GREENHOUSE AIR TEMPERATURE 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 de Trás-os-Montes e Alto Douro para a Investigação e Desenvolvimento
Agro-Industrial, 5000 Vila Real, Portugal.
b. Universidade de Trás-os-Montes e Alto Douro, Dep. Engenharias
5001-911 Vila Real, Portugal.
c. Instituto de Ciências e Tecnologias Agrárias e Agro-Alimentares da UTAD
5001-911 Vila Real, Portugal

Abstract: The particle swarm optimisation algorithm is proposed as a new method to design a model based predictive controller subject to restrictions. Its performance is compared with the one obtained by using a genetic algorithm for the environmental temperature control of a greenhouse. 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 one hour with a one-minute sampling period. *Copyright © 2002 IFAC*

Keywords: Model predictive control, agriculture, particle swarm optimisation algorithms.

1. INTRODUCTION

Greenhouses are building structures that allow the creation of an indoor microclimate for crop development, protecting it from adverse outdoor conditions. Moreover this microclimate can be modified by artificial actuations such as heating, ventilation and CO_2 supply in order to provide the best environmental conditions. This non-natural environmental conditions are achieved by additional energy spend 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 controllers (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 the evolution of the 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 are considered for the control actions and level constraints are considered for the outputs. Model predictive control cost functions, when subjected to restrictions, defines a very complex, non-linear, non-convex search space, hence suitable for evolutionary algorithms optimisation.

2. MODEL PREDICTIVE CONTROL OVERVIEW

Model Predictive Control (Clarke, *et al.*, 1987), comprise a collection of control methods having in common the fact 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 structure and on the objective function to be minimised (Camacho and Bordons, 1994).

The performance of MPC depends largely on the accuracy of the process model. This performance increases as process-model mismatch decreases. The estimated model must be as simplest 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 type of controllers is characterized by the strategy illustrated in figure 1 (Camacho and Bordons, 1994).

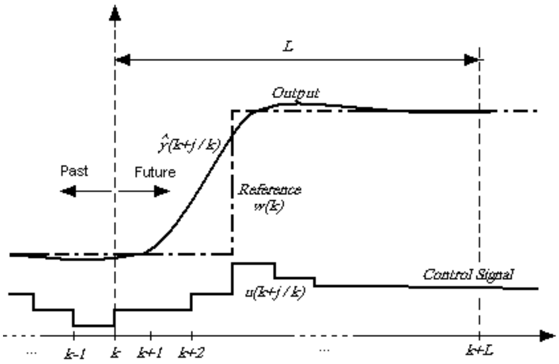

Fig. 1. Basic strategy of a model based predictive controller.

Future outputs for a horizon *L* (prediction horizon), are predicted for each sample *k* using the process model. The predicted output *y(k+j/k)* for *j=1,...,L* is based on past inputs and outputs as well as future values of the control signal. The collection of the future control signals are 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:

$$J = \lambda_1 \sum_{j=a}^{b} [\varepsilon(k+j \mid k)]^2 + \lambda_2 \sum_{j=1}^{c} [\Delta u(k+j-1)]^2 \qquad (1)$$

$$\varepsilon(k+j \mid k) = \hat{y}(k+j \mid k) - w(k+j) \qquad (2)$$

where *ε(k+j|k)* is the prediction error between the future trajectory and output, *Δu(k+j-1)* represents the control effort, λ_1 and λ_2 are weights for each component, *a* and *b* represent the maximum and minimum prediction horizon, and *c* characterize the control horizon. Constants *a* and *b* 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 equations (3) to (4).

$$w(k) = y(k) \qquad (3)$$

$$w(k+j) = \alpha \cdot w(k+j-1) + (1-\alpha) \cdot r(k+j) \qquad (4)$$

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

3. 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, which represent a potential solution for a given problem. Each particle moves through a *n*-dimensional search space (as birds in a flock), with an associated position vector $X_i(t)=\{x_{i1}(t),x_{i2}(t),...,x_{in}(t)\}$ and velocity vector $V_i(t)=\{v_{i1}(t),v_{i2}(t),...,v_{in}(t)\}$ for the current evolutionary iteration *t*.

The original PSO model integrates two types of knowledge acquisition by a particle: through it's own experience and from social sharing from other population members. The former was termed cognition-only model and the latest 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. Kennedy modelled particle behaviour by using the following equations:

$$co_{id}(t) = \left(p_{id}(t) - x_{id}(t)\right) \qquad (5)$$

$$so_{id}(t) = \left(p_{gd}(t) - x_{id}(t)\right) \qquad (6)$$

$$v_{id}(t+1) = v_{id}(t) + \varphi_1 . co_{id}(t) + \varphi_2 . so_{id}(t) \qquad (7)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \qquad (8)$$

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

An additional parameter was incorporated into equation (7) (Shi and Eberhart, 1999) resulting in equation (9):

$$v_{id}(t+1)=\omega(t).v_{id}(t)+\varphi_1.co_{id}(t)+\varphi_2.so_{id}(t) \qquad (9)$$

in which $\omega(t)$ 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. The 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* and *Wheel* topologies (Kennedy and Eberhart, 2001).
(ii) the physical information which considers distance between neighbour individual evaluated using some metric definition.

A simplified version of equation (7) was proposed by Clerc and Kennedy (2000) by considering $\varphi_1=\varphi_2=\varphi$ and defining an intermediate position p_{ig} in between the best previous position p_i and the global best p_g defined by equation (10):

$$p_{igd}(t)=p_{igd}(t)\frac{p_{id}(t)-p_{gd}(t)}{2} \qquad (10)$$

resulting in a modified velocity governed by equation (11):

$$v_{id}(t+1)=v_{id}(t)+\varphi.\left(p_{igd}(t)-x_{id}(t)\right) \qquad (11)$$

Clerc (1999) proposed the use of a constriction coefficient χ that is incorporated in the simplified velocity equation by:

$$v_{id}(t+1)=\chi\left(v_{id}(t)+\varphi.\left(p_{igd}(t)-x_{id}(t)\right)\right) \qquad (12)$$

with $\varphi \ge 4$. The constriction coefficient can be evaluated by using the following equation:

$$\chi=\frac{2k}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|} \quad 0 \le k \le 1 \qquad (13)$$

The effect of this coefficient is to promote convergence over time. Parameters k=1 and φ=4.1 are suggested (Kennedy and Eberhart, 2001) as good values to use. Another version of the constriction method results in the following modification of velocity equation (9):

$$v_{id}(t+1)=\chi.\left[\omega(t).v_{id}(t)+\varphi_1.co_{id}(t)+\varphi_2.so_{id}(t)\right] \qquad (14)$$

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 $[X_{min},X_{max}]$.

4. PROBLEM STATEMENT

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 is solved iteratively using the PSO algorithm and the results compared with the one obtained by using a genetic algorithm (GA).

If a MPC control strategy is to be included within a greenhouse, it is essential to have dynamic models that describe the greenhouse crop production process evolution as well as the control and exogenous inputs. The dynamic changes in the greenhouse are determined by differences in energy and mass contents between the inside and outside air, from exogenous energy as solar radiation or outdoor temperature and through the control actions taken. The energy balance of the greenhouse air 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 latest due to transmission through greenhouse cover and forced ventilation. Other energy and mass transport phenomena, for instance at the greenhouse soil are neglected due to its unimportant contribution to the overall air temperature.

Assuming that the greenhouse climate can be described by a linear system around an operating point, the greenhouse air temperature model will be described by the following first order auto-regressive parametric equation with exogenous inputs.

$$T_i(kT) = \frac{\left[\beta_1 \cdot q^{-1} \quad \beta_2 \cdot q^{-1} \quad \beta_3 \cdot q^{-1} \quad \beta_4 \cdot q^{-1}\right]}{1+\alpha \cdot q^{-1}} \cdot \begin{bmatrix} T_o(kT) \\ R_o(kT) \\ V(kT) \\ H(kT) \end{bmatrix} \quad (15)$$

where T is the sampling interval (in this case 1 minute), q^{-1} is the backward shift operator, T_i and T_o indoor and outdoor temperatures, R_o the outdoor solar radiation and V and H the artificial ventilation and heating. The model parameters α, β_1, β_2, β_3 and β_4 represent the partial contributions of each physical variable in the overall greenhouse air temperature. Since the model parameters are time varying (Boaventura Cunha *et al.*, 1998), recursive identification techniques associated with the U-D factorisation algorithm (Åström and Wittenmark, 1989) were applied to estimate their values.

Auto-regressive models, described generically by equation (16), were applied to describe the outside air temperature and solar radiation [9].

$$y_{TS}(kT) \cdot A(q^{-1}) = \xi(kT) \quad (16)$$

In which A is a 4th order polynomial in q^{-1} and y_{TS} is the time series to be modelled.

The meteorological data used was acquired with a sampling period of one minute in a greenhouse located at the Universidade de Trás-os-Montes e Alto Douro in the North of Portugal. The air temperature control in that particular greenhouse is accomplished by using two actuators, a ventilator with a flow rate of 38000 m^3/h and a gas heating system with a heating power of 100416 KJ/h.

In order 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 $n=100$ was found to be suitable. The convergence rate of the first gene/coordinate was used as a stop criterion. The search algorithm stops if the convergence rate does not change in 30 generations/epochs.

5. SIMULATION RESULTS

In this section, simulated results obtained for indoor greenhouse temperature control using a MPC strategy are reported. The quadratic programming problem with linear restrictions is solved using the PSO and a GA. Tuning parameters for both algorithms are described in tables 1 and 2.

Table 1 Particle swarm algorithm settings

Population size	100
φ_1	N(0.0, 2.0)
φ_2	N(0.0, 2.0)
ω	Linear decay
Coding scheme	Float

Table 2 Genetic algorithm settings

Population size	100
Mutation probability	0.1
Crossover probability	0.75
Selection strategy	Tournament
Coding scheme	Float
Elitism	Yes

For control purposes, the objective function to be minimized is described by:

$$\Gamma(u) = \lambda_1 \cdot \sum_{j=1}^{60} \varepsilon(k+j)^2 + \lambda_2 \cdot \sum_{j=1}^{60} \Delta u(k+j)^2 \quad (17)$$

After an enormous number of experiments, λ_1 =0.6 and λ_2 =0.4 are found to be suitable to solve the addressed problem. The performance of each optimisation algorithms is analysed in three different aspects: The set-point accuracy (18), the energy consumption (19 and 20) and the computing time required (21).

$$SPe = \sum_{k=1}^{N} [T_i(k) - w(k)]^2 \quad (18)$$

$$E_V = \sum_{k=1}^{N} V(k) \quad (19)$$

$$E_H = \sum_{k=1}^{N} H(k) \quad (20)$$

$$C_T = \frac{Elapsed \quad Time}{Time \quad Unit} \quad (21)$$

The set-point have a square shape with different indoor temperature levels for the night and day periods.

Table 3 shows the results obtained for the criteria defined by equations (18,19,20 and 21). Figures 2 to 5 represent the simulated set-point tracking responses for the period of one day obtained with PSO and GA, respectively.

Table 3 Simulation results using PSO and GA

	PSO	GA
SPe	0.0035	0.0085
E_H	380.34	384.52
E_V	228.20	228.45
C_T	258.04	302.18

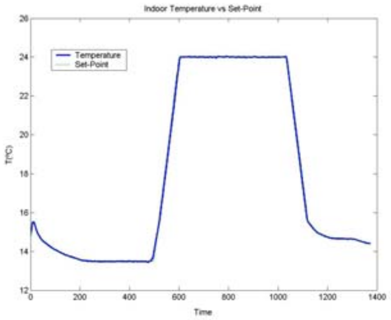

Fig. 2. Set-point tracking results using the PSO algorithm.

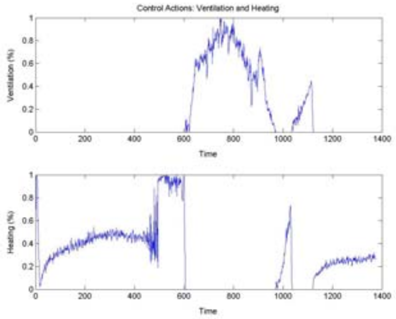

Fig. 3. Ventilation and heating actuations computed with the PSO.

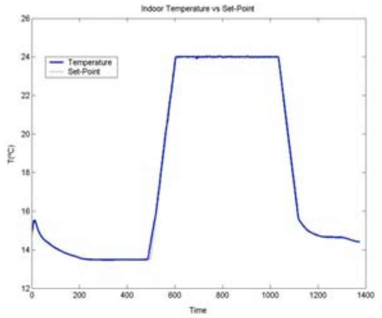

Fig. 4. Set-point tracking results using the GA.

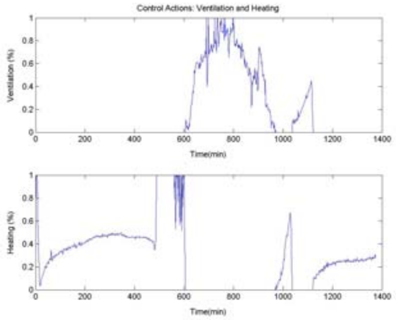

Fig. 5. Ventilation and heating actuations computed with the GA.

6. CONCLUSION

The particle swarm optimisation algorithm was proposed as a new method to design a greenhouse air temperature model predictive controller subject to restrictions. The 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 one hour with a one-minute sampling period. By observation of the simulation results, one can conclude that the PSO algorithm was able to reduce the set-point tracking error in approximately 40% relatively to the minimum error achieved by the genetic algorithm. Simultaneously it was able to decrease the heating consumption in 1.2%, the ventilation requirements by 0.1% and the algorithm run time in 14%.

REFERENCES

Åström, K. J. and B. Wittenmark (1989). *Adaptive Control*. Addison-Wesley, ISBN: 0-201558-66.

Boaventura Cunha, J., C. Couto and A. Ruano (1998) A greenhouse climate multivariable predictive controller, *Acta Horticulture N.534, ISHS*, pp: 269-276.

Camacho, E. F. and C. Bordons (1994). *Model Predictive Control in the Process Industry*, Springer, Sevilla.

Clarke, D. W., C. Mohtadi and P.S. Tuffs (1987). Generalized Predictive Control – Part I. The Basic Algorithm, *Automatica*, **Vol 23**, No 2, pp.137-148.

Clerc M., 1999. The Swarm and the Queen Towards a Deterministic and Adaptive Particle Swarm Optimization. *Proceedings of the 1999 Congress of Evolutionary Computation*, IEEE Press, **Vol. 3**, pp. 1951-1957.

Kennedy, J. and R.C. Eberhart (1995). Particle Swarm Optimization, *Proc. Of the 1995 IEEE Int. Conf. On Neural Networks,* pp. 1942-1948. IEEE Service Center, Piscataway, NJ.

Kennedy J. (1997). The Particle Swarm: Social Adaptation of Knowledge. *Proceedings of the 1997 IEEE Int. Conference on Evolutionary Computation, Indianapolis*, US, pp. 303-308.

Kennedy, J. and R. C. Eberhart (2001). *Swarm Intelligence.* Academic Press, ISBN: 1-55860-595-9, pp. 344-345.

Nielsen, B. and H. Madsen (1996). Predictive Control of Air Temperature in Greenhouses, *IFAC 13th Triennial World Congress, San Francisco*, pp. 399-404.

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

Shi Y. and R. C. Eberhart (1999). Empirical Study of Particle Swarm Optimization, *Proceedings of the 1999 IEEE Congress of Evolutionary Computation*, **Vol 3**, pp. 1945-1950.