Data-Driven Control of Thermoforming Machines

Eva Masero, *Member, IEEE*, Walter Zoff, and Riccardo Scattolini

***Abstract*—Thermoforming processes are widely used in plastics manufacturing and involve heating a plastic sheet to give it a specific shape. Changing the plastic material and the operating points in which the thermoforming machine operates requires an adjustment of its parameters so that the desired plastic piece is obtained and no failures in form occur. In particular, the heating phase parameters are the most critical, as the power and heating time have a significant impact on the final quality of the piece. Parameter tuning is usually done by an operator and often results in a waste of time, energy, and resources that should be minimized. This work investigates two sequential methods based on input-output data to optimize machine parameters that lead to the desired plastic form. Specifically, we propose two data-driven control stages for the thermoforming tuning problem. Both approaches have been validated in simulation, showing their ability to optimize the heating phase parameters to obtain the desired plastic shape.**

***Index Terms*—Industrial processes, machine learning, model identification, model predictive control (MPC), thermoforming.**

I. DESCRIPTION

THERMOFORMING processes are used in plastic manufacturing and generally involve heating a plastic sheet to give a specific shape using a mold [1]. An example of a sheet-processing thermoforming machine during the heating phase is shown in Fig. 1. The main hardware components of typical thermoforming machines are as follows.

1) *Human–Machine Interface (HMI):* For setting the parameters of the thermoforming machine.
2) *Loader:* A mobile carrier that transfers the sheet from storage to the thermoforming zone and then also returns the final thermoformed product.
3) *Frame:* A mobile structure that secures the plastic sheet in place during the process.
4) *Heaters:* Infrared lamp grids that regulate the heating phase. The machine is equipped with upper and lower heaters composed of a given number of lamps.
5) *Pyrometer:* A noncontact temperature sensor installed at a specific location, which monitors the sheet's temperature and determines when the heating should stop.

Fig. 1. Picture of a sheet-processing thermoforming machine generated using ChatGPT (OpenAI, 2025), and scheme of the main phases of the thermoforming cycle: heating, blowing, and stamping.

6) *Thermographic Camera:* It provides a thermal image that accurately captures the sheet's temperature map at the end of the heating phase.
7) *Mold and Counter-Mold:* The shaping elements that mold the final product.
8) Reduction plate: It regulates the open area beneath the sheet according to the desired product size.
9) *Coolers:* The machine has coolers for solidifying the formed plastic to maintain its shape.

The thermoforming cycle consists of sequential steps [2], [3] typically grouped into three phases: heating, blowing, and stamping. Initially, the sheet is transferred from storage to the thermoforming area and heated by the upper and lower heaters to the desired temperature. Once finished, the thermographic camera takes a photo of the heated sheet. A pressurization airflow then expands the heated sheet, while a mobile cart positions the mold. Subsequently, an inverter reverses the airflow, depressurizing the chamber and causing the sheet to

Received 2 February 2026; revised 20 April 2026; accepted 30 May 2026. This work was supported by the Next-Generation European Union (EU) [Italian Piano Nazionale di Ripresa e Resilienza (PNRR)-M4 C2, Invest 1.3-D.D.1551.11-10-2022, PE00000004] for the Made in Italy—Circular and Sustainable (MICS) under Grant CUP: MICS D43C22003120001. Recommended by Associate Editor T. Yang. *(Corresponding author: Eva Masero.)*

The authors are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, 20133 Milan, Italy (e-mail: eva.masero@polimi.it; walter.zoff@mail.polimi.it; riccardo.scattolini@polimi.it).

Digital Object Identifier 10.1109/TCST.2026.3700705

conform to the mold. Finally, the shaped sheet is cooled (and trimmed if necessary) and returned to storage.

II. STATE-OF-THE-ART IN THERMOFORMING

The thermoforming plastic market has experienced significant growth in recent years due to the increasing demand for recycled, lightweight, cost-effective, and customizable plastic components in industries such as food packaging, medical, automotive, and consumer goods [4]. This trend is further supported by thermoforming's ability to achieve high-volume production with reduced manufacturing costs [5]. The thermoforming plastic market size in Europe represented 23% of global thermoformed plastic revenue in 2023, with a market size of 3407 $ million. It is expected to reach 4498 $ million by 2030, registering a compound annual growth rate of 4.1% from 2024 to 2030 [6]. As the market continues to grow, it highlights the need for efficient and adaptable manufacturing processes, including advanced thermal control and real-time optimization tools.

A. Challenges and Technological Developments

Thermoforming machines have a series of adjustable parameters [2, Table 1], commonly referred to as *recipe parameters* (e.g., heating power, process times, vacuum pressure, etc.), which influence the characteristics of the final thermoformed product. These parameters are typically adjusted through trial and error via the machine HMI based on operator experience, which often leads to significant material waste, machine downtime, and suboptimal product quality. Moreover, recipe tuning cannot be performed once at the beginning of the machine's life cycle, but it must be repeated whenever material, ambient conditions, or tooling change to maintain product quality. Therefore, this makes manual tuning both time-consuming and costly, motivating the need for more systematic and automated optimization methods.

From a modeling perspective, the thermoforming process is governed by complex heat-transfer phenomena. Foundational principles of heat transfer, including conduction, convection, and radiation of the process, are well established [7] and form the basis of detailed physics-based models. Some traditional thermoforming models rely on partial differential equations and finite element methods to simulate the heating phase [8], [9]. Although these models provide detailed physical insight [10], their high computational cost and strong dependence on material properties and conditions severely limit their use in real-time optimization and control.

From a control perspective, and given the repetitive nature of the thermoforming process, cycle-to-cycle control approaches [11], such as iterative learning control (ILC) and terminal ILC (TILC), have been widely adopted. These methods update control inputs from one production cycle to the next based on the observed performance of previous cycles. For instance, Gauthier and Boulet [12] used a linear heating model combined with TILC to automatically tune heater temperature setpoints, while Girard et al. [13] applied cycle-to-cycle control to regulate heating profiles using sensor feedback. These approaches are effective at compensating for repeatable disturbances and model mismatch, but they rely on relatively simple linear models and typically converge slowly when large recipe parameter changes are required. Alternatively, Erchiqui [14] combined genetic algorithms with simulated thermal annealing to optimize the heating phase, achieving improved uniformity.

Model predictive control (MPC) offers a more systematic way to handle multivariate interactions and operational constraints [15], [16]. Given an accurate dynamic model, MPC is able to predict the future system output over a finite time horizon and compute optimal control actions [17]. For example, Chy et al. [18] proposed an MPC-based approach to manage the temperature distribution of thermoformed plastic sheets using infrared sensors, later extended in [19] by incorporating an ILC strategy to improve cycle-to-cycle performance. In both studies, the authors rely on a complex nonlinear physical model that must be linearized and solved repeatedly, leading to significant modeling effort and computational overhead. As a result, the use of offline multiparametric MPC is often required to make real-time implementation feasible [20].

More recently, hybrid and data-based approaches have been proposed to alleviate these limitations. For example, Hosseinionari and Seethaler [21] combine MPC based on a linear physics-based model with reinforcement learning (RL) to guide and accelerate the training of RL agents, thereby reducing online computational burden while preserving constraint handling for thermoforming temperature control. Moreover, Marchal et al. [22] use embedded sensors and design of experiments (DoE) to build statistical models for predicting sheet thickness in real time. This method demonstrates that data-driven models can provide fast and accurate predictions without requiring detailed physical modeling, but they typically do not address full recipe optimization.

Despite these advances, existing approaches still exhibit important limitations in terms of modeling effort, adaptability, and computational efficiency. Physics-based and MPC-based methods require significant modeling effort, adaptability, and computational resources, limiting their practical implementation. Cycle-to-cycle and ILC-based strategies rely on simplified linear models and may converge slowly when large recipe parameter changes are required. The existing data-driven approaches, although computationally efficient, typically focus on prediction tasks and do not address the full problem of recipe optimization. These limitations highlight the need for approaches that combine low modeling effort, computational efficiency, and adaptability to varying operating conditions.

B. Research Gap and Contributions

As pointed out in [22], a promising future direction, which remains an open gap, is the integration of prediction models with real-time numerical simulations to reduce tuning time and assist operators in configuring recipe parameters. Furthermore, prediction targets may also include the visual quality of the final plastic sheet. However, existing methods rely on computationally demanding physical models or address only partial aspects of the problem (e.g., prediction or cycle-to-cycle correction), lacking a unified and practical framework for full recipe optimization.

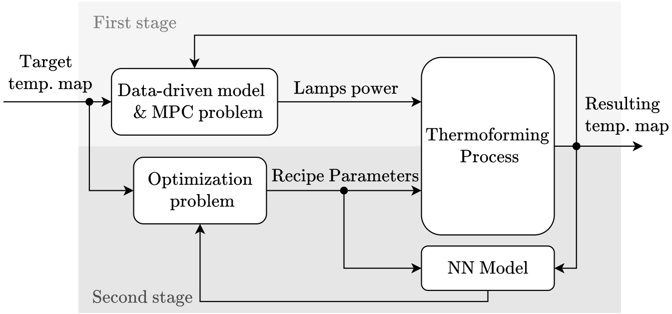


Fig. 2. Two-stage data-driven control scheme: the first stage optimizes lamp power, while the second stage fine-tunes remaining parameters based on the target temperature map.

In this context, we aim to address this gap by proposing a data-driven strategy for efficient thermoforming parameter tuning and adaptation. The thermoforming process naturally decomposes into two tasks: lamp power configuration and fine-tuning of the remaining recipe parameters, which must adapt to varying operating conditions (e.g., ambient temperature, humidity, and plastic characteristics). Accordingly, this article proposes a data-driven approach composed of two stages, as illustrated in Fig. 2.

1) The first stage determines the initial lamp power configuration according to the desired thermoformed characteristics, providing a tool for fast computation of lamp powers to reduce operator effort and setup time.
2) The second stage refines the remaining recipe parameters using a neural network (NN) model that enables adaptation to changing operating conditions through periodic retraining with new production data.

Preliminary results on the second stage of the proposed approach were presented in [23]. This article substantially extends that work with a more detailed data preparation pipeline for NN training, a more comprehensive qualitative evaluation of the NN model accuracy, a complete image preprocessing workflow, and additional optimization results.

Given the difficulty of physical modeling, two complementary data-driven models (see Fig. 2) are considered to balance accuracy and simplicity.

1) At the first stage, a linear model derived from a superposition analysis of lamp power, which provides a simple and physically interpretable relation between heating power and temperature contribution for the initial setup.
2) At the second stage, a nonlinear NN model was trained on historical data to capture complex thermal interactions in the thermoforming process.

This hybrid strategy balances modeling effort, computational efficiency, and adaptability, addressing key limitations of existing thermoforming control approaches. This study was conducted in collaboration with an Italian thermoforming machine manufacturer, using both simulated and real data from the same industrial machine. Specifically, the first model relies on data generated with a reliable, deterministic thermoforming simulator developed in OpenFOAM,[1] while the second model is trained on real data obtained from the physical machine.

TABLE I

NOTATIONS

Symbol	Description	Unit
$\mathcal{N} \in \mathbb{R}^N$	Set of N lamps	–
$c_i \in \mathbb{Z}_0^+$	Discrete power level (from 0 to 5) of lamp $i \in \mathcal{N}$	–
$C \in \mathbb{R}^N$	Power level configuration (vector of N levels)	–
P	Total number of spatial points in the 2D image	–
$T \in \mathbb{R}^P$	Temperature map (vector of P points)	°C
T_{sim}	Temperature map obtained from the simulator	°C
T_{model}	Temperature map obtained from the model	°C
$T(0)$	Nominal temperature map, for all $c_i = 0$	°C
$T(C), T^C$	Temperature map for configuration C	°C
T_{targ}	Target temperature map	°C
ΔT^C	Temperature map increment: $T^C - T(0)$	°C
ΔT^C_{sim}	Temperature map increment of simulator	°C
$\Delta T^C_{\text{model}}$	Temperature map increment of model	°C
$f_{\text{model}}(\cdot)$	Static model of temperature variation	°C
ΔT_{targ}	Target temperature map increment	°C
ρ	Temperature threshold	°C
e_{abs}	Absolute error	°C
e_{rel}	Relative error	%
T_{s}	Sampling time	s
τ	Time constant of the plastic sheet warming	s
J	Cost function	–
t	Continuous time index	s
k, k_{fin}	Discrete time index, final time index	–
N_{p}	Prediction horizon of the MPC	–
ϵ	Slack variable	–
λ, ϕ, φ	Weights in the cost function	–
$e^{\max}_{\Delta T} \in \mathbb{R}^P$	Vector of the maximum absolute error allowed	°C
$f(x)$	ReLU activation function $\max(0, x)$	–
$\sigma(x)$	Sigmoid activation function	–
$\mathcal{M} \in \mathbb{R}^M$	Set of M pixels in the image	–
y_m	Intensity of pixel $m \in \mathcal{M}$	–
$\hat{y}_m$	Predicted intensity of pixel $m \in \mathcal{M}$	–
$y_{\max}$	Maximum intensity of pixels	–
$\mu(\cdot)$	Mean value function	–
$\delta(\cdot)$	Standard deviation function	–

In thermoforming machines, the heating phase plays a critical role, as the desired shape is linked to its temperature map. Spatial temperature gradients directly influence the gradient of the resulting 3-D form, highlighting the importance of thermal distribution in shaping the final product. By considering both a model and a target temperature map, an optimization problem can be formulated to determine the optimal parameters for a given desired shape. Specifically, we propose two complementary optimization problems, each corresponding to one of the previously introduced models.

1) The first optimization problem, referred to as the first stage, is based on MPC and provides an offline tool to assist the machine operator during setup. It determines the optimal steady-state lamp power configuration for the desired thermoformed shape.
2) The second optimization problem, referred to as the second stage, aims to fine-tune recipe parameters other than lamp power. It is conceived as an automatic refinement mechanism that adjusts these parameters in real time in response to production variations.

Our contribution is primarily empirical and application-driven, showing that a data-driven approach can achieve accurate and reliable performance in a complex thermoforming process where first-principles modeling is impractical. The plastic sheet geometry is assumed to be rectangular, as in standard industrial practice, with variation limited to the temperature distribution across its surface; however, the proposed methodology can be extended to arbitrary geometries (e.g., square, circular, or irregular) by redefining the spatial discretization and the lamp-to-sheet mapping.

---

[1] https://www.openfoam.com/

For clarity, Table I summarizes the notation used throughout the article. The remainder of the article is organized as follows. Section III presents the first stage, including model identification, MPC-based optimization, and simulation results. Section IV describes the NN-based refinement of the second stage and shows its validation in simulations. Section V concludes with the main findings and potential future research.

III. LAMP MANAGEMENT

This section analyzes the role of heating lamps to achieve the desired temperature map. We use input-output data from a reliable thermoforming simulator and proceed as follows. First, we identify a data-based linear model of the heating phase based on lamps. Specifically, we perform a superposition analysis of the temperature on the plastic sheet when the lamp powers are increased by a certain percentage with respect to their baseline setting (i.e., set at 50% of their maximum power). The motivation behind this approach is to derive a data-driven model that can be obtained with reduced experimental effort, thus facilitating its implementation in thermoforming machines and its retuning whenever needed. This model is then employed to optimize the lamp powers to achieve the target temperature map while minimizing power costs.

A. Model Identification for Lamps

Since the characteristics of the heated sheet are mainly dependent on the lamp's power, we first investigate how lamp settings influence the temperature map at the end of the heating phase. The heating system comprises an upper bank and a lower bank of heaters, each consisting of a set of independently controllable lamps indexed by $\mathcal{N} \in \{1, \ldots, N\}$, whose power levels are specified as a percentage. In standard operation, both lower and upper heaters are initially set to a nominal power level (typically around 50%) to establish a stable baseline thermal field [24], while subsequent spatial refinement is performed to shape the final temperature distribution. In this work, the refinement is performed by adjusting only the upper heater bank, while the lower heaters remain fixed at their baseline setting (50%). This procedure simplifies control actions while still providing sufficient degrees of freedom for spatial correction. Accordingly, we focus on the management of the upper heater bank, which consists of $N = 120$ lamps uniformly distributed. Note that the proposed optimization framework is not restricted to a single heater and can be extended to jointly optimize both upper and lower heaters.

The powers of upper lamps are the simulator inputs and can be set to certain setpoints (in %), while the simulator output is the temperature map relative to the 2-D thermal image provided by the thermographic camera. In this case, the temperature map is a vector that contains P values (in °C), each corresponding to a spatial point in the 2-D image. The control inputs are defined as the power level increments of the upper lamps with respect to the nominal setting. These increments are collected in the vector $C = [c_i]_{i \in \mathcal{N}}$ and used as inputs to the simulator. Each control variable c_i can take the following integer values:

$$c_i \in \{0, 1, 2, 3, 4, 5\} \tag{1}$$

whose value is related to a power level increment

$$\begin{aligned} &0 \rightarrow +0\%, \quad 2 \rightarrow +20\%, \; 4 \rightarrow +40\% \\ &1 \rightarrow +10\%, \quad 3 \rightarrow +30\%, \; 5 \rightarrow +50\%. \end{aligned}$$

The actual % power of ith lamp is computed as $50 + 10\,c_i$.

The power level configuration $C \in \mathbb{R}^N$ provides a temperature map collected in the vector $T(C) \in \mathbb{R}^P$. In this case, we define as output the vector $\Delta T^C = T(C) - T(0)$, i.e., the difference between the temperature map $T(C)$ and the nominal temperature map $T(0)$, which is the result of applying the power level $c_i = 0$ to all i lamps. Note that these discrete power levels $c_1, c_2, \ldots, c_N$ will be the optimization variables of the control problem. In the next subsections, we denote by T^C_{model} and T^C_{sim} the temperature maps obtained from the approximate model and the simulator, respectively, for a given power level configuration C.

B. Dataset and Temperature Superposition Analysis

The dataset was generated by modifying the upper lamps one by one with a power increase of 10% from 50% to 100%, while the lower heating lamps were kept at their nominal power of 50%. This helps us understand and compute the temperature contribution of the single lamps at each spatial point in the temperature map. Therefore, the dataset contains the temperature generated by each lamp $i \in \{1, \ldots, N\}$ at each spatial point $p \in \{1, \ldots, P\}$ depending on its power level $c_i \in \{0, \ldots, 5\}$. This information can serve as a data-based model for predicting the final temperature map, provided it is well generalized to the heating-phase behavior. In the following, the combined contribution of lamp pairs is examined as part of the analysis.

For the temperature superposition analysis, we generated data samples in which pairs of lamps act simultaneously, each with a random power level between 1 and 5. Here, the objective is to analyze how power level increments are translated into temperature increases and to check if the principle of temperature superposition holds for any pair of lamps $i, j \in \mathcal{N}$ with $i \neq j$. We conducted tests to evaluate the validity of the superposition principle, assessing model accuracy through both graphical results and numerical performance indicators

$$e_{\text{abs}} = \max_p \left(\left| \Delta T^C_{\text{model,th}}[p] - \Delta T^C_{\text{sim,th}}[p] \right| \right)$$

$$e_{\text{rel}}\,(\%) = \max_p \left(\frac{\left| \Delta T^C_{\text{model,th}}[p] - \Delta T^C_{\text{sim,th}}[p] \right|}{T^C_{\text{sim,th}}[p]} \right) \cdot 100$$

where the temperature variations for the model and the simulator are defined, respectively, as $\Delta T^C_{\text{model}} = T^C_{\text{model}} - T(0)$ and $\Delta T^C_{\text{sim}} = T^C_{\text{sim}} - T(0)$ for a given power level configuration C. To reduce noise, points in ΔT^C_{sim} below a threshold of $\rho = 0.5$ °C were discarded, resulting in $\Delta T^C_{\text{sim,th}}$. The same points were also removed from $\Delta T^C_{\text{model}}$ and T^C_{sim} yielding $\Delta T^C_{\text{model,th}}$ and $T^C_{\text{sim,th}}$, respectively.

The maximum absolute error, e_{abs}, and the maximum relative error, e_{rel}, are used to quantify how accurately the superposition-based model reproduces the simulator response under the assumption of linear additive behavior. In particular, the model predicts the temperature distribution by summing the individual contributions of each lamp, whereas the simulator provides the ground-truth response when those

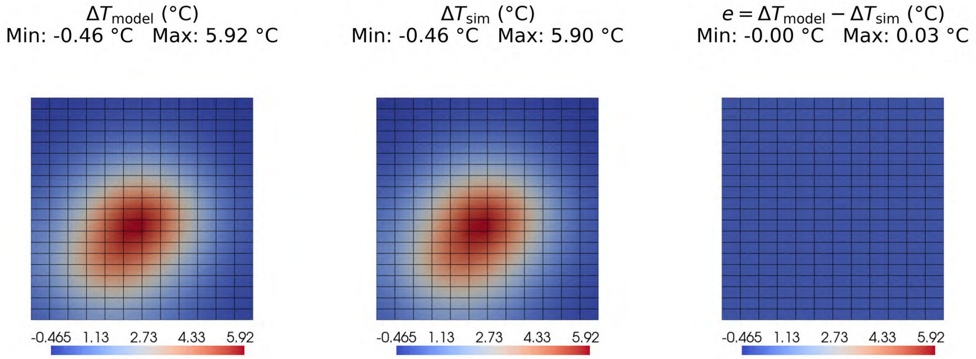


Fig. 3. Results for Test #1. The same color scale is applied for consistent comparison. The left plot (model) the temperature increment from the superposed effects of two individual lamps (ΔT_{model} in °C); the center plot (simulator) the result from simultaneous lamp operation in the simulator (ΔT_{sim} in °C); and the right plot shows the error ($e = \Delta T_{\text{model}} - \Delta T_{\text{sim}}$ in °C).

TABLE II
MODEL ACCURACY

	Lamps ID (power [%])	e_{abs} [°C]	e_{rel} [%]
Test #1	ID: 59 (80 %) and 56 (100 %)	0.032	0.0189 %
Test #2	ID: 74 (100 %) and 71 (100 %)	0.145	0.0871 %
Test #3	ID: 13 (70 %) and 16 (70 %)	0.004	0.0024 %
Test #4	ID: 55 (70 %) and 52 (80 %)	0.14	0.0877 %

lamps are activated simultaneously. Table II reports these error metrics for four representative tests. For each test, the analyzed lamp pairs are indicated by their IDs together with their corresponding power levels. In all tests, both absolute and relative errors remain small, even at high power, showing that the temperature field generated by two simultaneously active lamps is well approximated by the linear superposition of their individual contributions. Additionally, Fig. 3 illustrates a representative example (Test #1). The temperature increment predicted by the superposition-based model, ΔT_{model}, is compared with the simulator output, ΔT_{sim}. The resulting error map $e = \Delta T_{\text{model}} - \Delta T_{\text{sim}}$ confirms the very small discrepancy.

To provide reliable statistical validation, we evaluated the model over 100 randomly generated configurations of lamp pairs with varying power levels. The absolute error remained consistently low, with a mean of 0.044, a median of 0.014, respectively, 90th and 95th percentiles of 0.152 and 0.164, respectively, and a maximum value of 0.181. Similarly, the relative error was low, with a mean of 0.026%, a median of 0.009%, 90th and 95th percentiles of 0.091% and 0.101%, respectively, and a maximum of 0.107%. These results demonstrate that the superposition assumption holds reliably for lamp pairs. Hence, interaction effects are negligible within the considered operating regime, supporting the validity of our superposition-based modeling approach. We would like to remark that this approximation significantly limits the experimental effort required for model identification in practical industrial settings, and it can also be used for easy retuning when required due to changing operating conditions.

C. Dynamic Model

The control objective is to minimize the power of the upper lamps while tracking the desired temperature map $T_{\text{targ}} \in \mathbb{R}^P$, which depends on the final shape of the plastic product. Following a similar notation as previously, we consider $\Delta T_{\text{targ}} = T_{\text{targ}} - T(0)$ as the variation of the target temperature with respect to the nominal temperature. Since the final product is required to have a specific shape, the target temperature map must be defined according to that shape. Consider, as examples, the cases shown in Fig. 4.

For any power level configuration $C = [c_i]_{i\in\mathcal{N}}$ with $c_i \in \{0, \dots, 5\}$, the proposed data-based model allows for calculating the temperature variation $\Delta T^C_{\text{model}}$. This variation is experimentally approximated as the superposition of the individual contributions generated by each lamp $i \in \mathcal{N}$, namely

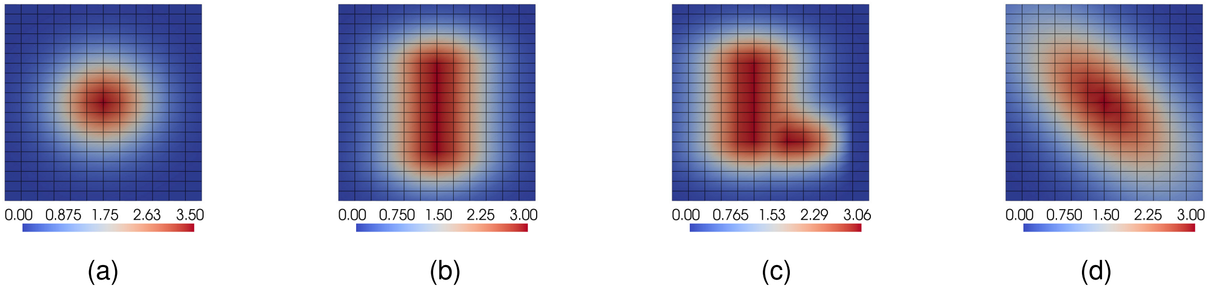
$$\Delta T^C_{\text{model}} = f_{\text{model}}(C) = \sum_{i\in\mathcal{N}} \Delta T_{\text{model}}(c_i) \tag{2}$$

where $f_{\text{model}}(C)$ denotes the static model that maps the lamp power configuration to the corresponding temperature variation. Here, $\Delta T_{\text{model}}(c_i) = T_{\text{model}}(c_i) - T(0)$ represents the temperature increase produced by a single lamp operated at power increment c_i relative to the nominal temperature. The temperature model (2) can be considered as a look-up table of size: (N lamps, 6 power levels, P points), which contains the temperature contribution of each lamp $i \in \mathcal{N}$ at each spatial point as a function of its power level c_i.

We would like to remark that the data-driven model (2) is a static model that relates the lamp power levels to the temperature variations. To derive a dynamic model that will be later needed for implementing the MPC approach, we also take into account the thermal inertia of the heating system. In particular, we consider first-order continuous dynamics to the steady temperature set by the lamp power [25]

$$\frac{\partial \Delta T^C_{\text{model}}(t)}{\partial t} = -\frac{1}{\tau}\left(\Delta T^C_{\text{model}}(t) - f_{\text{model}}(C(t))\right) \tag{3}$$

where τ represents the dominant thermal time constant associated with the heating dynamics, which depends on the material properties (e.g., thermal conductivity, heat capacity, and thickness) and the heat transfer conditions. Therefore, τ is not an intrinsic constant, but varies with the specific material and operating conditions considered.



(a) (b) (c) (d)

Fig. 4. Examples of target temperature map (in °C) according to the final product shape. (a) O-shape. (b) I-shape. (c) L-shape. (d) Diag-shape.

TABLE III

SUMMARY OF THE MPC RESULTS AT TIME INSTANT $k = k_{\text{fin}}$

	O-shape	**I-shape**	**L-shape**	**Diag-shape**
Total Power: $\sum_{i\in\mathcal{N}} c_i^*$	6	8	7	8
Max error: $\|\Delta T_{\text{targ}} - \Delta T^C\|_\infty$	0.22°C	0.45°C	0.82°C	0.19°C
Cost value: J	145.5	619.4	2260.63	231.36
Slack variable: $[\epsilon(1), \ldots, \epsilon(k_{\text{fin}})]$	$[2.5, 0, 0, \ldots, 0]$	$[2, 0, 0, \ldots, 0]$	[2.05, 0.11, 0.07, 0.07, 0.15, 0.15, 0.07, 0.07, 0.15, 0.15]	$[2, 0, 0, \ldots, 0]$
Non-zero power levels: c_i^*	Lamp 3: Level 1 Lamp 54: Level 1 Lamp 56: Level 3 Lamp 112: Level 1	Lamp 3: Level 1 Lamp 52: Level 1 Lamp 54: Level 2 Lamp 58: Level 3 Lamp 112: Level 1	Lamp 53: Level 2 Lamp 54: Level 1 Lamp 57: Level 2 Lamp 77: Level 1 Lamp 79: Level 1	Lamp 1: Level 1 Lamp 6: Level 1 Lamp 53: Level 2 Lamp 56: Level 2 Lamp 77: Level 1 Lamp 80: Level 1

For a sampling time of $T_s = 1$, the continuous-time model (3) can be discretized as

$$\Delta T^C_{\text{model}}(k+1) = \alpha\, \Delta T^C_{\text{model}}(k) + \beta\, f_{\text{model}}(C(k)) \tag{4}$$

where k is the discrete-time index, and $\alpha = e^{-T_s/\tau}$ and $\beta = 1 - \alpha$ are the corresponding discrete-time parameters. The experimental evaluation using the simulator under different operating conditions allows the time constant to be set to $\tau = 1$ s, yielding $\alpha = 0.37$ and $\beta = 0.63$.

D. Model Predictive Control

The optimization problem aims to minimize both the deviation of the sheet temperature from a desired target map and the associated power cost. To this end, we define the cost function as follows:

$$J = \sum_{i\in\mathcal{N}} c_i(k) + \left(\Delta T_{\text{targ}} - \Delta T^C_{\text{model}}(k)\right)^2 + \lambda \cdot \epsilon(k) \tag{5}$$

where the first term accounts for the control effort, with each decision variable $c_i(k)$ incurring the same unit power cost per ±10% increment. The second term penalizes the quadratic temperature tracking error. The slack variable $\epsilon \in \mathbb{R}^+$ is introduced to soften the temperature constraint and is weighted by $\lambda \in \mathbb{R}^+$. The penalty λ must be chosen large enough so that, whenever the original hard-constrained problem is feasible, the solution of the relaxed formulation coincides with that of the hard-constrained one. Note that the inclusion of the slack variable in (5) provides flexibility to the optimization by relaxing the temperature constraint, which is expressed as

$$\left|\Delta T_{\text{targ}} - \Delta T^C_{\text{model}}(k)\right| \le e^{\max}_{\Delta T} + \epsilon(k) \tag{6}$$

where $\epsilon(k)$ acts element-wise on the maximum absolute temperature error allowed, $e^{\max}_{\Delta T} \in \mathbb{R}^P$.

The control objective is to determine the optimal power levels of the upper lamps, i.e., their values resulting from the MPC problem: $C^*(k_{\text{fin}}) = [c_1^*(k_{\text{fin}}), c_2^*(k_{\text{fin}}), \ldots, c_N^*(k_{\text{fin}})]$, such that the resulting temperature map tracks the desired target map, ΔT_{targ}. The simulation length, k_{fin}, is chosen sufficiently long to ensure that the closed-loop system reaches steady state and that the optimal power levels converge.

Therefore, at each discrete-time instant $k = 1, \ldots, k_{\text{fin}}$, the following mixed-integer MPC problem is solved over a prediction horizon N_p:

$$\min_{C,\epsilon} \sum_{n=k}^{k+N_p-1} \Bigg(\phi \sum_{i\in\mathcal{N}} c_i(n|k) + \varphi \left\|\Delta T_{\text{targ}} - \Delta T^C_{\text{model}}(n|k)\right\|_\infty + \lambda \cdot \epsilon(k) \Bigg) \tag{7a}$$

$$\text{s.t.:}\quad c_i \in \{0,1,2,3,4,5\} \quad \forall i \in \mathcal{N} \tag{7b}$$

$$\Delta T^C_{\text{model}}(n+1|k) = \alpha\, \Delta T^C_{\text{model}}(n|k) + \beta\, f_{\text{model}}(C(n|k)) \tag{7c}$$

$$\left|\Delta T_{\text{targ}} - \Delta T^C_{\text{model}}(n|k)\right| \le e^{\max}_{\Delta T} + \epsilon(k) \tag{7d}$$

where the weighting parameters $\phi, \varphi \in \mathbb{R}^+$ were initially selected by normalizing the cost terms and subsequently fine-tuned to achieve a tradeoff between tracking performance and control effort, resulting in $\phi = 1$ and $\varphi = 50$. The weight λ was set to 100, and the maximum absolute error vector was set to $e^{\max}_{\Delta T} = \mathbb{1}_P$, considering a limit of 1 °C on the maximum error allowed. Regarding constraints, (7b) restricts the power

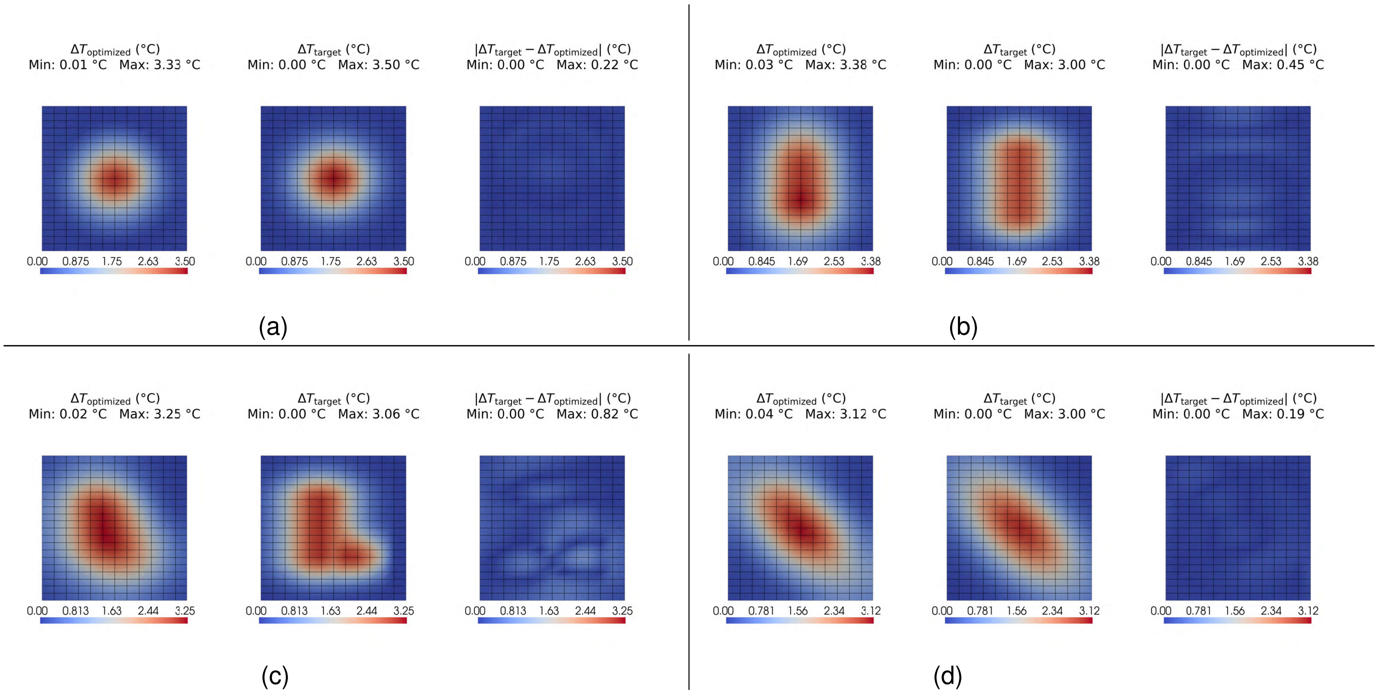


Fig. 5. Graphical results for the considered scenarios using the proposed MPC method. The same color scale is applied across each scenario for consistent comparison. For each scenario (a)–(d), **Left plot:** Temperature map by applying the optimized power level configuration C^*. **Center plot:** Target temperature map. **Right plot:** Absolute error between the optimized and target temperature maps. (a) O-shape scenario. (b) I-shape scenario. (c) L-shape scenario. (d) Diag-shape scenario.

level to allowable integer values, (7c) enforces the dynamic model, and (7d) imposes that the temperature deviation from the target distribution remains below the prescribed limit.

Remark 1: Enforcing (7d) as a soft constraint using the slack variable ϵ avoids infeasibility issues in situations where its strict satisfaction with $\epsilon = 0$ is impossible, e.g., when the available heating power is insufficient to counteract sudden thermal loads or environmental disturbances.

E. Results of Lamp Power Optimization

The optimization problem was solved using the commercial Gurobi solver in Python, which supports mixed-integer programming, on a laptop equipped with an Intel i7-13800H CPU, 32 GB RAM, and running Windows OS. The performed simulations correspond to four different scenarios: O-shape, I-shape, L-shape, and Diag-shape, each defined by the corresponding target temperature map shown in Fig. 4. The simulation length was set to $k_{\text{fin}} = 10$ steps, and the prediction horizon was selected to be $N_{\text{p}} = 2$ steps, as the control objective focuses on achieving the desired steady-state temperature distribution, and longer horizons did not yield noticeable performance improvements but increased computational cost. The average runtime of solving the proposed MPC problem over all the scenarios was 9.67 s. The problem size is fixed by the considered industrial setup and associated simulator. Nevertheless, the observed computation times confirm the practical applicability of the proposed approach for the considered industrial setup. Note that the runtime increases with the number of temperature points P, as each introduces additional linear constraints linking lamp intensities to temperatures, and with the number of controllable lamps N. Therefore, increasing either P or N will lead to an increase in runtime due to the proportional growth in constraints. However, the proposed MPC formulation is primarily intended for offline use, such as initial lamp configuration, rather than in-cycle control. Moreover, alternative implementations (e.g., warm-starting techniques or relaxation with rounding strategies [26]) could be considered to further reduce computational cost.

The total power cost and optimized power level configuration C^* for each scenario are summarized in Table III, while the resulting temperature maps at time instant k_{fin} are illustrated in Fig. 5. A subtle diagonal temperature pattern is visible on the optimized temperature maps, particularly in the L shape of Fig. 5c, which arises from the spatial arrangement of the lamps. Despite this minor effect, the results are satisfactory, and the optimization process successfully achieves the desired temperature profiles with minimal error. Once the lamp power adjustment is completed, the next objective is to develop an online tool for fine-tuning the rest of the recipe parameters.

IV. RECIPE PARAMETERS REFINEMENT

This section presents the second stage of the data-driven method for online control and automatic fine-tuning of the remaining recipe parameters, i.e., pyrometer setpoint, blowing time, vacuum time, etc. We used a dataset collected from an industrial thermoforming machine and developed a tool that analyzes thermal images captured at the end of the heating phase to adjust the recipe in real time. An NN model was first trained to learn the relationship between recipe parameters and thermal images. This modeling approach could be efficiently updated daily using a limited number of new experimental samples, enabling the NN to adapt to gradual changes in operating conditions while preserving predictive accuracy. Afterward, an optimization problem was solved to compute the optimal recipe parameters that best reproduce the target temperature map. The temperature map of the target thermal image, which influences the resulting 3-D shape, was considered "uniform." This choice was driven by limitations in the experimental data rather than by a modeling assumption. However, the approach could be extended to accommodate target temperature maps, such as the ones shown in Fig. 4.

A. Dataset Preparation

The second stage focuses on modeling the heating phase of the thermoforming process using NNs. The model was trained on experimental data provided by a thermoforming machine manufacturer. The dataset consisted of 544 samples, each corresponding to a full thermoforming cycle, and was collected over three working days. Each sample included the recipe parameters and the thermal image delivered at the end of the heating phase. Before training the NN model, we performed data analysis and preprocessing steps to ensure data quality and model suitability.

1) *Recipe Parameters*: The data analysis revealed a large number of input parameters, many of which exhibited no variation across the dataset. These noninformative features were discarded, reducing the set down to 21 key input features. A deeper examination of the input correlation matrix indicated that several features exhibited strong correlations, leading to multicollinearity issues. To overcome these issues, principal component analysis (PCA) was applied. PCA is a dimensionality reduction method that transforms a set of correlated variables into a new set of uncorrelated components, known as principal components, which capture the most important patterns in the data [27]. This transformation not only mitigates multicollinearity but also simplifies the dataset by reducing its dimensionality. To retain 95% of the original variance, five principal components were ultimately selected. We opted for standard PCA given the efficient performance observed, but kernel-based PCA [28] can be alternatively used to capture stronger nonlinearities in the data.

2) *Thermal Images*: To categorize thermal conditions, K-means clustering [29] was applied using the global mean intensity of each thermal image as the feature, computed over height, width, and RGB channels. The resulting 1-D feature vector was clustered into two groups, labeling low-intensity images as "Blue" and the others as "Purple." This revealed a noticeable class imbalance (520 blue vs. 24 purple). To mitigate bias of the model toward the majority class and improve generalization, the blue class was undersampled to 240, retaining all purple images and resulting in a 10:1 ratio. Since many blue images were highly redundant, this step reduced overlap while maintaining sufficient diversity for training, resulting in a final dataset of 264 images. The analysis of the mean pixel intensity over time (see Fig. 6) revealed a recurring daily pattern, with an initial preheating phase followed by stable thermal conditions. To capture this behavior, an additional feature encoding the cycle count within each workday was included among the recipe parameters. Finally, images were converted to RGB, normalized to the $[0, 1]$ range by dividing pixel values by 255, and then downscaled by a factor of 10 to a resolution of 100 × 150 to reduce computational cost. The preprocessed images of size $(100, 150, 3)$ (height, width, RGB channels) preserve the spatial and relative thermal information required for model training.

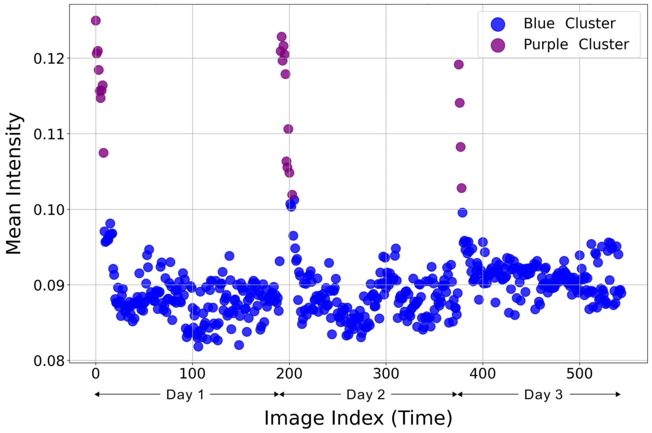


Fig. 6. Temporal trend of mean intensity over sample index.

After these preprocessing steps, the dataset was constructed by pairing the PCA-transformed input features with their corresponding thermal images, which served as the target output for the model. To avoid capturing unintended temporal patterns, the samples were randomly shuffled [30]. The dataset was then divided into 70% for training and 30% for validation during hyperparameter tuning, while maintaining the 10:1 class ratio between blue and purple images to ensure balance. Note that only training and validation splits were used, without a separate test set, due to the limited dataset size.

B. Neural Network Design and Training

For learning the nonlinear mapping between the input parameters and the resulting output thermal distribution, we use a multilayer perceptron (MLP), a fully connected NN composed of input, hidden, and output layers. As summarized in Fig. 7, the input layer receives the five principal components derived from PCA. Then, it is followed by two hidden layers with 128 and 64 neurons, respectively, that use the ReLU activation function

$$f(x) = \max(0, x)$$

to improve computational efficiency, mitigate the vanishing gradient problem, and promote sparsity. Moreover, after each hidden layer, batch normalization [31] and dropout [32] techniques are applied, respectively, to adjust activations and

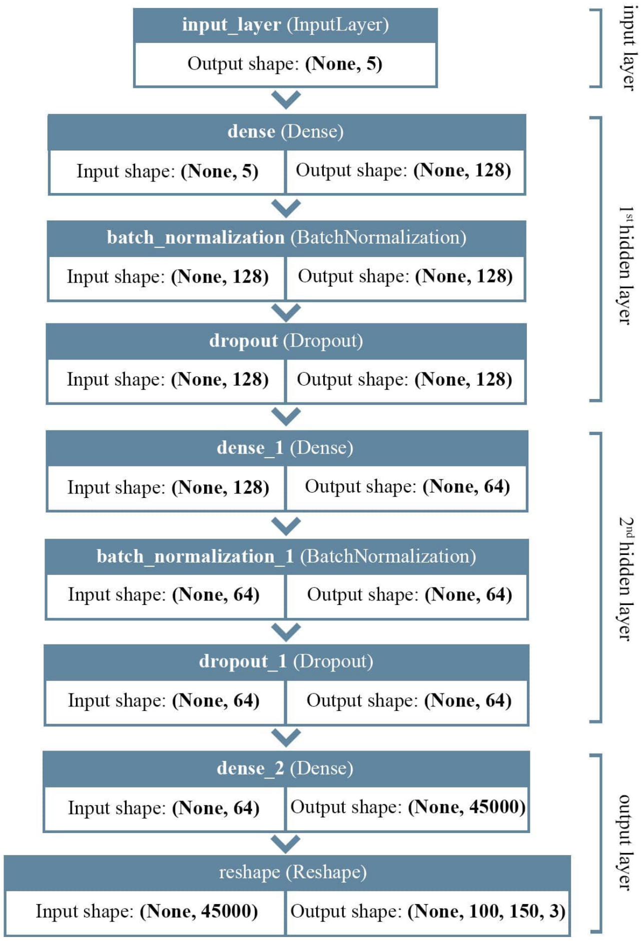


Fig. 7. Scheme of the MLP network architecture.

mitigate data overfitting. The final output layer applies a sigmoid activation function

$$\sigma(x) = \frac{1}{(1 + e^{-x})}$$

that constrains predicted pixel intensities within the range $[0, 1]$, aligning with the normalized thermal image format. Hence, the output is a vector representing the total number of pixels in the processed image. To reconstruct spatial patterns, the RGB output vector is scaled by 255 to reverse the normalization and restore the original intensity range. Then, it is reshaped to the original image dimensions using a scaling factor of 10, allowing for a direct mapping from the input features to the corresponding thermal image.

The hyperparameter tuning was performed using Bayesian optimization, resulting in a well-balanced tradeoff between training and validation loss. The optimal settings included a learning rate of 0.001, a batch size of 16, and a dropout rate of 0.3. The model was trained using the Adam optimizer [33] for 300 epochs. Fig. 8 shows the progression of training and validation loss over the epochs. A steep decline in loss during the initial phase indicates effective learning of the thermal patterns. The close alignment between the training and validation loss curves indicates good generalization, with no evident signs of overfitting. This conclusion is further supported by the fact that both losses remain consistently low throughout the training process (see Fig. 8) despite the absence of a separate test set.

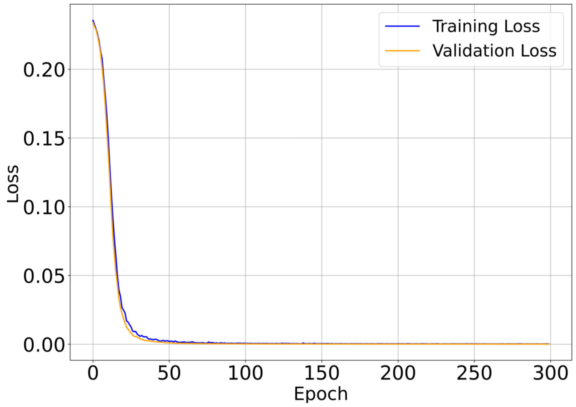


Fig. 8. Training and validation loss curves over 300 epochs.

The evaluation process involved comparing the thermal images predicted by the trained model with the corresponding ground-truth images from the validation set. The effectiveness of the trained NN model for predicting thermal distributions was assessed using the following key performance indicators (KPIs).

1) *Mean Squared Error* (MSE) and *Mean Absolute Error* (MAE) quantify, respectively, the average squared and average absolute differences between predicted pixel intensity $\hat{y}_m$ and actual value y_m in the ground image, for all pixels $m \in \mathcal{M}$

$$\mathrm{MSE} = \frac{1}{M}\sum_{m=1}^{M}(y_m - \hat{y}_m)^2$$

$$\mathrm{MAE} = \frac{1}{M}\sum_{m=1}^{M}|y_m - \hat{y}_m|\,.$$

Lower MSE and MAE values indicate better model prediction accuracy, as they quantify the deviation between predicted and actual pixel intensities.

2) *R2 Score* (R^2), also known as the coefficient of determination, evaluates the proportion of variance in the ground-truth images that is explained by the model predictions. It is calculated as

$$R^2 = 1 - \left(\sum_{m=1}^{M}(y_m - \hat{y}_m)^2 \Big/ \sum_{m=1}^{M}(y_m - \bar{y})^2\right)$$

where $\bar{y}$ is the mean intensity value of all M pixels computed from the ground-truth image. An R^2 value close to 1 indicates a high level of accuracy, while negative values reflect poor predictive performance.

3) *Peak Signal-to-Noise Ratio* (PSNR) measures the perceptual quality of reconstructed images as the ratio

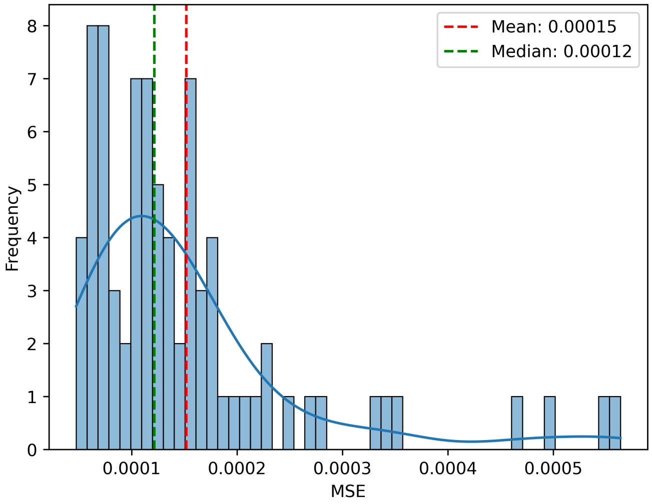


Fig. 9. Distribution of mse on the validation dataset.

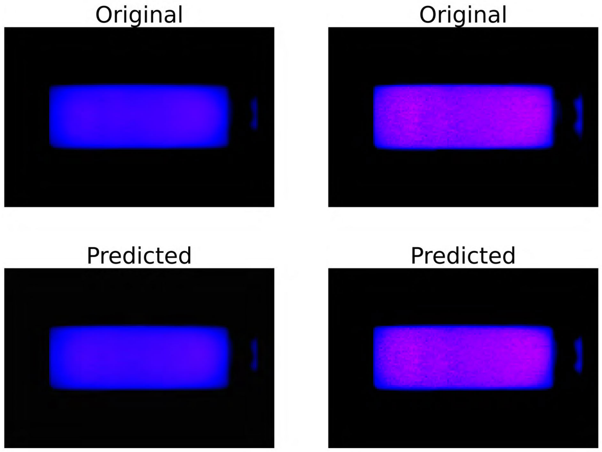


Fig. 10. Comparison between predicted and original thermal images: blue images (left column) and purple images (right column).

TABLE IV
NEURAL NETWORK MODEL ACCURACY

Statistic	MSE	MAE	R^2	PSNR
Mean	0.00015	0.00903	0.99782	38.9561
Median	0.00012	0.00930	0.99823	39.1591

between the maximum possible pixel value ($y_{\max}$) and the root mse. It is computed using the formula

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{y_{\max}}{\sqrt{\text{mse}}}\right).$$

Higher PSNR values correspond to better image quality and lower perceptual differences from the ground truth.

Fig. 9 shows the distribution of the mse value in the validation set, illustrating the frequency of mse scores, with the mean and median values indicated by dashed lines. The remarkably low mse demonstrates the model's accuracy in reconstructing thermal images with minimal deviation from the ground truth. In summary, Table IV displays the mean and median of the four KPIs in the validation set. The low mse and MAE values, an R^2 score close to 1, and a PSNR of 39 approximately, confirm the effectiveness of the model in capturing complex thermal patterns. Finally, a direct visual comparison of the predicted and actual thermal images is also provided in Fig. 10. The predicted images exhibit a strong resemblance to the ground truth, highlighting the model's ability to accurately replicate the spatial distribution of temperature and indicating satisfactory prediction within the considered operating range. More advanced techniques, such as cross-validation or data augmentation, could be considered to further improve model robustness.

C. Optimization of Recipe Parameters

This section addresses an inverse modeling problem aimed at determining the optimal set of recipe parameters that produces a temperature output closely matching a predefined target distribution. To this end, the following cost function is formulated to minimize the discrepancy between the thermal distribution predicted by the NN model and the target distribution

$$J = \left(\mu\left(T_{\text{model}}(x)\right) - \mu\left(T_{\text{targ}}\right)\right)^2 + \left(\delta\left(T_{\text{model}}(x)\right) - \delta\left(T_{\text{targ}}\right)\right)^2 \tag{8}$$

where $x \in \mathbb{R}^{n_x}$, with $n_x = 22$, denotes the vector of recipe parameters to be optimized. The NN model predicts the resulting temperature map $T_{\text{model}}(x)$ from which the mean temperature $\mu(T_{\text{model}}(x))$ and standard deviation $\delta(T_{\text{model}}(x))$ are calculated. The cost function penalizes deviations of these quantities from their respective target values $\mu(T_{\text{targ}})$ and $\delta(T_{\text{targ}})$, promoting both accurate temperature tracking and spatial uniformity of the thermal distribution. Both predicted and target thermal images (see Fig. 10) are processed using identical preprocessing steps before quantitatively evaluating their temperature maps via the cost function, namely.

1) Each thermal image is represented as an RGB array of shape $(100, 150, 3)$, obtained after the downscaling and normalization steps described in Section IV-A. Pixel coordinates are indexed by rows $i = 1, \ldots, 100$ and columns $j = 1, \ldots, 150$.
2) The image is then converted into a single-channel map by computing the channel-wise mean temperature across the three RGB channels: $T(i, j) = 1/3(R(i, j) + G(i, j) + B(i, j))$ for all i and j. This operation is applied to both predicted and ground-truth images.
3) Pixels with temperature values below a fixed threshold of 0.01 are considered nonheated (corresponding to the black background in all images) and are removed from the temperature map, i.e., $T_{\text{clip}}(i, j) = \{T(i, j) \text{ if } T(i, j) \geq 0.01\}$.
4) A fixed cropping mask is applied to isolate the plastic sheet region and remove irrelevant background areas. Specifically, only the first 130 columns are retained, i.e., $T_{\text{crop}} = T_{\text{clip}}[\,:\,, 0 : 130]$, eliminating the colorful background region that appears on the right side of the

plastic sheet (see Fig. 10). The same cropping mask is applied to all images in the dataset.

5) The resulting cropped map T_{crop} is used to compute the mean $\mu(\cdot) = \text{mean}(T_{\text{crop}})$ and the standard deviation $\delta(\cdot) = \text{std}(T_{\text{crop}})$ required by the cost function J.

This procedure ensures that all predicted and target thermal images undergo the same transformation pipeline, enabling the consistent computation of the mean temperature and standard deviation used in the cost function.

Remark 2: For more complex plastic geometries, a more sophisticated cost function may be required. In such cases, in addition to deviations in the mean temperature and standard deviation, a point-wise error metric such as the sum of temperature differences across the entire map [as considered in (5)] can be included. This would allow for more precise alignment between predicted and target distributions at each spatial location, which is essential for accurately reproducing complex thermal patterns.

To ensure realistic and practical parameter adjustments, several constraints are incorporated into the optimization problem. Only the subset of input parameters $x = [r,\ q]$ that directly influence the heating phase can vary freely, as these parameters have the most significant impact on the resulting thermal distribution. In particular, the seven free parameters r_i are bounded within ±20% of their historical mean values $\mu(r_i)$, providing a conservative window that ensures the resulting temperature map remains within safe operating conditions for the material and the process

$$0.8\,\mu\left(r_i\right) \le r_i \le 1.2\,\mu\left(r_i\right), \quad i = 1, 2, \ldots, 7$$

while the other input parameters, q_j, are fixed to their historical values $\mathcal{Q}_j$ to preserve alignment with known feasible process settings

$$q_j \in \mathcal{Q}_j \ , \quad j = 1, 2, \ldots, 15.$$

This strategy guarantees that the optimization explores only realistic and valid parameter configurations, preventing the generation of impractical solutions. Moreover, the model inherently respects hard constraints such as maximum lamp power and maximum sheet temperature. In a potential on-machine implementation, additional failsafes (e.g., real-time monitoring and automatic shutdown triggers) could be added to prevent thermal damage or excessive energy consumption.

The optimization problem is solved using a genetic algorithm (GA), summarized in Algorithm 1. The GA initializes a population of `POP_SIZE` = 70 individuals, combining prior knowledge (one high-quality solution from historical data) with randomly sampled individuals within predefined parameter bounds. Each individual's fitness is evaluated using the cost function (8), which quantifies deviations from the target mean temperature and standard deviation. Parent selection is performed via tournament selection, favoring low-cost solutions while preserving population diversity. Offspring is generated using Blend Crossover (`CXPB` = 0.5), which interpolates between parent genes to explore new regions of the solution space. To avoid premature convergence, Gaussian mutation is applied to the free parameters, and constrained parameters are resampled from historical values with mutation probability `MUTPB` = 0.2. The evolutionary cycle is repeated for `N_GEN` = 100 generations, yielding optimized recipe parameters aligned with the target thermal distribution.

Algorithm 1 Optimization of Recipe Parameters

```
Require: Population size POP_SIZE, mutation probability MUTPB, crossover probability CXPB, maximum number of generations N_GEN.
Ensure: Optimized input parameters minimizing $J$.
Initialize population of size POP_SIZE:
  – Include one high-quality individual from the dataset.
  – Sample free parameters $r$ within $\pm 20\,\%$ of their historical mean values.
  – Sample constrained parameters $q$ from historical data.
for 1 to N_GEN do
    Evaluate fitness of each individual by using the cost function (8).
    Select parent individuals using tournament selection.
    Apply Blend Crossover with probability CXPB.
    Apply mutation:
      – Gaussian mutation on free parameters $r$.
      – Re-sample parameters $q$ with probability MUTPB.
    Replace least fit individuals with the newly generated offspring, ensuring progressive improvement.
end for
return The best-performing individual found after N_GEN generations as the optimized set of recipe parameters.
```

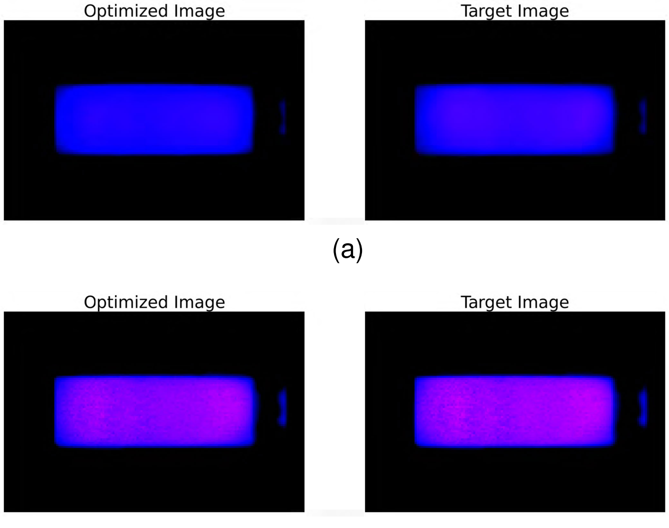
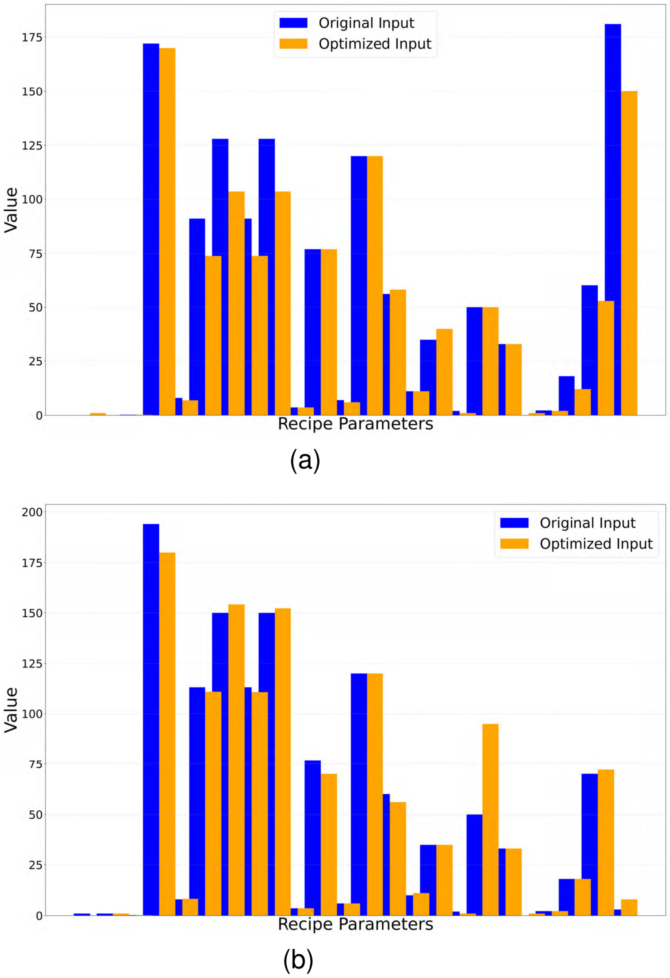

Optimized Image
Target Image
(a)
Optimized Image
Target Image
(b)

Fig. 11. Comparison between target and optimized images. (a) Blue scenario. (b) Purple scenario.

D. Results From Recipe Parameter Tuning

The optimization results are analyzed according to the target for two different heating scenarios.

1) The blue target image is associated with a uniform, medium-temperature distribution across the sheet.



(a)

(b)

Fig. 12. Comparison of original and optimized recipe parameters. Parameter names (x-axis) and units (y-axis) have been omitted for confidentiality reasons. (a) Blue scenario. (b) Purple scenario.

2) The purple target image represents a high-temperature distribution without pronounced spatial patterns.

Fig. 11 illustrates the results for both scenarios, comparing the target thermal image with the corresponding predicted image obtained using the recipe parameters optimized via Algorithm 1. The GA optimization yielded a cost value function of $J = 2.77\text{E} - 4$ for the blue scenario and $J = 3.05\text{E} - 4$ for the purple scenario. Although minor discrepancies remain, the resulting thermal distributions were well preserved, indicating the robustness of the optimization strategy. While additional accuracy could potentially be achieved through extended optimization time or more refined hyperparameter tuning, the method prioritizes computational efficiency, which is an essential requirement in industrial settings. The optimization process required approximately 3 min per run in a Google Colab environment equipped with TPU acceleration. This runtime could be significantly reduced using parallel implementations or deployment on dedicated hardware, further enhancing its suitability for time-sensitive applications.

Additionally, the original recipe parameters are compared with their optimized counterparts in Fig. 12 for the blue and purple scenarios. Due to a lack of access to the physical thermoforming machine, it is currently not possible to perform an on-machine experimental study. Nevertheless, these results illustrate how the proposed method adjusts the recipe parameters with respect to the expert's manual configuration. The GA effectively modifies key variables while maintaining values close to historical configurations, thereby ensuring feasibility within practical process constraints. The proposed data-driven strategy yields a well-initialized, near-optimal set of input parameters and provides a solid foundation for enabling rapid, adaptive decision-making in industrial environments. Furthermore, the proposed NN model could be retrained daily with new production data, enhancing its ability to adapt to and compensate for exogenous disturbances during manufacturing, including equipment aging.

V. CONCLUSION

This article presents a data-driven strategy for efficient tuning of thermoforming parameters, structured into two complementary stages corresponding to lamp configuration and fine-tuning of recipe parameters. The first stage relies on an MPC method to determine optimal lamp power settings for the desired thermoformed shape, offering operators a reliable baseline and reducing setup time. The second stage introduces a NN-based approach to automatically fine-tune the remaining parameters in real time. The proposed dual-model framework, combining a linear lamp model and a nonlinear NN model, leverages experimental and simulation data to capture process dynamics without requiring complex physical modeling. This framework allows for effective optimization of thermal distributions, directly impacting the quality and accuracy of the final thermoformed products.

By combining both stages, the strategy enables efficient setup and real-time adjustments during production, addressing the inefficiencies inherent in traditional trial-and-error parameter tuning. Furthermore, the models' capability for retuning with new production data ensures adaptation to evolving manufacturing environments. Results on simulated and real data demonstrate the potential of the proposed methodology to improve process efficiency and flexibility in industrial thermoforming applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. C. Mantega, his team, and SCM Group S.p.A for providing the dataset and for many interesting discussions on this topic.

REFERENCES

[1] S. Engelmann, *Advanced Thermoforming: Methods, Machines and Materials, Applications, Automation, Sustainability, and the Circular Economy*. Hoboken, NJ, USA: Wiley, 2023.

[2] E. Turan et al., "Digital twin modelling for optimizing the material consumption: A case study on sustainability improvement of thermoforming process," *Sustain. Comput., Informat. Syst.*, vol. 35, Sep. 2022, Art. no. 100655.

[3] K. A. Athanasiou, “Thermoforming process overview,” in *Fundamentals of Plastics Thermoforming*. Cham, Switzerland: Springer, 2009, pp. 17–26.

[4] *Thermoformed Plastics Market. Market Analysis, 2018–2030*. Accessed: Jul. 21, 2024. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/thermoformed-plastics-market

[5] M. N. Subramanian, *Thermoforming: Processing and Technology*. Hoboken, NJ, USA: Wiley, 2024.

[6] *Europe Thermoformed Plastic Market Size*. Accessed: Jul. 21, 2024. [Online]. Available: https://www.grandviewresearch.com/horizon/outlook/thermoformed-plastic-market/europe

[7] T. L. Bergman, *Fundamentals of Heat and Mass Transfer*. Hoboken, NJ, USA: Wiley, 2011.

[8] M. K. Warby, J. R. Whiteman, W.-G. Jiang, P. Warwick, and T. Wright, “Finite element simulation of thermoforming processes for polymer sheets,” *Math. Comput. Simul.*, vol. 61, nos. 3–6, pp. 209–218, Jan. 2003.

[9] A. Ragoubi, G. Ducloud, A. Agazzi, P. Dewailly, and R. L. Goff, “Modeling the thermoforming process of a complex geometry based on a thermo-visco-hyperelastic model,” *J. Manuf. Mater. Process.*, vol. 8, no. 1, p. 33, 2024.

[10] M. Ajersch, “Modeling and real-time control of sheet reheat phase in thermoforming,” M.Sc. thesis, Dept. Elect. Comput. Eng., McGill Univ., Montreal, QC, Canada, 2004.

[11] A. K. Rzepniewski, “Cycle-to-cycle control of multiple input-multiple output manufacturing processes,” Ph.D. dissertation, Massachusetts Inst. of Technol., Cambridge, MA, USA, 2005.

[12] G. Gauthier and B. Boulet, “Terminal iterative learning control design with singular value decomposition decoupling for thermoforming ovens,” in *Proc. Amer. Control Conf.*, Jun. 2009, pp. 1640–1645.

[13] P. Girard, R. di Raddo, V. Thomson, and B. Boulet, “*Advanced in-cycle and cycle-to cycle on-line adaptive control for thermoforming of large thermoplastic sheets*,” SAE Tech. Paper 2005-01-1520, 2005.

[14] F. Erchiqui, “Application of genetic and simulated annealing algorithms for optimization of infrared heating stage in thermoforming process,” *Appl. Thermal Eng.*, vol. 128, pp. 1263–1272, Jan. 2018.

[15] E. F. Camacho, C. Bordons, and J. M. Maestre, *Model Predictive Control* (Advanced Textbooks in Control and Signal Processing). Cham, Switzerland: Springer, 2026.

[16] J. A. Rossiter, *Model-based Predictive Control: A Practical Approach*. Boca Raton, FL, USA: CRC Press, 2017.

[17] E. Masero, J. M. Maestre, A. Ferramosca, M. Francisco, and E. F. Camacho, “Robust coalitional model predictive control with predicted topology transitions,” *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 4, pp. 1869–1880, Dec. 2021.

[18] M. M. I. Chy, B. Boulet, and A. Haidar, “A model predictive controller of plastic sheet temperature for a thermoforming process,” in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 4410–4415.

[19] M. Chy and B. Boulet, “Estimation and control of plastic temperature in heating phase of thermoforming process,” Ph.D. thesis, Dept. Elect. Comput. Eng., McGill Univ., 2014.

[20] P. Tøndel, T. A. Johansen, and A. Bemporad, “An algorithm for multi-parametric quadratic programming and explicit MPC solutions,” *Automatica*, vol. 39, no. 3, pp. 489–497, Mar. 2003.

[21] H. Hosseinionari and R. Seethaler, “The integration of model predictive control and deep reinforcement learning for efficient thermal control in thermoforming processes,” *J. Manuf. Processes*, vol. 115, pp. 82–93, Apr. 2024.

[22] N. Marchal, G. Ducloud, A. Agazzi, and R. Le Goff, “Data-based model applied to thermoforming process control,” *Int. J. Adv. Manuf. Technol.*, vol. 129, nos. 11–12, pp. 5347–5358, Dec. 2023.

[23] E. Masero, W. Zoff, and R. Scattolini, “Data-driven modeling and optimization of the thermoforming heating phase,” *IFAC-PapersOnLine*, vol. 59, no. 26, pp. 109–114, 2025.

[24] *Heating Temperature Guidelines for Forming Boltaron Sheet*. Accessed: Jan. 21, 2026. [Online]. Available: https://www.boltaron.com/thermoforming/heating-forming/

[25] H. Hosseinionari, M. Ramezankhani, R. Seethaler, and A. S. Milani, “Development of a computationally efficient model of the heating phase in thermoforming process based on the experimental radiation pattern of heaters,” *J. Manuf. Mater. Process.*, vol. 7, no. 1, p. 48, Feb. 2023.

[26] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[27] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[28] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *Proc. Int. Conf. Artif. Neural Netw.*, 1997, pp. 583–588.

[29] B. Chong, “K-means clustering algorithm: A brief review,” *Academic J. Comput. & Inf. Sci.*, vol. 4, no. 5, pp. 37–40, 2021.

[30] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” 2012, *arXiv:1206.5533*.

[31] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] D. Kinga, “A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent.*, 2015, vol. 5, no. 6.

Eva Masero (Member, IEEE) received the B.Sc. degree in industrial electronics from the University of Extremadura, Badajoz, Spain, in 2016, the M.Sc. degree in electronics, robotics, and automation engineering from the University of Seville, Seville, Spain, in 2018, and the Ph.D. degree (cum laude) in automation engineering from the University of Seville in 2023.

She worked as a Predoctoral Researcher within the ERC Advanced Grant OCONTSOLAR, where she developed coalitional predictive control strategies for large-scale solar power plants. She was a Visiting Researcher at the University of Sheffield, Sheffield, U.K., in 2021. From 2023, she is an Assistant Professor at the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milan, Italy, where she extends her research toward data-driven control. Her research focuses on model predictive control, with emphasis on cooperative and decentralized architectures. Her interests include multiagent systems, predictive control, and data-driven methods applied to energy systems.

Walter Zoff received the B.Sc. degree in automation engineering and the M.Sc. degree in automation and control engineering from Politecnico di Milan, Milan, Italy, in 2022 and 2025, respectively.

He is currently a Consultant with Reply, Milan, where he works on data analytics and AI-driven applications, including LLM-based systems and retrieval-augmented generation (RAG). His research interests include data-driven modeling, machine learning, neural-network-based control, and the integration of AI techniques in industrial and software systems.

Riccardo Scattolini was born in Milano, Italy, in 1956.

He is a Full Professor of automatic control at the Politecnico di Milano, Milan, Italy. During the academic year 1984/1985, he was a Visiting Researcher at the University of Oxford, Oxford, U.K. He also spent one year in industry working on the simulation and control of chemical plants. He is the author of more than 130 journal papers and over 150 conference papers published in major international venues in control engineering. His research interests include modeling, identification, simulation, diagnosis, and control of industrial and electrical systems, with emphasis on the theory and application of model predictive control and fault detection methods.

Dr. Scattolini received the Quazza Prize and the Heaviside Prize from the Institution of Electrical Engineers, U.K. He has served as an Associate Editor of the IFAC journal *Automatica* and the *International Journal of Adaptive Control and Signal Processing*.

Open Access funding provided by ‘Politecnico di Milano’ within the CRUI CARE Agreement