

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A two-layer distributed MPC approach to thermal control of Multiprocessor Systems-on-Chip

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Tilli A., Garone E., Conficoni C., Cacciari M., Bosso A., Bartolini A. (2022). A two-layer distributed MPC approach to thermal control of Multiprocessor Systems-on-Chip. CONTROL ENGINEERING PRACTICE, 122, 1-14 [10.1016/j.conengprac.2022.105099].

Availability:

This version is available at: <https://hdl.handle.net/11585/868929> since: 2024-09-22

Published:

DOI: <http://doi.org/10.1016/j.conengprac.2022.105099>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

A two-layer distributed MPC approach to thermal control of Multiprocessor Systems-on-Chip

Andrea Tilli^a, Emanuele Garone^b, Christian Conficoni^a, Matteo Cacciari^a, Alessandro Bosso^a, Andrea Bartolini^a

^aDepartment of Electrical, Electronic and Information Engineering, University of Bologna, Viale del Risorgimento 2, Bologna, Italy. Email: {andrea.tilli, christian.conficoni3, matteo.cacciari, alessandro.bosso, a.bartolini}@unibo.it.

^bUniversité Libre de Bruxelles, 50 Av. F.D. Roosevelt, B-1050 Brussels, Belgium. Email: egarone@ulb.ac.be.

Abstract

Next-generation Multiprocessor, or Multicore, Systems-on-Chip offer very high computing performance at the expense of a very high power density unevenly distributed on the chip. The hot spots thus generated represent a significant source of performance and reliability degradation, as well as power consumption increase. In recent years, run-time thermal control strategies have been developed to deal with this issue by acting on some “computational knobs” (e.g., clock frequencies and supply voltages). In this context, schemes based on Model Predictive Control (MPC) are particularly suitable due to their capability to deal with constraints explicitly. In this paper, we first discuss relevant properties for the design of predictive controllers for thermal systems. Starting from the Partial Differential Equation (PDE) describing heat diffusion in a solid, we prove meaningful feasibility properties that can be leveraged for constraint reduction. We then present a procedure to derive approximated but effective modular thermal models intended to build an efficient distributed MPC. Finally, a two-layer control solution is proposed to maximize performance while preserving feasibility despite model approximations. The effectiveness of this approach is validated through extensive and realistic numerical simulations.

Keywords:

Distributed MPC, Many-core Systems, Thermal Modeling

1. Introduction

In recent years, the growing demand for efficient and high-performance computing has fostered the diffusion of Multiprocessor Systems-on-Chip (MPSoCs) in every sector of the worldwide economy, ranging from industry to everyday products such as laptops, tablets, and mobile phones. From the early 70’s up to now, the performance trend in microprocessors has been accurately described by Moore’s law, according to which performance is doubled every 18 months. Until the early 2000’s the main factors enabling performance improvements were micro-architectural enhancements (such as instruction-level parallelism) and technology scaling. Then, the so-called “Power Wall” was hit. Ever since, silicon power density could not be kept invariant with respect to technology scaling and improvement, owing to leakage and dynamic power terms that do not scale as well as the transistors’ size.

For this reason, single CPUs gradually disappeared in favor of multi-core architectures, with multiple cores integrated on the same silicon die. Exploiting parallelism, multi-core processors increase the computational throughput without increasing frequency. Nevertheless, power remains a concern for MP-SoC improvement. Because of the speed disparity between the (quadratic) scaling of transistor size and the reduction of power consumption, power density still grows. As a consequence, the temperature on the chip unevenly increases, creating hot spots and thermal gradients. High temperatures adversely affect the

chip’s operating speed by degrading the carrier mobility in transistors, whereas the power consumption is increased since leakage power exponentially increases with temperature. In addition, components’ lifetime shortens as a consequence of high temperatures and hot spots, while thermal gradients accelerate failure mechanisms such as electro-migration, stress migration, and dielectric breakdown. These effects cause a bottleneck in enhancing MPSoC performance and reliability, commonly called “Thermal Wall”. Lately, this problem has received an increasing level of attention.

In general, the approaches that address such thermal issues can be grouped into two prominent families: Static Thermal Management (STM) and Dynamic Thermal Management (DTM) techniques. The former approach allows increasing the power consumption sustained by the chip (the so-called “thermal design power”) by acting on architectural design, e.g., through a suitable sizing of heatsinks, fans, and, in some cases, liquid cooling. However, for today’s MPSoCs, STM strategies can no longer remove heat in worst-case conditions. Consequently, DTM techniques have become more and more crucial to bound the operating temperature by “run-time active control”, exploiting temperature sensors along with Dynamic Voltage and Frequency Scaling (DVFS) (Kang et al., 2011), thread migration/scheduling (Coskun et al., 2009), (Liu et al., 2014), throttling (Tsai and Chen, 2014), and clock gating. Then, standard cooling systems can be designed to handle the average case, leaving the management of peaks to active control.

This work focuses on DTM techniques based on frequency scaling. Early methods relied on simple empirical “ON-OFF” strategies: in case of temperature-bound violations, all core frequencies were set to the minimum value. Later on, the benefits of feedback control have been exploited, leading to the adoption of classic regulators such as PIDs (Skadron et al., 2002; Kadin et al., 2009; Wang et al., 2008).

More recently, the framework of Model Predictive Control (MPC) has been recognized as very promising due to the capability of explicitly dealing with inputs (i.e., core frequencies) and state-space constraints (i.e., temperature bounds) (Zanini et al., 2009; Wang et al., 2009; Bartolini et al., 2012), while pursuing performance optimization. In this context, frequency-scaling is typically used as the control input, and a linear-quadratic formulation of the MPC is considered. This control problem can be either managed online (implicitly) by solving a constrained Quadratic Programming (QP) problem at each sampling instant or off-line (explicitly) by solving a multi-parametric constrained QP problem (Bemporad et al., 2002). In both cases, the main impact on performance is the problem complexity, which depends on the dimension of inputs and states and is strongly affected by the number of constraints. In implicit formulations, the computational burden scales with dimensionality, whereas in explicit ones, the criticality is the memory usage. Additionally, the system model should be simple enough to comply with computational and storage limitations, yet sufficiently accurate to produce reliable predictions. This task is increasingly difficult in modern and near-future MPSoCs, which, for High-Performance Computing (HPC) and Datacenter applications, could integrate hundreds of cores (McKeown et al., 2018). A common way to deal with controller complexity in such conditions is through distributed solutions (Bartolini et al., 2012). Indeed, the MPSoC thermal model can be split into several simpler core-centric interacting modules. This way, the centralized MPC problem can be cast into a set of simpler interconnected problems, one for each core. Then, an MPC is implemented separately for each core, leading to a distributed and scalable solution.

Recent works in the MPC framework applied to multicore systems are (Wang et al., 2016), (Wang et al., 2019). In the first contribution, DVFS and task migration are combined, by solving a centralized unconstrained problem (the temperature threshold is penalized in the objective function), and partitioning the task migration part between the core blocks. The second work proposes a nonlinear MPC solution to account for power leakage terms, yet again in a centralized fashion. Distributed MPC solutions are presented in (Camponogara et al., 2021), (Scherer et al., 2015) for resource constrained thermal management of large scale systems such as buildings. In this case, the decomposition is applied to the optimization problem, with a centralized master updating the coupling variables (e.g., Lagrange multipliers) for a set of distributed sub-problems. Other approaches include power budget constraints, along with thermal capping, via heuristic, multi-layer optimization techniques either centralized (Iranfar et al., 2017; Dey et al., 2019), or partially distributed (Bambini et al., 2020). Another popular framework that tackles this class of problems is machine learn-

ing, specifically reinforcement learning. Indeed, there is a considerable literature of reinforcement learning applications to power management and thermal control of computational platforms (Das et al., 2014), (Shafik et al., 2015), (Mandal et al., 2019). As discussed in (Pagani et al., 2018), the main drawbacks of this class of solutions are : significant complexity (leading to processing overhead), and convergence issues (it is not trivial to tune the learning rate for balancing the exploration versus exploitation phases).

In general, the aforementioned methods, lack formal feasibility guarantees, especially in case of model uncertainties.

1.1. Contribution of this paper

This paper proposes a two-layer fully distributed (no centralized component is needed) MPC strategy for reliable thermal capping and optimization of the MPSoC performance. To build an effective MPC scheme, we begin with a careful model analysis of the conditions for feasibility. To the best of our knowledge, this issue is usually disregarded in the specific literature on MPSoC thermal control, i.e., no formal proofs are provided to guarantee constraint satisfaction at all times. In this context, we employ the *heat equation*, i.e., the Partial Differential Equation (PDE) representing the heat conduction (Evans, 1998; Strauss, 1992), to study constraint reduction for both centralized and distributed scenarios. This approach allows defining simple control policies with guaranteed feasibility that need few measurements, no accurate knowledge of the model, and no information exchange among cores (i.e., in a decentralized fashion).

The obtained strategies can be implemented as an *ultimate thermal capping control layer* that is triggered whenever dangerous working conditions occur (i.e., the temperature is too high in some regions). We remark that the thermal capping does not boil down to the expected trivial solution: “whenever the critical temperature is approached, turn off all of the power sources”. Indeed, the PDE analysis allows to extend the basic intuition to the case of decentralized control, i.e., “turn off only the sources whose temperature is too high”, and to the realistic case of power dissipation that cannot be zeroed, i.e., “turn to minimum power the sources whose temperature is dangerous”. To this aim, the thermal constraints are reformulated to show that the dangerous temperature for a source may be different from the silicon critical temperature, as its effects on the neighborhood have to be taken into account.

With the above result at hand, feasibility is always preserved, but no performance optimization can be achieved. As already mentioned, MPC strategies have been proposed to handle this aspect. These approaches modulate the computation (related to the dissipated power) to keep performance as close as possible to the required level while satisfying thermal constraints. In this respect, a crucial aspect concerns the accuracy of the control model, i.e., a finite-dimensional model used to predict the thermal dynamics. To maximize performance, one should stay close to the maximum temperatures. However, using a reduced-order model, one must trade-off between its complexity and the safety margin that must be considered to account for the modeling inaccuracies. Therefore, we propose an identification-based

procedure to perform state- and time-discretization of the original PDE, aiming to obtain for each core a prediction model slightly biased toward temperature overestimation. The result of this procedure is a set of low-complexity thermal models that physically interact between each other.

Finally, a *distributed MPC layer* is defined to achieve performance optimization. Unlike the decentralized thermal capping layer, this controller is regarded as distributed because requires some connectivity properties between cores, i.e., some communication infrastructure to allow information exchange among neighbor cores. Combining this layer with the aforementioned thermal capping strategy, we obtain a *two-layer distributed control strategy* with a low computational burden in performance optimization (given by the distributed MPC) and model-independent guaranteed temperature constraint preservation (thanks to the thermal capping strategy). This way, the MPC layer is not designed to be conservatively robust to unavoidable modeling approximations. Indeed, even if the proposed identification method is polarized toward temperature overestimation, we do not formally guarantee that the MPC preserves feasibility for the actual system. For this, we rely on the ultimate thermal capping layer. Nevertheless, the distributed MPC based on the simplified, discretized models must admit feasible solutions at any time. To this aim, we carry out a feasibility test based on (Löfberg, 2011) and check the consistency of the identified models w.r.t. the properties of thermal systems.

The paper is organized as follows. In Section 2, we formulate the thermal management problem in MPSoCs. In Section 3, after presenting the PDE thermal model and its properties, constraint reduction and feasibility issues are treated to derive the ultimate thermal capping strategy. In this respect, very preliminary results were given (Tilli et al., 2012) for the centralized scenario. Here we address the decentralized scenario and non-zero minimum power, leading to relevant upgrades in the control strategy. In Section 4, we define an \mathcal{H}_∞ identification/model-reduction problem to obtain distributed models and to improve feasibility preservation. In this regard, the approach proposed in (Löfberg, 2011) has been integrated and adapted to our discretization process to discard models that cannot ensure MPC feasibility. The architecture of our two-layers distributed control solution is then presented in Section 5, proposing an efficient and reliable, even if heuristic, tuning procedure to obtain performance optimization and feasibility preservation. Finally, Sections 6 and 7 respectively provide simulation results and a conclusive discussion.

2. Thermal control of MPSoCs

MPSoCs integrate on a single chip substrate memory hierarchies, I/O components, and cores, i.e., processing units connected all together by on-chip networks. From a constructive point of view (see Fig. 1b), the chip architecture is designed to favor heat flow from the active silicon device to the heat sink, that is, from where heat is generated to where it is removed towards the ambient (Mallik et al., 2008). The primary heat sources are localized on cores where the highest power is consumed and where thermal criticalities usually occur (Hamann

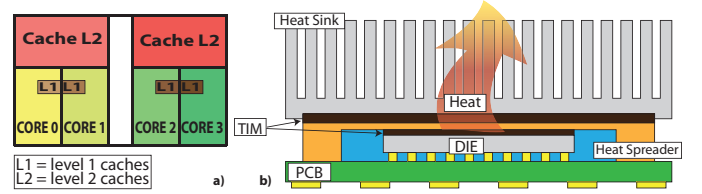


Figure 1: Chip thermal architecture.

et al., 2007). This motivates the choice, quite common in literature, of considering simple thermal models with only cores and cache components for simulations (see Fig. 1).

As previously mentioned, architecture-level design solutions (STM) cannot entirely remove the heat dissipated on the chip due to high power densities. This limit can be addressed with run-time techniques (DTM), which use introspective monitors, sensors, and performance knobs available in modern MPSoCs to maximize computational performance, maintaining the chip temperature below a preset threshold. Recent literature presented many variants of MPC-based DTM schemes for managing thermal issues in MPSoCs (see (Zanini et al., 2009) (Wang et al., 2009) and references therein). The common idea is to act on core frequencies with the twofold objective of meeting the temperature constraints and tracking the frequency targets requested by a high-level SoC manager.

For an MPSoC composed of n_C cores, the prototype of the MPC optimization problem can be formulated in discrete-time and discrete-space, with n_T points selected to represent the die thermal state, as follows:

$$\begin{aligned} \min_{f_C(t), \dots, f_C(t+N-1)} & \sum_{k=0}^{N-1} |f_T(t+k|t) - f_C(t+k|t)|_Q^2 \\ \text{subj. to: } & T_j(t+k+1|t) \leq T_{\text{CRIT}}, \quad j \in \{1, \dots, n_T\}, k \in \{0, \dots, N-1\}, \\ & \mathbf{1}_{n_C} f_{\min} \leq f_C(t+k|t) \leq \mathbf{1}_{n_C} f_{\max}, \quad k \in \{0, \dots, N-1\}, \end{aligned} \quad (1)$$

where N is the prediction horizon (i.e. the number of steps ahead to be evaluated to make an optimal decision at time t), $f_C = [f_{C,1}, \dots, f_{C,n_C}]^T \in \mathbb{R}^{n_C}$ are the frequencies assigned to the cores, $f_T = [f_{T,1}, \dots, f_{T,n_T}]^T \in \mathbb{R}^{n_T}$ are the core target frequencies, $Q \in \mathbb{R}^{n_C \times n_C}$ is a symmetric positive definite matrix, and $|\cdot|_Q$ denotes the norm induced by it, i.e., $|\bar{x}|_Q := \sqrt{\bar{x}^T Q \bar{x}}$. T_{CRIT} and T_j , $j = 1, \dots, n_T$, are respectively the critical temperature and the states of the reduced-order model. Finally, \leq is used to denote the element-wise inequality and $\mathbf{1}_{n_C}$ is a column vector of all ones of dimension n_C .

In problem (1), we implied that a model is available to relate the core frequencies (and the ambient temperature) with future core temperatures (see, e.g., (Zanini et al., 2009; Ladenheim et al., 2018)). Specifically, the notation $T_j(t+k|t)$, $f_C(t+k|t)$, and $f_T(t+k|t)$, $k \in \mathbb{N}$, is used to indicate, respectively, the estimated temperature, the assigned frequencies, and the target frequencies for the future time $t+k$. The prediction model is also required to perform such estimation, and thus, formulate the MPC problem. Such a model can be written as:

$$x(t+1) = \alpha(x(t)) + \beta(f_C(t), w(t), e(t))$$

with state $x \in \mathbb{R}^{n_T}$ collecting all the n_T points temperatures, $\alpha(\cdot) : \mathbb{R}^{n_T} \rightarrow \mathbb{R}^{n_T}$ a state function taking into account the coupling of one state to the others, $\beta(\cdot)$ an input function of proper dimensions, taking into account the effect of controlled inputs (like $f_C \in \mathbb{R}^{n_c}$), and other not directly controllable inputs or external factors (e.g., the ambient conditions), denoted with vectors $w \in \mathbb{R}^{n_c}$ and $e \in \mathbb{R}^{n_c}$ respectively. While $\alpha(\cdot)$ is usually linear, $\beta(\cdot)$ is not, although with some change of coordinates the thermal model can be considered linear. However, we do not further explicit this reasoning here, because we pursue a distributed approach also at the modeling level (i.e., not just decomposing the optimization problem), and some of these concepts are presented later on concerning the proposed solution.

The above approach is usually referred to as *centralized* because the decision variables (i.e., the frequencies assigned to each core) result from the solution of a unique problem running on a single processor. However, the complexity of problem (1) does not scale well with the number of cores (Christofides et al., 2012). For this reason, in the following, we provide a distributed reformulation characterized by a set of local controllers, each one supervising the temperature of one core or a group of cores. Choosing $Q = I_{n_c}$ and assuming to use a controller for each core i , we can consider a local optimization problem of the form

$$\begin{aligned} \min_{f_{C,i}(t), \dots, f_{C,i}(t+N-1)} & \sum_{k=0}^{N-1} |f_{T,i}(t+k|t) - f_{C,i}(t+k|t)|^2 \\ \text{subj. to: } & T_i(t+k+1|t) \leq T_{\text{CRIT}}, \quad k \in \{0, \dots, N-1\}, \\ & f_{\min} \leq f_{C,i}(t+k|t) \leq f_{\max}, \quad k \in \{0, \dots, N-1\}, \end{aligned} \quad (2)$$

where T_i is the temperature representing the i -th core's thermal status. Although the local problems appear to be independent, they are actually coupled since the temperature of each core influences the thermal state of the neighbors.

Following (Bartolini et al., 2012), in this work we consider a slightly modified formulation of problem (2) where the target and commanded frequencies ($f_{T,i}$ and $f_{C,i}$) are replaced with the target and commanded power consumptions, denoted respectively with $P_{T,i}$ and $P_{C,i}$. In general, the power consumption P_i of each core i can be expressed as a nonlinear function depending on the core frequency f_i and other measurable parameters w_i related to core voltages and workloads. In particular, it holds that

$$P_i = g(f_i, w_i), \quad (3)$$

where $g(\cdot, w_i)$ is assumed to be known and invertible in $[f_{\min}, f_{\max}]$, for all $w_i \in \mathcal{W}$, where \mathcal{W} is the range of all possible workloads and voltages.

By inverting the static map (3) with known w_i , the power commands can be translated into frequencies, which are the actual MPSoC control inputs. We emphasize that the map $g(\cdot, w_i)$ is assumed to be known, as its identification is not the focus of this work. However, its definition is not a trivial task for MPSoC thermal control (we refer to (Bartolini et al., 2012) and references therein for details). Therefore, the prototype of the

control problem becomes, for all i :

$$\begin{aligned} \min_{P_{C,i}(t), \dots, P_{C,i}(t+N-1)} & \sum_{k=0}^{N-1} |P_{T,i}(t+k|t) - P_{C,i}(t+k|t)|^2 \\ \text{subj. to: } & T_i(t+k+1|t) \leq T_{\text{CRIT}}, \quad k \in \{0, \dots, N-1\}, \\ & P_{\min} \leq P_{C,i}(t+k|t) \leq P_{\max}, \quad k \in \{0, \dots, N-1\}, \end{aligned} \quad (4)$$

where the power consumption bounds are given as follows to account for the worst-case scenario of the workload:

$$\begin{aligned} P_{\min} & := \max_{w \in \mathcal{W}} \min_{f \in [f_{\min}, f_{\max}]} g(f, w) \\ P_{\max} & := \min_{w \in \mathcal{W}} \max_{f \in [f_{\min}, f_{\max}]} g(f, w). \end{aligned} \quad (5)$$

Compared to the formulation (2) the advantage of (4) is that, after the input transformation, the chip thermal model can be considered reasonably linear. **Indeed, almost all the nonlinearity is contained in the algebraic function $g(\cdot)$, because, applying thermodynamic first principles, the temperature dynamics is linear w.r.t. the power balance. In particular, see (Bartolini et al., 2012), the discrete-time discrete-space model included as equality constraint in (4) is a linear time-invariant (LTI) system having the following structure:**

$$\begin{aligned} x_i(t+1) & = A_i x_i(t) + B_i [P_{C,i}(t) \quad T_{\text{amb}}(t) \quad T_{n,i}(t)]^T \\ T_i(t) & = C_i x_i(t) \end{aligned} \quad (6)$$

where x_i is the state, A_i , B_i , and C_i are matrices of appropriate dimension, T_{amb} is the ambient temperature, and $T_{n,i}$ are the temperatures of neighbor cores that influence the temperature of the i -th core. These neighbors are usually selected according to the chip floorplan, but the discretization sampling time needs to be considered as well.

In the general framework, the local MPC selects at each time step t the sequence of powers $P_{C,i}(t), \dots, P_{C,i}(t+N-1)$ that minimizes the cost and meets temperature constraints defined by problem (4) over an horizon of N time steps, using model (6) to predict the temperature evolution over such horizon given the state at time t . The first element of the sequence is then applied and, repeating the procedure at the next time step, based on the new state measurement and moving forward the prediction, in a typical receding horizon fashion. An important aspect to note is that parameters w_i , particularly the workload, usually are not fully known over the prediction horizon. The design of the controller must account for these issues. This unpredictability can be handled assuming low variability between sampling instants, that is, choosing $w_i(t) = w_i(t-1)$. As regards the number of inputs and the order of the model, which characterize its complexity, they can be set according to experimental tests and physics-based considerations (Bartolini et al., 2012). In spite of the noticeable literature related to problems (1), (2), and even (4), to the best of the authors' knowledge, the following essential issues have not been entirely addressed, and they should be more rigorously assessed to obtain more effective control solutions.

- Feasibility of the control problem must be verified over different prediction horizons, taking into account that the

constraint on maximum temperature must be met over all the die, not only for the points considered in the spatial discretization. Here, we leverage the PDE thermal model properties to draw general results, which are not influenced by potential side-effects due to time- and space-discretization adopted in control-oriented modeling.

- Rules for model discretization must be stated. A prediction model should be simple, yet accurate, to limit the computational effort and capture the die temperatures. Discretization allows model simplification, but accuracy is penalized. In this paper, we exploit the structural properties of thermal systems to capture the points where the die temperature attains its maximum, keeping the model and problem complexity as low as possible.

3. Feasibility properties in the PDE framework

In this section, we present properties useful for designing a reliable and efficient controller for thermal systems subject to uniform maximum temperature constraints. Firstly, we use such results to simplify the control problem formulation, reduce the number of constraints, and prove control feasibility. To generally treat the whole class of thermal systems, independently of time- and space-discretization or other approximations, we model heat conduction in continuous-time for a generic open volume V using the following PDE:

$$\begin{cases} \rho c \frac{\partial T(x,t)}{\partial t} - \alpha \nabla^2 T(x,t) = q(x,t), & (x,t) \in V \times [0, \tau], \\ T(x,0) = T_0(x), & x \in V, \\ T(x,t) = T_{\partial V}(x,t), & (x,t) \in \partial V \times [0, \tau], \end{cases} \quad (7)$$

where the first line is the well-known heat equation, α is the material conductivity, ρ its density, and c its specific heat. In addition, $\partial T(x,t)/\partial t$ is the temperature variation over time, $\nabla^2 T$ denotes the divergence of the temperature gradient (Laplacian), while $[0, \tau]$ is the time interval, and q is the thermal power generated by internal sources. The second and third equations define the initial temperature and conditions on the boundary of the volume ∂V , respectively. These equations are usually given as Dirichlet boundary conditions (DBC), where $T(x,t)$ is completely known on the boundary (i.e., for all $x \in \partial V$ and all $t \geq 0$). Such conditions need to be set according to the Third Principle of Thermodynamics, i.e., forcing the temperature evolution to be always positive. Hence, expressing the temperature in Kelvin degrees, the following additional constraints have to be considered:

$$\begin{aligned} T_0(x) &\geq 0, & \forall x \in V, \\ T_{\partial V}(x,t) &\geq 0, & \forall (x,t) \in \partial V \times [0, \tau]. \end{aligned} \quad (8)$$

Remark 1. *It is also worth noting that the classical Fourier-based heat equation, used in this work to model thermal systems, assumes an infinite speed of heat propagation. To account for the boundedness of the propagation speed, we should use hyperbolic instead of parabolic equations, considering second-order time derivative, or use a nonlinear parabolic equation*

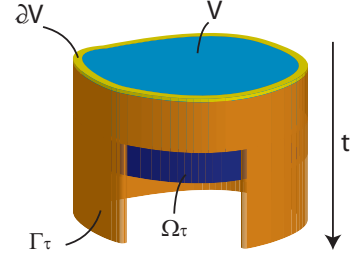


Figure 2: Parabolic cylinder for the 2D volume V .

like the Porous Medium Equation. However, considering the transmission speed of the heat in silicon and the small sizes of chips, the classical linear PDE is a valid approximation.

In the following, we mention a fundamental property of thermal systems that allows to infer where the maximum temperature occurs in the absence of heat sources. This result, known as Maximum Principle, will be useful for the successive proofs.

Maximum Principle (Evans, 1998; Strauss, 1992): For a heat equation of the form (7), consider the parabolic cylinder Ω_τ and the parabolic boundary Γ_τ (both portrayed in Fig. 2) respectively defined as:

$$\Omega_\tau := V \times (0, \tau], \quad \Gamma_\tau := \bar{\Omega}_\tau \setminus \Omega_\tau = \bar{V} \times [0, \tau] \setminus \Omega_\tau, \quad (9)$$

where $\bar{(\cdot)}$ indicates the closure of a set. Let $T(x,t)$ be the solution to the heat equation for $q(x,t) = 0$, and suppose that $T(x,t)$ is sufficiently smooth, i.e., $T(x,t) \in C^2(\Omega_\tau) \cap C(\bar{\Omega}_\tau)$. Then,

1. (weak maximum principle):

$$\max_{(x,t) \in \bar{\Omega}_\tau} T(x,t) = \max_{(x,t) \in \Gamma_\tau} T(x,t), \quad \text{i.e., } \Gamma_\tau \cap [\arg \max_{(x,t) \in \bar{\Omega}_\tau} T(x,t)] \neq \emptyset.$$

2. (strong maximum principle): if V is connected and there exists a point $(x_m, t_m) \in \Omega_\tau$ such that,

$$T(x_m, t_m) = \max_{(x,t) \in \bar{\Omega}_\tau} T(x,t), \quad \text{i.e., } \Omega_\tau \cap [\arg \max_{(x,t) \in \bar{\Omega}_\tau} T(x,t)] \neq \emptyset,$$

then $T(x,t)$ is constant in $\bar{\Omega}_\tau$.

In other words, the maximum temperature is found at the boundaries of the parabolic cylinder.

3.1. Constraint reduction

The key target of the predictive controller is to limit the temperature of the volume V under a specified threshold. This would lead to an optimization problem with an infinite number of constraints, one for each infinitesimal volume element. Starting from the maximum principle, we show how to reduce constraints to a finite number, making the controller implementable, and reducing its complexity.

The next result generalizes the Maximum Principle for the scenarios where the heat equation is affected by n_s heat sources.

Proposition 1. For problem (7), (8), suppose that $q(x, t) \neq 0$ only if $x \in \mathcal{V}_s = \bigcup_{i=1}^{n_s} V_{s,i}$, where $V_{s,i} \subset V$ is a compact connected set, for all $i \in \{1, \dots, n_s\}$. Then, the maximum temperature is found in Γ_τ (see (9)) or in $\mathcal{V}_s \times [0, \tau)$.

Proof. Since (7) is a Cauchy problem, it admits a unique solution $\bar{T}(x, t)$. Consider now the Cauchy problem restricted to the volume outside the heat sources belonging to \mathcal{V}_s :

$$\begin{cases} \rho c \frac{\partial T}{\partial t}(x, t) - \alpha \nabla^2 T(x, t) = 0, & x \in V \setminus \mathcal{V}_s, t \in [0, \tau), \\ T(x, 0) = T_0(x), & x \in V \setminus \mathcal{V}_s, \\ T(x, t) = T_{\partial V}(x, t), & x \in \partial V, t \in [0, \tau), \\ T(x, t) = \bar{T}(x, t), & x \in \partial \mathcal{V}_s, t \in [0, \tau). \end{cases} \quad (10)$$

Applying the maximum principle, the highest temperature is at $t = 0$ or on $\partial V \cup \partial \mathcal{V}_s$, i.e.:

$$\max_{(x,t) \in \overline{V \setminus \mathcal{V}_s} \times [0, \tau)} \bar{T}(x, t) = \max_{(x,t) \in \{\overline{V \setminus \mathcal{V}_s} \times \{0\}\} \cup \{\partial V \cup \partial \mathcal{V}_s\} \times [0, \tau)} \bar{T}(x, t),$$

hence:

$$\max_{(x,t) \in \bar{\Omega}_\tau} \bar{T}(x, t) = \max_{(x,t) \in \{V \times \{0\}\} \cup \{\partial V \times [0, \tau)\} \cup \{\mathcal{V}_s \times [0, \tau)\}} \bar{T}(x, t).$$

□

We present an immediate consequence of the above proposition that will be useful for control design.

Corollary 1. Consider problem (7), (8) with n_s heat sources as in Proposition 1, and suppose that the initial and boundary temperatures are lower than T_{CRIT} . Then, the solution $T(x, t)$ satisfies $T(x, t) \leq T_{\text{CRIT}}$, for all $x \in V$, if and only if the maximum temperature on sources is always lower than T_{CRIT} .

Fig. 3 illustrates the temperature distribution in two subsequent time instants. In Fig. 3-(a) the maximum temperatures are attained on the two sources, in which we applied a 20W power. Then, after removing powers (see Fig. 3-(b)), the maximum temperature moves to a middle point, not corresponding to any sources, but with a magnitude far lower than the initial one. The main consequence of the propositions above is that constraints violations will happen on sources first, provided that the boundary temperatures and the initial condition are feasible (i.e., below the temperature threshold). According to this result, and with the further assumption that each source can be treated as an isolated point (usually reasonable in MPSoCs), i.e.:

$$\mathcal{V}_s = \bigcup_{i=1}^{n_s} x_{s,i}, \quad x_{s,i} \in V, \forall i \in \{1, \dots, n_s\}, \quad (11)$$

it follows that we can convert the infinite-dimensional constraints into a finite-dimensional formulation given by:

$$T(x_{s,i}, t) \leq T_{\text{CRIT}}, \quad \forall i \in \{1, \dots, n_s\}. \quad (12)$$

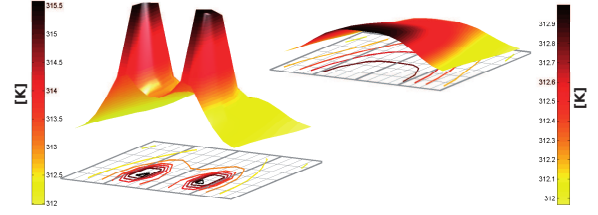


Figure 3: Two sources simulation: a) 20W per source; b) 0W per source.

3.2. Feasibility

A fundamental requirement for MPC schemes is recursive feasibility, i.e., if a control input sequence meeting the constraints exists at time $t = 0$, then it will also exist for all $t > 0$. In the MPC literature, such property is enforced by using adequate prediction horizons, terminal cost, invariant terminal sets, etc. (Bemporad and Morari, 1999; Rawlings and Mayne, 2009). However, these indirect methods come with some limitations, since they augment the controller complexity (Gondhalekara et al., 2009; Löfberg, 2011). In this subsection, our aim is to prove that for both the centralized problem (1) and the distributed problems (2), (4) we can find feasible solutions that do not rely on accurate model knowledge and are effective for any initial temperature distribution in the admissible set. In this perspective, we propose a reasonable requirement for the boundary conditions.

Assumption 1. For the problem (7), (8), it holds that:

- $T_0(x) \leq T_{\text{CRIT}}$, for all $x \in V$;
- $T_{\partial V}(x, t) \leq T_{\text{CRIT}}$, for all $(x, t) \in \partial V \times [0, \tau)$.

The first step in our development is captured by the following result, obtained in the absence of internal heat sources.

Proposition 2. For the problem (7), (8), let Assumption 1 hold, and suppose that

$$q(x, t) = 0, \quad \forall (x, t) \in V \times [0, \tau). \quad (13)$$

Then, it holds that $T(x, t) \leq T_{\text{CRIT}}$, for all $(x, t) \in V \times [0, \tau)$.

Proof. The result follows from Corollary 1 by letting $\mathcal{V}_s = \emptyset$. □

In the MPC context, the above statement implies that if the constraints are not violated at a given time, then the null input action (i.e., zeroing all the sources) always ensures constraint satisfaction. This way, feasibility of the centralized control problem (1) is guaranteed for any prediction horizon.

Remark 2. A physical interpretation of this property is given recalling that, according to the Second Principle of Thermodynamics, heat flow cannot be directed as the temperature gradient. Hence, no inductive storage elements can be present in thermal systems and, consequently, no free resonant or double integrative behavior can arise in the absence of heat sources.

For what concerns the feasibility of decentralized/distributed solutions, a crucial point is to define the “perimeter” of the local controllers and the information available to each of them. By Proposition 1, the requirement to prevent temperatures larger than T_{CRIT} all over the system can be achieved by preventing over-temperature on the sources (assuming suitable initial and boundary conditions). Then, following the formulation of (4), a local controller is employed for each source $x_{s,i}$ in (11): each controller can read its own source temperature and act on its local power, while it has to comply with the local constraint $T(x_{s,i}, t) \leq T_{\text{CRIT}}$.

At this stage, no information on the rest of the system is considered available, i.e., we consider a decentralized control scenario. In such scenario, the previous result for the centralized problem (1) suggests that shutting down a single core should guarantee no over-temperature for that source, assuming that the others have no over-temperature as well. In other words, whenever a core is approaching dangerous temperature values, the local decision of reducing its power should be enough to prevent local constraint violation, without requiring the reduction of power of all the other cores, provided that each of them is meeting its local thermal bound. This property, crucial for decentralized feasibility, holds true as stated in the following proposition.

Proposition 3. Consider problem (7), (8), with heat sources defined as in (11). Let Assumption 1 hold, and suppose that

$$q(x, t) = 0, \quad \forall (x, t) \in \{V \setminus \mathcal{V}_s\} \times [0, \tau). \quad (14)$$

Then, for each source $x_{s,i} \in \mathcal{V}_s$, the local decision of imposing $q(x_{s,i}, t) = 0$, $t \in [0, \tau)$, guarantees $T(x_{s,i}, t) \leq T_{\text{CRIT}}$ for all $t \in [0, \tau)$ if $T(x_{s,j}, t) \leq T_{\text{CRIT}}$ for all of the other sources $j \in \{1 \dots n_s\}$, $j \neq i$, and all $t \in [0, \tau)$.

Proof. As soon as $q(x_{s,i}, t)$ is zeroed in a source $x_{s,i}$, this becomes equivalent to a no-source point in V , then the result follows from Proposition 1. \square

In the analysis above, the possibility of imposing zero power on sources is a key element to achieve important properties in the path toward the feasibility of both centralized and distributed control problems (1), and (4). In real chips, it is hard and possibly harmful to have a sudden zeroing of the core power consumption (i.e., abruptly halting the core activities). The standard procedure is to slow down the clock frequency of the cores to a lower limit f_{min} , thus reducing the source powers to a value which can be upper bounded, considering the worst-case scenario for w_i . Here, we refer to this upper bound of the minimal source power as q_{min} .

This situation cannot be covered by the results of Propositions 2 and 3. However, taking inspiration from them, and introducing a suitable thermal constraint review, similar results with non-zero minimum power can be obtained as shown in the following. Define $T_{\text{EQ}}(x)$ as the solution of the following Cauchy problem

$$\begin{cases} -\alpha \nabla^2 T(x) = q_{\text{min}}(x), & x \in V, \\ T(x) = T_{\partial V, \text{max}}(x), & x \in \partial V, \end{cases} \quad (15)$$

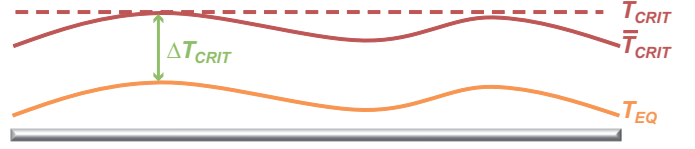


Figure 4: Graphical interpretation of the new bound \bar{T}_{CRIT}

where $q_{\text{min}}(x)$ is the minimum power which can be dissipated in any source, and $T_{\partial V, \text{max}}(x)$ is the largest ambient temperature that can be experienced by the chip on its boundary. According to its definition, $T_{\text{EQ}}(x)$ corresponds to the steady-state temperature distribution for the system (7) under minimum power consumption and highest ambient temperature. Obviously, $T_{\text{EQ}}(x) < T_{\text{CRIT}}$, for all $x \in \bar{V}$ is guaranteed by a proper sizing of the chip, otherwise the thermal constraints could not be met. In addition, by Proposition 1, it follows that the maximum value of $T_{\text{EQ}}(x)$ is located on sources or boundaries, i.e.:

$$\left\{ \arg \max_{x \in \bar{V}} T_{\text{EQ}}(x) \right\} \cap \{\mathcal{V}_s \cup \partial V\} \neq \emptyset.$$

Then, defining

$$\Delta T_{\text{CRIT}} := \min_{x \in \bar{V}} (T_{\text{CRIT}} - T_{\text{EQ}}(x)), \quad (16)$$

it results

$$\Delta T_{\text{CRIT}} = \min_{x \in \partial V \cup \mathcal{V}_s} (T_{\text{CRIT}} - T_{\text{EQ}}(x)) > 0. \quad (17)$$

At this point, a crucial modification of the temperature bound needs to be introduced. We define the constraint $\bar{T}_{\text{CRIT}}(x)$ as follows

$$\bar{T}_{\text{CRIT}}(x) := T_{\text{EQ}}(x) + \Delta T_{\text{CRIT}}, \quad x \in \bar{V}. \quad (18)$$

The meaning of this new bound is graphically represented in Fig. 4, for a simple one-dimensional thermal system. Note that the new bound $\bar{T}_{\text{CRIT}}(x)$ is tighter than T_{CRIT} , but this restriction is fundamental for feasibility in real scenarios. Firstly, we modify Assumption 1 to ensure desirable boundary conditions.

Assumption 2. For the problem (7), (8), it holds that:

- $T_0(x) \leq \bar{T}_{\text{CRIT}}(x)$, for all $x \in V$;
- $T_{\partial V}(x, t) \leq \bar{T}_{\text{CRIT}}(x)$, for all $(x, t) \in \partial V \times [0, \tau)$.

The following results are generalizations of Propositions 2 and 3.

Proposition 4. For the problem (7), (8), let Assumption 2 hold, and suppose that

$$q(x, t) = q_{\text{min}}(x), \quad \forall (x, t) \in V \times [0, \tau). \quad (19)$$

Then, it holds that $T(x, t) \leq \bar{T}_{\text{CRIT}}(x)$, for all $(x, t) \in V \times [0, \tau)$.

Proof. Without loss of generality, suppose that $T_{\partial V}(x, t) = T_{\partial V, \text{max}}(x)$ as in (15), for all $(x, t) \in \partial V \times [0, \tau)$. Applying

the superposition principle, we can rewrite the heat equation $\rho c \dot{T}(x, t) - \alpha \nabla^2 T(x, t) = q_{\min}(x)$ as follows

$$\rho c \frac{\partial \Delta T}{\partial t}(x, t) - \alpha \nabla^2 \Delta T(x, t) = 0, \quad (20)$$

where $\Delta T(x, t) := T(x, t) - T_{\text{EQ}}(x)$. The following boundary conditions hold: $\Delta T_0(x) \leq \Delta T_{\text{CRIT}}$, for all $x \in V$ and $\Delta T_{\partial V}(x, t) \leq \Delta T_{\text{CRIT}}$, for all $(x, t) \in \partial V \times [0, \tau)$. Note that system (20) has the same structure of (7). Applying the Maximum Principle, we obtain $\Delta T(x, t) \leq \Delta T_{\text{CRIT}}$, and $T(x, t) \leq T_{\text{EQ}}(x) + \Delta T_{\text{CRIT}} = \bar{T}_{\text{CRIT}}(x)$ for all $(x, t) \in V \times [0, \tau)$. \square

Proposition 5. Consider problem (7), (8), with heat sources defined as in (11). Let Assumption 2 hold, and suppose that

$$q(x, t) = q_{\min}(x), \quad \forall (x, t) \in \{V \setminus \mathcal{V}_s\} \times [0, \tau). \quad (21)$$

Then, for each source $x_{s,i}$, the local decision of imposing $q(x_{s,i}, t) = q_{\min}(x_{s,i})$, $t \in [0, \tau)$, guarantees $T(x_{s,i}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,i})$, for all $t \in [0, \tau)$, as long as $T(x_{s,j}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,j})$ for all other sources $j \in \{1 \dots n_s\}$, $j \neq i$, and all $t \in [0, \tau)$.

Proof. As in the proof of Proposition 4, suppose that $T_{\partial V}(x, t) = T_{\partial V, \max}(x)$ as in (15), for all $(x, t) \in \partial V \times [0, \tau)$. By defining $\Delta T(x, t) := T(x, t) - T_{\text{EQ}}(x)$ and $\Delta q(x, t) := q(x, t) - q_{\min}(x)$ (where $\Delta q(\cdot)$, $q(\cdot)$ and $q_{\min}(\cdot)$ can be non-zero in $x_{s,i}$, $i \in \{1, \dots, n_s\}$), the temperature dynamics can be rewritten as follows, again exploiting the superposition principle

$$\rho c \frac{\partial \Delta T}{\partial t}(x, t) - \alpha \nabla^2 \Delta T(x, t) = \Delta q(x, t). \quad (22)$$

This equation has the same structure of (7), thus it inherits all of its features. In addition, according to (18), the bound $\bar{T}_{\text{CRIT}}(x)$ is mapped into a constant threshold ΔT_{CRIT} . Therefore, as soon as $q(x_{s,i}, t) = q_{\min}(x_{s,i})$ is imposed in a source point $x_{s,i}$, $\Delta q(x_{s,i}, t)$ is zeroed, and this point becomes equivalent to a no-source point in V for (22). Then, according to Proposition 1, the maximum of $\Delta T(x, t)$ occurs on the remaining sources $x_{s,j}$, $j \in \{1 \dots n_s\}$, $j \neq i$ or at the initial condition, or on the boundaries. In these regions, $\Delta T(x, t)$ is always lower than ΔT_{CRIT} by assumption, therefore the result follows. \square

With the above results at hand, the feasibility properties of the centralized and distributed problems (1) and (4) are formally stated as follows.

Proposition 6. Centralized Feasibility.

Consider problem (7), (8), with heat sources defined as in (11). Let Assumption 2 hold with $\tau = \infty$ for simplicity, and suppose that

$$q(x, t) = q_{\min}(x), \quad \forall (x, t) \in V \times [0, \infty). \quad (23)$$

Consider the control input $u(t) := [q(x_{s,1}, t), \dots, q(x_{s,n_s}, t)] \in \mathbb{R}^{n_s}$ and suppose that there exist a scalar $\Delta > 0$ and a controller $\bar{u}(\cdot)$ such that the constraints (12) are satisfied for $t \in [0, \Delta]$ if $u(t) = \bar{u}(t)$, for all $t \in [0, \Delta]$. Then, for any $t' \in [0, \Delta]$, it is possible to compute a controller $\bar{u}'(\cdot)$ such that the constraints (12) are satisfied over $[0, t' + \Delta]$ if $u(t) = \bar{u}(t)$ for $t \in [0, t')$ and $u(t) = \bar{u}'(t)$ for all $t \in [t', t' + \Delta]$.

Proof. For $t \in [t', t' + \Delta]$, define $\bar{u}'(\cdot)$ as follows:

$$\bar{u}'(t) = \begin{cases} \bar{u}(t), & t \in [t', \Delta) \\ [q_{\min}(x_{s,1}), \dots, q_{\min}(x_{s,n_s})]^\top, & t \in [\Delta, t' + \Delta]. \end{cases} \quad (24)$$

The result follows from Proposition 4. \square

Proposition 7. Decentralized Feasibility.

Consider problem (7), (8), with heat sources defined as in (11). Let Assumption 2 hold with $\tau = \infty$ for simplicity, and suppose that

$$q(x, t) = q_{\min}(x), \quad \forall (x, t) \in \{V \setminus \mathcal{V}_s\} \times [0, \infty). \quad (25)$$

For all $i \in \{1, \dots, n_s\}$, define the control input $u_i(t) := q(x_{s,i}, t)$ and consider the constraints (12). Suppose that for a generic source $i \in \{1, \dots, n_s\}$ there exist $\Delta_i > 0$ and a controller $\bar{u}_i(\cdot)$ such that the local decision of imposing $u_i(t) = \bar{u}_i(t)$, for all $t \in [0, \Delta_i]$, ensures $T(x_{s,i}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,i})$ over $[0, \Delta_i]$ as long as $T(x_{s,j}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,j})$, for all other sources $j \in \{1 \dots n_s\}$, $j \neq i$, and all $t \in [0, \Delta_i]$. Then, for any $t' \in [0, \Delta_i]$ it is possible to compute a controller $\bar{u}'_i(\cdot)$ such that:

1. if $u_i(t) = \bar{u}_i(t)$ for $t \in [0, t')$ and $u_i(t) = \bar{u}'_i(t)$ for $t \in [t', t' + \Delta_i]$, then the local constraint $T(x_{s,i}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,i})$ is satisfied over $[0, t' + \Delta_i]$ if $T(x_{s,j}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,j})$, for all other sources $j \in \{1 \dots n_s\}$, $j \neq i$, and all $t \in [0, t' + \Delta_i]$;
2. the local control strategy applied to the source i does not prevent the other sources from meeting their local constraint $T(x_{s,j}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,j})$ exploiting the same strategy;
3. $\bar{u}'_i(\cdot)$ can be selected in a decentralized way, i.e., without any information on the variables and the controllers of the other nodes.

Proof. For $t \in [t', t' + \Delta_i]$, define $\bar{u}'_i(\cdot)$ as follows:

$$\bar{u}'_i(t) = \begin{cases} \bar{u}_i(t), & t \in [t', \Delta_i) \\ q_{\min}(x_{s,i}), & t \in [\Delta_i, t' + \Delta_i], \end{cases} \quad (26)$$

which is decentralized and ensures feasibility in view of Proposition 5. Concerning point 2, by Proposition 4 it follows that all of the local controllers can apply $u_i(t) = q_{\min}(x_{s,i})$, for all $t \in [t', t' + \Delta_i]$, guaranteeing feasibility. Moreover, considering a generic $j \in \{1 \dots n_s\}$ and defining two disjointed subsets of sources, S_0 and S_s , not including j , but covering all of the sources (i.e., $S_0 = \{n : n \in \{1 \dots n_s\}, n \neq j\}$, $S_s = \{m : m \in \{1 \dots n_s\}, m \neq j\}$, $S_0 \cap S_s = \emptyset$, $S_0 \cup S_s = \{1 \dots j-1, j+1 \dots n_s\}$), if all the sources in S_0 apply $u_n(t) = q_{\min}(x_{s,n})$, for all $t \in [t', t' + \Delta_i]$, and those in S_s meet their own local constraints $T(x_{s,m}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,m})$, for all $t \in [t', t' + \Delta_i]$, then, in the source j it is possible to apply $u_j(t) = q_{\min}(x_{s,j})$, for all $t \in [t', t' + \Delta_i]$, guaranteeing $T(x_{s,j}, t) \leq \bar{T}_{\text{CRIT}}(x_{s,j})$, for all $t \in [t', t' + \Delta_i]$. Thus, also point 2 is proven. \square

It is further to remark that the above results hold for any positive prediction horizon, which can be selected arbitrarily small without impairing feasibility. Hence, there is room to preserve feasibility until the current temperature reaches the bound, both in the centralized and distributed approach.

Another relevant point to highlight is the use of $\bar{T}_{\text{CRIT}}(x) \leq T_{\text{CRIT}}$ instead of T_{CRIT} . This is not just a technical point related to the available theoretical tools, but it accounts for possible practical troubles in the thermal behavior. As a matter of fact, when a core (or any heat source) needs an indirect path through its neighbors to expel its heat and keep a safe temperature (i.e., it has no direct path toward the external ambient that is efficient enough), the neighbors are required to keep temperature lower than the maximum, to give room to the heat flow coming from the above-mentioned core. Clearly, this characteristic strongly depends on the considered system's layout and thermal resistivity and should be carefully addressed in the multicore design.

3.3. Model-free temperature capping control

The above results guarantee feasibility by applying minimum power consumption at each core. No accurate thermal model knowledge is required to define the considered control strategy. The centralized feasibility of Proposition 6 is essentially based on the observation that, by switching all the cores to the minimum power, the maximum temperature along the chip will never increase. The distributed feasibility of Proposition 7 allows to recover this property locally to each source, without needing any information exchange but just assuming, in a local controller, that all of the others are working correctly. This can be achieved at the cost of replacing the uniform bound T_{CRIT} with the variable bound $\bar{T}_{\text{CRIT}}(x)$.

At first glance, these results could seem of little practical interest, since they just provide the trivial solution of switching to minimum power the cores to meet the temperature constraints, but they do not define any algorithm to find other feasible solutions which maximize the computing power, as reported in problems (1) and (4). To this aim, in the following, algorithms based on approximated discretized models will be presented, also exploiting information exchange between controllers for the distributed case. Nevertheless, the above feasibility results play a fundamental role for practical applications. Indeed, by using algorithms based on approximated discrete models, temperature constraint violation in the real system could be experienced. This inconvenience can be prevented, according to the above feasibility results, by adding an *ultimate model-free decentralized temperature capping control layer* on each local i -th source. As soon as $T(x_{s,i}, t)$ approaches $\bar{T}_{\text{CRIT}}(x_{s,i})$, this control layer has to override the optimizing controller and impose $q(x_{s,i}, t) = q_{\min}(x_{s,i})$ for a suitable time interval to obtain $T(x_{s,i}, t)$ sufficiently lower than $\bar{T}_{\text{CRIT}}(x_{s,i})$.

4. Finite-dimensional discrete-time modeling oriented to distributed MPC

In the remainder of this work, we focus on the distributed control problem (4). For each core, we implement a local, discrete-time MPC controller that supervises its temperature. The objectives are to ensure feasibility through the results of Section 3 and to maximize performance. To pursue the latter goal, in contrast to the ultimate model-free temperature capping

control strategy described in the previous section, each local controller needs a thermal model of the core to predict future temperatures and a framework to exchange information with the neighbor cores. In this respect, designing the local thermal model is a crucial task: it must be simple to limit the computational burden yet accurate enough to minimize temperature prediction errors and meet the feasibility properties of the original system. Indeed, whereas the linear, continuous-time PDE model presented in the previous section always ensures feasibility, no similar features exist, in general, for its discrete-time discrete-space approximations. The following assumptions are made to design the local models (Bartolini et al., 2012):

- as mentioned in Section 2, a function of the form $P_i = g(f_i, w_i)$ as in (3) maps the frequency f_i and the workload w_i of each core i into the corresponding power consumption. It follows that the thermal model used for prediction in the i -th local controller is linear;
- according to the literature on MPSoCs, a second-order model is sufficiently rich to capture the main dynamics of each core;
- each local controller is only affected by the North, South, East, and West adjacent cores (if present), according to heat diffusion and sampling time considerations. By (7), interaction among cores is directly proportional to time intervals and inversely related to spatial distance. Thus, if the discretization time is short enough, it can be assumed that the thermal state of a core between two sampling instants will be mainly affected by its closest neighbor, its thermal power, and the ambient temperature.

To obtain an accurate finite-dimensional discrete-time model from the original heat equation (7), we adopt system identification tools. According to the assumptions above, the structure selected for the identification of the core local model is a second-order Multi-Input Single-Output (MISO) system

$$\hat{T}_i(t) = a_{1,i}T_i(t-1) + a_{2,i}T_i(t-2) + b_{1,i}^\top u_i(t-1) + b_{2,i}^\top u_i(t-2) \quad (27)$$

where \hat{T}_i is the temperature prediction, T_i is the measured temperature of the i -th core in its hottest point, $u_i := [P_{C,i} \ T_{\text{amb}} \ T_{n,i}]^\top$ is the vector of inputs, where $P_{C,i}$ is the power consumed by the core, T_{amb} is the ambient temperature, while $T_{n,i} := [T_{n1,i} \ \dots \ T_{nq,i}]^\top$ are the temperatures of the q cores adjacent to i . Scalars $a_{1,i}$, $a_{2,i}$ and vectors $b_{1,i}$, $b_{2,i}$ are to be determined with the identification process, based on the available output T_i and inputs u_i , with the aim of minimizing a predefined objective function, possibly under constraints. Define the prediction error $e_i(t) := T_i(t) - \hat{T}_i(t)$. Then, exploiting (27), we adopt an \mathcal{H}_∞ -like identification problem formulated as

$$\begin{aligned} & \min_{s_1, \dots, s_M, a_{1,i}, a_{2,i}, b_{1,i}, b_{2,i}} \sum_{k=1}^M s_k \\ & \text{subj. to: } \left. \begin{aligned} e_i(t-k) &\geq -s_k \\ s_k &\geq 0 \end{aligned} \right\} k \in \{1, \dots, M\}, \end{aligned} \quad (28)$$

where M denotes the time window selected for the identification. The model parameters computed from (28) ensure $\hat{T}_i \geq T_i$ at all times. This result is instrumental in achieving robust and reliable thermal control since overestimated temperatures reflect more aggressive control actions, preventing the system from hitting critical temperatures.

4.1. Feasibility check for the discretized model

The local model obtained in the previous section is simple and can provide good accuracy (as shown in Section 6). Unfortunately, the feasibility properties of the physical system (modeled by the heat equation) are not necessarily preserved for the identified model. Therefore, it is crucial to verify the control feasibility of each discrete local model. To do so, we resort to the concept of recursive feasibility and the approach to verify it, reported in Löfberg (2011). According to the definition, recursive feasibility is guaranteed if, for every initially feasible condition (in our case, the core state, ambient, and neighbors thermal status), applying the optimal control input does not steer the system to an operational point where the control problem becomes unfeasible. It is well known that such property is affected by the prediction horizon N . According to the properties of thermal systems described in Section 3, the subsequent analysis is carried out for $N = 1$, as we would like the identified model to match the feasibility properties of the underlying physical system.

Firstly, we express the discrete-time model of each core i in the following state-space observable canonical form

$$\begin{aligned} \begin{bmatrix} x_{1,i}(t+1) \\ x_{2,i}(t+1) \end{bmatrix} &= \begin{bmatrix} a_{1,i} & 1 \\ a_{2,i} & 0 \end{bmatrix} \begin{bmatrix} x_{1,i}(t) \\ x_{2,i}(t) \end{bmatrix} + \begin{bmatrix} b_{1,i}^\top \\ b_{2,i}^\top \end{bmatrix} u_i(t) \\ &= A_i \begin{bmatrix} x_{1,i}(t) \\ x_{2,i}(t) \end{bmatrix} + B_{1,i} P_{C,i}(t) + B_{2,i} d_i(t), \quad (29) \\ T_i(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_c \begin{bmatrix} x_{1,i}(t) \\ x_{2,i}(t) \end{bmatrix}, \end{aligned}$$

where $x_{1,i}$ is the core temperature T_i (available from measurements), while $x_{2,i}$ a second state completing the second-order system description, but with no physical interpretation. Also, we split the input vector u_i into the control input $P_{C,i}$ and the other contributions, collected in $d_i := [T_{\text{amb}} \ T_{n,i}^\top]^\top$.

For simplicity, in the following we drop the subindex i denoting the specific core. Using (29), we can study the feasibility of optimization problem (4), with $N = 1$, analyzing the following quadratic problem

$$\begin{aligned} \min_{P_C} P_C^2 + H(t)P_C \\ \text{subj. to: } Ex(t) + FP_C \leq G(d(t)), \quad (30) \end{aligned}$$

where $H(t) := -2P_T(t)$, and

$$E := \begin{bmatrix} CA \\ 0_{1 \times 2} \end{bmatrix}, \quad F := \begin{bmatrix} CB_1 \\ 1 \\ -1 \end{bmatrix}, \quad G(d) := \begin{bmatrix} \bar{T}_{\text{CRIT}} - CB_2 d \\ P_{\text{max}} \\ -P_{\text{min}} \end{bmatrix}, \quad (31)$$

where $d(t)$ is available for measurement, assuming that at the discrete-time instant t each core i can communicate $T_i(t)$ with its neighbors.

In the following, we check recursive feasibility for problem (30) by ensuring that, for a feasible initial state $x(t)$ and a disturbance $d(t)$ ranging in a compact set \mathbb{D} , it is possible to find $P_C(t+1)$ such that:

$$E(Ax(t) + B_1 P_C^*(t) + B_2 d(t)) + FP_C(t+1) \leq G(d(t)), \quad (32)$$

where, for simplicity, we assumed that the disturbance $d(\cdot)$ is constant over $\{t, t+1\}$, thus the constraints (30) can be applied in $t+1$ as $Ex(t+1) + FP_C(t+1) \leq G(d(t))$. Exploiting (32), we can thus verify if there exist $x(t)$, $d(t)$ such that (30) is recursively feasible. Equivalently, it is possible to invoke Farkas' Lemma as in (Löfberg, 2011) to ensure that there is no vector $y(t)$ such that the following conditions are verified:

$$\begin{aligned} y(t) \geq 0 \quad y(t)^\top F = 0 \\ y(t)^\top [G(d(t)) - E(Ax(t) + B_1 P_C^*(t) + B_2 d(t))] < 0. \quad (33) \end{aligned}$$

Therefore, starting from a feasible state $x(t)$, with admissible disturbances $d(t) \in \mathbb{D}$, and applying $P_C^*(t)$, if there exists $y(t)$ satisfying (33), the control problem is unfeasible at the time instant $t+1$. According to (Löfberg, 2011), we define the following two-level optimization problem (with an outer problem containing an inner one in the constraints) to check the consistency of Farkas' Lemma conditions

$$\min_{y,x,d} y^\top [G(d) - E(Ax + B_1 P_C^* + B_2 d)] \quad (34a)$$

$$\text{subj. to: } y \geq 0, \quad y^\top F = 0, \quad (34b)$$

$$x \in \mathbb{X}, \quad d \in \mathbb{D}, \quad (34c)$$

$$P_C^* = \arg \min_{P_C} P_C^2 + H(t)P_C \quad (34d)$$

$$\text{subj. to: } Ex + FP_C \leq G(d), \quad (34e)$$

where \mathbb{X} is the set of allowed values for $x(t)$, and the cost function of the outer problem is the third condition of Farkas' Lemma applied to our problem. Notice that, in this context, it is simple to determine \mathbb{D} , since the ambient temperature can be assumed contained in an interval of the form $[T_{\text{amb,min}}, T_{\text{amb,max}}]$, while the temperatures of the neighbor cores are constrained to be less than the corresponding critical thresholds. This way, we can define \mathbb{D} as follows:

$$\begin{aligned} \mathbb{D} := \{d = [T_{\text{amb}} \ T_{n1} \ \dots \ T_{nq}]^\top : T_{\text{amb,min}} \leq T_{\text{amb}} \leq T_{\text{amb,max}}, \\ T_{\text{amb,min}} \leq T_{nj} \leq \bar{T}_{\text{CRIT},n,j}, j \in \{1, \dots, q\}\}. \quad (35) \end{aligned}$$

Concerning \mathbb{X} , x_1 corresponds to the core temperature, thus its bounds are given by an interval of the form $[T_{\text{amb,min}}, \bar{T}_{\text{CRIT}}]$, while the bounds for x_2 are such that x_1 at the next time sample is lower than \bar{T}_{CRIT} :

$$\begin{aligned} \mathbb{X} := \{x := [x_1 \ x_2]^\top \in \mathbb{R}^2 : T_{\text{amb,min}} \leq x_1 \leq \bar{T}_{\text{CRIT}}, \\ a_1 x_1 + x_2 + b_1^\top \begin{bmatrix} P_C \\ d \end{bmatrix} \leq \bar{T}_{\text{CRIT}}, \forall P_C \in [P_{\text{min}}, P_{\text{max}}], \forall d \in \mathbb{D}\}. \quad (36) \end{aligned}$$

Replacing the inner convex problem, defined by (34d), (34e), with the corresponding Karush-Kuhn-Tucker optimality conditions, we obtain the following equivalent problem

$$\min_{y,x,d,\lambda,P_C} y^\top [G(d) - E(Ax + B_1P_C + B_2d)] \quad (37a)$$

$$\text{subj. to: } y \geq 0, \quad y^\top F = 0, \quad y^\top \mathbf{1}_3 = 1 \quad (37b)$$

$$x \in \mathbb{X}, \quad d \in \mathbb{D}, \quad (37c)$$

$$2P_C + H(t) + \lambda^\top F = 0, \quad \lambda \geq 0, \quad (37d)$$

$$G(d) - Ex - FP_C \geq 0, \quad (37e)$$

$$\lambda^\top [G(d) - Ex - FP_C] = 0, \quad (37f)$$

where λ is the vector of Lagrange multipliers associated with inequality constraints of the inner problem, and a normalization constraint on y has been added for numerical reasons.

If Farkas' conditions hold, then recursive feasibility is not guaranteed. Indeed, finding a negative (even local) solution to the problem above, implies that there exists a feasible initial condition which, after the optimal input is provided, moves to a point where feasibility can no longer be achieved for the future steps. This case would stand in clear contrast with the properties of thermal systems, which have been analyzed in Section 3. Therefore, if this situation occurs with a set of identified model parameters, the corresponding model has to be discarded, and a new identification process must be executed, taking different exciting traces, inputs, model order, or sampling time, until the Farkas' condition is violated, i.e., feasibility is ensured. Note that, intuitively, recursive feasibility is expected to be achieved by increasing the model order and reducing the sampling time, since the resulting discrete-time models become a better approximation of the original PDE.

With such iterative procedure, the model used in each local MPC controller "ideally" ensures a feasible control solution that prevents \bar{T}_{CRIT} constraint violations. However, many issues make such an ideal thermal capping method not straightforwardly applicable in real-world applications: (i) the obtained discrete models are approximations of the actual continuous-time system, (ii) the workload of a core is unpredictable and, even though the variations between two-time steps are usually limited, there may be cases for which significant variations cause wrong predictions, (iii) the identification procedure uses a set of data collected from the physical system that may not capture all possible behaviors, causing unexpected outputs (even if we considered a conservative approach we could not guarantee the absence of constraint violations), (iv) the behavior of the system temperature in between the sampling time is unobservable. Therefore, although we build feasible and conservative (in terms of constraints) models, the feasibility for the actual system is not guaranteed. In the next section, we present a control solution that guarantees feasibility and robustness to model uncertainties in any circumstance.

5. Two-layer distributed control

The challenges related to model uncertainties and workload unpredictability can be addressed either with a model-free temperature capping control strategy, as proposed in Section 3.3,

or with a fully conservative MPC solution. Both strategies have pros and cons.

The model-free strategy, not being driven by the model, results completely immune to uncertainties and unpredictabilities, guaranteeing feasibility, provided that a continuous-time implementation of the control strategy is assumed. However, this solution has two main limitations: (i) it strongly impacts performance due to the intense control chattering between the desired power and the minimum power, which may also lead to premature failure of the processor; (ii) it needs a safety margin on \bar{T}_{CRIT} to avoid exceeding the critical temperature during the sampling interval if a discrete-time implementation is considered. Note that a discrete-time solution fits well with MPSoC digital technology and that the safety margin must be increased with the sampling time value. However, assuming a hardware-based thermal capping solution, the sampling time of such part can be taken very small, and, in turn, the temperature margin can be thin.

On the other hand, we can adopt a local MPC controller for each core that, exploiting the identified model of the core and predicting its temperature behavior, guarantees more accurate and chattering-free performance compared to the previous solution. However, feasibility cannot be ensured due to unavoidable uncertainties on the model and future workload. Besides, as in the model-free solution, time discretization imposes a safety margin on \bar{T}_{CRIT} to prevent constraint violations between two sampling instants. It is worth noting that this solution appears hard to be implemented on hardware. Therefore, it needs a larger sampling time and a significantly larger margin that strongly affects performance. Indeed, as the temperature threshold decreases, also power consumption and core speed reduce as well. As an illustrative example, consider a scenario in which all cores have a temperature close, but not equal, to the limit decreased by the safety margin, namely τ_{MPC} . Each local controller would exploit the discrete model of the core to define the optimal control action that keeps the future temperature close to τ_{MPC} , according to the information of the current temperature of the core and its neighbors. However, during the sampling interval, the temperature of all the cores varies, possibly determining an overestimation of the control actions and, as a consequence, triggering thermal emergencies.

As shown in Fig. 5, our solution adopts a hierarchical architecture in both the methods above. Indeed, such strategies cooperate to exploit the benefits of both. The higher layer, namely the MPC layer, addresses the thermal capping issue, maximizing performance by exploiting the previously mentioned distributed MPC scheme. The lower one, namely the safety layer, guarantees control feasibility, the imposition of the bound \bar{T}_{CRIT} , and the reduction of the MPC margin, favoring better performance.

5.1. MPC layer

The MPC layer is a discrete-time software layer designed with a distributed structure. Such configuration has the twofold benefit of improving the computational efficiency of the control algorithm (as the computational burden is distributed among cores) and the system reliability (as the breakdown of one core

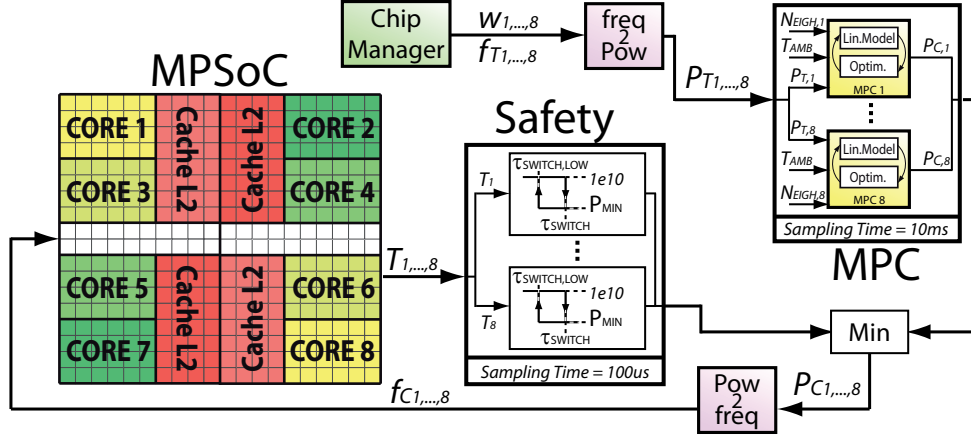


Figure 5: Control architecture.

does not compromise the whole system performance). Each controller uses the local model identified in Section 4 to predict the future temperature of the core and solves the optimization problem shown in (4) to obtain the best control decisions that keep the i -th core temperature, T_i , close to the bound. We called this bound $\tau_{MPC,i}$ to highlight that each i -th controller has a different bound that may differ from $\bar{T}_{CRIT,i}$. It is reasonable to expect $\tau_{MPC,i} < \bar{T}_{CRIT,i}$ due to safety margins introduced to prevent overtemperatures between two sampling instants. A discussion on how to determine the value of $\tau_{MPC,i}$ is reported in Section 5.3.

An implementation issue for the MPC layer concerns the estimation of the state x_2 , which, as already mentioned, is not available from measurements (whereas x_1 corresponds to the temperature sensed on the core). Due to the complete observability of the model, we can estimate the unknown state by storing past control inputs and temperatures, following a dead-beat observer approach. From (29), we have

$$x_2(t) = a_2 T(t-1) + b_2^T \begin{bmatrix} P_C(t-1) \\ d(t-1) \end{bmatrix}.$$

With the core temperature sensor reading, the estimation above, the identified model parameters, and information about its neighbors, each core can solve the following problem

$$\begin{aligned} & \min_{P_{C,i}(t)} |P_{T,i}(t) - P_{C,i}(t)|^2 \\ \text{subj. to: } & \begin{bmatrix} x_{1,i}(t+1) \\ x_{2,i}(t+1) \end{bmatrix} = A_i \begin{bmatrix} x_{1,i}(t) \\ x_{2,i}(t) \end{bmatrix} + B_{1,i} P_{C,i}(t) + B_{2,i} d_i(t) \\ & T_i(t+1) = C \begin{bmatrix} x_{1,i}(t+1) \\ x_{2,i}(t+1) \end{bmatrix} \leq \tau_{MPC,i}, \\ & P_{\min} \leq P_{C,i}(t) \leq P_{\max}. \end{aligned} \quad (38)$$

5.2. Safety layer

The safety layer includes a set of hardware-based hysteresis controllers, one for each core, completely independent from the MPC layer. At this level, a model-free temperature capping

control strategy is adopted, as proposed in Section 3.3. During nominal operation, the MPC layer keeps the temperature of each i -th core below $\tau_{MPC,i}$. However, if such constraint is violated and the i -th core temperature reaches $\bar{T}_{CRIT,i}$, the corresponding hysteresis controller bypasses the MPC layer, immediately providing to the core the minimum power, P_{\min} , until the temperature decreases to a given lower value, $\tau_{SWITCH,LOW,i}$, according to the hysteresis mechanism.

In Section 3 we proved that a set of *local* continuous-time regulators ensures feasibility, keeping the temperature below $\bar{T}_{CRIT,i}$. However, the clock-driven nature of MPSoCs imposes the use of a discrete-time controller. As a result, to avoid temperature violations during the sampling intervals, it is necessary to provide a margin, i.e., to decrease the critical temperature $\bar{T}_{CRIT,i}$ to a value $\tau_{SWITCH,i}$, depending on the selected sampling period. Nevertheless, since the safety layer is implemented in hardware, the sampling time can be extremely small (few microseconds). An effective choice of $\tau_{SWITCH,i}$ can be obtained inverting the discrete models (29) of each core, and subtracting from \bar{T}_{CRIT} the maximum temperature increase occurred in a single sample interval in the worst-case scenario (i.e., critical neighbor temperature, maximum ambient temperature, and maximum input power). Next to the upper threshold, the lower threshold, $\tau_{SWITCH,LOW,i}$, must be defined to deactivate the layer and resume the nominal behavior. Such value is lower than $\tau_{MPC,i}$ to make the system restart from a feasible state. In this regard, we set $\tau_{SWITCH,LOW,i} = \tau_{MPC,i} - \Delta$ where Δ is a small arbitrary value. This way, we obtain a hardware-based, discrete-time, hysteresis-based controller capable of guaranteeing feasibility.

5.3. $\tau_{MPC,i}$ threshold

In the proposed hierarchical solution, the combined use of a safety and an MPC layer improves the global system performance and guarantees feasibility. This result, a key point for manufacturers whose profit is strictly related to performance, can be obtained by suitably selecting $\tau_{MPC,i}$. In this regard, a trade-off needs to be pursued, as explained in what follows. In principle, setting $\tau_{MPC,i} = \tau_{SWITCH,i}$ would maximize performance. Indeed, the higher is the temperature limit, the

more power can be allocated by the controller to increase the core speed. However, uncertainties would cause frequent interventions of the safety controller that would set the power to the minimum value and contribute to performance degradation. Thus, a margin between $\tau_{\text{MPC},i}$ and $\tau_{\text{SWITCH},i}$ must be provided. The greater the margin, the more conservative the controller becomes, again impairing performance (since the local MPC would maintain the core speed to a lower level w.r.t. what strictly necessary). In our analysis, we consider as maximum margin the one preventing the use of the safety layer. In this case, the local MPC controller is completely conservative, and we denote the corresponding maximum temperature threshold as $\tau_{\text{MPC,CONS},i}$.

The central idea to improve performance is to select $\tau_{\text{MPC,CONS},i} < \tau_{\text{MPC},i} < \tau_{\text{SWITCH},i}$, i.e., to design a less conservative MPC controller, leaving the safety layer to manage critical situations. Indeed, it is reasonable to expect a smaller performance degradation from limited use of the safety layer rather than having a completely conservative/robust MPC layer. To this aim, a trade-off problem between the conservativeness of the MPC controller and the frequency of the safety controller's activation must be solved. Due to the significant number of factors affecting the controller, as external inputs and the already cited model uncertainties, rigorous analytical estimation of $\tau_{\text{MPC},i}$ is difficult. Therefore, we propose two empirically-based methods to impose this margin.

A first simple method consists of running typical benchmarks, e.g., PARSEC (Bienia et al., 2008), and calibrating $\tau_{\text{MPC},i}$ as the value that reduces the violations of $\tau_{\text{SWITCH},i}$ under an arbitrarily predetermined percent of time. This solution lets the user the freedom of choosing the degree of exploitation of the safety layer. However, if the goal is maximizing the computing performance, an optimization problem should be formulated and solved. In this respect, the second proposed approach searches for the $\tau_{\text{MPC},i}$ that maximizes an objective function given by the integral of the core frequencies, under the constraint $\tau_{\text{MPC},i} \in [\tau_{\text{MPC,CONS},i}, \bar{T}_{\text{CRIT},i}]$, that is:

$$\max_{\tau_{\text{MPC},i}} \sum_{i=1}^{n_C} \int_0^{t_B} f_{C,i}(\tau_{\text{MPC},i}) dt \quad (39a)$$

$$\text{subj. to: } \tau_{\text{MPC,CONS},i} \leq \tau_{\text{MPC},i} \leq \bar{T}_{\text{CRIT},i} \quad i \in \{1, \dots, n_C\}, \quad (39b)$$

where n_C is the number of cores, f_C are the corresponding frequencies, and $[0, t_B]$ is the benchmarks' time interval. We solved the problem for each benchmark, and we selected the optimal values $\tau_{\text{MPC},i}$ as the average of each problem's result.

Remark 3. *The MPC controller imposes constraints on temperature predictions obtained with the identified model, not on the real temperature of the core. As a result, the actual core temperature can be lower than $\tau_{\text{SWITCH},i}$ even if $\tau_{\text{MPC},i} > \tau_{\text{SWITCH},i}$. This fact motivates the use of $\bar{T}_{\text{CRIT},i}$ as an upper bound in (39b). In case $\tau_{\text{MPC},i} > \tau_{\text{SWITCH},i}$, we set the lower layer of the local safety controller as $\tau_{\text{SWITCH,LOW},i} = \tau_{\text{SWITCH},i} - \Delta$ to avoid a deactivation threshold greater than the activation one.*

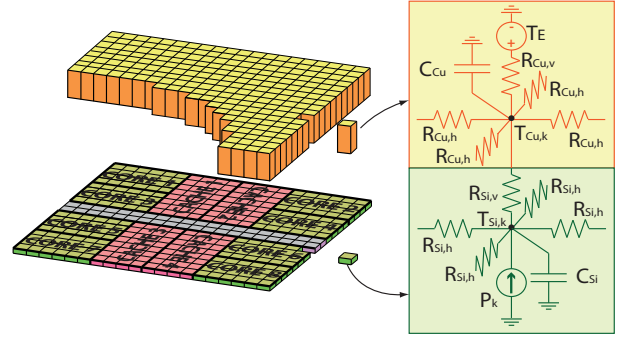


Figure 6: Chip thermal simulator.

Remark 4. *The rigorous solution of (39) would require an infinite number of simulations for computing the integral in the objective function (one for each $\tau_{\text{MPC},i}$ value, and for each benchmark). To reasonably address this issue, we search the $\tau_{\text{MPC,CONS},i}$ preventing the intervention of the ultimate capping controller, then we solve the optimization problem for a discrete set of value in the interval $[\tau_{\text{MPC,CONS},i}, \bar{T}_{\text{CRIT},i}]$.*

6. Simulation results

In this section, we report significant simulation results. Firstly, we provide some information on the accurate model used to emulate the real system thermal behavior. Then, we summarize the design steps to build the two-layer controller. Finally, we show its effectiveness for a typical benchmark.

Fig. 6 shows the plant used for simulations. We adopted as a reference the chip layout of the Enterprise Xeon[®] Processor presented in (Rusu et al., 2010), doubling it to obtain an 8-core chip. According to (Paci et al., 2006), we performed thermal simulations by using a finite element approach. We divided the chip into two layers, representing the silicon and the heat spreader, and each of them into cells (with 360 cells per layer). To each cell, we associated a small thermal system composed by a thermal resistor for the vertical thermal dissipation ($R_{\text{Si},v} = 1.6\text{K/W}$, $R_{\text{Cu},v} = 290\text{K/W}$)¹, four resistors for the horizontal thermal dissipation ($R_{\text{Si},h} = 22.9\text{K/W}$, $R_{\text{Cu},h} = 1.2\text{K/W}$), a thermal capacitance ($C_{\text{Si}} = 1 \times 10^{-3}\text{J/K}$, $C_{\text{Cu}} = 1.2 \times 10^{-2}\text{J/K}$) and, exploiting the analogy with the electrical framework, a current generator and a voltage generator, depending on the layer. The former represents the power dissipated by the active silicon, while the latter is associated with the heat spreader's ambient temperature. The network associated with the chip is shown in Fig. 6. The resulting model is

$$\begin{aligned} \dot{T}_{\text{Si},k} &= \frac{P_k}{C_{\text{Si}}} + \frac{T_{\text{Cu},k} - T_{\text{Si},k}}{C_{\text{Si}}R_{\text{Si},v}} + \sum_{i=1}^q \frac{T_{\text{Si},n,i} - T_{\text{Si},k}}{C_{\text{Si}}R_{\text{Si},h}} \\ \dot{T}_{\text{Cu},k} &= \frac{T_{\text{Si},k} - T_{\text{Cu},k}}{C_{\text{Cu}}R_{\text{Si},v}} + \frac{T_{\text{Cu},k} - T_{\text{amb}}}{C_{\text{Cu}}R_{\text{Cu},v}} + \sum_{i=1}^q \frac{T_{\text{Cu},n,i} - T_{\text{Cu},k}}{C_{\text{Cu}}R_{\text{Cu},h}} \end{aligned} \quad (40)$$

¹Border cells have lower resistance values to simulate lateral heat dissipation.

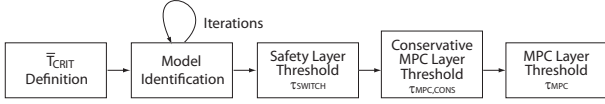


Figure 7: Scheme of the control design and implementation steps.

where $T_{Si,n,i}$ and $T_{Cu,n,i}$, $i \in \{1, \dots, q\}$ are respectively the neighbors of the k -th silicon and copper cell.

The admissible power consumption of each core ranges from $P_{\min} = 4.38W$ to $P_{\max} = 25W$, corresponding to frequencies $f_{\min} = 1.6GHz$ and $f_{\max} = 2.97GHz$ respectively. The idle power, i.e., the power consumed when the core is on but not running any workload, is $P_{IDLE} = 0.25W$. The power dissipated by the caches is the 30% of the power consumed by the cores directly connected to them. Fig. 7 shows the control design steps listed below

1. We **defined** $\bar{T}_{CRIT,i}$ as shown in Fig. 4, where T_{EQ} has been obtained by applying to each core the minimum power.
2. We **identified the prediction models** of each core performing the iterative procedure of Section 4.
3. We **defined** $\tau_{SWITCH,i}$ in the safety layer by inverting the local model.
4. We **determined** $\tau_{MPC,CONS,i}$ (the conservative MPC threshold) by running a set of benchmarks on the system supervised by the two-layer controller. For each benchmark and core, we set as initial threshold $\tau_{MPC,i} = \tau_{SWITCH,i}$, and we decreased the MPC threshold of the same amount until core temperatures remained confined below $\tau_{SWITCH,i}$.
5. We solved the optimization problem to **find** $\tau_{MPC,i}$ and maximize performance. Starting from $\tau_{MPC,i} = \tau_{MPC,CONS,i}$, we increased all the $\tau_{MPC,i}$ simultaneously², looking for the values that maximize the integral of the cores frequencies, then we averaged the results of all the benchmarks.

According to the previous points, we set $\bar{T}_{CRIT} = [358.89, 359.07, 359.80, 360.00, 359.74, 359.95, 359.29, 359.47]K$. As an example, we report below the model of the form (29) for core 3, obtained from the identification procedure with sampling time 1ms:

$$\begin{bmatrix} T_3(t+1) \\ x_{2,3}(t+1) \end{bmatrix} = \begin{bmatrix} 1.5 & 1 \\ -0.5 & 0 \end{bmatrix} \begin{bmatrix} T_3(t) \\ x_{2,3}(t) \end{bmatrix} + \begin{bmatrix} 3.2 \times 10^{-2} & -6.0 & 3.4 \times 10^{-4} & -3.9 \times 10^{-5} & 3.9 \times 10^{-5} \\ -3.0 \times 10^{-2} & 6.0 & 2.0 \times 10^{-4} & -3.6 \times 10^{-5} & 3.8 \times 10^{-4} \end{bmatrix} \begin{bmatrix} P_{C,3}(t) \\ T_{amb}(t) \\ T_1(t) \\ T_4(t) \\ T_5(t) \end{bmatrix}.$$

We checked feasibility of the obtained system by solving problem (37) with \mathbb{D} given by (35), with $T_{amb,\min} = 293K$ and $T_{amb,\max} = 330K$, and \mathbb{X} defined as in (36). **We solved the problem using the tool YALMIP (Löfberg, 2004). In particular,**

²Increasing $\tau_{MPC,i}$ by different values is impractical because of the significant amount of possibilities it would determine.

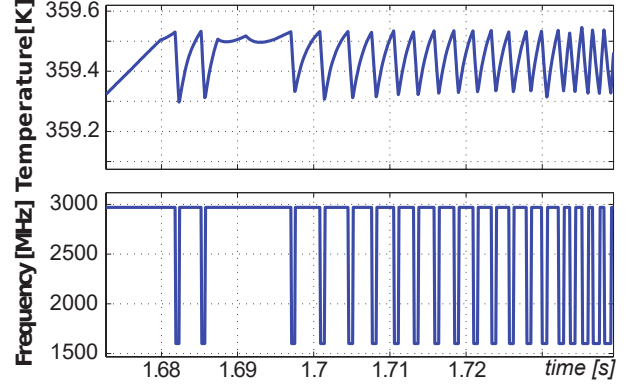


Figure 8: Response with only safety layer controller.

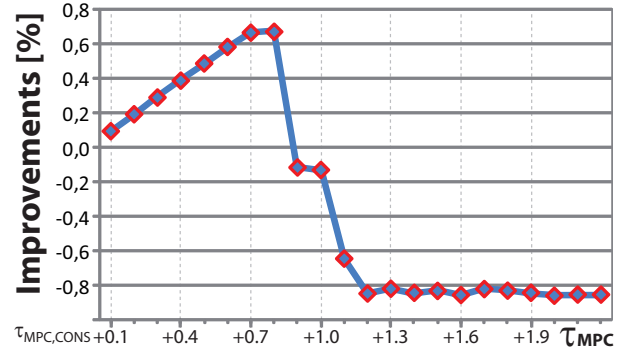


Figure 9: Performance comparison with different τ_{MPC} .

we adopted the local solver Ipopt, started from different initial conditions, and a more rigorous global solver (bminb). In both cases, we obtained a positive objective function value equal to 10.5, corresponding to the optimal vector $y^* = [0 \ 0.5 \ 0.5]^T$. Therefore, according to the discussion reported in Section 4.1, we can state the identified model satisfies the feasibility properties of the original PDE, and can be then profitably used for control purposes.

Then, we proceeded inverting the model to obtain $\tau_{SWITCH} = [358.63, 358.80, 359.53, 359.73, 359.47, 359.68, 359.02, 359.20]K$, whereas by iteratively decreasing the τ_{MPC} until no safety layer interventions were detected³, we determined the conservative MPC thresholds $\tau_{MPC,CONS} = [357.83, 358.00, 358.73, 358.93, 358.67, 358.88, 358.22, 358.40]K$.

Fig. 8 shows the temperature response of core 3 when only the safety layer is active. As expected, the temperature is bounded below $T_{CRIT} = 360K$.

The procedure to define τ_{MPC} is similar to the one used for $\tau_{MPC,CONS}$. We set $\tau_{MPC} = \tau_{MPC,CONS}$, then we iteratively increased its value by 0.1K. For each simulation, we stored the integral of the frequency as performance metric.

In Fig. 9 we compared the performance w.r.t. the chosen τ_{MPC} , which is indicated as a $\tau_{MPC,CONS}$ offset, owing to space limitation. The plots shown in Fig. 9 correspond to the average values obtained over the tests with different benchmarks. As result of the optimization problem, $\tau_{MPC} = [358.63, 358.80,$

³At each iteration $\tau_{MPC,CONS}$ is decreased by 0.1K.

359.53, 359.73, 359.47, 359.68, 359.02, 359.20]K that is equal to τ_{SWITCH} . As already mentioned, these values could also be greater due to the model conservativeness obtained with the identification procedure. The improvements in performance are about 1% on average. The lower hysteresis threshold of the switch controllers, $\tau_{\text{SWITCH,LOW}}$, is equal to $\tau_{\text{SWITCH}} - 0.1\text{K}$.

For what concerns the online solution of problem (38) in the MPC layer, we leveraged the core thermal model properties and its low dimensionality to design a simple strategy:

1. Set $P_{C,i} = P_{T,i}^A$, and use the model to predict $T_i(t+1) = x_{1,i}(t+1)$.
2. If $T_i(t+1) \leq \tau_{\text{MPC},i}$ then keep $P_{C,i} = P_{T,i}$ and compute the corresponding core frequency using (3).
3. If, $T_i(t+1) > \tau_{\text{MPC},i}$, with $P_{C,i} = P_{T,i}$, set $P_{C,i}$ so that $T_i(t+1) = \tau_{\text{MPC},i}$ by using (29), i.e.,

$$P_{C,i} = \frac{\tau_{\text{MPC},i} - a_{1,i}x_{1,i}(t) - x_{2,1}(t) - B_{2,i}(1, :)d_i(t)}{B_{1,i}(1, 1)}$$

where round brackets indicate the entries of the matrices B_1 and B_2 , and $:$ denotes all the row/column elements. Applying the procedure above in Matlab/Simulink, the solution to each core MPC problem can be computed in just $3\mu\text{s}$ on a PC with an *i7 8th* gen. CPU. Therefore, it could be deployed on many-core systems without impacting significantly on the cores computational effort, which could be directed to carry out the applications' specific workload. Fig. 10 shows the response of core 3 when Fluidanimate, a Parsec benchmark related to simulation (via Smoothed Particle Hydrodynamics method) of incompressible fluids for interactive animation purposes (Bienia et al., 2008) is assigned to the controlled system. The temperature is perfectly bounded below $\bar{T}_{\text{CRIT},3} = 359.80\text{K}$, and the safety layer intervenes only when the temperature crosses $\tau_{\text{SWITCH},3} = 359.53\text{K}$, setting the frequency to 1.6GHz. Fig. 11 shows the simulation results, with the same application benchmark, for core 3 when different MPC thresholds were applied: the τ_{MPC} that maximizes the performance ($\tau_{\text{MPC,MAX}}$), the τ_{MPC} that reduces the violation under the 0.1% ($\tau_{\text{MPC},0.1\%}$) and the completely conservative $\tau_{\text{MPC}} = \tau_{\text{MPC,CONS}}$. It can be noticed how, even though the safety layer intervenes more frequently than in the other cases, the frequency for ($\tau_{\text{MPC,MAX}}$) is the highest on average.

We validated our control strategy with several Parsec benchmarks. Owing to space constraints, here we report results for another scenario, corresponding to Bodytrack, a computer vision application. Figs. 12, 13 show the corresponding results. Again, the ability of the two layer solution to keep the core temperature under the critical threshold is confirmed, with the help of the safety layer when needed, while applying the requested power and frequency, whenever the thermal conditions allow it (see the first and third plots in Fig. 12). Also, the speculative strategy to select a less conservative MPC thresholds pays off even in this condition, as shown in Fig. 10.

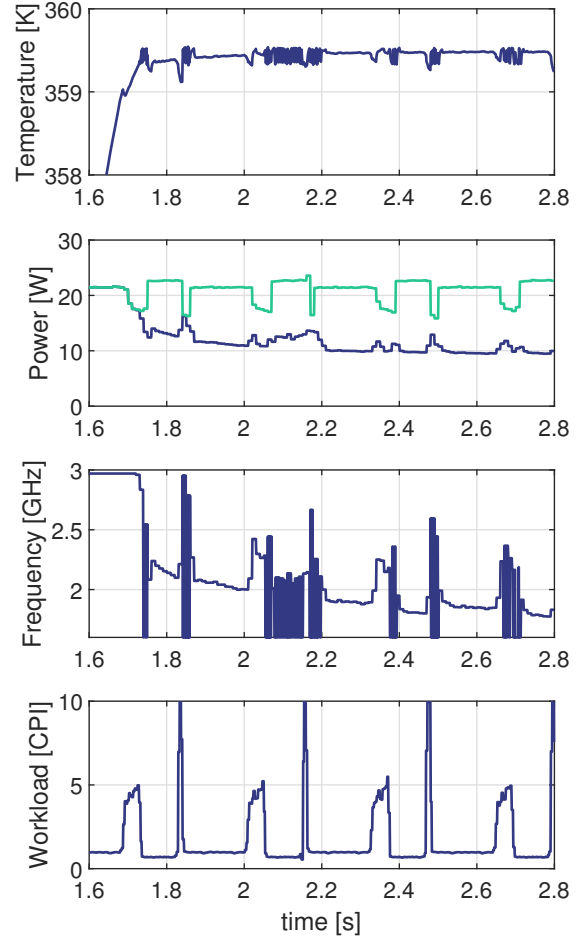


Figure 10: Simulation results for benchmark Fluidanimate.

6.1. Comparison with other solutions.

We conclude this Section with some considerations where we compare the proposed approach with some of the recent works, mentioned in Section 1. We remark that our two-layer solution formally guarantees feasibility, thanks to the safety layer, also in case of heavy model uncertainties. Moreover, the MPC layer is verified to ensure recursive feasibility for the considered system model. To the best of our knowledge, none of the works presented in the literature has fully addressed such issues. Also, the proposed model does not require any centralized step, with the safety layer being decentralized (only local temperature information is needed), and the MPC layer fully distributed (only local information exchange among neighbours core is needed). Other recent solutions in the MPC framework requires some sort of centralized coordination (Camponogara et al., 2021), (Wang et al., 2016), or adopt a fully centralized approach (Wang et al., 2019). Compared to machine learning-based approaches (Mandal et al., 2019), (Das et al., 2014), our approach can be implemented online with less complexity, and it does not require extensive offline exploration to guide an ef-

⁴We assume the target $P_{T,i}$ is within the power bounds, if not, it can be saturated before solving the problem.

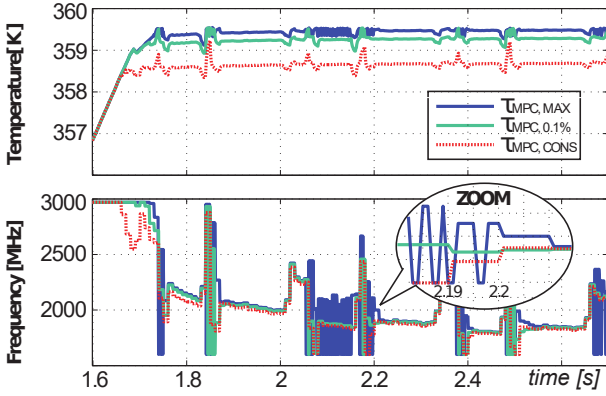


Figure 11: Performance comparison with benchmark Fluidanimate and three different τ_{MPC} .

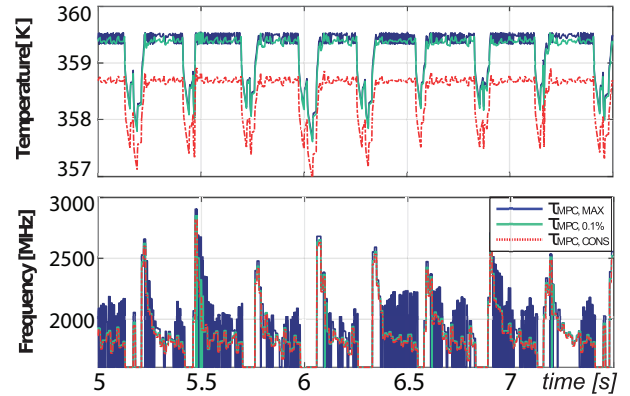


Figure 13: Performance comparison with benchmark Bodytrack and three different τ_{MPC} .

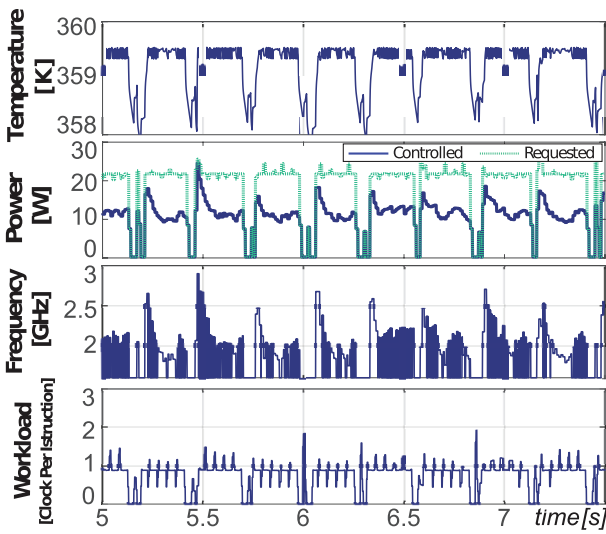


Figure 12: Simulation results for benchmark Bodytrack.

efficient runtime manager (Gupta et al., 2017).

7. Discussion and future works

In this work, we have exploited the features of a PDE-based thermal model, describing heat conduction phenomena, to study the feasibility of predictive controllers for managing Multiprocessors Systems-on-Chip temperature. Infinite-dimensional PDE description is crucial to avoid possible side effects related to time and space discretization. In this context, we showed that predictive controllers are intrinsically feasible when applied to whatever thermal system for any prediction horizon greater than or equal to zero. The proof of such a result relies on the Maximum Principle, which is also instrumental in deriving a rule for reducing the number of constraints from infinite to finite, greatly simplifying the controller design. Despite the final results looking quite intuitive, in the authors' opinion, the presented study is helpful to provide a rigorous confirmation of the intuition and revise the non-oscillating properties of thermal systems in an MPC context.

These achievements have been exploited to design a two-layer control architecture which, adopting time- and space-discretized approximated modeling (and discrete-time control actuation), ensures robust feasibility preserves performance as much as possible and keeps the controller complexity at a treatable level. To this aim, a distributed MPC layer manages the chip temperature maximizing performance, while a safety layer (a set of hysteresis controllers) guarantees feasibility.

From the practical application's viewpoint, we provided a step-by-step empirical tuning procedure for the proposed control solutions. Specifically, we described a method to derive a tradeoff between the frequency of intervention of the safety layer and the temperature bound of the MPC controllers. Compared to a completely conservative solution, which would always avoid safety intervention, we obtained up to 1% of performance enhancement.

References

- Bambini, G., Balas, R., Conficoni, C., Tilli, A., Benini, L., Benatti, S., Bartolini, A., 2020. An open-source scalable thermal and power controller for HPC processors, in: 2020 IEEE 38th International Conference on Computer Design (ICCD), IEEE, pp. 364–367.
- Bartolini, A., Cacciari, M., Tilli, A., Benini, L., 2012. Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller. *IEEE Trans. Parallel Distrib. Syst* 24(1), 170–183.
- Bemporad, A., Morari, M., 1999. Robust model predictive control: A survey', in: Garulli, A., Tesi, A., Vicino, A. (Eds.), *Robustness in identification and control*, Lecture Notes in Control and Information Sciences. Springer, p. 207226.
- Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E., 2002. The explicit linear quadratic regulator for constrained systems. *Automatica* 38, 3–20.
- Bienia, C., Kumar, S., Singh, J.P., Li, K., 2008. The parsec benchmark suite: Characterization and architectural implications. *Proc. of the 17th International Conference on Parallel Architectures and Compilation Techniques*.
- Camponogara, E., Scherer, H., Biegler, L., Grossmann, I., 2021. Hierarchical decompositions for mpc of resource constrained control systems: applications to building energy management. *Optimization and Engineering* 22, 187–215.
- Christofides, P., Scattolini, R., de la Peña, D., Liu, J., 2012. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering* 0, 56–65.
- Coskun, A.K., Ayala, J.L., Atienza, D., Rosing, T.S., Leblebici, Y., 2009. Dynamic thermal management in 3d multicore architectures, in: *Proceedings*

- of the Conference on Design, Automation and Test in Europe, European Design and Automation Association. pp. 1410–1415.
- Das, A., Shafik, R.A., Merrett, G.V., Al-Hashimi, B.M., Kumar, A., Veeravalli, B., 2014. Reinforcement learning-based inter-and intra-application thermal optimization for lifetime improvement of multicore systems, in: Proceedings of the 51st Annual Design Automation Conference, pp. 1–6.
- Dey, S., Singh, A.K., McDonald-Maier, K.D., 2019. P-EdgeCoolingMode: an agent-based performance aware thermal management unit for DVFS enabled heterogeneous MPSoCs. *IET Computers & Digital Techniques* 13, 514–523.
- Evans, L.C., 1998. *Partial Differential Equations*. American Mathematical Society.
- Gondhalekara, R., Imura, J., Kashima, K., 2009. Controlled invariant feasibility - a general approach to enforcing strong feasibility in mpc applied to move-blocking. *Automatica* 45(12), 2869–2875.
- Gupta, U., Patil, C.A., Bhat, G., Mishra, P., Ogras, U.Y., 2017. Dypo: Dynamic pareto-optimal configuration selection for heterogeneous mpsoacs. *ACM Transactions on Embedded Computing Systems (TECS)* 16, 1–20.
- Hamann, H.F., Weger, A., Lacey, J.A., Hu, Z., Bose, P., Cohen, E., Wakil, J., 2007. Hotspot-limited microprocessors: Direct temperature and power distribution measurements'. *IEEE Journal of Solid-State Circuits* 4, 56–65.
- Iranfar, A., Kamal, M., Afzali-Kusha, A., Pedram, M., Atienza, D., 2017. Thespot: Thermal stress-aware power and temperature management for multiprocessor systems-on-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 1532–1545.
- Kadin, M., Reda, S., Uht, A., 2009. Central vs. distributed dynamic thermal management for multi-core processors: which one is better? *ACM Great Lakes Symposium on VLSI*, New York, NY, USA, , 137–140.
- Kang, K., Kim, J., Yoo, S., Kyung, C.M., 2011. Runtime power management of 3-d multi-core architectures under peak power and temperature constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30, 905–918.
- Ladenheim, S., Chen, Y.C., Mihajlović, M., Pavlidis, V.F., 2018. The mta: An advanced and versatile thermal simulator for integrated systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37, 3123–3136.
- Liu, Z., Tan, S.X.D., Huang, X., Wang, H., 2014. Task migrations for distributed thermal management considering transient effects. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 23, 397–401.
- Löfberg, J., 2004. Yalmip: A toolbox for modeling and optimization in matlab, in: Proceedings of the CACSD Conference, Taipei, Taiwan.
- Löfberg, J., 2011. Oops! i cannot do it again: Testing for recursive feasibility in mpc. *Automatica* 48(3), 550–555.
- Mallik, D., Mahajan, R., Wakharkar, V., 2008. Flip chip packaging for nanoscale silicon logic devices: Challenges and opportunities, in: Morris, J.E. (Ed.), *Nanopackaging: nanotechnologies and electronics packaging*. Springer, pp. 491–516.
- Mandal, S.K., Bhat, G., Patil, C.A., Doppa, J.R., Pande, P.P., Ogras, U.Y., 2019. Dynamic resource management of heterogeneous mobile platforms via imitation learning. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27, 2842–2854.
- McKeown, M., Lavrov, A., Shahrada, M., Jackson, P.J., Fu, Y., Balkind, J., Nguyen, T.M., Lim, K., Zhou, Y., Wentzlaff, D., 2018. Power and energy characterization of an open source 25-core manycore processor., in: *HPCA*, pp. 762–775.
- Paci, G., Morari, M., Dua, V., Pistikopoulos, E., 2006. Exploring temperature-aware design in low-power mpsoacs. *Design, Automation and Test in Europe Conference* , 1–6.
- Pagani, S., Manoj, P.S., Jantsch, A., Henkel, J., 2018. Machine learning for power, energy, and thermal management on multicore processors: A survey. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 101–116.
- Rawlings, J., Mayne, D., 2009. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin.
- Rusu, S., Tam, S., Muljono, H., Stinson, J., Ayers, D., Chang, J., Varada, R., Ratta, M., Vora, S., 2010. A 45 nm 8-core enterprise xeon@processor. *IEEE Journal of solid-state circuits* 45, 7–14.
- Scherer, H., Camponogara, E., Normey-Rico, J., Álvarez, J.D., Guzmán, J.L., 2015. Distributed mpc for resource-constrained control systems. *Optimal Control Applications and Methods* 36, 272–291.
- Shafik, R.A., Yang, S., Das, A., Maeda-Nunez, L.A., Merrett, G.V., Al-Hashimi, B.M., 2015. Learning transfer-based adaptive energy minimization in embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 877–890.
- Skadron, K., Abdelzaher, T., Stan, M.R., 2002. Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management. the 8th International Symposium on High-Performance Computer Architecture, Anaheim, CA, USA , 17–28.
- Strauss, W.A., 1992. *Partial Differential Equations: An Introduction*. Wiley.
- Tilli, A., Garone, E., Cacciari, M., Bartolini, A., 2012. Thermal models characterization for reliable temperature capping and performance optimization in multiprocessor systems on chip. *Proc. of American Ctrl. Conf., Montreal, Canada, , 4721–4726*.
- Tsai, T.H., Chen, Y.S., 2014. A thermal-throttling server in 3d multicore chips, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, ACM. pp. 1425–1430.
- Wang, H., Guo, X., Tan, S.X.D., Zhang, C., Tang, H., Yuan, Y., 2019. Leakage-aware predictive thermal management for multicore systems using echo state network. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 1400–1413.
- Wang, H., Ma, J., Tan, S.X.D., Zhang, C., Tang, H., Huang, K., Zhang, Z., 2016. Hierarchical dynamic thermal management method for high-performance many-core microprocessors. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 22, 1–21.
- Wang, Y., Ma, K., Wang, X., 2009. Temperature-constrained power control for chip multiprocessors with online model estimation. *Proc. of the 36th annual international symposium on Computer architecture*, Austin, TX, USA, .
- Wang, Z., Zhu, X., McCarthy, C., Ranganathan, P., Talwar, V., 2008. Feedback control algorithms for power management of servers. 3rd International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks, Annapolis, MD .
- Zanini, F., Atienza, D., Benini, L., Micheli, G.D., 2009. Multicore thermal management with model predictive control. *Proc. of the 19th European Conference on Circuit Theory and Design* , 90–95.