The 6th Asian Control Conference
July 18-21, 2006
Bali, Indonesia

# On-Line Control using the Particle Swarm Optimisation Algorithm

**João Paulo Coelho[1] , P.B. de Moura Oliveira[2,3] , J. Boaventura Cunha[2,3] and Damir Vrancic[4]**
[1] IPB - ESTIG - Instituto Politécnico de Bragança- Escola Superior de Tecnologia e Gestão
5301-854 Bragança, Portugal
[2] UTAD - Universidade de Trás-os-Montes e Alto Douro, Departamento de Engenharias
5001-911 Vila Real, Portugal
[3] CETAV -Centro de Estudos e Tecnologias do Ambiente e da Vida da UTAD
5001-911 Vila Real, Portugal
[4] Department of Systems and Control, J. Stefan Institute
Jamova, 39,1000 Ljubljana, Slovenia
e-mail: jpcoelho@ipb.pt; oliveira@utad.pt; jboavent@utad.pt;damir.vrancic@ijs.si

## Abstract

In the last two decades, evolutionary based algorithms have proved to be an important tool in solving optimisation problems in many disciplinary areas, namely in control system design. However one of its limitations for some type of applications is the usually high computational load required, which restricts its use for on-line control. This paper proposes the use of a stochastic search algorithm, known as particle swarm, as an optimisation tool for an on-line model predictive control of a custom made laboratory thermodynamic system. Preliminary results are presented.

## 1. Introduction

A model predictive controller is applied to regulate the air temperature of a thermodynamic laboratory process. This type of control strategy has become very popular since the 80's [1] and, comparing to other control algorithm, has the advantage to provide the system with the ability to react before any deviations in the controlled variable take place. This anticipatory behaviour of the controller is achieved by using a model of the process in order to predict the system response in view of a given set of future control actions. The control values to be injected in the system for a specific time horizon are usually computed by minimising a quadratic cost function of the form [2]:

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

with,

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

In the above formulae, $\Delta u(k+j-1)$ represents the control effort, $\lambda_1$ and $\lambda_2$ are weights for each expression component, *c* represents the control horizon, constants *a* and *b* represent the instant limits in which it is desirable that the output follows the reference and $\varepsilon(k+j\,|\,k)$ is the prediction error between the future trajectory $w(k+j)$ and the predicted output $\hat{y}(k+j\,|\,k)$.

Usually the cost function $J$ must be minimised regarding a set of design and physical constraints. It is common to consider magnitude and rate constraints for the control actions and level constraints for the output signal.

Discarding the problems associated with model fitting, predictive control resumes the minimisation of a function subject to constraints. Due to the nature of this function, the controller states are obtained by iterative numerical procedures. These function minimisation strategies can be based in either deterministic or stochastic search methods. However, since the search space defined by the

ISBN 979-15017-0

restricted cost function is, generally, very complex, non-linear and non-convex, stochastic search algorithms are suitable for this application.

## 2. The Particle Swarm Optimisation Algorithm

In real-time control, any control algorithm must be fast enough to run completely between sample instants. In this context, the application of a given control strategy depends on the sampling frequency, on the computational power of the hardware and on the complexity of the control algorithm. The model predictive control is, by itself, a computational heavy algorithm. The computational load depends on several factors like the prediction horizon, the complexity of the model and the performance of the search algorithm.

Among the fastest stochastic search algorithms is the particle swarm optimisation algorithm (PSO). This search strategy has a decade of existence and was firstly proposed by Kennedy and Eberhart [3]. Since then several works have been published on this subject concerning his mathematical proprieties, or application in a specific problem like greenhouse environment control [5] and PID control [6].

Conceptually, the PSO is an algorithm 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 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 $i$ particle and 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 [3,4]. The behaviour of each particle is based on these two types of knowledge and their current position regarding the search. In this context, the behaviour of particle *i* in the search space is governed by the following two equations:

$$v_{id}\left(t+1\right)=\omega(t)\cdot v_{id}\left(t\right)+\varphi_1.\left[p_{id}\left(t\right)-x_{id}\left(t\right)\right]+\varphi_2.\left[p_{gd}\left(t\right)-x_{id}\left(t\right)\right] \tag{3}$$

$$x_{id}\left(t+1\right)=x_{id}\left(t\right)+v_{id}\left(t+1\right) \tag{4}$$

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 $\varphi_1$ is known as the cognitive constant and $\varphi_2$ as the social constant, that represent uniformly distributed random numbers generated in a pre-defined interval. The $\omega(t)$ variable represents the inertia weight and his value affects the type of search. A large $\omega$ value will direct the PSO for a global search while a small $\omega$ will direct the PSO for a local search. In order to make a global search in the early run and more local in the end, the inertia weight can be made to vary linearly from a larger value to a smaller one.

The velocity of the particles is bounded by a maximum value $V_{max}$. The value of this constraint is intimately connected with the maximum "jump" in the search each particle can make in each iteration. The value selected for $V_{max}$ should not be too high to avoid oscillations, or too low to avoid search traps. Additionally the particle's position should be, if necessary, relocated to a point inside the defined search space.

## 3. Experimental Setup

In order to apply the addressed control strategy in a real physical system, a thermodynamic process was built. The plant is composed by a PVC tube with an inner diameter of 63 mm and a length of 60 cm. Additionally two actuators, a fan and a heating resistor grid, were embedded in the tube. Air is forced to circulate by a fan through a pipe and it is heated at its inlet by an electric heater. The purpose of this system is to allow the air temperature control in a specific spot of the tube. In order to do that three temperature sensors have been installed. One to measure the temperature of the heating element, one to measure the environment temperature and the other installed at ten centimetres away from the tube outlet.

The proposed system has two degrees of freedom, meaning that it is possible to manipulate the mass of air at room temperature entering the tube by regulating the fan speed and the heat produced by the resistor by controlling the mean power applied to this element. A diagram of the building blocks of the process and a picture of its final aspect are illustrated in the following figure.

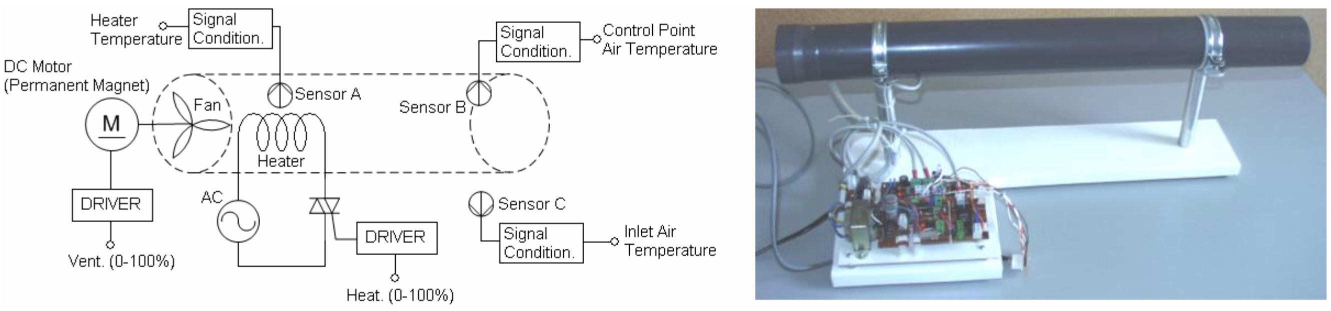

Figure 1: Block diagram of the process to be controlled (left) and picture of the process final aspect (right)

The control and measured signals are manipulated in a PC compatible digital computer with a Intel Pentium II processor running at 450 MHz. The communication between the process and the computer was handled by a custom made ISA bus data acquisition card with an 8 bit resolution. Due to the time constants involved in the process, a 1Hz sampling frequency was found suitable.

## 4. System Model

Regardless of the system's two degrees of freedom, in this work the air flow rate is kept constant. Hence the system input is a voltage that controls the mean power applied to the heater and the output is the outlet air temperature. In order to use a MPC control strategy a model of the plant is required. For that propose, a preliminary system identification process was carried out using as an excitation input signal random both in amplitude and period.

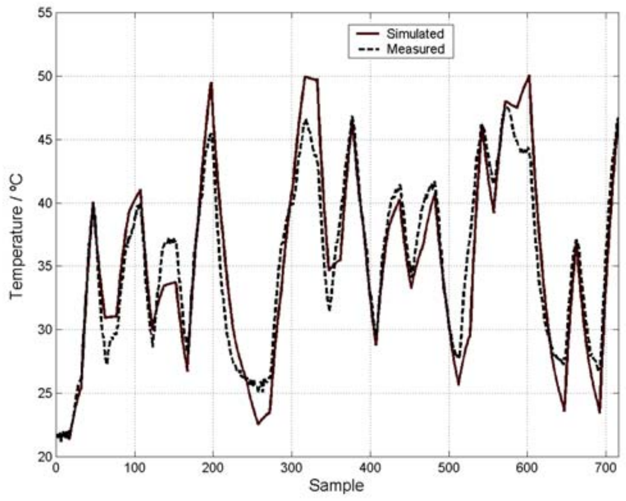

Figure 2: Open-Loop simulation of the ARX plant model under validation data.

Considering that the outdoor temperature and ventilation rate is approximately constant, the model founded sufficiently accurate to model the plant dynamic behaviour is:

$$T_{pipe}\left[kT\right]=0.9579\cdot T_{pipe}\left[\left(k-1\right)T\right]\text{-}0.194\cdot Heat^{2}\left[\left(k-1\right)T\right]+2.1684\cdot Heat^{2}\left[\left(k-2\right)T\right]+0.749 \qquad (6)$$

where $T_{pipe}$ is the outlet air temperature, $Heat$ is the relative voltage applied to the phase control hardware that drives the heater and $T$ is the sampling period. The model incorporates the square value of the $Heat$ variable. This is because the relation between the heat generated and the applied voltage has a quadratic proportionality as stated by the well known Jules law. Although the model seems non-linear it is still linear in the parameters. Hence the values of the model coefficients were obtained using the least squares method. In figure 2, the open-loop simulation results for the proposed model under validation data are presented.

## 5. Experimental Results

In the following figure, the controller performance regarding the set-point accuracy and the control signal of the heater is shown.

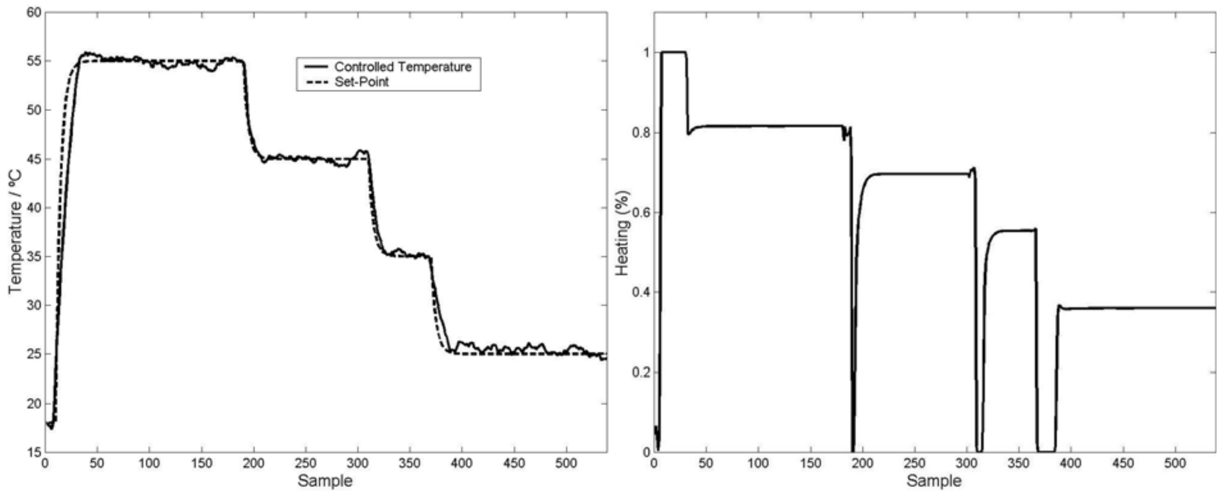

Figure 3: Experimental results regarding set-point accuracy (left) and control signal (right).

Those results were accomplished using a fifty particles swarm size and the PSO algorithm was evolved during two hundred iterations. In this experiment the prediction horizon and the control horizon was set to ten steps ahead and the weight factors $\lambda_1$ and $\lambda_2$ was set to 0,6 and 0,4 respectively. As it can be observed the quality of the set-point tracking is good, as expected from this type of control.

## 6. Conclusion

In this paper, the preliminary results of real-time control on a physical plant using the particle swarm optimisation algorithm have been presented. From the results obtained it can be concluded that this tool has real practical use outside the simulation environment. Indeed, evolutionary algorithms great potential it's their ability to evolve a set of possible solutions in a highly sophisticated multimodal search space with large number of discontinuities. Moreover its application in multiobjective problems allied to automatic decision mechanisms can be of great interest in applications outside the computer environment. With an obsolete computer it was shown that it is possible to use this kind of evolutionary search tool in practical real-time control applications.

## References

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

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

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

[4] Kennedy J. and Eberhart R. C., 2001,"Swarm Intelligence", Academic Press.

[5] Coelho, J. P., De Moura Oliveira, P. B and Cunha, J. B. *Greenhouse Air Temperature Control using the Particle Swarm Optimisation Algorithm*, 15th IFAC World Congress, Barcelona, Spain (2002)

[6] De Moura Oliveira P. B., Boaventura Cunha J. e Coelho J. P., (2002), "Design of PID Controllers using the Particle Swarm Algorithm", *23rd IASTED International Conference in Modelling, Identification and Control (MIC 2002)*, pp. 263-268, 18-21 de February, Innsbruck, Austria.