Predictive Control of Hydronic Floor Heating Systems using Neural Networks and Genetic Algorithms

Vinther, Kasper; Green, Torben; Østergaard, Søren; Bendtsen, Jan Dimon

Published in:
IFAC-PapersOnLine

DOI (link to publication from Publisher):
10.1016/j.ifacol.2017.08.1477

Publication date:
2017

Document Version
Accepted author manuscript, peer reviewed version

Predictive Control of Hydronic Floor Heating Systems using Neural Networks and Genetic Algorithms ⋆

Kasper Vinther * **Torben Green** ** **Søren Ø. Jensen** **
Jan D. Bendtsen *

* *Section of Automation and Control, Aalborg University, Denmark (e-mail: {kv,dimon}@es.aau.dk).*
** *Danish Technological Institute, Denmark (e-mail: {tog,sdj}@teknologisk.dk.)*

Abstract: This paper presents the use a neural network and a micro genetic algorithm to optimize future set-points in existing hydronic floor heating systems for improved energy efficiency. The neural network can be trained to predict the impact of changes in set-points on future room temperatures. Additionally, weather disturbances such as solar heat gain can be anticipated and compensated for, while taking into account the slow dynamics of the floor. Together with a genetic algorithm, they provide a way to search for optimal future set-point sequences, when convexity and continuity in the solution space is not guaranteed. Evaluation of the performance of multiple neural networks is performed, using different levels of information, and optimization results are presented on a detailed house simulation model.

Keywords: Modeling for control optimization, Evolutionary algorithms, Nonlinear predictive control, House modeling, Floor heating.

1. INTRODUCTION

Hydronic floor heating has become a popular source of heat in houses. This can be attributed to the fact that floor heating can be distributed more evenly throughout the room (less cold/hot spots), gives you warm feet, can reduce the required forward temperature (potentially more efficient), is unobtrusive, and is quiet. However, floor heating also has a limitation in reaction time due to slow thermal dynamics of the floor. Furthermore, adjustment of the forward temperature of the water circulated in the floor heating, to match a particular house in terms of size, level of insulation, floor type, etc., can be quite cumbersome. In this respect, is it very important to secure identical pressure losses in the different circuits of an underfloor heating system, as a too high flow in one circuit will lead to the need of an increased forward temperature for the entire system. An additional challenge is to compensate for changes in weather, e.g., seasonal and daily variation in ambient temperature and solar heat gain.

A common approach in commercial solutions, using heat pumps as heating source, is to set the forward temperature based on a measurement of the ambient temperature in a feed-forward manner. A preadjusted heating curve, with a slope and offset calibrated for the particular house, is used to map the two temperatures to provide enough heating in cold weather and to save energy during warmer weather (lower forward temperature gives better heat pump efficiency). However, some conservatism is needed to ensure enough heating in all situations, which leads to sub-optimality in terms of energy efficiency. A potentially more optimal solution is to use the actual heat demand in a feedback control setting. Furthermore, taking future predictions of the weather dependent heat demand into account makes sense, due to the slow dynamics of the floor.

Control oriented predictive models for buildings have received a great deal of attention in the literature, e.g, see (Atam and Helsen, 2016) and references therein. A common grey-box approach is to make a resistive-capacitive thermal network model of the house (Atam and Helsen, 2016; Bacher and Madsen, 2011; Ma et al., 2012). This gives physically meaningful parameters and linear models can easily be derived, e.g., for model predictive control (MPC). However, the hydronic floor heating introduces bilinear terms (Atam and Helsen, 2016; Ma et al., 2012), because the heat input to the floor is a product of mass flow, inlet temperature, and the temperature/state of the floor. For control it is also impractical and expensive to derive resistive-capacitive models for each encountered house. An interesting black-box model alternative is to use artificial neural networks (ANN), which can provide a data-driven nonlinear model of an entire house with floor heating. Additionally, ANNs do not require any a priori knowledge of the particular house in terms of size/orientation/location of room, windows, doors, etc. A comparison with other types of black-box models in (Morel et al., 2001; Salque et al., 2012) also indicate the strength of ANN.

A drawback of ANN is its nonlinear nature, which can be difficult to handle in control and optimization, because convexity of the problem is often not guaranteed. ANNs

⋆ This work was financially supported by the Danish Energy Agency through the EUDP project OpSys (jn:64014-0548) and the Faculty of Engineering and Science at Aalborg University

are used to optimize the operation of a ground source heat pump in (Salque et al., 2012), where brute force computation is used to find an optimal on/off sequence over a future horizon. Optimization of a house heating system is also performed in (Argiriou et al., 2000), with ANNs and a simple rule based controller that determines if the system should be on or off in the next time step. A Newton-Rahpson algorithm and an ANN is used in (Ng et al., 2014) to iteratively find an optimal input sequence for an automotive airconditioning system. However, this requires computation of the Jacobian and Hessian, do not guarantee a global optimum, and do not handle mixed integer problems. A dynamic programming solution is suggested in (Morel et al., 2001), where house heating optimization is divided into easier-to-solve sub-problems. This potentially also leads to sub-optimal solutions, but the implementation called NEUROBAT has undergone commercialization in resent years, which demonstrate the potential of ANNs for predictive control of buildings. The authors in (Chow et al., 2002; Nassif, 2014) suggests using the ANN with a genetic algorithm (GA) to globally solve the optimization of absorption chillers and HVAC systems. Furthermore, a differential evolution (DE) algorithm is suggest in (Harasty et al., 2016) to do predictive control of room temperature in buildings. GAs and DE both belong to a larger class of evolutionary algorithms, which can handle non-convex mixed-integer problems.

In this paper, we investigate the use of a micro GA (μGA), together with ANN prediction, for optimization of room temperature tracking and energy use in houses with floor heating. The μGA, introduced by K. Krishnakumar (Krishnakumar, 1989), does not require tuning of parameters like population size, mutation rate, or crossover probability. The method is more robust towards premature convergence problems encountered in standard big population GA and for some problems it is also quicker, e.g, see diverse examples on controller tuning (Krishnakumar, 1989), seismic modeling (Alvarez, 2002), and building design (Wright and Alajmi, 2005). A multilayer perceptron (MLP) neural network (NN) structure is used for the predictor as done in (Morel et al., 2001; Salque et al., 2012; Argiriou et al., 2000; Ng et al., 2014; Chow et al., 2002; Nassif, 2014; Harasty et al., 2016). Additionally, NNs with increasing number of input/information is compared in terms of prediction accuracy. Finally, a detailed resistice-capacitive house model, with realistic disturbance input, is implemented in the Modelica modeling language for evaluation of the NNs and the optimization algorithm.

The rest of the paper is organized as follows. Section 2 describes the house model with hydronic floor heating loops and the disturbance data used in simulations. Section 3 then provides the analysis and comparison of NN house models. The proposed μGA optimization algorithm is then presented in Section 4 along with discussion of results. Finally, concluding remarks are provided in Section 5.

2. MODEL DESCRIPTION

A realistic simulation environment is required to evaluate different control strategies under the exact same load conditions and disturbances. The simulation model used for evaluation in this paper consists of a resistive-capacitive network house model, a dynamic staggered grid pipe model for the hydronic floor heating, valve and circulation pump models with associated control loops, yearlong weather data, and weekly schedules for occupancy and appliances. Each of these components are described in the following and implemented in the non-proprietary, object-oriented, equation-based modeling language Modelica.

2.1 House Model

A 150 m^2 Danish single family detached house is chosen for the simulation, with materials and parameter values chosen in accordance with the Danish building regulations from 2010 (Hansen, 2012). The layout is illustrated in Fig. 1 and contains four rooms (both small and large) with a height of 2.5 m. Rooms 1, 2, and 4 have wooden flooring and Room 3 has a light concrete floor. The transparent area of the windows (double layered low-E glazing) is equal to 80% of the values provided in the figure. The walls mainly consist of lightweight concrete and insulation, and the ceiling mainly consists of gypsum and insulation.

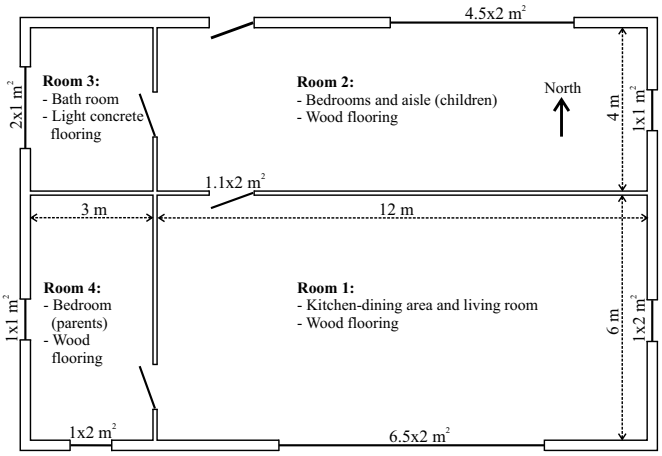

Fig. 1. Illustration of the simulated house with indication of sizes, flooring type, and window/door locations.

The models of walls, windows, floors, and ceiling are based on resistive-capacitive networks (examples of such type of model can also be found in, e.g., (Bacher and Madsen, 2011; Atam and Helsen, 2016)). The floor models have a capacity connected with two resistances in series going from lower to upper floor and then to the room air temperature T_a. Resistances also connect the floor capacity with the hydronic heating pipe water temperature T_w and with the ground temperature T_g. An illustration of the resistive-capacitive network for Room 1, along with all parameter values, is provided in Fig. 2.

Ventilation of air in the house model happens both through infiltration (set to 0.1 l/s/m^2) and mechanical ventilation with heat recovery (set to 0.19 l/s/m^2, assuming 70% heat recovery). The associated heat losses Q_v are distributed among the rooms based on their floor sizes in m^2.

2.2 Pump, Valve, and Pipe Models

Each room has its own hydronic floor heating pipe loop connected to inlet and outlet manifolds, with lengths 360, 240, 40, and 90 m for Room 1-4, respectively. The water

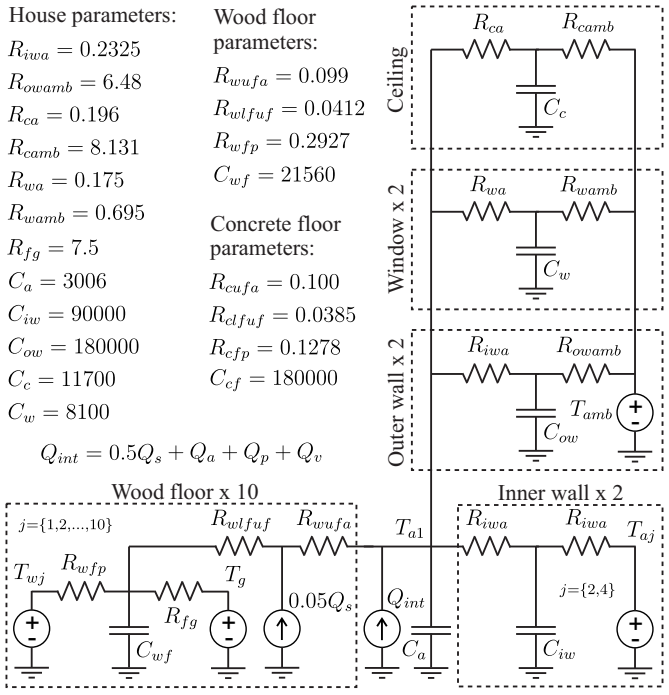

Fig. 2. Resistive-capacitive network model for Room 1, with internal heat gain Q_{int} coming from the sun Q_s, appliances Q_a, people Q_p, and ventilation Q_v. The other rooms have similar models connected together through the inner walls. Thermal resistance is in m^2K/W, capacity is in $J/(m^2K)$, and the floor is divided in 10 to account for the discretization.

flow in each loop is controlled by a manual adjustment valve (to account for different pressure losses in the loops) and an electronic valve that controls the flow to ensure the desired room temperature. Additionally, a pump is installed to ensure that enough hot water is circulated and to obtain a pressure potential in the system. A schematic of the setup with control loops is illustrated in Fig. 3.

The standard Modelica libraries Fluid and Media (Modelica Association, 2016) are used to obtain models for the pump, pipes, and valves and to get media property functions for water and air. A quadratic flow characteristic is chosen for the pump based on empirical data for a Grundfos Magna3 25-60 pump with a nominal rotational speed of 4000 RPM. Quadratic characteristics are also chosen for the valves and the opening degree of the manual adjustment valves are set according to the size in m^2 of each room relative to Room 1. Furthermore, a quadratic turbulent flow model is chosen for the pipes, which are discretized into 10 serially connected segments using a staggered grid approach. For more detail on Fluid library models see (Franke, 2009) and references therein. Finally, the floors are discretized into 10 volumes, without any horizontal heat transfer, and the inner pipe diameters are set to 12 and 16 mm for the wood and concrete floors, respectively.

2.3 Existing Control Loops

PI controllers are used to maintain the desired room temperatures in the house. The temperatures are sampled each minute and the control output is limited to 0-100%

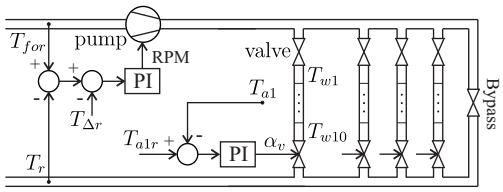

Fig. 3. Floor heating system with existing control loops (temperature control only shown for Room 1). Each of the floor pipes are discretized into 10 segments along the flow direction and the resistive-capacitive floor model separates the water and air temperatures.

valve opening α_v with anti-windup. Furthermore, 2nd order filters with 300 second rise time are applied to α_v to get a realistic slow dynamic response of the valves.

The temperature difference between the inlet/forward and outlet/return flow in the floor heating pipes ($T_{for} - T_r$) is used to control the pump. The desired temperature difference $T_{\Delta r}$ is set to 7°C, and a higher difference (larger cool-down) means that the water loops needs more flow. Additionally, a bypass valve ensures a minimal flow passage in case all loops have closed valves. The pump speed is set in the range 0-5000 RPM, again using PI control with anti-windup, and the temperature difference is sampled each minute.

The heat source in the house is not a part of this study. We will in the following therefore assume that the forward temperature is controlled separately to our desired set-point and fast compared to the rest of the system.

2.4 Disturbance Input Data

Hourly values for global solar radiation Q_{sg} (station 6141 Abed) and ambient temperature T_{amb} (station 6156 Tytofte) are loaded into the simulation based on a design reference year for Denmark (Wang et al., 2012). The solar radiation is used to generate heat inputs for each window in the house, taking both direct and diffuse radiation, the angle to the sun, and the solar heat transmittance for the window into account. The solar heat input is then split in half between the air and floor capacities. Furthermore, a fixed ground temperature $T_g = 10$°C is assumed.

Weekly schedules are generated for the heat input from people Q_p and appliances Q_a based on typical usage of a single family house, with a different schedule during the weekend and an hourly resolution. The schedules are adjusted to give an average heat input of 1.5 W/m^2 from people and 3.5 W/m^2 from appliances (Hansen, 2012). Normally distributed noise is added to each value in the schedules with a standard deviation of 10% of the value.

The inter-zonal air flow through doors V_d (mixing flow) is approximated using an empirical relationship (e.g, also used in DesignBuilder (Design Builder Software Ltd, 2016)) calculated as

$$V_d = A_d \alpha_d V_f, \tag{1}$$

where A_d is the door area (set to 2 m^2), α_d door opening in %, and V_f is the Farea-flow (set to 0.1 $m^3/m^2/s$). Furthermore, a schedule is used to set how the door between room 1 and 2 is open based on occupancy information (50% open from 7 to 21 on weekdays, 50% open from 9

to 21 in the weekend, and 0.25% open in-between due to leakage beneath the door). The two other doors are mostly closed and a fixed opening of 1.25% is used all the time (averaged occasional opening and leakage).

Fig. 4 shows the different sources of heat input to the house model during a week. The average total heat inputs from the sun, occupants and appliances, and floor heating are 594 W, 749 W, and 1842 W, respectively. This gives an average heat loss to the surroundings of 3185 W during the first week of January. Furthermore, note that some days have more sun than others and that each day has a high peak in heat input from appliances during dinner preparation.

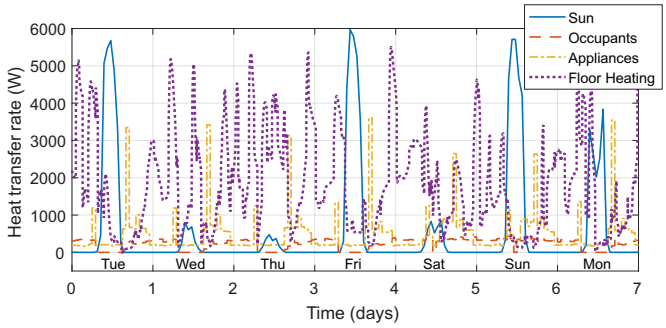

Fig. 4. Different sources of heat input to the house during a seven day period from 1st of January 00:00.

3. NEURAL NETWORK MODELS

3.1 Multi-Layer Perceptron Neural Network Structure

The general fully connected MLP NN structure is illustrated in Fig. 5. The outputs are calculated as

$$\mathbf{y} = \mathbf{w}_2 tansig(\mathbf{w}_1\mathbf{u} + \mathbf{b}_1) + \mathbf{b}_2, \tag{2}$$

where the *tansig* function is a common choice as nonlinear neuron function f, although any smooth function could in theory be used. The NN contains $nm + n + pn + p$ parameters (dimension of weights and biases, see Fig. 5) that needs to be trained, to give a mapping from $\mathbf{u}$ to $\mathbf{y}$ that approximates the measured data well. The Levenberg-Marquardt algorithm with bayesian regularization was chosen for training of the NN, which minimizes a combination of squared errors and weights in order to produce a NN that generalizes well (Matlab function *trainbr*). Note

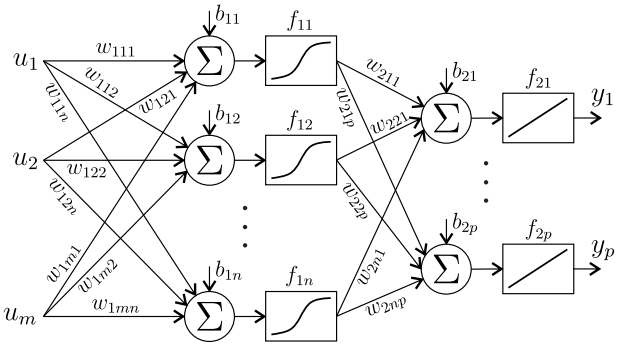

Fig. 5. MLP NN with a nonlinear hidden layer and a linear output layer.

that validation data is not needed during training with this method, and the default stopping criteria was used with a maximum of 500 epochs/repetitions. Additionally, all data provided to the NNs are normalized to the interval -1 to 1. This improves the training and an inverse mapping can then be applied on the output.

3.2 Comparison of Potential Neural Network House Models

The example house introduced in Section 2 has four rooms and the goal is to produce an estimator for the room air temperatures. The output of the NN is thus

$$\hat{\mathbf{y}}(k) = [\hat{T}_{a1}(k+1), \hat{T}_{a2}(k+1), \hat{T}_{a3}(k+1), \hat{T}_{a4}(k+1)]. \tag{3}$$

The time between each sample k was set to 10 minutes, which is slow enough to be able to see changes in air temperature when changes occur in the hydronic floor heating system, yet fast enough to capture the air dynamics. The four first weeks in January are used as training data and the following two weeks are used as test data for comparison. Furthermore, since the training procedure only guarantees to find a local minimum, each tested NN was trained 20 times and the best performing NN on the test data was picked for comparison. Note that Matlab uses the Nguyen-Widrow algorithm to randomly initialize weights and biases.

The network inputs can be split up into three categories, which are disturbance inputs, manipulated inputs, and past values of the output. The inputs that can be manipulated are the same for all tested networks and consist of the forward temperature T_{for} and the room air temperature references $T_{a1r-a4r}$. The entire system with existing control loops is thus considered a black box we would like to optimize through manipulation of set-points. This also means that the optimization can be installed in existing systems as retrofit. Amplitude modulated pseudo random binary sequences (APRBS) were applied to these inputs in order to have enough excitation in the potential input space. The minimum and maximum period time was set to 0.5 and 3 hours, respectively, based on the system dynamics. The forward temperature change was limited to ± 4°C from the nominal value of 35°C and the room temperature references were allowed to change ± 1°C from the user defined setpoint of 22°C (these limits are also used during optimization).

In general, the choice of inputs depend on the available measurements and how strongly they influence the outputs. The final list of inputs for seven selected NNs are summarized in Table 1 and comparison of their performance in terms of one-step-ahead (OSA) and multi-step-ahead (MSA) is shown in Fig. 6.

All the NNs perform better than a simple prediction (baseline), which just use future room temperature set-points as the prediction, e.g., $\hat{T}_{a1}(k+1) = T_{a1r}(k)$. The performance also improve with increasing number of inputs. It can also be concluded that at least four neurons are needed, although having too many neurons actually makes the performance worse. Note that more uncertainty in the results are observed with larger prediction horizon, but the models are still able to predict the general trends.

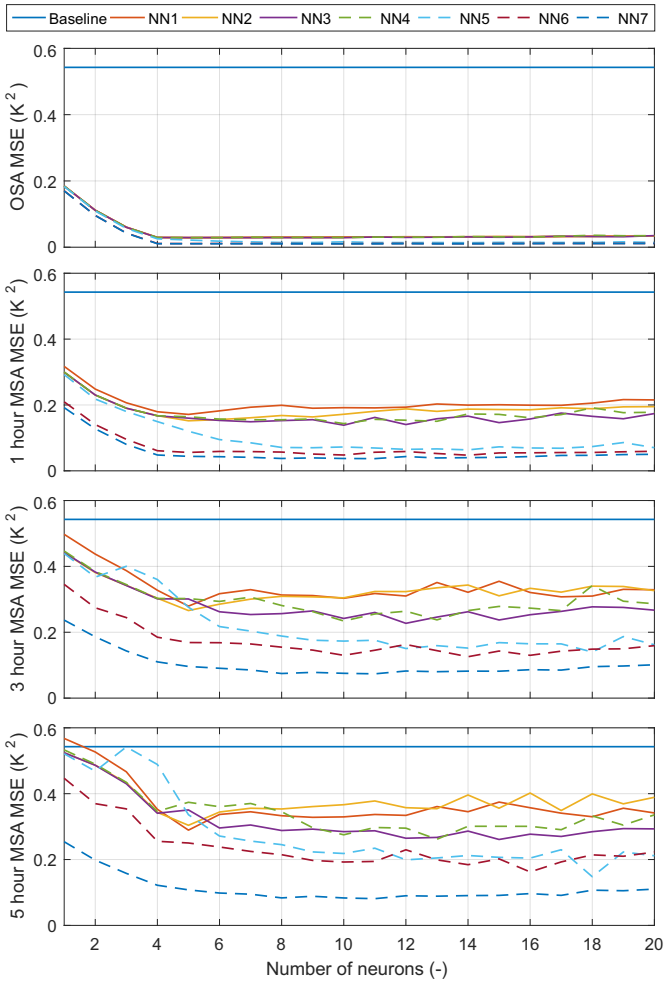

Fig. 6. Test data comparison of the mean square error performance of each NN for different predict ahead lengths and increasing number of neuron functions. The baseline predict room temperatures solely based on future room temperature set-points.

All presented NNs have the output vector two samples back in time as input. Having less gives worse performance and more did not give significant improvements (less parameters also makes it easier to train the NN). Having more input delays than two did not improve the results either and less input delay do not help (NN1 vs NN2).

The number of disturbances we assume available increase with each NN. Most NNs encountered in the literature has ambient temperature T_{amb} and global solar radiation Q_{sg} from weather forecasts as input, which is also common for the selected NNs, except for NN7, which has solar radiation into each room as separate inputs Q_{s1-s4}. NN3 has a time of day signal N_d and NN4 also has a weekday/weekend indicator N_w. The last two signals are replaced by occupancy measurements for each room N_{o1-o4} in NN5 and by direct knowledge of internal heat gains Q_{i1-i4} and ventilation through the door between room 1 and 2 V_{12} in NN6-7 (the other doors are mostly closed).

NN7 has an unrealistic amount of input information, but can be used as a benchmark for performance. It can be seen that information about the solar radiation into each room allows it to outperform NN6, which only has a single global solar radiation input. NN5 perform almost as well as NN6, but only relies on occupancy sensors. If occupancy information is not available then a time of day signal is useful, as illustrated by NN3. The weekday/weekend indicator signal fed to NN4 did not provide any performance improvement for the considered scenario. In general, NN3 seem to be a good candidate for optimization, as it only requires a weather forecast.

Comparison of time series data for NN3 with 12 neurons and NN7 with 8 neurons is shown in Fig. 7. Both NNs predict well for short time horizons and are used in the following.

Table 1. Different combinations of input for seven selected NNs. Note that the input at sample k is taken as the average value between sample k and $k+1$, all networks have the predicted room temperatures $\hat{\mathbf{y}}$ at time $k+1$ as output, and the input vector $\mathbf{y}$ is replaced by $\hat{\mathbf{y}}$ during MSA predict.

Input	NN1	NN2	NN3	NN4	NN5	NN6	NN7
$T_{amb}(k)$	x	x	x	x	x	x	x
$T_{amb}(k-1)$		x	x	x	x	x	x
$Q_{sg}(k)$	x	x	x	x	x	x	
$Q_{sg}(k-1)$		x	x	x	x	x	
$N_d(k)$			x	x			
$N_d(k-1)$			x	x			
$N_w(k)$				x			
$N_w(k-1)$				x			
$N_{o1-04}(k)$					x		
$N_{o1-04}(k-1)$					x		
$Q_{i1-i4}(k)$						x	x
$Q_{i1-i4}(k-1)$						x	x
$V_{12}(k)$						x	x
$V_{12}(k-1)$						x	x
$Q_{s1-s4}(k)$							x
$Q_{s1-s4}(k-1)$							x
$T_{for}(k)$	x	x	x	x	x	x	x
$T_{for}(k-1)$		x	x	x	x	x	x
$T_{a1r-a4r}(k)$	x	x	x	x	x	x	x
$T_{a1r-a4r}(k-1)$		x	x	x	x	x	x
$\mathbf{y}(k-1)$	x	x	x	x	x	x	x
$\mathbf{y}(k-2)$	x	x	x	x	x	x	x
Total inputs	15	22	24	26	30	32	38

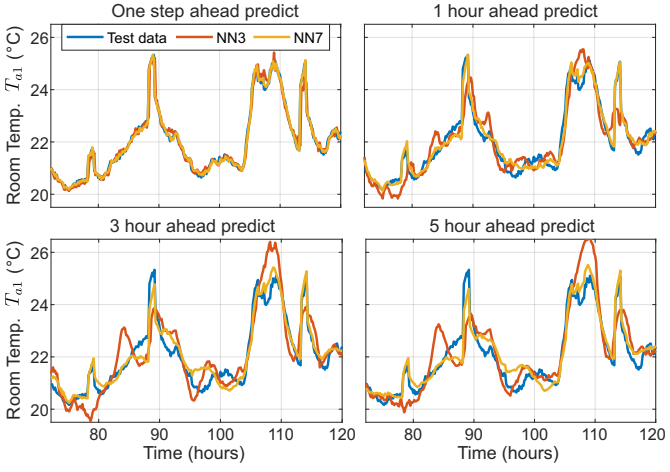

Fig. 7. Time series comparison of selected NNs with test data (only two days is shown). Similar performance is observed for the other three rooms.

4. GENETIC ALGORITHM OPTIMIZATION

4.1 Optimization Problem Formulation

The optimization problem at sample k can be formulated as a constrained minimization problem, i.e., solve

$$\begin{aligned}\underset{\mathbf{u}_r(k)}{\text{minimize}} \sum_{i=0}^{N_h} & (\Delta T_{for}(k+i) \\ & + v(T_{a1}(k+1+i) - \overline{T}_{a1r}(k+1+i))^2 \\ & + v(T_{a2}(k+1+i) - \overline{T}_{a2r}(k+1+i))^2 \\ & + v(T_{a3}(k+1+i) - \overline{T}_{a3r}(k+1+i))^2 \\ & + v(T_{a4}(k+1+i) - \overline{T}_{a4r}(k+1+i))^2), \quad (4)\end{aligned}$$

subject to

$$\hat{\mathbf{y}}(k+i) = NN(\mathbf{u}(k+i)), \quad (5)$$
$$\Delta T_{for,min} \leq \Delta T_{for}(k+i) \leq \Delta T_{for,max}, \quad (6)$$
$$\Delta T_{a1r,min} \leq \Delta T_{a1r}(k+i) \leq \Delta T_{a1r,max}, \quad (7)$$
$$\Delta T_{a2r,min} \leq \Delta T_{a2r}(k+i) \leq \Delta T_{a2r,max}, \quad (8)$$
$$\Delta T_{a3r,min} \leq \Delta T_{a3r}(k+i) \leq \Delta T_{a3r,max}, \quad (9)$$
$$\Delta T_{a4r,min} \leq \Delta T_{a4r}(k+i) \leq \Delta T_{a4r,max}, \quad (10)$$

where N_h is the prediction horizon (set to 29 or 5 hours), v is a positive scalar weight, $\Delta T_{for} = T_{for} - \overline{T}_{for}$ is the deviation from the nominal forward temperature, $\overline{T}_{a1r}$ to $\overline{T}_{a4r}$ denote the user defined room temperature set-points, ΔT_{a1r} to ΔT_{a4r} denote the deviation from the user defined set-points (e.g., $T_{a1r} = \overline{T}_{a1r} + \Delta T_{a1r}$), NN is the NN map including scaling of data, $\mathbf{u}$ is the input vector defined in Table 1, and the optimization variables

$$\begin{aligned}\mathbf{u}_r(k) = [&\Delta T_{for}(k), \Delta T_{for}(k+6), ...\Delta T_{for}(k+24), \\ &\Delta T_{a1r}(k), \Delta T_{a1r}(k+6), ...\Delta T_{a1r}(k+24), \\ &\Delta T_{a2r}(k), \Delta T_{a2r}(k+6), ...\Delta T_{a2r}(k+24), \\ &\Delta T_{a3r}(k), \Delta T_{a3r}(k+6), ...\Delta T_{a3r}(k+24), \\ &\Delta T_{a4r}(k), \Delta T_{a4r}(k+6), ...\Delta T_{a4r}(k+24)], \quad (11)\end{aligned}$$

are the hourly deviations from the nominal set-points five hours ahead (note that $\Delta T_{for}(k)$ is the set-point change applied between sample k to $k+6$, etc.).

The optimization problem is limited to 25 variables (hourly resolution) to reduce the computational demand. Quadratic terms are used on the room temperature in the cost function to penalize large deviations from the set-point. The tuning weight v could be used in a user interface as a knob to either favor comfort or savings. Its value was set to 100 in the optimization presented in the following, placing more emphasis on comfort than economy.

4.2 General Purpose Global Optimization Solver

A μGA is proposed in the following to globally solve the constrained optimization problem formulated in Subsection 4.1, i.e., the μGA functions as a nonlinear optimizer in a NMPC-like fashion. A typical population size of $N = 5$ individuals is used, where each of the individuals represent a possible choice of $\mathbf{u}_r(k)$ in (11), i.e., a potential solution to the optimization problem. A flowchart of the μGA is shown in Fig. 8; it involves the following main operations:

- *Initialization*, where new individuals are picked at random using a normal (Gaussian) distribution with center defined by an elite individual and standard deviation set to be 50% of the distance between min/max bounds for each parameter. The picked value is also truncated to be within the min/max bounds. This provides a way to increase the odds of picking values close to the current best solution and values at the bounds. The best solution from the previous time step can be used as the elite individual to warm-start the search.
- *Automatic restart*, where new individuals are randomly picked as in the initialization (the elite individual is kept unchanged). Restart is triggered when the individuals have "clumped up" (converged) in order to avoid getting stuck in a local optimum.
- *Fitness calculation*, where the fitness of an individual is calculated based on the cost function in (4). This step includes five hour ahead prediction with the NN using an input vector modified by the individual.
- *Deterministic tournament selection*, where two different individuals are picked at random from the population and the most fit is chosen as parent for mating/crossover.
- *Gaussian fitness based box crossover*, where children (new individuals) are produced by randomly picking values based on a Guassian distribution with center closest to the most fit parent and standard deviation set to be 50% of the distance between the parents. This approach is closely related to parent centric normal crossover (Ballester and Carter, 2004) with the exception that child values are restricted to be within the interval defined by the parents, which is similar to non-extended blend crossover (Herrera et al., 2003).

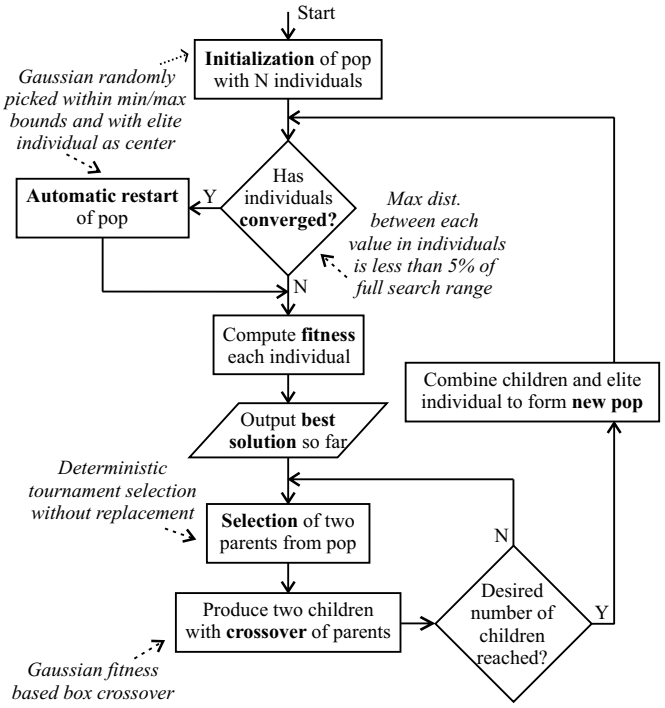

Fig. 8. Flowchart description of the proposed μGA optimization algorithm.

4.3 Optimization Results

The potential of NN3 together with μGA optimization is tested during two weeks in February. The μGA was set to run through 500 generations, which gives a computation time of less than 10 seconds on a standard laptop PC. This is well within the 10 minutes available (potentially also on smaller dedicated microcontrollers), and 1000 generations did not provide any noticeable improvement in performance. The result is compared with three other strategies; just keeping the forward temperature and room air temperature set-points constant, using heating curves to calculate the forward temperature, and optimization based on NN7 instead of NN3. The heating curve sets the forward temperature based on the outdoor ambient temperature and is adjusted to provide 35°C when the ambient is 0°C to provide sufficient heating of all the rooms. Fig. 9 shows time series data from the simulations and Table 2 summarizes key performance metrics.

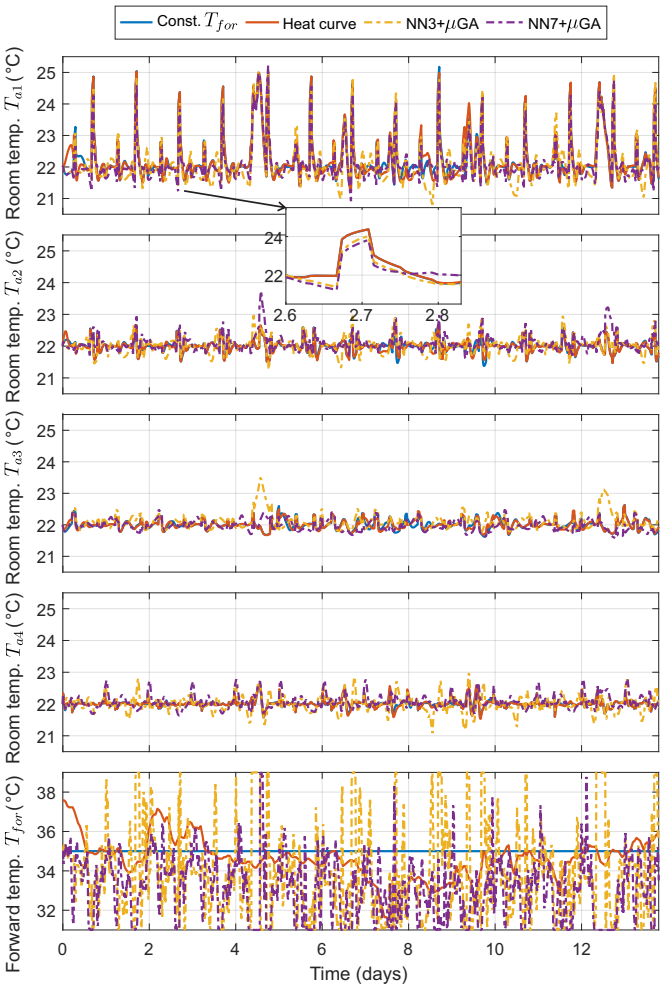

Fig. 9. Comparison of forward and room temperatures using different optimization algorithms. The desired room temperature is 22°C.

All strategies use the same average amount of floor heating Q_f during the test period to keep the room temperatures close to 22°C. The RMSE results for each temperature are similar, but slightly worse for the NN+μGA approaches (depends on the choice of v). However, most of the large peaks in the temperature in Room 1 have been reduced a bit by turning off heating preemptively causing a precooling of the room, which is enabled by the ability of the NN to predict ahead (see zoomed plot on Fig. 9). The peaks are caused by heat input from dinner cooking and/or large solar radiation and will cause the largest thermal discomfort. Furthermore, the forward temperature was lowest with NN+μGA, which can result in operation at better COP values if, e.g., heat pumps are used as energy source. Fig. 10 gives a clearer view of the changes by showing how much heat was required at different forward temperature intervals. NN7 provide the largest shift downwards, but NN3 also has a clear tendency to move consumption down to lower forward temperature, while using much less information than NN7. Using less information in NN3 occasionally gives wrong predictions of future heating needs and sets the forward temperature too high, but this does not seem to hurt the room temperature tracking too much in general.

Table 2. Performance using different optimization algorithms.

Parameter	Const.	Curve	NN3+μGA	NN7+μGA
Mean T_{for} (°C)	35	34.6	33.86	33.08
T_{a1} RMSE (°C)	0.705	0.721	0.721	0.625
T_{a2} RMSE (°C)	0.189	0.197	0.263	0.271
T_{a3} RMSE (°C)	0.155	0.149	0.272	0.172
T_{a4} RMSE (°C)	0.117	0.127	0.246	0.227
Mean Q_f (W)	1875	1876	1876	1875

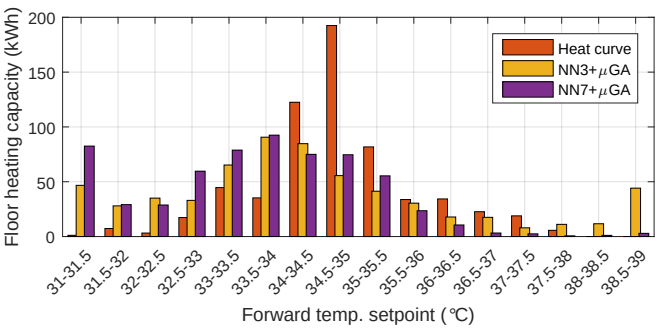

Fig. 10. Total amount of required floor heating capacity at different intervals of forward temperature for each optimization algorithm.

5. CONCLUSION

Multiple NNs has been tested on a detailed simulation model of a single family detached house. The results reveal that at least 4 neurons are needed and between 8-12 neurons seem appropriate, for for a 4 room building, in order to have decent prediction of indoor room temperatures several hours ahead in time. A time of day signal can be an cheap way to get a performance boost and combined with occupancy information they can provide a NN that performs similarly to a NN that has exact knowledge of internal heat gains. Additionally, knowing the solar heat input in each room separately gives the best performance, but this is likely too impractical and expensive to implement. Furthermore, μGA optimization of set-points, using both a simple NN and the best NN,

were able to lower the average forward temperature, while maintaining thermal comfort. This can potentially improve the efficiency of heat pumps. An added benefit of the proposed optimization algorithm is also that there are no tuning parameters, which gives it a high plug and play potential for large scale deployment. Future optimization considerations could include incorporation of a heat pump model and mixed integer optimization with minimization of on/off cycles and energy cost through price signals (can also be solved with μGA).

REFERENCES

Alvarez, G. (2002). Can we make genetic algorithms work in high-dimensionality problems? *Stanford Exploration Project*, 112, 195–212.

Argiriou, A.A., Bellas-Velidisb, I., and Balarasa, C.A. (2000). Development of a neural network heating controller for solar buildings. *Neural Networks*, 13(7), 811–820.

Atam, E. and Helsen, L. (2016). Control-Oriented Thermal Modeling of Multizone Buildings: Methods and Issues. *Control Syst. Mag.*, 36(3), 86–111.

Bacher, P. and Madsen, H. (2011). Identifying suitable models for the heat dynamics of buildings. *Energy and Build.*, 43(7), 1511–1522.

Ballester, P.J. and Carter, J.N. (2004). An effective real-parameter genetic algorithm with parent centric normal crossover for multimodel optimization. In *Proc. GECCO*, 901–913. Seattle, WA, USA.

Chow, T.T., Zhang, G.Q., Lin, Z., and Song, C.L. (2002). Global optimization of absorption chiller system by genetic algorithms and neural network. *Energy and Build.*, 34(1), 103–109.

Design Builder Software Ltd (2016). Designbuilder. `http://www.designbuilder.co.uk/`.

Franke, R. (2009). Standardization of Thermo-Fluid Modeling in Modelica.Fluid. In *7th Int. Modelica Conf.* Como, Italy.

Hansen, E.J. (2012). *Guidelines on Building Regulations 2010*. Statens Byggeforskningsinstitut.

Harasty, S., Lambeck, S., and Cavaterra, A. (2016). Model Predictive Control for Preventive Conservation using Artificial Neural Networks. In *12th REHVA World Congress*. Aalborg, Denmark.

Herrera, F., Lozano, M., and Sanchez, A.M. (2003). A Taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: An Experimental Study. *Int. J. Intell. Syst.*, 18, 309–338.

Krishnakumar, K. (1989). Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization. In *Proc. SPIE 1196, Intell. Control and Adapt. Syst.*, 289–296. Philadelphia, PA, USA.

Ma, Y., Kelman, A., Daly, A., and Borrelli, F. (2012). Predictive Control for Energy Efficient Buildings with Thermal Storage. *Control Syst. Mag.*, 32(1), 44–64.

Modelica Association (2016). Modelica libraries. `https://modelica.org/libraries`.

Morel, N., Bauer, M., El-khoury, M., and Krauss, J. (2001). Neurobat, a Predictive and Adaptive Heating Control System using Artificial Neural Networks. *Solar Energy J.*, 21, 161–201.

Nassif, N. (2014). Modeling and optimization of HVAC systems using artificial neural network and genetic algorithm. *Build. Sim.*, 7(3), 237–245.

Ng, B.C., Darus, I.Z.M., Jamaluddin, H., and Kamar, H.M. (2014). Application of adaptive neural predictive control for an automotive air conditioning system. *Appl. Therm. Eng.*, 73(1), 1244–1254.

Salque, T., Riederer, P., and Marchio, D. (2012). Development of a Neural Network-based Building Model and Application to Geothermal Heat Pumps Predictive Control. In *The 4th Int. Conf. on Advances in Syst. Sim.*, 24–29. Seattle, WA, USA.

Wang, P.R., Scharling, M., and Nielsen, K.P. (2012). 2001-2010 Design Reference Year for Denmark. Tech. rep. 12-17, Danish Meteorological Institute.

Wright, J. and Alajmi, A. (2005). The Robustness of Genetic Algorithms in Solving Unconstrained Building Optimization Problems. In *Proc. Ninth Int. IBPSA Conf.*, 1361–1368.