Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

Computers and Electronics in Agriculture 49 (2005) 330–344

Computers and electronics in agriculture

www.elsevier.com/locate/compag

Greenhouse air temperature predictive control using the particle swarm optimisation algorithm

J.P. Coelho[a], P.B. de Moura Oliveira[b,c], J. Boaventura Cunha[b,c,*]

[a] *Instituto Politécnico de Bragança, Escola Superior de Tecnologia e Gestão, 5301-854 Bragança, Portugal*
[b] *Universidade de Trás-os-Montes e Alto Douro, Dep. Engenharias, 5001-911 Vila Real, Portugal*
[c] *Centro de Estudos e Tecnologias do Ambiente e da Vida da UTAD, 5001-911 Vila Real, Portugal*

Received 16 January 2005; received in revised form 1 August 2005; accepted 2 August 2005

Abstract

The particle swarm optimisation algorithm is proposed as a new method to design a model-based predictive greenhouse air temperature controller subject to restrictions. Its performance is compared with the ones obtained by using genetic and sequential quadratic programming algorithms to solve the constrained optimisation air temperature control problem. Controller outputs are computed in order to optimise future behaviour of the greenhouse environment, regarding set-point tracking and minimisation of the control effort over a prediction horizon of 1 h with 1-min sampling period, for a greenhouse located in the north of Portugal. Since the controller must be able to predict the greenhouse environmental conditions over the specified time interval, it is necessary to use mathematical models that describe the greenhouse climate, as well as to predict the outside weather. These requirements are met by using auto regressive models with exogenous inputs and time series auto-regressive models to simulate the inside and outside climate conditions, respectively.

These models have time variant parameters and so, recursive identification techniques are applied to estimate their values in real-time. The models employ data from the climate inside and outside the greenhouse, as well as from the control inputs. Simulations with the proposed methodology to design the model-based predictive air temperature controller are presented. The results indicate a better efficiency of the particle swarm optimisation algorithm as compared with the efficiencies obtained with a genetic algorithm and a sequential quadratic programming method.

Keywords: Agriculture; Greenhouse climate; Model predictive control; Particle swarm optimisation algorithms

* Corresponding author. Tel.: +351 259350339; fax: +351 259350480.
E-mail address: jboavent@utad.pt (J.B. Cunha).

0168-1699/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2005.08.003

1. Introduction

Greenhouses are building structures that allow the creation of an indoor microclimate for crop development, protecting plants from adverse outdoor conditions. The greenhouse climate can be modified by artificial actuations, such as heating, ventilation and CO_2 supply, in order to provide the best environmental conditions. These modifications must be accomplished by spending additional energy in the production, requiring a regulator that minimises the energy consumption while keeping the state variables as close as possible to the optimum crop physiological reference. The use of model predictive control (MPC) for greenhouse indoor environment control has the advantage of providing the system with the ability to react before any deviations in the controlled variable take place, avoiding delays in the system response (Nielsen and Madsen, 1996). This class of control algorithms must employ models to describe and predict variables required for crop development over a specified time horizon. The MPC operation, within a process with bounded signals, usually involves the solution of a quadratic programming problem. This optimisation procedure is a fundamental part of model-based predictive control. The controller states are obtained by iterative numerical procedures that can be based on deterministic or stochastic algorithms. The optimiser must be able to handle constraints to model physical bounds, such as actuator saturation. Commonly, magnitude and rate constraints for the control actions and level constraints for the outputs are considered. Model predictive control cost functions, when subjected to restrictions, define a complex, non-linear, non-convex search space (Camacho and Bordons, 1994); hence, suitable for evolutionary algorithms optimisation.

This paper reports the use of the particle swarm optimisation algorithm (Kennedy and Eberhart, 1995) to optimise a predictive controller within a greenhouse temperature control application. The results achieved with the proposed methodology are compared with the ones obtained with a genetic algorithm and a sequential quadratic programming method.

2. Methods and theoretical background

2.1. Experimental set-up

This section describes the experimental set-up and the methods used to simulate and control the air temperature inside a greenhouse located on the UTAD University Campus in the north of Portugal.

The greenhouse had a floor area of 210 m^2 and was covered with a 200 μm polyethylene film. To control its climate, several actuators and sensors were installed and connected to an acquisition and control system based on a personal computer and a data acquisition card (PCL-818, Advantech). This device had 16 analogue channels with 12-bit resolution, 31 digital input/output channels, one 16-bit counter, one analogue output and a real-time clock.

The weather and the greenhouse climate physical variables were acquired and/or controlled, with a sampling interval of 1 min. Inside and outside air and soil temperatures, relative humidities and solar radiation were measured using, respectively: PT100 sensors, capacitor sensors (F22H, Rotronic) and low-cost optoelectronic sensors (TSL230, Texas Instruments). Inside air CO_2 concentration was measured with an infrared gas analyser

(GMP111, Vaisala) and the wind speed was measured with an anemometer (PV01-02, Electromatic).

Outside sensors were located 2 m above the soil. Inside the greenhouse, the air environmental data were measured at the central point 1 m above the soil, and soil temperature was measured at a depth of 5 cm. The data-acquisition system resolutions, including the sensors, were ±0.2 °C, ±2%, ±0.2 m/s and ±10 W/m^2 for temperatures, relative humidities, wind speed and solar radiation, respectively. The installed actuators were a ventilator with a flow rate of 38,000 m^3/h, a shadow/thermal screen, a gas heating system with a heating power of 100,416 kJ/h, CO$_2$ injectors and an irrigation system.

Inside air temperature set-points, $w(k)$, were constant during the daytime period and computed on-line for the night period. The desired air temperature for daytime conditions was 24 °C. For the night period, the desired air temperature depended on the outside air temperature, T_o, and wind speed, W_s, in order to reduce the heating system energy consumption. These values are computed each minute using Eq. (1),

$$w(k) + 0.1\,W_s(k) = \begin{cases} 12\,(^\circ\mathrm{C}); & T_o(k) < 0\,^\circ\mathrm{C} \\ 0.5T_o(k) + 12\,(^\circ\mathrm{C}); & 0\,^\circ\mathrm{C} \le T_o(k) \le 8\,^\circ\mathrm{C} \\ 16\,(^\circ\mathrm{C}); & T_o(k) > 8\,^\circ\mathrm{C} \end{cases} \tag{1}$$

where $W_s(k)$ and $T_o(k)$ are the wind speed and outside air temperature measured at sample k.

The effective set-points used by the controller were obtained as the mean value of the last 10 calculations, which prevented high-frequency fluctuations in the air temperature caused by fast load variations in the wind speed or the transitions between night and day periods.

2.2. Air temperature model

To design model-based control algorithms to compute suitable control inputs in such a way that a cost performance function is optimised, it is necessary to use software models to compare different scenario outcomes. This requirement implies the use of dynamic models to predict the inside greenhouse climate, as well as the weather, in order to infer the optimal solution of a set of control sequences.

The dynamic changes in the greenhouse are determined by differences in energy and mass contents between inside and outside air, from exogenous energy as solar radiation or outdoor temperature and through the control actions taken. The greenhouse air energy balance is affected by energy supply and energy losses. The former is due to an artificial heating system and heat load imposed by the sun and the latter is due to transmission through the greenhouse cover and forced ventilation. Other energy and mass transport phenomena, for instance at the greenhouse soil, are neglected due to their small contribution to the overall air temperature dynamics.

Assuming that the greenhouse climate can be described by a linear system around an operating point, the greenhouse air temperature model is described by a parametric equation with exogenous inputs, u (Ljung, 1987),

$$T_i(kT) \cdot A(q^{-1}) = u(kT - nT) \cdot B(q^{-1}) + \varepsilon(kT) \tag{2}$$

where T_i is the air temperature, T the sampling interval, q^{-1} the backward-shift operator, n the number of delays from input to output, u the input signals vector and A and B are polynomials in q^{-1}.

To select a model, a set of possible models, with different orders (degrees of polynomials A and B) and delays were first tested. There are several criteria that can be employed for this purpose, but the one most commonly used is Aikaike's Information Theoretic Criterion (AIC) (Akaike, 1974).

More explicitly, to simulate and predict the air temperature, T_i, the model described by Eq. (3) is used,

$$T_\mathrm{i}(kT) = \frac{[b_1 q^{-1}\ b_2 q^{-1}\ b_3 q^{-1}\ b_4 q^{-1}\ b_5]}{1 + a q^{-1}} \cdot \begin{bmatrix} \Delta T_{\mathrm{i,o}}(kT) \\ R_\mathrm{o}(kT) \\ \Delta T_{\mathrm{p,i}}(kT) \cdot H(kT) \\ \Delta T_{\mathrm{i,o}}(kT) \cdot V(kT) \\ 1 \end{bmatrix} \tag{3}$$

where a denotes the transfer function denominator parameters, b_i the coefficients of the B_i polynomials in the delay operator, $\Delta T_{\mathrm{i,o}}(kT) = T_\mathrm{i}(kT) - T_\mathrm{o}(kT)$ the difference between inside (T_i) and outside (T_o) air temperatures, $\Delta T_{\mathrm{p,i}}(kT) = T_\mathrm{p}(kT) - T_\mathrm{i}(kT)$ is the difference between the heating pipe water (T_p) and inside air (T_i) temperatures, R_o is the outside solar radiation, V and H the ventilation and heating control input signals, respectively, and the constant input 1 relates to the offset component.

The model parameters a, b_1, b_2, b_3 and b_4 represent the partial contributions of each physical variable in the overall greenhouse air temperature and b_5 is the contribution of the continuous component. Since the model parameters are time varying (Boaventura Cunha et al., 2000), recursive identification techniques must be employed to estimate their values.

There are several methods that can be used to estimate the time-varying parameters of polynomials A and B (Åström and Wittenmark, 1995). For the case where the real parameters are slow time variant, as in this type of process, a recursive least-squares algorithm with exponential forgetting can be used (Boaventura Cunha et al., 1997). For the system described in Eq. (3), with delay $n = 1$, the model output, $\hat{T}_\mathrm{i}(k)$, can be expressed as:

$$\hat{T}_\mathrm{i}(k) = \varphi^\mathrm{T}(k)\theta + \varepsilon(k) \tag{4}$$

where: $\varphi(k)$ is the data or regression vector and θ is the vector that contains the estimated parameters,

$$\varphi(k) = \begin{bmatrix} T_\mathrm{i}(k-1) \\ \Delta T_{\mathrm{i,o}}(k-1) \\ R_\mathrm{o}(k-1) \\ \Delta T_{\mathrm{p,i}} \cdot H(k-1) \\ \Delta T_{\mathrm{i,o}} \cdot V(k-1) \\ 1 \end{bmatrix} \tag{5}$$

$$\theta = [a\ b_1\ b_2\ b_3\ b_4\ b_5]^\mathrm{T} \tag{6}$$

The recursive computations of the vector parameters can be realized using Eqs. (7.1)–(7.3) known as Recursive Least-Squares algorithm (RLS),

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)[y(k) - \varphi^{\mathrm{T}}(k)\hat{\theta}(k-1)] \tag{7.1}$$

$$K(k) = P(k-1)\varphi(k)[I + \varphi^{\mathrm{T}}(k)P(k-1)\varphi(k)]^{-1} \tag{7.2}$$

$$P(k) = [I - K(k)\varphi^{\mathrm{T}}(k)]P(k-1) \tag{7.3}$$

where $K(k)$ denotes a gain matrix and $P(k)$ is the covariance matrix of the estimated parameters.

This RLS algorithm assumes that the process model parameters are constant. Since the real processes have time-varying parameters, the least-squares method could lead to inadequate estimations. To cope with time-variant cases, some adjustment mechanisms must be introduced in the previous basic equations. Several solutions to this problem have been proposed (Åström and Wittenmark, 1995; Ljung, 1987). When the real parameters have abrupt changes, the covariance matrix P must be periodically reset to αI, with α being a large number. If the real parameters are slowly time-variant, an RLS with exponential forgetting can be used. In the RLS with exponential forgetting algorithm, the least-squares criterion is given by,

$$V(\theta, k) = \sum_{j=1}^{k} \lambda^{k-i}(T_i(j) - \varphi^{\mathrm{T}}(j)\theta(j-1))^2 \tag{8}$$

where $0 < \lambda < 1$, designates the forgetting factor. This factor implies a time varying weighting of the data.

This method has the main disadvantage that when the input is not persistent, and as the old data is discarded in the estimation procedure, the matrix P increases exponentially with rate λ, which in this case was chosen as 0.9. This phenomenon is called estimator windup.

To solve the problem of estimator windup, an estimator that forgets the information only in the directions in which new information is gathered was implemented. For the estimator described by Eqs (9.1)–(9.8), the estimations converge avoiding large changes in the parameters. The parameters that minimise the cost function in Eq. (8) are obtained recursively from Parkum (1992):

$$\alpha(k) = P(k)\varphi(k+1) \tag{9.1}$$

$$\beta(k) = \varphi^{\mathrm{T}}(k+1)\alpha(k) \tag{9.2}$$

$$\begin{cases} \xi(k) = \dfrac{\beta(k)}{\lambda\beta(k) - (1-\lambda)} & , \beta(k) > 0 \\ \xi(k) = 1 & , \beta(k) \le 0 \end{cases} \tag{9.3}$$

$$\delta(k) = \frac{\alpha(k)}{\xi(k) + \beta(k)} \tag{9.4}$$

$$K(k) = \delta(k)\varphi^{\mathrm{T}}(k+1) \tag{9.5}$$

$$P(k+1) = [I - K(k)]P(k)[I - K^{\mathrm{T}}(k)] + \delta(k)\delta^{\mathrm{T}}(k) \tag{9.6}$$

$$\varepsilon(k+1) = y(k+1) - \varphi^{\mathrm{T}}(k+1)\cdot\hat{\theta}(k) \tag{9.7}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + P(k+1)\varphi(k+1)\cdot\varepsilon(k+1) \tag{9.8}$$

This method can be combined with other methods and techniques aiming to achieve a robust estimation algorithm. A simple way to improve the robustness of this method is to substitute for the prediction error used in the estimator law, $\varepsilon(k+1) = T_{\mathrm{i}}(k+1) - \varphi^{\mathrm{T}}(k+1)\theta(k)$, with a function, $f(\varepsilon(k))$, in order to decrease the effects of infrequent large noise signals, such as in the case of intermittent sensor failures, over the computed parameters. In this way, the parameter adaptation law can be replaced by (Åström and Wittenmark, 1995),

$$\theta(k+1) = \theta(k) + P(k+1)\cdot\varphi(k+1)\cdot f(\varepsilon(k+1)) \tag{10}$$

where $f(\varepsilon(k+1))$ is a linear function for small prediction errors but increases slowly for large ones,

$$f(\varepsilon(k+1)) = \frac{\varepsilon(k+1)}{1 + d|\varepsilon(k+1)|} \tag{11}$$

in which d is a positive constant.

Figs. 1–3 show the parameters computed on-line for the day 5 November of 2000. These parameters, computed using Eqs. (9)–(11), together with the inside and outside greenhouse climate data, were used to compare the described temperature adaptive control algorithms, regarding both set point accuracy and control effort. These figures allow the visualisation of the model time-varying parameters, estimated with a time step of 1 min. Detailed information about the greenhouse climate model and the estimation techniques used to compute their values can be found in Boaventura Cunha et al. (1997).

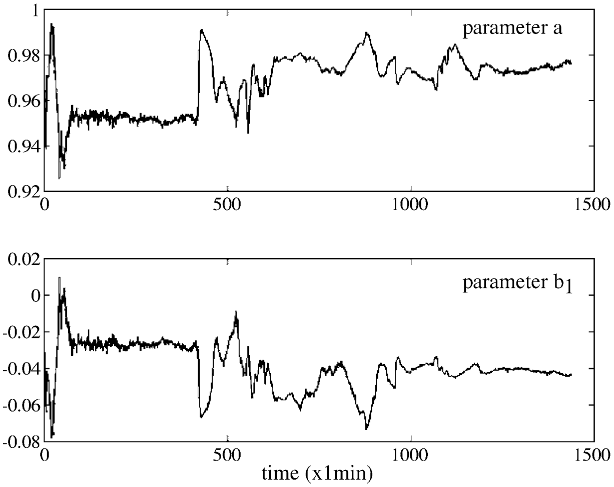

Fig. 1. Evolution of the computed parameters a and b_1 of the air temperature model Eq. (3).

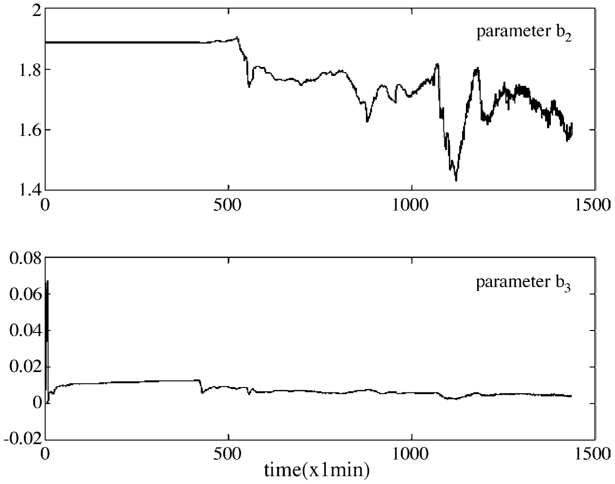

Fig. 2. Evolution of the computed parameters b_2 and b_3 of the air temperature model Eq. (3).

2.3. Model predictive control

Model Predictive Control (Clarke et al., 1987) comprises a collection of control methods having in common that the controller is based on the future predictions of the system behaviour using a mathematical model of the plant. There are several predictive control algorithms based on process models. These algorithms differ from each other only in the system or disturbances model's structures and in the objective function to be minimised (Camacho and Bordons, 1994), and have been applied to a wide set of control engineering

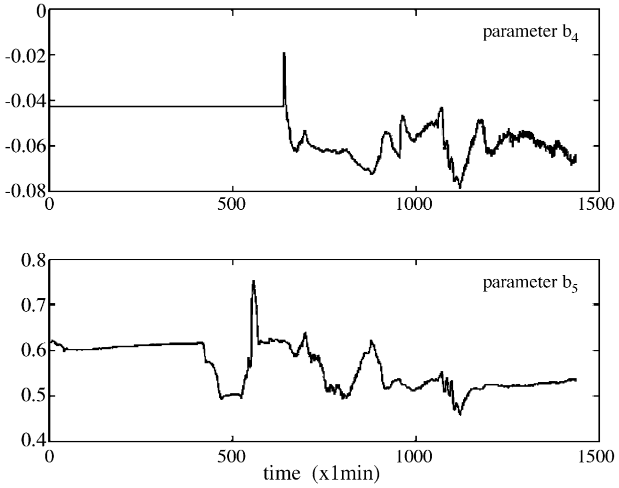

Fig. 3. Evolution of the computed parameters b_4 and b_5 of the air temperature model Eq. (3).

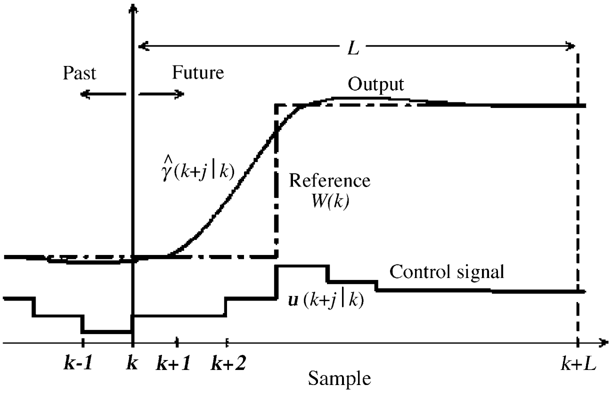

Fig. 4. Basic strategy of a model-based predictive controller. Reference signal, w, control signal, u, and output signal, $\hat{y}$, vs. number of samples.

applications including greenhouse environmental control (Pinon et al., 2000; Kyriannakis et al., 2002; El Ghoumari et al., 2002).

The performance of MPC depends largely on the accuracy of the process model. This performance increases as the process-model mismatch decreases. The estimated model must be as simple as possible and capable of describing the system dynamics in a way to predict, with some precision, future outputs. So, a large part of the design effort is related to system modelling and identification. MPC involves the computation of a sequence of future control values for which it is expected that the system output tracks a given input reference. The methodology underlying these types of controllers is characterized by the strategy illustrated in Fig. 4 (Camacho and Bordons, 1994).

Future outputs for a prediction horizon L are computed for each sample k using the process model. The predicted output $\hat{y}(k+j|k)$ for $j=1, \ldots, L$ is computed using past inputs and outputs as well as the computed future values of the control signal. The collection of future control signals is computed by optimising a predefined criterion in order to maintain the process output as close as possible to the reference $w(k)$. This criterion normally takes the form of a quadratic function of the error between the predicted output and the set point. In most cases, the control effort is included in the objective function in order to avoid abrupt changes in the control action.

Future control actions are computed optimising a specified cost function governed by the following expressions:

$$J = \lambda_1 \sum_{j=h_{\min}}^{h_{\max}} [\varepsilon(k+j|k)]^2 + \lambda_2 \sum_{j=1}^{h_c} [\Delta u(k+j-1)]^2 \tag{12}$$

$$\varepsilon(k+j|k) = \hat{y}(k+j|k) - w(k+j) \tag{13}$$

where $\varepsilon(k+j|k)$ is the prediction error between the future trajectory and expected output, $\Delta u(k+j-1)$ represents the control effort, λ_1 and λ_2 the weighing factors for each component, $h_{\max}$ and $h_{\min}$ represent the maximum and minimum prediction horizon and h_c is the control horizon. Constants $h_{\max}$ and $h_{\min}$ represent the instant limits in which it is desirable that the output follows the reference.

The reference trajectory $w(k+j)$ is sometimes different from the real reference (Clarke et al., 1987). Normally, a soft approximation from the actual value of the output towards the known reference is considered. This approach avoids abrupt changes in the control action by means of less aggressive responses. The shaped reference $w(k+j)$ is often approximated by using a first-order lag model as described by Eqs. (14) and (15).

$$w(k) = y(k) \tag{14}$$

$$w(k+j) = \alpha \cdot w(k+j-1) + (1-\alpha) \cdot r(k+j) \tag{15}$$

with $\alpha \in [0, 1]$, $j = 1, 2, \ldots$, and $r(k)$ denoting the real reference signal.

2.4. The particle swarm optimisation algorithm

Kennedy and Eberhart (1995) proposed the Particle Swarm Optimisation (PSO) algorithm, conceptually based on the social behaviour of groups of organisms, such as herds, schools and flocks. As an evolutionary technique, the PSO is a population-based algorithm, formed by a set of particles, representing potential solutions for a given problem. Each particle moves through a n-dimensional search space, with an associated position vector $X_i(t) = \{x_{i1}(t), x_{i2}(t),\ldots, x_{in}(t)\}$ and velocity vector $V_i(t) = \{v_{i1}(t), v_{i2}(t), \ldots, v_{in}(t)\}$ for the current evolutionary iteration t.

The original PSO model integrates two types of knowledge acquisition by a particle: through its own experience and from social sharing from other population members. The former was termed cognition-only model and the latter was termed social-only model (Kennedy, 1997). The behaviour of each particle is based on these two types of knowledge and their current position regarding the search. The particle behaviour is modelled by using the following equations:

$$\mathrm{co}_{id}(t) = p_{id}(t) - x_{id}(t) \tag{16}$$

$$\mathrm{so}_{id}(t) = p_{gd}(t) - x_{id}(t) \tag{17}$$

$$v_{id}(t+1) = v_{id}(t) + 2\varphi_1 \cdot \mathrm{co}_{id}(t) + 2\varphi_2 \cdot \mathrm{so}_{id}(t) \tag{18}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{19}$$

in which d represents the dimension index, $1 \leq d \leq n$, $p_{id}(t)$ represents the best previous position of particle i in the current iteration t, $p_{gd}(t)$ represents the global best position in the current iteration for a pre-defined neighbourhood type, and the terms co, so, v denote cognition-only, social-only and velocity. Parameter φ_1 is known as the cognitive constant and φ_2 is known as the social constant, which represent uniformly distributed random numbers generated in a pre-defined interval.

An additional parameter was incorporated into Eq. (18) by Shi and Eberhart (1999), resulting in the relation (20):

$$v_{id}(t+1) = \omega(t) \cdot v_{id}(t) + \varphi_1 \cdot \mathrm{co}_{id}(t) + \varphi_2 \cdot \mathrm{so}_{id}(t) \tag{20}$$

in which ω represents the inertia weight. The value given to the inertia weight will affect the type of search in the following way: a large ω will direct the PSO for a global search while a small ω will direct the PSO for a local search. This parameter can vary linearly from a larger value to a smaller value in order to make the search global in the early run and local in the end of the run. Constants φ_1, φ_2 and ω can be interpreted as the confidence that each particle has in the current position, its own experience and its neighbours experience, respectively.

The neighbourhood can be of different size and topology. Each particle can take into account either:

(i) The social information from a list of particles pre-defined in the beginning of the simulated evolution. The list can incorporate all the population individuals, with an individual being able to use the best solution found by every other member. This full-connected social network structure was termed *Star*. In other list definitions, an individual uses only k adjacent neighbours organised in a *Circle* or *Wheel* topology (Kennedy et al., 2001).

(ii) The physical information which considers distance between neighbouring individuals, evaluated using some metric definition.

The velocity is limited by a maximum, $V_{\max}$, meaning the maximum jump that each particle can make in one iteration. The selected value for $V_{\max}$ should not be too high to avoid oscillations, or too low to avoid search traps. The inertia weight and maximum velocity parameters selection in the PSO algorithm was studied and reported by Shi and Eberhart (1998). Each particle position should also be located within its dynamic range $[X_{\min}, X_{\max}]$.

The PSO algorithm has been applied successfully to solve several optimisation problems (Kennedy et al., 2001), including control tuning and optimisation (De Moura Oliveira et al., 2002; Coelho et al., 2002a,b).

2.5. *Methodology of the optimisation application*

The problem addressed in this report is to control the air temperature within a greenhouse using a MPC strategy. The quadratic programming (QP) problem underlying this type of controller was solved iteratively using the PSO algorithm and the results were compared with the ones obtained by using genetic algorithms (GA) and sequential quadratic programming (SQP) (Biggs, 1975) methods.

Besides the modelling technique described in Section 2.2, the MPC requires models to predict the outside climate during the prediction horizon. Several approaches were successfully used with this propose, such as the lazy men's prediction (Van Straten, 1996). Here, auto-regressive models, described by Eq. (21), are applied to predict the outside air temperature and solar radiation (Coelho et al., 2002a,b),

$$y_{\mathrm{TS}}(kT)\cdot A_{\mathrm{TS}}(q^{-1}) = \xi(kT) \tag{21}$$

in which A_{TS} is a fourth-order polynomial in q^{-1}, y_{TS} the time series to be modelled and ξ is the model error.

Table 1
Particle swarm algorithm settings

Population size	30
φ_1	N (0.0, 2.0)
φ_2	N (0.0, 2.0)
ω	Linear decay (1.4, 0.7)
Coding scheme	Floating point

To use evolutionary algorithms as design tools within the predictive control framework, it is necessary to modify them accordingly. The prediction steps are represented by population members that correspond to genes and space coordinates in the GA and PSO algorithms, respectively. Thus, control actions Δu to be applied to the system in a specified future time are encoded into corresponding data structures that form the population. In each generation/epoch the best two solutions found are shifted one position toward the present instant and introduced in the population of the next generation. The size of the population must be related to the size of the search space, ensuring a sufficient number of points for the evolutionary algorithm prospect. In the present case, a population of size $n = 30$ was selected in order to reduce the computational time. The best population/swarm solution was used as a stop criterion; i.e., the search algorithm stops if its convergence rate does not change in 30 generations/epochs.

The quadratic programming problem with linear restrictions was solved using the PSO, GA and SQP for indoor greenhouse temperature control using the described MPC strategy. Tuning parameters for both algorithms (Tables 1 and 2) are typical values used in these types of optimisation algorithms. Further study involving different parameters is underway, which will be reported in future publications, in which a comparison will be made with some results published by (Krink et al., 2001; Ursem et al., 2002; Ursem, 2003). For control purposes, the objective function to be minimized is described by:

$$\Gamma(u) = \lambda_1 \cdot \sum_{j=1}^{60} \varepsilon(k+j)^2 + \lambda_2 \cdot \sum_{j=1}^{60} \Delta u(k+j)^2 \qquad (22)$$

The cost function factors, λ_1 and λ_2, define the set-point accuracy and control effort relative weights, respectively, and are set to $\lambda_1 = 0.6$ and $\lambda_2 = 1$. The performance of each optimisation algorithm was analysed in three different aspects: set-point accuracy (23), energy consumption ((24) and (25)), for ventilation and heating signals, respectively, and the normalized computing time (26) spent by each algorithm in relation to the maximum

Table 2
Genetic algorithm settings

Population size	30
Mutation probability	0.1
Crossover probability	0.75
Selection strategy	Tournament
Coding scheme	Floating point
Elitism	Yes

computation time required by the least efficient algorithm. Also, the standard deviations of the control signals ($\sigma_{\triangle H}$ and $\sigma_{\triangle V}$) were computed in order to evaluate the control effort for each case.

$$\mathrm{SPe} = \sum_{k=1}^{N} [T_i(k) - w(k)]^2 \tag{23}$$

$$E_V = \sum_{k=1}^{N} V(k) \tag{24}$$

$$E_H = \sum_{k=1}^{N} H(k) \tag{25}$$

$$C_\mathrm{T} = \frac{\text{elapsed time}_{\mathrm{GA,PSO,SQP}}}{\max(\text{elapsed time}_{\mathrm{GA,PSO,SQP}})} \tag{26}$$

3. Results and discussion

Simulations using two data sets, for the 5th of November of 2000 and the time period from the 1st of December to the 12th December of 2000, are presented to illustrate the performances of the PSO, GA and SQP algorithms regarding set-point accuracy, energy consumption of the ventilation and heating systems, actuator effort (standard deviations of the heating and ventilation control signals, $\sigma_{\triangle H}$ and $\sigma_{\triangle V}$) and computation time.

Table 3
Simulation results using PSO, GA and SQP for the 5th day of November, 2000

	SQP	PSO	GA
SPe	23.01	4.3	6.1
E_H	344.4	351.8	356.1
E_V	252.8	225.8	225.9
$\sigma_{\triangle H}$	0.055	0.053	0.055
$\sigma_{\triangle V}$	0.078	0.070	0.071
C_T (%)	26	74	100

Table 4
Simulation results using PSO, GA and SQP for the time period from 1 December to 12 December, 2000

	SQP	PSO	GA
SPe	248.3	56.4	77.2
E_H	4216.4	4278.1	4314.3
E_V	2160.1	2029.7	2011.4
$\sigma_{\triangle H}$	0.061	0.059	0.060
$\sigma_{\triangle V}$	0.072	0.065	0.066
C_T (%)	30	82	100

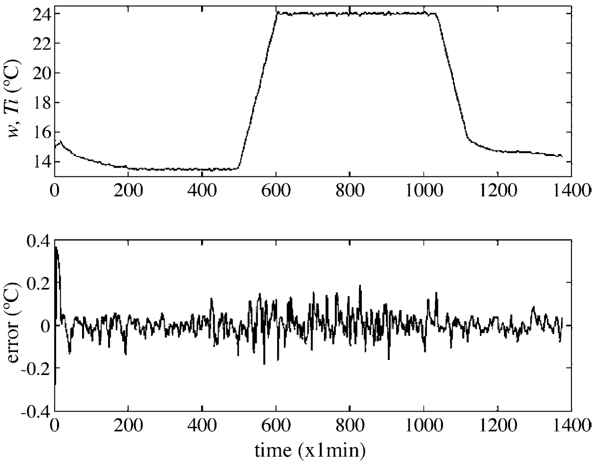

Fig. 5. Air temperature set-point tracking results using the PSO algorithm.

Tables 3 and 4 show the results obtained with PSO, GA and SQP algorithms for the criteria defined by equations (23)–(26) using the data sets mentioned. Tables 3 and 4 simulation results, indicate that the PSO algorithm was able to reduce the set-point tracking error in approximately 30% (27%) and 81% (77%) relative to the set-point accuracies achieved with the genetic and SQP algorithms, respectively. Simultaneously, the PSO algorithm was able to decrease the variations in the heating and ventilation control signals, reducing the effort over the actuators. Regarding the energy consumption computed with the three methods, it can be concluded from Tables 3 and 4 that they are very similar for the heating and ventilation

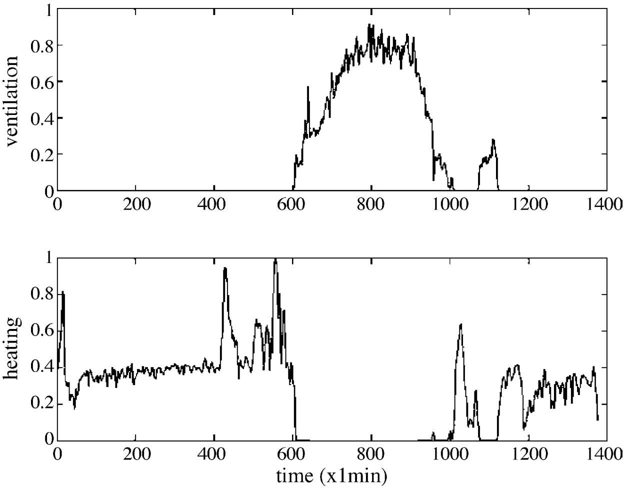

Fig. 6. Ventilation and heating control signals computed with the PSO algorithm. These signals are scaled between 0 and 1, corresponding to actuator power inputs range, from 0 to 100% of their maximum power.

actions. The major drawback of the PSO algorithm is due to the large computation run time, about 2.8 times greater than that required by the SQP method.

Due to the difficulty in visualizing the simulation curves for the largest data set, Figs. 5 and 6 show the simulated set-point tracking and actuating signals computed with the PSO algorithm only for the 5th of November, 2000.

Fig. 5 indicates a close match between the system response (T_i) and set-point (w) due to the time scale used, as can been seen from the error plot clearly indicating error amplitudes comprised between $\pm 0.2\,^{\circ}$C. The heating and ventilation control signals are shown in Fig. 6, scaled between 0 and 1, corresponding to actuator power inputs ranging from 0 to 100% of their maximum power. It can be seen from the air temperature and the controller output curves (Figs. 5 and 6), that the PSO multivariable predictive controller has a good performance in respect to set-point accuracy, energy saving and reduction of control effort.

4. Conclusions and further work

The particle swarm optimisation algorithm was proposed as an alternative method to the design of a greenhouse air temperature model predictive controller subject to constraints. The controller outputs are computed to optimise the greenhouse environment future behaviour, regarding air temperature set-point tracking and control effort reduction. The prediction horizon considered was of 1 h with a 1-min sampling period. From the simulation results, we can conclude that the PSO algorithm, using a small population size, was able to significantly reduce the set-point tracking error using a lower control effort. As an evolutionary inspired optimisation algorithm, the PSO requires a larger computation run time than a classical optimisation technique, which still restricts its use in real-time applications. However, future research involving the use of micro-sized populations and parallel computational means will be directed to real-time greenhouse environmental control. Multiple tests were also conducted using data sets for other periods during the year, and the results are in agreement with those presented in this report. Moreover, the proposed techniques will be implemented and tested not only for greenhouse temperature control, but also for air humidity and CO_2 concentration control.

References

Akaike, H., 1974. A new look at the statistical model identification. IEEE Trans. Automation Control, AC-19, 716–723.

Åström, K.J., Wittenmark, B., 1995. Adaptive Control, second ed. Reading MA, Addison-Wesley.

Biggs, M.C., 1975. Constrained minimization using recursive quadratic programming. in: L.C.W. Dixon, G.P. Szergo (Eds.), Towards Global Optimisation. North-Holland, pp. 341–349.

Boaventura Cunha, J., Couto, C., Ruano, A.E.B., 1997. Real-time parameter estimation of dynamic temperature models for greenhouse environmental control. Control Eng. Pract., Journal Code: 123, 5/10, 1473–1481.

Boaventura Cunha, J., Couto, C., Ruano, A.E.B., 2000. A greenhouse climate multivariable predictive controller. Acta Horticult. N. 534, ISHS, 269–276.

Camacho, E.F., Bordons, C., 1994. Model Predictive Control in the Process Industry. Springer-Verlag, Sevilla.

Clarke, D.W., Mohtadi, C., Tuffs, P.S., 1987. Generalized predictive control—part I. The basic algorithm. Automatica 23 (2), 137–148.

Coelho, J.P., Boaventura Cunha, J., de Moura Oliveira, P.B., 2002a. Solar radiation prediction methods applied to improve greenhouse climate control. In: World Congress of Computers in Agriculture and Natural Resources, pp. 154–161.

Coelho, P.C., de Moura Oliveira, P.B., Boaventura Cunha, J., 2002b. Greenhouse Air Temperature Control using the Particle Swarm Optimisation Algorithm. In: Camacho, E.F., Basanez, L., de la Puente, J.A. (Eds.), Preprints of 15th IFAC World Congress, 21–26 July. Barcelona, Spain, p. 65 (Abstract).

De Moura Oliveira, P.B., Boaventura Cunha, J., Coelho, J.P., 2002. Design of PID controllers using the particle swarm algorithm. In: 23rd IASTED International Conference in Modelling, Identification and Control (MIC 2002), 18–21 February, Innsbruck, Austria, pp. 263–268.

El Ghoumari, M.Y., Tantau, H.J., Megas, D., Serrano, J., 2002. Real-time non linear constrained model predictive control of a greenhouse. In: Camacho, E.F., Basanez, L., de la Puente, J.A. (Eds.), Proceedings of the 15th IFAC World Congress, Barcelona. Barcelona Spain.

Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia). IEEE Service Center, vol. 4, Piscataway, NJ, pp. 1942–1948.

Kennedy, J., 1997. The particle swarm: social adaptation of knowledge. In: Bäck, Thomas, Michalewicz, Zbigniew, Yao, Xin (Eds.), Proceedings of the 1997 IEEE International Conference on Evolutionary Computation. Indianapolis, US, pp. 303–308.

Kennedy, J., Eberhart, R.C., Shi, Y., 2001. Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco, pp. 344–345.

Kyriannakis, E., Arvanitis, K.G., Sigrimis, N.A., 2002. On-Line improvement for the decentralized predictive control of the heat dynamics of a greenhouse. In: Camacho, E.F., Basanez, L., de la Puente, J.A. (Eds.), 15th IFAC Triennial World Congress on Automatic Control, July. Barcelona, Spain, pp. 21–26.

Krink, T., Ursem, R.K., Filipic, B., 2001. Evolutionary algorithms in control optimisation: the greenhouse problem. In: Beyer, H., Cantu-Paz, E., Goldberg, D., Parmee, Spector, L., Whitley, D. (Eds.), Proceedings of the Third Genetic and Evolutionary Computation Conference (GECCO-2001), pp. 440–447.

Ljung, L., 1987. System Identification—Theory for the User. Prentice-Hall, Englewood Cliffs, NJ.

Nielsen, B., Madsen, H., 1996. Predictive control of air temperature in greenhouses. In: IFAC 13th Triennial World Congress, San Francisco, pp. 399–404.

Parkum J., 1992. Recursive identification of time-varying systems. Ph.D. Thesis. Technical University of Denmark, Lyngby, Denmark.

Pinon, S., Pena, M., Schgurensky, C., Kuchen, B., 2000. Constrained predictive control of a greenhouse. In: Proceedings of International Conference on Modelling and Control in Agriculture, Horticulture and Post-Harvest Processing (Agricontrol 2000), July 10–12, Wageningen, The Netherlands, pp. 190–195.

Shi, Y., Eberhart, R.C., 1998. Parameter Selection in Particle Swarm Optimisation, Evolutionary Programming VII. Lecture Notes in Computer Science 1447. Springer-Verlag, pp. 591–600.

Shi, Y., Eberhart, R.C., 1999. Empirical study of particle swarm optimization. In: Proceedings of the 1999 IEEE Congress of Evolutionary Computation, vol. 3, pp. 1945–1950.

Ursem, R.K., Filipic, B., Krink, T., 2002. Exploring the performance of an evolutionary algorithm for greenhouse control. J. Comput. Inf. Technol. 10 (3), 195–201.

Ursem, R.K., 2003. Models for evolutionary algorithms and their applications in system identification and control optimization. Ph.D. Thesis. Department of Computer Science, University of Aarhus, Denmark.

Van Straten, G., 1996. On-line optimal control of greenhouse crop cultivation. Acta Horticult. (ISHS) 406, 203–212.