstrate that the self-triggered MPC is recursively feasible and convergent within each epoch, we add terminal equality constraints to the optimization problem (8a)–(8g) and apply the concept of the theorem and proofs as provided in Reference [49].

The modified self-triggered MPC optimization at time $t_{m,k}$ is

$$V^*(x(t_{m,k}), r(t_{m,k})) = \min_{\hat{u}_j, \hat{\Delta}_j} \int_{\hat{t}_0}^{\hat{t}_N} l(\hat{x}(t), \hat{u}(t))\, \mathrm{d}t \tag{10}$$
$$s.t. \quad (8b)–(8f)$$
$$\hat{x}_N = 0 \quad \hat{r}_N \geq 0$$

THEOREM 4.4. *Suppose at the beginning of each epoch $m \in \mathbb{Z}_{\geq 0}$ the modified optimization problem (10) is feasible for $(x(t_{m,0}), r(t_{m,0}))$, then, by applying Algorithm, 1 the optimization problem (10) remains recursively feasible for all $t_{m,k}$ with $k \in \mathbb{Z}_{>0}$.*

Furthermore, with the thus constructed sequence of feasible solutions, the modified self-triggered MPC is close-loop convergent.

THEOREM 4.5. *Suppose at the beginning of each epoch $m \in \mathbb{Z}_{\geq 0}$ the modified optimization problem (10) is feasible for $(x(t_{m,0}), r(t_{m,0}))$ and the cost does not increase between epochs, then by applying the proposed Algorithm 1 the close-loop process state converges to zero, i.e., $x(t) \to 0$ for $t \to \infty$.*

In conclusion, we provided provable properties for our proposed hierarchical control approach combining self-triggered MPC and finite-horizon energy control that ensure the control system behaves correctly and reliably.

## 5 EVALUATION

Over the following two sections, the proposed concept is evaluated using a double integrator plant, and a room climate control, including both temperature and air quality, case study. In both scenarios, we show that the proposed approach is able to optimize the control cost by dynamically adjusting the time between control updates in accordance with the control demand and within the provided energy and its flexibility. The double integrator's evaluation places additional focus on the impact energy prediction errors have on the system and control performance. And in the room climate control case study, we also evaluate the system performance under model inaccuracies of the MPC's model and sensor measurement noise.

### 5.1 Setup

*Model Predictive Control.* The process dynamics of a double integrator are characterized by

$$\dot{x}(t) = Ax(t) + Bu(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

with input constraints $u(t) \in [-100, 100]$ and output constraints $y(t_{m,k}) = [1\,0]\, x(t_{m,k}) \in [-2, 2]$, which are comparable to Reference [29]. The cost penalizes both deviations of the output from a reference and the magnitude of the input [49],

$$l(x(t), u(t)) = 10 \cdot (y(t) - y_{\text{ref}})^2 + u_{\text{norm}}(t)^2$$

where $u_{\text{norm}}(t) = \frac{u(t)}{100}$ is the normalized input. During the simulation the reference changes unexpectedly [29] every 12 h. The horizon of the self-triggered MPC spans $N = 20$ control updates, as in Reference [49], and the time between two control updates is bounded by $\Delta \in [0.001, 1]$ h. The upper bound coincides with the length of an epoch, as the MPC updates at least at the synchronized times $T_m$. The self-triggered MPC has an energy usage flexibility that is equivalent to 20 control updates. This is on the order of how many updates are necessary to follow a change in the reference.

*Sensors.* Two energy harvesting sensor nodes measure the control plant state, one for each state. Each node has a solar panel with a size of 50mm by 33mm identical to the one employed in Reference [42]. The energy controller manages their resources. Its optimization horizon spans one day following the daily periodicity of both natural light and human-driven indoor environments. The length of an epoch is set to 1 h, since the dynamics of a double integrator are fast. Because the reference is not expected to change, the harvesting nodes are dimensioned to support only a few control updates per epoch. The actions associated with each control update include directly communicating with the central unit. The point-to-point communication between each node and

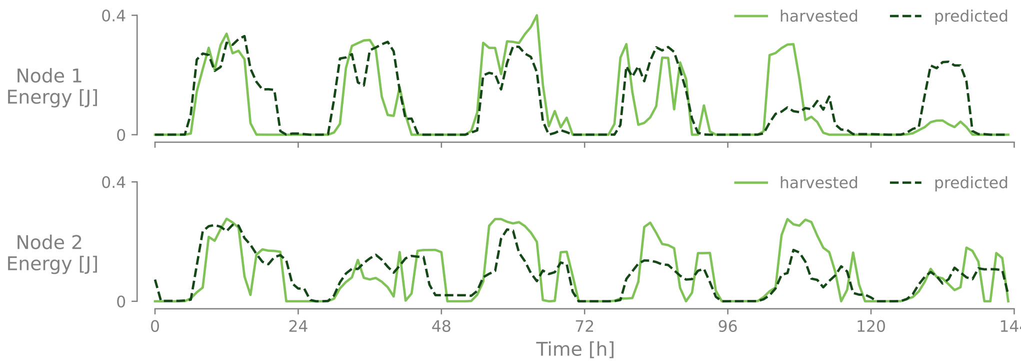



Fig. 4. Each node is exposed to a unique environment, harvesting energy at different rates and times. The *ewma* prediction follows the general trend of the harvested energy, yet still experiences prediction errors.

the central unit can for example be achieved with LoRa with which communication ranges can span up to multiple kilometers [48]. Each node requires $\mu = 10\,\text{mJ}$ per control update, which is comparable to wireless sensors performing long-range LoRa communication [6]. The energy storage of each node is furthermore augmented by the specified flexibility. Thus, both nodes have an energy storage with a capacity of $B_R = 3\,\text{J}$ that is initially half full.

*Harvesting Environment and Energy Predictors.* The energy harvesting nodes are exposed to indoor environments with different harvesting characteristics. We use two energy traces from the dataset presented in Reference [42] from two offices starting in September 2018. Although the harvested energy in indoor environments is challenging to predict [45], the energy controller requires a prediction for each node. We evaluate the system behavior and performance for two predictors with different accuracies. The first predictor (*const*) estimates the harvested energy to be the same as the energy that was harvested on the same weekday during the same hour in the previous week. The second predictor (*ewma*) from Reference [27] estimates with an exponentially weighted moving average of the same weekdays at the same hour with a weighting factor of $\alpha = 0.5$. The harvested energies and the *ewma* predictions for the two nodes are shown in Figure 4. For indoor scenarios, the **mean absolute deviation percentage (MADP)** is a suitable metric to describe a prediction's accuracy [44]. The *const* predictor has an MADP = 79 % at node 1 and the *ewma* predictor MADP = 66 %. At node 2, the *const* and *ewma* performances are MADP = 60 %, respectively MADP = 65 %.

*Baselines.* We compare our proposed approach (*self-triggered*) to two baselines. The first baseline is a model predictive controller with a constant period (*periodic*). The period of this approach is set to the highest frequency that can be sustained by both harvesting-based sensor nodes without any system failures throughout the previous year. The second baseline (*duty cycled*) uses the finite-horizon energy controller's optimization problem (2a)–(2e) in Section 3.1 with $R = 0$ to determine how much energy is available in every epoch at each node separately. Then the minimum of the two energies according to Equation (3) is used to derive a duty cycle and thus a period for the MPC during the respective epoch $m$, namely $p_m = \frac{1}{\lfloor \frac{u^*_{\text{sep}}(0)}{\mu} \rfloor}$.

## 5.2 Performance Evaluation

The above-described setup was used to simulate the system for the two baseline controllers, *duty cycled* and *periodic*, and the proposed *self-triggered* scheme for 6 days. The *self-triggered* and

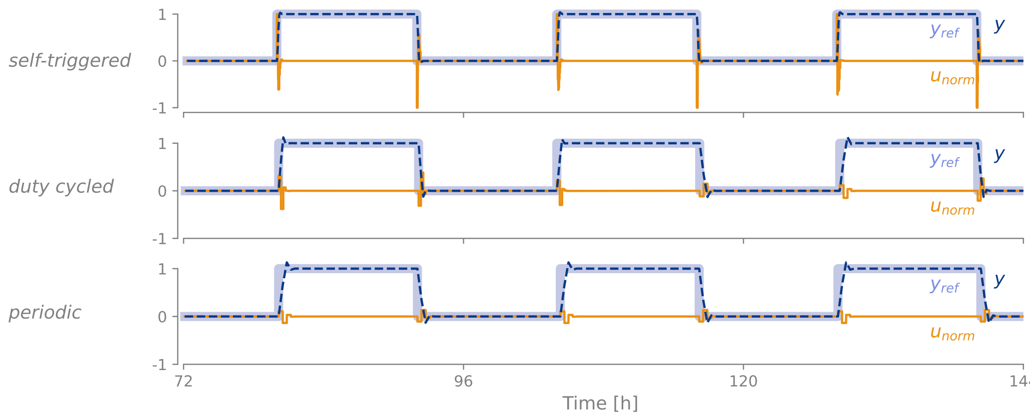
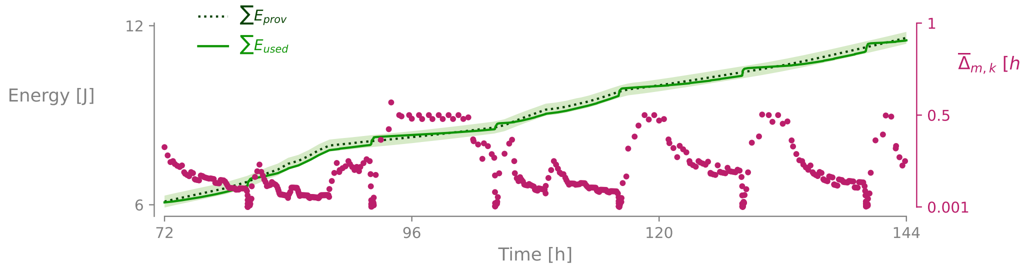


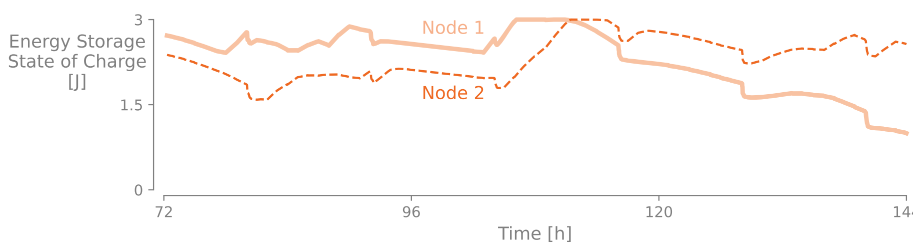

Fig. 5. The control system reference $y_{\text{ref}}$, output $y$ and normalized input $u_{\text{norm}}$ for the three approaches *periodic*, *duty cycled*, and *self-triggered*. The reference signal changes unexpectedly multiple times. All approaches adapt the input such that the output follows the reference. However, the proposed *self-triggered* approach adapts best to these changes.

*duty-cycled* methods are simulated in combination with each predictor where the same predictor is used to generate energy predictions for both nodes.

The simulation results over the last 2 days for the three controllers, where the *self-triggered* and *duty cycled* approaches rely on the *ewma* predictions, are shown in Figure 5. For all three approaches, the double integrator output follows the reference by adapting the input accordingly. The *periodic* baseline adapts slowly because of its large period, yet it does not require an energy prediction. For both predictors, the harvesting conditions allowed the *duty cycled* baseline to have a smaller period than the *periodic* one and thus adapt better to the unexpected changes. Moreover, when employing the proposed scheme, the plant adapts best regardless of the predictor that the energy controller relies on. It achieves this by decreasing the time between two control updates. Figure 6(a) shows the averaged time between two control updates on the right axis (averaged $\Delta_{m,k}$) for the excerpt in Figure 5. The small oscillatory variations in the averaged $\Delta_{m,k}$ are a result of applying a simple scheme to synchronize the two control loops. The simple scheme sets the subsequent control update to occur at $T_{m+1}$ if at time $t_{m,k}$ the next control update $t_{m,k+1}$ is after $T_{m+1}$. This ensures that a control update and the energy controller are executed at the periodic synchronized times $T_m$ for $m \in \mathbb{Z}_{\geq 0}$ despite the unevenly spaced times $t_{m,k}$. When the time between control updates increases to a length comparable to that of an epoch, this synchronization scheme noticeably impacts the time between control updates, as seen in Figure 6(a). The pronounced dips in the time between control updates at the times of the changes in the reference demonstrate how the self-triggered MPC performs more control updates precisely when they are necessary. In doing so, the self-triggered MPC temporarily uses more than the provided energy, fully exploiting the flexibility in energy usage it has. Afterward, it uses less energy than provided by the energy controller and thus compensates for the over-used energy. Figure 6(b) shows the corresponding evolution of the energy storage state of charge for both nodes. The state of charge of both nodes experiences strong dips when the *self-triggered*'s MPC performs frequent control updates. While the energy cost per control update is the same for both nodes, their harvesting characteristics are different due to the temporally and spatially varying primary energy. This results in different evolutions of energy storage fill levels, highlighting the importance of considering nodes jointly when optimizing their resource usage as well as addressing unpredictable variability in energy.




(a) The accumulated provided energy with the allowed flexibility and the used energy determined by the self-triggered MPC on the left axis, and the averaged time between consecutive control updates on the right axis.

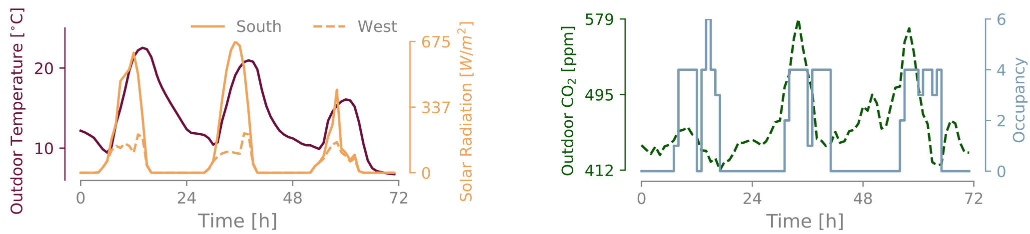



(b) The energy storage state of charge evolution of each node.

Fig. 6. When the reference changes, the *self-triggered* MPC performs frequent control updates and thus quickly regulates the output to the new reference value. This results in higher energy usage at the harvesting-based nodes and therefore a dip in their energy storage state of charge. The short time between control updates and exploited energy flexibility are subsequently compensated by longer time intervals between control updates.

With these behaviors, the proposed *self-triggered* approach achieves improvements in the control cost function compared to the *duty cycled*, respectively *periodic* baseline. The *duty cycled* and *self-triggered* performance and thus control cost depend on the predictor. For the *const* predictor, the *self-triggered* approach improves the control cost by 61.8 % in comparison to the *periodic* method and the *duty cycled* approach has a cost that is 18.4 % lower than that of the *periodic*. With the *ewma* predictor, the *self-triggered* approach improves the control cost also by 61.8 % and the *duty cycled* approach by 40.8 % in comparison to the *periodic* method. While the *duty cycled* approach's performance changes greatly between the two predictors, the *self-triggered*'s remains the same. This is a result of the *duty cycled* approach's energy directly dictating when control updates occur. Its energy controller determines different energies and thus times for control updates for the two predictors. Conversely, the *self-triggered* concept allows control updates to occur also in accordance with the control demand balancing it with the provided energy. This resulted in large and consistent improvements with this approach regardless of the predictor.

It is important to highlight the interplay between the energy controller and the MPC. During the simulations, for which typical values for indoor energy harvesting were used, the *self-triggered* method operated correctly and efficiently. The energy controller, self-triggered MPC, and their interface ensured that the energy constraints from the energy harvesting subsystems are respected, the energy controller is optimal, and available energy is efficiently used for control updates when needed. This would not be possible if the energy controller and MPC are designed in isolation.




(a) The outdoor temperature and solar radiation on the South- and West-facing building facade.

(b) The outdoor $CO_2$ concentration and occupancy number.

Fig. 7. A number of different disturbances affect the room's temperature and air quality.

## 6 CASE STUDY: ROOM TEMPERATURE AND AIR QUALITY CONTROL

In this section, we evaluate our proposed hierarchical scheme in a room climate case study where the room's temperature and air quality are regulated. We build a room temperature model, rely on a model of the room's $CO_2$ concentration as an indicator for air quality, and use real data traces for the ambient outdoor conditions. To complete the scenario, the energy harvesting sensor model uses indoor harvesting data from a long-term deployment. The goal is to maintain the room temperature within a user-defined comfort zone and the $CO_2$ concentration below a threshold value while minimizing the effort associated with heating, cooling, and ventilating. The proposed combination of self-triggered MPC and energy controller achieves an excellent control performance by increasing the frequency of control updates when required and decreasing the frequency otherwise. We furthermore explore the impact of sensor measurement noise and inaccuracies in the model of the control plant that the MPC uses.

### 6.1 Setup

*Room Model.* The considered room has two windows and is located in the southwest corner of a building. An air handling unit supplies fresh air from outside that can be heated or cooled. Since the temperature and $CO_2$ dynamics are independent of one another [36], we model them separately. The model of the room's temperature is generated with the BRCM toolbox [46]. The boundary between the room and the remainder of the building is modeled with adiabatic boundary conditions as used in Reference [46]. The temperature is influenced by the outdoor temperature[1] and solar radiation on the two facades. The outdoor temperature and incident solar radiation are depicted in Figure 7(a) for the simulated time frame. The latter is calculated from horizontal solar radiation data[2] using a solar radiation model with an assumed albedo factor of 0.2 [20]. People in the room contribute with a heat gain of 150 W per person to the room's temperature [3]. We define the room's occupancy to follow the curve shown in Figure 7(b). For the dynamics of the $CO_2$ concentration, we use the linear approximation of the non-linear model presented in Reference [8]. Real measurement traces of ambient outdoor $CO_2$ concentrations from Reference [19] reflect the $CO_2$ concentration of the fresh air the air handling unit supplies. Figure 7(b) depicts the outdoor $CO_2$ concentration for the simulated time frame. Occupants in the room increase the $CO_2$ concentration on average by approximately 12 g/h [31].

*Model Predictive Control.* The temperature range comfortable for people is specified to span from 20 °C to 25 °C and the $CO_2$ concentration should remain below 1,500 ppm. As these are not hard

[1] http://www.soda-pro.com/web-services/meteo-data/merra, retrieved on August 3, 2021.
[2] https://ads.atmosphere.copernicus.eu, retrieved on September 3, 2021.

constraints but objectives, slack variables are introduced. The slack variables capture temperatures that violate the comfort zone and $CO_2$ concentrations that exceed the threshold. The control plant process has three inputs: the ventilation rate, heating, and cooling power. The maximum ventilation rate is 200 kg/h, and for both heating and cooling the maximum power is set to 2 kW. The control cost function captures the goal of maintaining a comfortable temperature and good air quality with the slack variables and minimizes the ventilation, heating, and cooling effort [2],

$$l(x(t), u(t)) = 10^3 \cdot \epsilon^T \epsilon + 10^2 \cdot u_{\text{norm}}(t)^T u_{\text{norm}}(t) \tag{11}$$

where $\epsilon$ is a vector containing the slack variables defined for temperature and $CO_2$ concentration and $u_{\text{norm}}(t)$ are the inputs normalized by their respective maximum values. Since the $CO_2$ concentration is on a much larger scale than temperature, the slack variable associated with $CO_2$ is divided by 1000. The MPC horizon spans ten control updates [30] and the time between two consecutive control updates is constrained to lie between 15 min and 3 h. These values span a typical range from fast to slow control for MPC in building management [39].

For the MPC, predictions of the outside temperature, solar radiation on the facades, outside $CO_2$ concentration, and occupancy are necessary. Methods to generate predictions of these disturbances are orthogonal to the main results of the present article and thus not considered in detail. Instead, the outside $CO_2$ concentration is predicted to be constant at 470 ppm. The occupancy prediction follows a simple curve assuming the room is an office for four people. Thus four people are estimated to be present between 07:00 and 17:00 each day. Predictions of the temperature and solar radiation on the facades underestimate the measured values by 10 % each day between 07:00 and 17:00 and are otherwise assumed to be accurate.

*Sensor.* An energy harvesting sensor node measures the room temperature and air quality and communicates this data to the central unit. Common air quality sensors that are integrated into embedded systems rely on metal oxides for sensing. They require significant resources for their operation even when they are turned on only for short periods of time [22]. Thus, such a node's energy cost per control update is high. We set it to $\mu = 275$ mJ, corresponding to sensing with a metal oxide sensor [22] and communication this data point-to-point with LoRa to a central unit [6]. The solar panel and energy storage capacity are dimensioned accordingly. The former has a size of 10cm by 6.6cm and the latter is on the order that it can support a few control updates per hour for a day. Including an energy flexibility of $R = 3$ J, the capacity of the energy storage is 20 J. Since the dynamics of the plant are relatively slow and the time between two control updates determined by the self-triggered MPC may be multiple hours apart, the length of an epoch is set to 6 h. The energy controller's optimization horizon spans one week, following the weekly periodicity of many indoor environments with distinct characteristics on weekdays and weekends. An estimate of the future harvestable energy is generated with the *ewma* predictor [27] with a weighting factor of $\alpha = 0.5$. Indoor harvesting data [42] from an office with two windows, one facing south and one facing west, from October 2019 is used.

*Modeling Inaccuracies and Measurement Noise.* It is challenging to generate an accurate model of a building's temperature and air quality. On the one hand, precise model parameters are difficult to obtain, and, on the other hand, the used models are a simplified representation of the underlying physical processes. Thus, in real-world deployed control systems the MPC typically relies on a model of the plant process that does not precisely capture the actual plant evolution. We explore the effect of such modeling inaccuracies on the proposed concept with simulations where the model the MPC uses during its optimization differs from the model with which the plant evolution is simulated. For the plant state evolution, we use non-linear $CO_2$ dynamics [8] and we introduce errors in the model's parameters related to both the temperature and $CO_2$ processes. These errors follow a multiplicative error model. The plant evolution model parameter $\tilde{g}$ is generated from the

MPC's model parameter $g$ according to $\tilde{g} = (1 + \eta) \cdot g$ where $\eta$ is a random sample from a Gaussian distribution with zero mean and standard deviation $\sigma = 0.001$.

We also consider the case where sensor nodes provide noisy measurements of the plant state. For example, turning on a metal oxide sensor only for a short measurement requires processing of the measurement and nonetheless leads to non-negligible errors in the sensed data [22]. We simulate the occurrence of noisy measurements by introducing errors in the plant states that the MPC bases its optimization on while the plant is simulated to evolve without these errors. A noisy measurement $\tilde{x}(t_{m,k})$ of $x(t_{m,k})$ at time $t_{m,k}$ is modeled as $\tilde{x}(t_{m,k}) = (1 + \eta) \cdot x(t_{m,k})$ where $\eta$ is a random sample from a Gaussian distribution with zero mean and standard deviation $\sigma = 0.001$.

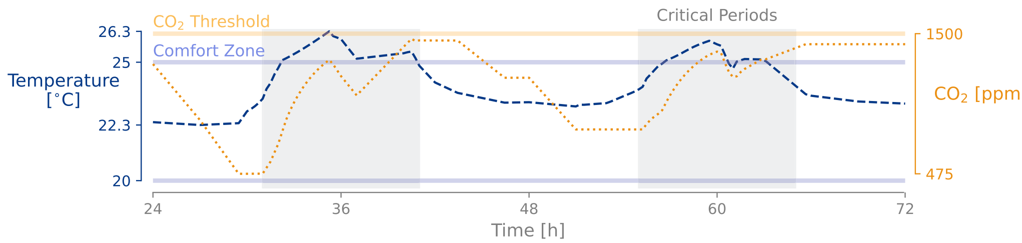
*Baselines and Metrics.* Equivalent to Section 5, the proposed *self-triggered* approach is compared to two baselines, *periodic* and *duty cycled*. The *periodic* baseline has a constant period, which is the shortest period that can be sustained by the harvesting-based sensor node during the previous month. The *duty cycled* baseline employs the energy controller from Section 3.1 with $R = 0$ to determine a control update frequency for each epoch. We evaluate and compare the control cost defined in Equation (11). To understand how the *self-triggered* approach is able to improve the control cost compared to the two baselines, we analyze the timing of the control updates. To this end, we consider the number of control updates that occur when frequent control updates are particularly important, e.g., when the control demand is high, disturbances are mispredicted, or reference values change. We denote such periods of time as critical periods. In the described room climate control scenario, each day between 07:00 and 17:00 is a critical period, since during this time period disturbances are mispredicted.

### 6.2 Simulations

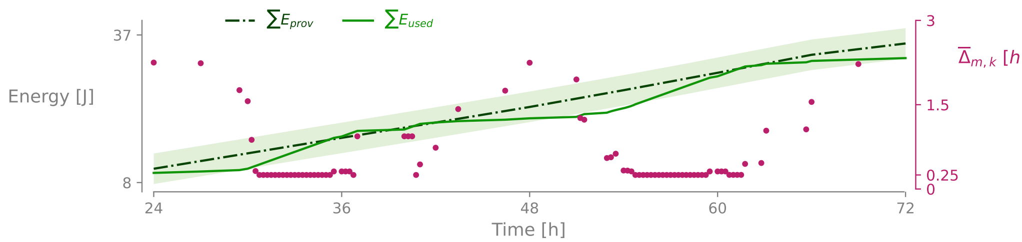
The three controllers are simulated for 3 days regulating the room's temperature and air quality when subject to the outdoor conditions and occupancy pattern shown in Figure 7. The resulting temperature and air quality evolution for the *self-triggered* method without considering model inaccuracies or measurement noise is shown in Figure 8(a) for a 2-day excerpt.

During the night, when there is little demand for control, the *self-triggered* approach executes infrequently, and the provided energy is under-used. This is shown in Figure 8(b) in the right axis, where the average time between two control updates ($\bar{\Delta}_{m,k}$) at night is large. The accumulated used energy ($\sum E_{\text{used}}$) is below the provided energy ($\sum E_{\text{prov}}$). The MPC determines more frequent control updates during the critical periods. The rate at which the room warms up and the $CO_2$ concentration evolves does not match the MPC's predicted behavior because of the prediction errors in the disturbances. This increases the demand for control and the self-triggered MPC nonetheless achieves good control performance by determining shorter times between control updates. The increased energy usage during the critical periods is made possible by the laxity afforded during the nights and the energy flexibility granted by the proposed approach.

The *self-triggered* approach improves the control cost by 68.2% and by 97.0% compared to the *duty cycled*, respectively, *periodic*, baseline. The *self-triggered* approach achieves these drastic improvements in control performance by leveraging key properties of the proposed hierarchical approach. The energy flexibility enables it to optimize its control cost better by performing more control updates when they are particularly important, i.e., during critical periods, and fewer otherwise. Figure 9 summarizes this behavior. It shows the number of control updates performed during the critical periods and the total throughout the 3-day simulation for all three approaches for the simulation without modeling inaccuracies or measurement noise. Although the *duty cycled* approach performs most control updates throughout the simulation horizon, the *self-triggered* method performs most of the control updates when they are important for control quality leading to significant improvements in control performance.




(a) Room temperature and $CO_2$ concentration controlled by the *self-triggered* approach.




(b) The accumulated provided energy with the allowed flexibility and the used energy determined by the self-triggered MPC on the left axis, and the averaged time between consecutive control updates on the right axis.

Fig. 8. The air quality remains below the threshold, while the room temperature slightly exceeds the comfort zone during the critical periods. However, the self-triggered MPC performs frequent control updates during these periods and nonetheless maintains the temperature close to the comfort zone. Before and after these periods, the time between control updates is longer.

We furthermore compare the performance of the *self-triggered* and *duty cycled* approaches when the MPC's model is inaccurate or measurements are noisy. Figure 10 shows the percentage of control updates performed during the critical periods and the control cost relative to the smallest observed control cost of the *duty cycled* baseline. For each type of error and baseline, the results of ten realizations of the random multiplicative errors are depicted. The *self-triggered* approach has a consistently lower control cost and thus better control performance. While for most realizations these improvements are significant, the worst observed control cost of the *self-triggered* approach is only 7 % better than the best of the *duty cycled* baseline. The control performances of both the *duty cycled* and *self-triggered* approach are influenced by model inaccuracies and measurement errors. The control cost varies by 42 % for the *duty cycled* and by 64 % for the *self-triggered* approach. Although these variations are on the same order of magnitude, the *self-triggered*'s performance is slightly more affected by errors. This may arise from errors impacting the performance when the time between two control updates is large. Between these sparse control updates, the *self-triggered* approach does not observe the plant state for longer periods of time and is thus unable to react to the consequences of noisy measurements or model inaccuracies. This behavior is inherent to the self-triggering control paradigm. On the contrary, the *duty cycled* method samples the plant state at periodic times. The control update timing is exclusively dictated by energy availability, which is reflected by the consistent control update percentage. Conversely, modeling inaccuracies and noisy measurements have an effect on the timing of the control updates determined by *self-triggered* approach. This is a result of the *self-triggered* approach optimizing the control update timing based on an inaccurate model or imprecise knowledge of the plant's current state. Nonetheless, in the proposed system concept the impact on the timing of the control updates is limited, because they

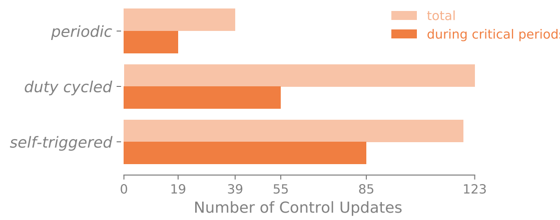



Fig. 9. The significant control performance improvements of the *self-triggered* approach are enabled by the timing of the control updates. The *self-triggered* approach adjusts to the dynamic changes in control demand and performs the most updates when important, i.e., during critical periods, and fewer otherwise. Employing an energy controller also allows the *duty cycled* approach to use more energy and thus perform more control updates than the *periodic* one.


Fig. 10. The *self-triggered* approach consistently results in a lower control cost than the *duty cycled* approach even when measurements are noisy or the MPC relies on an inaccurate model. For both approaches, noisy measurements and model inaccuracies result in variability in the control cost and thus performance.

are optimized not only considering the plant state and control demand but also energy availability. As such the bounds on the resource $r$ ensure that regardless of model inaccuracies or noisy measurements the timing of control updates results in an energy usage that satisfied Equation (6).

### 6.3 Application Scenarios and Limitations of the Proposed System Concept

The proposed control approach is well suited for non-critical application scenarios where the control plant is measured with energy harvesting nodes that are limited in when and how often they can sense. The latter implies that the nodes' energy requirements for sensing, possibly pre-processing, and communicating are non-negligible in comparison to the primary energy they harvest. This can, on the one hand, be a result of severely limited primary energy because of energy-scarce deployment locations or desired small form factor of the nodes. On the other hand, this may be due to energy requirements when communication is long-range or sensors are power hungry. The applicability of the proposed approach is limited to scenarios where there is point-to-point communication between each node and the central unit. The extension to support multi-hop communication protocols like Tsch [18] is considered future work. Nonetheless, efficient long-range communication such as LoRa facilitates communication ranges of multiple kilometers in city environments [48], thus supporting numerous applications. Examples of application scenarios include building automation for retrofitting older buildings that are not equipped with sensors. In indoor environments, photovoltaic energy harvesting nodes only have limited energy available [45]. Additionally, air quality and $CO_2$ sensors, where the latter are for example used to estimate occupancy [21, 52], are power-hungry components [22]. This scenario is evaluated in Section 6. Further application examples include irrigation and environment control for smart agriculture in greenhouses [16] or scheduling and routing in a solid waste management system where a scalable and long-term self-sustainable approach is desirable for sensing the current capacity of garbage bins [26]. In these scenarios, the required long-range communication limits when and how often nodes can make data available to the central unit.

Despite the numerous application scenarios and the proposed concept's efficient use of harvesting-based nodes while accommodating a varying control demand, it suffers from some limitations. The non-deterministic environment might not provide sufficient energy in every epoch. A discussion on the energy requirements of the proposed system concept and methods that can be applied when those requirements are not met is given in Section 3.4. In any case, the energy controller always determines a provided energy that is optimal with respect to maximizing the minimal available energy. Furthermore, the system concept guarantees recursively feasible and convergent control as shown in Section 4 as long as the energy provided by the energy controller satisfies the condition in Equation (9) and nodes are able to support the provided energy and energy flexibility despite energy prediction errors. In this case, the limited energy availability is merely reflected in improved control performance and not in the correct system behavior. Moreover, these two key properties, i.e., recursive feasibility and convergence, hold independent of how much of the finite energy flexibility the MPC has exploited so far and thus how much energy flexibility remains. However, the currently available energy flexibility much like the provided energy impacts the controller's performance. The self-triggered MPC can increase the control update frequency in accordance with control demand to achieve a good control performance when the energy flexibility is not exhausted. Conversely, the MPC can expand the time between control updates to replenish the available energy flexibility once the environment provides more energy or the control demand is lower. Furthermore, an inherent limitation of self-triggered control is that between control updates the system does not sense the plant state and cannot react to important changes that require a different control update timing and actuation than what was planned. Such situations may, for example, arise when the system is subject to model inaccuracies or noisy measurements. A representative study and discussion of the effect model inaccuracies and noisy measurements have on the proposed concept's performance is provided in Section 6.2.

## 7 RELATED WORK

The proposed work lies in the intersection between control systems and distributed energy harvesting embedded systems. First, we summarize the state of the art in distributed embedded systems in conjunction with control, a subject that is highly investigated in the context of cyber-physical systems. Subsequently, we identify existing research in energy management for harvesting-based systems. Last, a brief overview of event-triggered and self-triggered control is presented.

*Distributed Embedded Systems and Control.* Distributed embedded sensing systems are invaluable data sources for many large-scale control systems, as, for instance, found in building control [34], public lighting control in smart cities [38], and peak energy scheduling in power systems [35]. Embedded systems are also leveraged in smaller control systems with faster dynamics, such as a cart pole with an inverted pendulum. In Reference [47], the authors propose a resource-aware control scheme using predictive triggering and demonstrate it with an inverted pendulum. The predictive triggering allows the wireless sensors to identify over-provisioned resources, for example, communication slots, and reallocate them in accordance with the controller. The authors of Reference [32] also studied an inverted pendulum and proposed a closed-loop feedback over a multi-hop wireless network with latency guarantees of 50 ms or less to achieve stability.

These works employ novel control methods in conjunction with wireless sensors and co-design aspects of embedded systems, for example, communication and controllers. Yet, they do not address the energy constraints of embedded systems or the challenges accompanying energy harvesting systems. Especially the latter has strong implications for the control system design. Integrating harvesting-based sensor nodes in automatic control systems requires the design of distinct controllers to address the influence and interplay of energy harvesting with the control system.

*Energy Management in Harvesting-based Embedded Systems.* Embedded energy harvesting systems are employed in numerous applications, including animal tracking, water quality, health, structural health, air quality, and disaster monitoring [25]. These systems require energy management strategies to orchestrate the flow of energy such that dynamic harvesting constraints are kept. Various such methods have been proposed [40]. They adapt a system's operation in accordance with its available energy, local environment, and its objective by for example changing the sensing, actuation, communication, and computation rate, data processing algorithms, or deciding on executing various implementations of the application [4, 12, 14, 50]. By exploiting energy predictions, these methods can optimize a system's long-term performance [24]. Such approaches can be data driven and employ machine learning techniques [33, 41] or provide provable optimal performance for their assumed system model and objective, i.e., References [11, 14]. Conversely, energy management methods can behave in a reactive manner and thus not rely on any predictive model of the harvesting-based system's environment [50]. This simplifies system design especially for unpredictable environments, for example indoors [45]. The described works, both based on predictions or reactive, typically focus on managing the resources of an individual energy harvesting system. Recently, in Reference [43], we discuss the importance of jointly considering multiple distributed energy harvesting systems collaborating in an application, and we propose a joint optimization for the resources of distributed energy harvesting systems.

In this work, we employ this joint optimization approach, since in automatic control applications multiple distributed embedded energy harvesting systems sense the control plant. To enable as many control updates as possible, the proposed energy controller provably optimizes the same objective as the one in Reference [11] provably did for an individual system. We propose a different formulation for the optimization than in Reference [11] and for the first time analytically bound the effect prediction errors have on an energy management strategy. Additionally, the presented finite-horizon energy controller integrates an interface to the self-triggered MPC.

*Event-triggered Control and Self-triggered Model Predictive Control.* Event-triggered and self-triggered control have been studied in the context of resource-constrained control [23, 30]. In both cases, control updates are performed at aperiodic times. In event-triggered control, an observable system property is continuously monitored and when a condition is satisfied, the next control update is performed. When a control update is triggered, data are not available from all distributed sensor nodes but merely from the node or subset of nodes that were triggered. It is thus necessary to instruct all sensors that they need to perform a measurement. Therefore, event-triggered control requires continuous observation and, for low latency, a continuously available communication link from all embedded systems to the control algorithm and vice versa. These prerequisites are at odds with low-power sensing and wireless communication rendering this approach unsuitable for automatic control with energy harvesting sensor nodes. In self-triggered control, the time to the next control update is determined ahead of time by the controller at the current control update while optimizing the process cost. This enables the next sensor sampling and communication to occur at a pre-determined future time, which in turn allows the nodes to remain in an energy-efficient low-power state and the wireless radios to be completely switched off until then.

Predictive triggering and self-triggering including various aspects of this control paradigm such as robustness [10] have been investigated thoroughly [23]. Self-triggered control has also been researched in the domain of wireless networks where imperfections and limitations of the shared network and thus communication are considered [5, 47]. In Reference [7], the authors propose the use of self-triggered MPC for sensor nodes where communication is more expensive than local computation. In Reference [28] self-triggered MPC is applied for constrained systems and the optimization objective combines communication demand and control cost. Self-triggered MPC has also been proposed for constrained linear systems for which only the system output is available when the MPC is triggered [51]. A more general resource-aware self-triggered MPC was investigated in Reference [49] for a resource with deterministic characteristics that is charged at a constant rate. The incorporated resource dynamics align with an established communication resource model of digital networks. A further example is provided that considers the energy requirement of solving the MPC optimization and constraints of the system's asymptotic power consumption. Such a scenario has deterministic static constraints as is pertinent in battery-based systems.

On the contrary, in energy harvesting systems the time-varying and non-deterministic nature of the primary energy source requires, on the one hand, control of the limited resources and, on the other hand, results in resource dynamics with distinct characteristics that need to be incorporated in the self-triggered MPC. Our work proposes a combined and co-designed control of the resources of energy harvesting nodes and self-triggered MPC thus enabling automatic control systems with energy harvesting sensor nodes.

## 8 CONCLUSION

We present a novel hierarchical concept for using energy harvesting sensor nodes in automatic control applications by joining self-triggered MPC and finite-horizon energy control. Self-triggering enables efficient use of available energy by dynamically adjusting the frequency of control updates to optimize the control cost. To fully leverage the potential of self-triggering with harvesting-based systems, the varying primary energy must be managed and the resulting energy constraints taken into account by the MPC. Our optimization formulation is the first to combine an energy controller and self-triggered MPC and provide a formal interface between the two. We formally prove the long-term correctness and reliability of our proposed system. We furthermore demonstrate the validity of our approach using a double integrator system and apply it to a room temperature and air quality control case study. In both scenarios, the proposed concept provides significant improvements in the control performance while respecting the resource constraints imposed by energy harvesting. In the room climate case study, our proposed scheme reduces the control cost function by up to 97%. During periods of mispredicted disturbances to the control plant, our approach performs frequent control updates to achieve good control performance. It thus utilizes the available energy to perform control updates when they are most needed and fewer otherwise. Although this can temporarily increase the energy harvesting sensor nodes' energy consumption, the MPC's flexible use of energy is entirely supported by the energy harvesting sensor nodes without affecting their reliability or energy efficiency.

## APPENDIX

## A PROOFS OF THEOREMS

In this section, we provide detailed proofs of the proposed hierarchical control approach's properties presented in Section 4.

Proof of Theorem 4.1. We first show that optimization problem (2a)–(2e) maximizes the minimal available energy for node $s$, i.e., for any $a_s(i)$ that satisfies the constraints of the optimization problem (2a)–(2e) it holds that $\min_i a_s(i) \geq \min_i a_s^*(i)$, where $a_s^*(i)$ is the optimal solution. Based on the separate optimizations for each node $1 \leq s \leq S$ the common planned energy follows Equation (3). Subsequently, we follow the same proof as in Reference [43]. In particular, we show that the minimum of the common planned energy as determined by the separate optimizations cannot be improved. Last, with constraint Equation (4c) this minimum does not decrease with the joint optimization problem (4a)–(4g), and therefore the energy controller maximizes the minimal common energy.

We now prove the first part, i.e., that optimization problem (2a)–(2e) maximizes the minimal planned energy. We assume $a_s^*(i)$ is the optimal solution to optimization problem (2a)–(2e) and its minimum is $A_s = \min_i a_s^*(i)$. If by adding a constraint $a_s(i) \geq A_s + \epsilon$ for all $0 \leq i < M$ for some infinitesimally small $\epsilon > 0$ the optimization becomes infeasible, then no provided energy with a higher minimum exists and the statement holds. However, if there is a feasible solution, then we consider the optimization problem (2a)–(2e) and add a constraint that at an index where the optimal energy has a minimum, the feasible solution is larger, i.e., $a_s(k) = a_s^*(k) + \epsilon$ for some $k$ for which $a_s^*(k) = A_s$. We further extend the optimization by adding the constraint $a_s(i) \geq A_s$ for all $0 \leq i < M$ and note that as there was a feasible solution for $a(i) \geq A_s + \epsilon$ for all $0 \leq i < M$ there is also a feasible solution to this modified optimization.

We now show that since there is a feasible solution for this modified optimization, we can construct a feasible solution that differs only at two indices, $j, k$, from the optimal solution, $a_s^*$, $a_s(i) = a_s^*(i) \quad \forall i \notin \{j, k\}$. Due to the constraints in Equations (2b) and (2e) any feasible solution of the optimization problem (4a)–(4g) satisfies $\sum_{i=0}^{M-1} a_s(i) = \sum_{i=0}^{M-1} E_{\mathrm{pred,m,s}}(i)$. Therefore, in addition to the above-described difference at index $k$, the constructed feasible solution differs by $a_s(j) = a_s^*(j) - \epsilon$ at index $j$ from the optimal solution. If because of $a_s(k)$ there is a violation of the constraint in Equation (2d) at index $i = c$ such that

$$b_s(c) = b_s^*(k) + \sum_{i=k}^{c-1} E_{\mathrm{pred,m,s}}(i) - \sum_{i=k}^{c-1} a_s^*(i) = -\epsilon$$

then the constructed feasible solution can be obtained by choosing index $j$ such that $k < j < c$ and $a_s^*(j) > A_s$ and setting $a_s(j) = a_s^*(j) - \epsilon$. If no such index exists, then $b_s^*(k)$ must be increased by $\epsilon$. If $b_s^*(k) = B_{R,s} - 2R$, then this cannot be achieved, and there is no feasible solution. Otherwise, index $j$ can be set to $j = k - 1$. There is thus a feasible solution for which $a_s(k) = a_s^*(k) + \epsilon = A_s + \epsilon$ and for a single additional index $a_s^*(j) = A_s + \Delta$ with $\Delta > 0$ is decreased by $\epsilon$, while all other provided energies are identical to the optimal one.

Finally, there is a contradiction in the optimality of the objective, Equation (2a). If there is such a feasible solution, then the objective was not optimal for the assumed optimal provided energies $a_s^*(i)$. The summands of the objectives of the optimal and constructed feasible solution differ only at two indices and thus, for their comparison, only these have to be considered. Therefore, the difference in the objectives is

$$(a_s^*(k)^2 - (a_s^*(k) + \epsilon)^2) + (a_s^*(j)^2 - (a_s^*(j) - \epsilon)^2) = (A_s^2 - (A_s + \epsilon)^2) + ((A_s + \Delta)^2 - (A_s + \Delta - \epsilon)^2) = 2\epsilon(\Delta - \epsilon)$$

As $\epsilon$ is infinitesimally small, we have $2\epsilon(\Delta - \epsilon) > 0$. Consequently, the objective was not minimal and thus not optimal. This concludes the proof that optimization problem (2a)–(2e) maximizes the minimal energy for node $s$.

By contradiction, we prove that the minimum of the resulting $u_{\mathrm{sep}}^*$ cannot be improved. Suppose there is a common energy $\hat{u} = \min_{s,i} a_s(i)$ with $\hat{u} > \min_i u_{\mathrm{sep}}^*(i)$. This implies there has to be at least one harvesting-based node $1 \leq s \leq S$ whose minimum energy is larger than what the optimization problem (2a)–(2e) determined. This directly contradicts the proven statement above. In conclusion, the energy controller determines a common energy to be used $u^*_{\text{joint}}$ whose minimum is maximized. □

Proof of Theorem 4.2. We first prove that the optimal solutions to the optimization problem (2a)–(2e) are Lipschitz continuous with respect to the prediction errors and, subsequently, determine the Lipschitz constant.

We reformulate the optimization problem (2a)–(2e) to a standard parametric quadratic programming problem,

$$\min_{a_s} \frac{1}{2} a_s^T D a_s \quad : \quad A a_s \geq \lambda_{s,m}$$

where $a_s = [a_s(0)\, a_s(1) \cdots a_s(M-1)]^T$, $D = I_M$, and $\lambda$ is the free parameter. The constant factor of $\frac{1}{2}$ does not influence the minimization objective. To specify $A$ and $\lambda$, we introduce a lower triangular matrix $G \in \mathbb{R}^{M \times M}$, a matrix $F \in \mathbb{R}^{M \times M+1}$, and a block matrix $H \in \mathbb{R}^{(2M+2) \times M}$,

$$G_{ij} = \begin{cases} 1 & \text{for } i \geq j \\ 0 & \text{for } i < j \end{cases} \qquad F_{ij} = \begin{cases} 1 & \text{for } i \geq j-1 \\ 0 & \text{for } i < j-1 \end{cases} \qquad H = \begin{bmatrix} h & 0 & \cdots & 0 \\ 0 & h & 0 & \dots \\ & & \cdots & \\ 0 & \cdots & 0 & h \\ 0 & \cdots & 0 & h \end{bmatrix} \quad \text{with} \quad h = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The boundary conditions and energy predictions are incorporated into two vectors,

$$v_s = \begin{bmatrix} 0 & -(B_{R,s} - 2R) & \cdots & 0 & -(B_{R,s} - 2R) & b_s(0) & -b_s(0) & 0 & \cdots & 0 \end{bmatrix}^T \in \mathbb{R}^{3M+2}$$
$$w_{s,m} = \begin{bmatrix} b_s(0) & E_{\text{pred,m,s}}(0) & \cdots & E_{\text{pred,m,s}}(M-1) \end{bmatrix}^T \in \mathbb{R}^{M+1}$$

An equivalent parametric reformulation of optimization problem (2a)–(2e) is defined by matrix $A$ and parameter $\lambda_{s,m}$ with

$$A = \begin{bmatrix} H \cdot G \\ I_M \end{bmatrix} \qquad \lambda_{s,m} = v_s - \begin{bmatrix} H \cdot F \cdot w_{s,m} \\ 0 \end{bmatrix}$$

and matrix $D$ as specified above. The parametric solution of the standard quadratic programming problem is Lipschitz continuous with respect to the parameter $\lambda$ [9, Chapter 4] and thus

$$\|a^*_{\text{s,pred}} - a^*_{\text{s,harv}}\|_2 \leq L \cdot \|\lambda_{\text{pred,s,m}} - \lambda_{\text{harv,s,m}}\|_2$$

By Reference [15, Theorem 3.1], the tightest Lipschitz constant $L$ is determined by $L = \gamma(D, A)$, where $\gamma(A, D) = \max\{\|S\|_2 \,:\, S \in P(D, A)\}$ and $\|S\|_2$ is the largest singular value of $S$. To determine the set of matrices $P(D, A)$, we first define the set of matrices $Q(D, A)$, which contains all invertible symmetric submatrices of $\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix}$ that contain $D$. This set is completely determined by $\begin{bmatrix} D & \hat{A}^T \\ \hat{A} & 0 \end{bmatrix}$, where $\hat{A}$ consists of at most $M$ selected rows of $A$, as otherwise these matrices are not invertible. The set $P(D, A)$ is made up of the first $M$ rows of the inverses of all matrices in $Q(D, A)$.

To simplify the calculation of $\gamma(A, D)$, we explicitly determine the inverse of $\begin{bmatrix} D & \hat{A}^T \\ \hat{A} & 0 \end{bmatrix}$ by noting that $D = I_M$. As a result, we obtain the elements $P(D, A)$ as the first $M$ rows of these inverses with $S = \begin{bmatrix} I - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A} & \hat{A}^T(\hat{A}\hat{A}^T)^{-1} \end{bmatrix}$. The singular values of such a matrix $S$ are the square roots of the eigenvalues of $SS^T$. We find $SS^T = I - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A} + \hat{A}^T(\hat{A}\hat{A}^T)^{-2}\hat{A}$. To relate the eigenvalues of $SS^T$ to the singular values of $\hat{A}$, we substitute $\hat{A} = U\Sigma V^T$ with orthogonal matrices $U$, $V$, and non-negative diagonal matrix $\Sigma$ and obtain $SS^T = V\Sigma^{-2}V^T$. We can thus state that

$$L = 1/(\text{smallest singular value of } \hat{A})$$

where all singular values of $\hat{A}$ need to be strictly positive. Therefore, we need to determine a subset of rows of $A$ that leads to the smallest singular value.

The singular values of a matrix do not change if rows are permuted or multiplied by $-1$. Moreover, due to the Cauchy interlacing theorem adding a row to a matrix can only decrease the minimal singular value. Consequently, to determine the smallest singular value of any $\hat{A}$ we only need to consider matrices $\hat{A}$ that for each row $i$ satisfy either $\hat{a}_{i,j} = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$ or $\hat{a}_{i,j} = \begin{cases} 1 & i \geq j \\ 0 & \text{otherwise} \end{cases}$. These matrices $\hat{A}$ are invertible.

We now show a lower bound on the smallest singular value of any $\hat{A}$ that needs to be considered. To this end, we determine the inverse of any $\hat{A}$ and show that $\|\hat{A}^{-1}\|_F \leq \sqrt{2M-1}$, where $\|\cdot\|_F$ denotes the Frobenius norm. It is well known that the Frobenius norm of a matrix equals the $l_2$-norm of its singular values and that the singular values of the inverse of a matrix are its reciprocals. Hence, it holds that $\|\hat{A}^{-1}\|_F = \sqrt{\sum_i \sigma_i^{-2}}$, where $\sigma_i$ are the singular values of $\hat{A}$. To determine the smallest singular value $\sigma_1$ that satisfies this equality, we set all other $\sigma_i^{-2} = 0$ for $i \neq 1$ and obtain the lower bound $\sigma_1 \geq \frac{1}{\|\hat{A}^{-1}\|_F} \geq \frac{1}{\sqrt{2M-1}}$. As a result, we can bound the Lipschitz constant as

$$L \leq \sqrt{2M-1}$$

It remains to show that $\|\hat{A}^{-1}\|_F \leq \sqrt{2M-1}$ for any $\hat{A}$. We prove that $\hat{A}^{-1}$ only has elements 0, 1, or $-1$, and at most $2M-1$ of them are 1 or $-1$. As the Frobenius norm of a matrix equals the $l_2$-norm of its elements, we obtain the desired result $\|\hat{A}^{-1}\|_F \leq \sqrt{2M-1}$. Let us denote the elements of $\hat{A}^{-1}$ as $\hat{a}^{-}_{i,j}$. By noting that $\hat{A}\hat{A}^{-1} = I_M$ it can be verified for the considered set of matrices $\hat{A}$ that

$$\hat{a}^{-}_{i,j} = \begin{cases} 1 & i = j \\ -1 & i > j \wedge \hat{a}_{i,j} = 1 \wedge \sum_{k=i-1}^{1} \hat{a}^{-}_{k,j} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Therefore, at most one element in every column of $\hat{A}^{-1}$ has the value $-1$ and exactly one element has the value 1 and thus $\|\hat{A}^{-1}\|_F \leq \sqrt{2M-1}$.

Finally, we exploit the structure of $\lambda$ to restate the right-hand side as $\|\lambda_{\text{pred,s,m}} - \lambda_{\text{harv,s,m}}\|_2 = \|H \cdot F \cdot (w_{pred,s,m} - w_{harv,s,m})\|_2$. We make use of the matrix norm induced by the Euclidean norm and their consistency to arrive at $\|H \cdot F \cdot (w_{s,m,pred} - w_{s,m,harv})\|_2 \leq \|H \cdot F\|_2 \cdot \|(w_{s,m,pred} - w_{s,m,harv})\|_2$. This allows us to use the following inequality between induced norms for any matrix C: $\|C\|_2 \leq \sqrt{\|C\|_1 \cdot \|C\|_\infty}$. Based on the structure and elements of the matrix $H \cdot F$, the one and infinity norm of $H \cdot F$ are $2M+2$ and $M+1$, respectively. Last, since $w_{s,m,pred}$ and $w_{s,m,harv}$ do not differ in the first element, the norm of their difference is equivalent to the norm of the difference between predicted and harvested energy. In conclusion, by combining these statements the difference in the optimal provided energies when using predictions as opposed to the harvested energy is

$$\|a^*_{\text{s,pred}} - a^*_{\text{s,harv}}\|_2 \leq \sqrt{2}(M+1)\sqrt{2M-1}\|E_{\text{pred,m,s}} - E_{\text{harv,s}}\|_2 \leq 2(M+1)^{3/2}\|E_{\text{pred,m,s}} - E_{\text{harv,s}}\|_2 \quad \square$$

Proof of Theorem 4.3. The energy harvesting subsystem model defines the energy storage state of charge iteratively by Equation (1). The physical energy storage cannot exceed its capacity or become negative even when the energy consumer uses or attempts to use energy that would lead to such a state of charge, the limits in Equation (1) ensure this. The theorem, however, states that under application of Algorithm 1 and with accurate predictions the energy consumer never attempts to use energy in a way that would lead to a negative energy storage. We therefore remove the lower limit in Equation (1) and show that the state of charge nonetheless remains larger or equal to zero. Hence, we consider the energy storage state of charge evolution as

$$E_{\text{stor,s}}(T_{m+1}) = E_{\text{stor,s}}(T_m) + E_{\text{harv,s}}(T_m) - E_{\text{used}}(T_m)\big]^{B_{R,s}}$$

This can be rewritten to consider the harvested energy as accurately predicted as well as expanding with summands that are zero and expanding one of the sums,

$$\begin{aligned} E_{\text{stor,s}}(T_{m+1}) = E_{\text{stor,s}}(T_m) - R - \sum_{j=0}^{m-1}(E_{\text{prov}}(T_j) - E_{\text{used}}(T_j)) + R + \sum_{j=0}^{m}(E_{\text{prov}}(T_j) - E_{\text{used}}(T_j)) \\ + E_{\text{pred,m,s}}(0) - E_{\text{prov}}(T_m)\big]^{B_{R,s}} \end{aligned}$$

The provided energy $E_{\text{prov}}(T_m)$ is determined according to Equation (5) based on the results of the energy controller executed at $T_m$. This implies for the As we assume the optimization problem (1) is feasible $E_{\text{prov}}(T_m) = u_{\text{joint}}(0) \le a_s^*(0)$. With this bound on the provided energy and Equations (4f) and (4b), we bound the previous equation,

$$\begin{aligned} E_{\text{stor,s}}(T_{m+1}) &= \hat{b_s}(0) + R + \sum_{j=0}^{m}(E_{\text{prov}}(T_j) - E_{\text{used}}(T_j)) + E_{\text{pred, m}}(0) - E_{\text{prov}}(T_m)\big]^{B_{R,s}} \\ &\ge \hat{b_s}(0) + E_{\text{pred,m,s}}(0) - a_s^*(0) + R + \sum_{j=0}^{m}(E_{\text{prov}}(T_j) - E_{\text{used}}(T_j))\big]^{B_{R,s}} = \hat{b_s}(1) + R + \delta E\big]^{B_{R,s}} \end{aligned}$$

where $\hat{b}(1) = \hat{b}(0) + E_{\text{pred,m,s}}(0) - a_s^*(0)$ is the first discrete-time step in the optimization at time $T_m$ according to Equation (4b) and $\delta E$ is the deviation of the used energy from the provided energy up to time $T_{m+1}$. The optimal solutions $a_s^*$ of optimization problem (4a)–(4g) guarantee that the constraint in Equation (4e) is satisfied and thus $\hat{b}(1) \in [0, B_R - 2R]$. In addition, the interface between the finite-horizon energy controller and self-triggered MPC ensures that the deviation between provided and used energy is bounded by $R$, as specified in Equations (6) and (8f), and therefore $\delta E \in [-R, R]$. As this holds recursively for all epochs and with the assumed initial state of charge, the state of charge of any node $1 \le s \le S$ thus never becomes negative. □

Proof of Theorem 4.4. Within each epoch, the resource has the same structure as in Reference [49], and we therefore apply the same proof as presented there. We show that if for $t_{m,k-1}$ a feasible solution exists, then a feasible solution also exists for $t_{m,k}$ within the same epoch. As we assume the optimization problem (1) is feasible at the beginning of the epoch, it is recursively feasible within each epoch.

Suppose that at time $t_{m,k-1}$ the input $\{\hat{u}_0(t_{m,k-1}), \hat{u}_1(t_{m,k-1}), \ldots, \hat{u}_{N-1}(t_{m,k-1})\}$ and time between control updates $\{\hat{\Delta}_1(t_{m,k-1}), \hat{\Delta}_2(t_{m,k-1}), \ldots, \hat{\Delta}_N(t_{m,k-1})\}$ resulted in a feasible solution. Then, at time $t_{m,k}$, the input and time between control updates are constructed as

$$\{\hat{u}_1(t_{m,k-1}), \ldots, \hat{u}_{N-1}(t_{m,k-1}), 0\} \quad \text{and} \quad \{\hat{\Delta}_2(t_{m,k-1}), \ldots, \hat{\Delta}_N(t_{m,k-1}), \tilde{\Delta}\}$$

where $\tilde{\Delta}$ is selected such that $E_{\text{prov}}(T_m)\frac{\tilde{\Delta}}{\Delta_{\text{epoch}}} - \mu = 0$ result in a feasible solution to the modified self-triggered MPC optimization (10). Because $E_{\text{prov}}(T_m)$ has an upper and lower bound as specified in Equation (9), there exists by design a $\tilde{\Delta} \in [\Delta_{\min}, \Delta_{\max}]$ that satisfies $E_{\text{prov}}(T_m)\frac{\tilde{\Delta}}{\Delta_{\text{epoch}}} - \mu = 0$. The thus constructed input and distribution of control updates is a feasible solution, because, under the assumptions in Section 3, $\hat{u}_j = 0$ is in $\mathcal{U}$. Furthermore, as the terminal state constraint is fulfilled at $t_{m,k-1}$ and the zero input is assumed to retain the state at zero, the terminal constraint is also fulfilled at $t_{m,k}$. In addition, $\tilde{\Delta}$ is chosen such that the final resource level does not change and thus the terminal constraint on the resource is also satisfied at $t_{m,k}$. Therefore, there exists a feasible solution for the optimization at time $t_{m,k}$, and within each epoch, the optimization is recursively feasible. □

Proof of Theorem 4.5. We construct a sequence of feasible solutions as described above for each epoch and thus apply the same convergence proof as in Reference [49]. We denote by $\tilde{}$ the system under application of the constructed sequence and by $^*$ the optimal solution to the optimization. The control cost function decreases between two subsequent control updates within an epoch as

$$V^*(x(t_{m,k+1}), r(t_{m,k+1})) - V^*(x(t_{m,k}), r(t_{m,k})) \leq V(\tilde{x}(t_{m,k+1}), \tilde{r}(t_{m,k+1})) - V^*(x(t_{m,k}), r(t_{m,k})) \\ = -\int_0^{\Delta_1^*(t_{m,k})} l(x(t), u_0^*(t))dt$$

At the end of an epoch, we have by induction for the last control update with index $k' = \max k$

$$V^*(x(t_{m,k'}), r(t_{m,k'})) \leq V^*(x(T_m), r(T_m)) - \int_{T_m}^{t_{m,k'}} l(x(t), u(t))dt$$

Since the cost does not increase between epochs, we can state the cost in the limit of epochs as

$$\lim_{m\to\infty} V^*(x(t_{m,k'}), r(t_{m,k'})) \leq V^*(x(0), r(0)) - \lim_{m\to\infty} \int_0^{t_{m,k'}} l(x(t), u(t))dt$$

In our proposed approach, the energy cost per control update is constant and larger than zero, and thus $\lim_{m\to\infty} t_{m,k'} = \infty$. By exploiting these results and the assumptions made in Section 3, one can thus conclude that the process state converges. □

## REFERENCES


[1] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. 2017. Understanding the limits of LoRaWAN. *IEEE Commun. Mag.* 55, 9 (2017), 34–40.

[2] Abdul Afram and Farrokh Janabi-Sharifi. 2014. Theory and applications of HVAC control systems—A review of model predictive control (MPC). *Build. Environ.* 72 (2014), 343–355.

[3] Kaiser Ahmed, Jarek Kurnitski, and Bjarne Olesen. 2017. Data for occupancy internal heat gain calculation in main building categories. *Data Brief* 15 (2017), 1030–1034.

[4] Patroklos Anagnostou, Andres Gomez, Pascal A. Hager, Hamed Fatemi, J. Pineda de Gyvez, Lothar Thiele, and Luca Benini. 2019. Energy and power awareness in hardware schedulers for energy harvesting IoT SoCs. *Integration* 67 (2019), 33–43.

[5] Dominik Baumann, Fabian Mager, Ulf Wetzker, Lothar Thiele, Marco Zimmerling, and Sebastian Trimpe. 2021. Wireless control for smart manufacturing: Recent approaches and open challenges. *Proc. IEEE* 109, 4 (2021), 441–467.

[6] Simon Benninger, Michele Magno, Andres Gomez, and Luca Benini. 2019. Edgeeye: A long-range energy-efficient vision node for long-term edge computing. In *10th International Green and Sustainable Computing Conference (IGSC'19).* IEEE, 1–8.

[7] J. D. J. Barradas Berglind, T. M. P. Gommans, and W. P. M. H. Heemels. 2012. Self-triggered MPC for constrained linear systems and quadratic costs. *IFAC Proc. Vol.* 45, 17 (2012), 342–348.

[8] Anass Berouine, Radouane Ouladsine, Mohamed Bakhouya, and Mohamed Essaaidi. 2020. Towards a real-time predictive management approach of indoor air quality in energy-efficient buildings. *Energies* 13, 12 (2020), 3246.

[9] J. Frédéric Bonnans and Alexander Shapiro. 2013. *Perturbation Analysis of Optimization Problems.* Springer.

[10] Florian David Brunner, Maurice Heemels, and Frank Allgöwer. 2016. Robust self-triggered MPC for constrained linear systems: A tube-based approach. *Automatica* 72 (2016), 73–83.

[11] B. Buchli, P. Kumar, and L. Thiele. 2015. Optimal power management with guaranteed minimum energy utilization for solar energy harvesting systems. In *2015 International Conference on Distributed Computing in Sensor Systems*, IEEE, 147–158.

[12] Bernhard Buchli, Felix Sutton, Jan Beutel, and Lothar Thiele. 2014. Dynamic power management for long-term energy neutral operation of solar energy harvesting systems. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys'14)*. 31–45.

[13] Bernhard Buchli, Felix Sutton, Jan Beutel, and Lothar Thiele. 2014. Towards enabling uninterrupted long-term operation of solar energy harvesting embedded systems. In *European Conference on Wireless Sensor Networks*. Springer, 66–83.

[14] A. Caruso, S. Chessa, S. Escolar, X. del Toro, and J. C. López. 2018. A dynamic programming algorithm for high-level task scheduling in energy harvesting IoT. *IEEE IoT J.* 5, 3 (2018), 2234–2248. https://doi.org/10.1109/JIOT.2018.2828943

[15] Lucian Coroianu. 2016. Best lipschitz constants of solutions of quadratic programs. *J. Optim. Theory Appl.* 170 (2016), 853–875. https://doi.org/10.1007/s10957-016-0966-2

[16] Ying Ding, Liang Wang, Yongwei Li, and Daoliang Li. 2018. Model predictive control and its application in agriculture: A review. *Comput. Electr. Agric.* 151 (2018), 104–117.

[17] Stefan Draskovic and Lothar Thiele. 2021. Optimal power management for energy harvesting systems with a backup power source. In *10th Mediterranean Conference on Embedded Computing (MECO'21)*. IEEE, 1–9.

[18] Simon Duquennoy, Atis Elsts, Beshr Al Nahas, and George Oikonomo. 2017. Tsch and 6tisch for contiki: Challenges, design and evaluation. In *Proceedings of the 13th International Conference on Distributed Computing in Sensor Systems (DCOSS'17)*. IEEE, 11–18.

[19] EMPA, Swisscom AG, Decentlab GmbH, and Swiss Data Science Center. 2019. Carbosense T and RH Data—Release October 2019. https://doi.org/10.18160/RW69-MP2Y

[20] L. M. Fernández-Ahumada, J. Ramírez-Faz, R. López-Luque, A. Márquez-García, and M. Varo-Martínez. 2019. A Methodology for buildings access to solar radiation in sustainable cities. *Sustainability* 11, 23 (2019). https://doi.org/10.3390/su11236596

[21] Alessandro Franco and Francesco Leccese. 2020. Measurement of CO2 concentration for occupancy estimation in educational buildings with energy efficiency purposes. *J. Build. Eng.* 32 (2020), 101714.

[22] Markus-Philipp Gherman, Yun Cheng, Andres Gomez, and Olga Saukh. 2021. Compensating Altered Sensitivity of duty-cycled MOX gas sensors with machine learning. In *Proceedings of the 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON'21)*. IEEE, 1–9.

[23] W. P. M. H. Heemels, Karl Henrik Johansson, and Paulo Tabuada. 2012. An introduction to event-triggered and self-triggered control. In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC'12)*. IEEE, 3270–3285.

[24] Roy Chaoming Hsu and Tzu-Hao Lin. 2018. A fuzzy q-learning based power management for energy harvest wireless sensor node. In *Proceedings of the International Conference on High Performance Computing & Simulation (HPCS'18)*. IEEE, 957–961.

[25] S. R. Jino Ramson and D. J. Moni. 2017. Applications of wireless sensor networks—A survey. In *Proceedings of the International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT'17)*. 325–329. https://doi.org/10.1109/ICIEEIMT.2017.8116858

[26] Ola M. Johansson and Rolf Johansson. 2008. Model predictive control for scheduling and routing in a solid waste management system. *IFAC Proc. Vol.* 41, 2 (2008), 4481–4486.

[27] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava. 2007. Power management in energy harvesting sensor networks. *ACM Trans. Embed. Comput. Syst.* 6, 4 (2007), 32–es.

[28] H. Li, W. Yan, and Y. Shi. 2018. Triggering and control codesign in self-triggered model predictive control of constrained systems: With guaranteed performance. *IEEE Trans. Automat. Contr.* 63, 11 (2018), 4008–4015. https://doi.org/10.1109/TAC.2018.2810514

[29] Yingzhao Lian, Yuning Jiang, Naomi Stricker, Lothar Thiele, and Colin N. Jones. 2021. Resource-aware stochastic self-triggered model predictive control. *IEEE Control Systems Letters* 6 (2021), 1262–1267.

[30] Y. Lian, S. Wildhagen, Y. Jiang, B. Houska, F. Allgöwer, and C.N. Jones. 2020. Resource-aware asynchronous multi-agent coordination via self-triggered MPC. In *Proceeding of the 59th IEEE Conference on Decision and Control (CDC'20)*. IEEE, 685–690.

[31] Marcel Macarulla, Miquel Casals, Matteo Carnevali, Núria Forcada, and Marta Gangolells. 2017. Modelling indoor air carbon dioxide concentration using grey-box models. *Build. Environ.* 117 (2017), 146–153.

[32] Fabian Mager, Dominik Baumann, Romain Jacob, Lothar Thiele, Sebastian Trimpe, and Marco Zimmerling. 2019. Feedback control goes wireless: Guaranteed stability over low-power multi-hop networks. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (ICCPS'19)*. ACM, 12 pages. https://doi.org/10.1145/3302509.3311046

[33] A. Murad, F. A. Kraemer, K. Bach, and G. Taylor. 2019. Autonomous management of energy-harvesting IoT nodes using deep reinforcement learning. In *Proceedings of the IEEE International Conferences on Self-Adaptive and Self-Organizing System (SASO'19)*, IEEE, 43–51. https://doi.org/10.1109/SASO.2019.00015
[34] Zoltán Nagy, Fah Yik Yong, Mario Frei, and Arno Schlueter. 2015. Occupant centered lighting control for comfort and energy efficient building operation. *Energy Build.* 94 (2015), 100–108.
[35] Truong X. Nghiem and Rahul Mangharam. 2015. Scalable scheduling of energy control systems. In *Proceedings of the SIGBED International Conference on Embedded Software (EMSOFT'15)*. IEEE, 137–146.
[36] Alessandra Parisio, Damiano Varagnolo, Marco Molinari, Giorgio Pattarello, Luca Fabietti, and Karl H. Johansson. 2014. Implementation of a scenario-based mpc for hvac systems: An experimental case study. *IFAC Proc. Vol.* 47, 3 (2014), 599–605.
[37] Pangun Park, Sinem Coleri Ergen, Carlo Fischione, Chenyang Lu, and Karl Henrik Johansson. 2017. Wireless network design for control systems: A survey. *IEEE Commun. Surv. Tutor.* 20, 2 (2017), 978–1013.
[38] Gianni Pasolini, Chiara Buratti, Luca Feltrin, Flavio Zabini, Cristina De Castro, Roberto Verdone, and Oreste Andrisano. 2018. Smart city pilot projects using LoRa and IEEE802. 15.4 technologies. *Sensors* 18, 4 (2018), 1118.
[39] Gianluca Serale, Massimo Fiorentini, Alfonso Capozzoli, Daniele Bernardini, and Alberto Bemporad. 2018. Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities. *Energies* 11, 3 (2018), 631.
[40] Amandeep Sharma and Ajay Kakkar. 2020. A review on solar forecasting and power management approaches for energy-harvesting wireless sensor networks. *Int. J. Commun. Syst.* 33, 8 (2020), e4366.
[41] Shaswot Shresthamali, Masaaki Kondo, and Hiroshi Nakamura. 2017. Adaptive power management in solar energy harvesting sensor node using reinforcement learning. *ACM Trans. Embed. Comput. Syst.* 16, 5s, Article 181 (September 2017), 21 pages. https://doi.org/10.1145/3126495
[42] Lukas Sigrist, Andres Gomez, and Lothar Thiele. 2019. Dataset: Tracing indoor solar harvesting. In *Proceedings of the 2nd Workshop on Data Acquisition to Analysis (DATA'19)*. Association for Computing Machinery, New York, NY, 47–50. https://doi.org/10.1145/3359427.3361910
[43] Naomi Stricker, Yingzhao Lian, Yunning Jiang, Colin Jones, and Lothar Thiele. 2021. Joint energy management for distributed energy harvesting systems. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys'21)*. ACM, 575–577.
[44] Naomi Stricker and Lothar Thiele. 2020. Analysing and improving robustness of predictive energy harvesting systems. In *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. ACM, 64–70. https://doi.org/10.1145/3417308.3430264
[45] Naomi Stricker and Lothar Thiele. 2022. Accurate onboard predictions for indoor energy harvesting using random forests. In *Proceedings of the 11th Mediterranean Conference on Embedded Computing (MECO'22)*. IEEE, 1–6.
[46] D. Sturzenegger, D. Gyalistras, V. Semeraro, M. Morari, and R. S. Smith. 2014. BRCM matlab toolbox: Model generation for model predictive building control. In *Proceedings of the American Control Conference*. 1063–1069. https://doi.org/10.1109/ACC.2014.6858967
[47] Sebastian Trimpe and Dominik Baumann. 2019. Resource-aware IoT control: Saving communication through predictive triggering. *IEEE IoT J.* 6, 3 (2019), 5013–5028.
[48] Roman Trüb. 2022. *Design and Testing of Mixed-Range Low-Power Wireless Networks*. Ph.D. Dissertation. ETH Zurich.
[49] Stefan Wildhagen, Colin N. Jones, and Frank Allgöwer. 2020. A resource-aware approach to self-triggered model predictive control: Extended version. (2020). arXiv:eess.SY/1911.10799. Retrieved from https://arxiv.org/abs/1911.10799.
[50] Fan Yang, Ashok Samraj Thangarajan, Gowri Sankar Ramachandran, Wouter Joosen, and Danny Hughes. 2021. AsTAR: Sustainable energy harvesting for the internet of things through adaptive task scheduling. *ACM Trans. Sens. Netw.* 18, 1 (2021), 1–34.
[51] Jingyuan Zhan, Xiang Li, and Zhong-Ping Jiang. 2017. Self-triggered robust output feedback model predictive control of constrained linear systems. In *Proceedings of the American Control Conference (ACC'17)*. IEEE, 3066–3071.
[52] M. S. Zuraimi, A. Pantazaras, K. A. Chaturvedi, J. J. Yang, K. W. Tham, and S. E. Lee. 2017. Predicting occupancy counts using physical and statistical Co2-based modeling methodologies. *Build. Environ.* 123 (2017), 517–528.

Received 6 January 2022; revised 28 March 2023; accepted 3 May 2023