3ème Conférence Internationale sur le Soudage, le CND et l'Industrie des Matériaux et Alliages
IC-WNDT-MI'12
Oran
26-28 novembre2012

COMPARAISON OF HEURISTIC AND HYBRID HEURISTIC FOR THE SOLUTION OF LINEAR MODEL BASED PREDICTIVE CONTROL

H. Merabti[1], *D.Boucherma*[2] *and K. Belarbi*[3]

[1]: *unité de recherche en technologie industrielle, URTI/CSC, BP1037, site université Badji Mokhtar, Chaiba, 23000, Annaba, Algerie. Email : merabtihalim@yahoo.fr*
[2]: *unité de recherche en technologie industrielle, URTI/CSC, BP1037, site université Badji Mokhtar, Chaiba, 23000, Annaba, Algerie. Email : djamelboucherma25@yahoo.com*
[3]: *University of Constantine, Route de Ain el bey, Constantine Algeria. Email: kbelarbi@yahoo.com*

ABSTRACT:

In this work, we compare the application of simple meta heuristics and hybrid for the on line optimal solution of the linear MPC. For these purpose, we consider two meta heuristics: particles swarm optimization, PSO, and gravitational search algorithm, GSA and a hybrid algorithm, PSO-GSA. Two linear systems are considered, a monovariable and multivariable system.

Keywords: predictive control, heuristics, optimization.

1 INTRODUCTION

Model based predictive control (MBPC) is based on the use of a model for predicting the future behavior of the system over a finite future horizon. The current control action is obtained by solving on-line, at each sampling instant, a finite horizon optimal control problem, using the current state of the plant as the initial state [1]. The optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. The solution of the optimization problem depends on the nature of the model and constraints. If the model is linear with linear constraints such as saturation of the control signal, the problem is cast as a quadratic problem which was solved using the active set method [2].
Later, a number of efficient solution procedures based on the so called multi parametric programming have been proposed [3]. In these methods, the state space is divided into regions and a control law is obtained off line for each region yielding a control as look up table. However the number of region may become prohibitively large for high dimension systems. Very recently, the canonical piecewise-affine (PWA) functions are used to approximate the linear MPC control law, leading to very fast computation times [4]. The developments associated with these approaches is however laborious. In this work, we investigate the application of the so called meta heuristics for the on line optimal solution of the linear MPC. For these purpose, we present a comparison between three different heuristics: particles swarm optimization, gravitational search algorithm, and hybrid PSO-GSA. Two linear systems are considered, a monovariable and multivariable system.
This paper is organized as follows: the following section introduces the MPC approach, heuristics uses in optimization are presented in section three, and in section four we present the simulation results.

2 LINEAR MODEL PREDICTIVE CONTROL

Consider the problem of regulating to the origin the following linear discrete system:

$$x(t+1) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) \quad (1)$$

Where $x(t) \in R^n$ is the state vector, $u(t) \in R^m$ is the input vector, $A \in R^{n*n}$, B $\in R^{n*m}$ and $(A; B)$ is a controllable pair.
Given $x(t)$ the value of the state obtained from measurement at time instant t , MPC solves the following optimization problem

$$min_u\{J(U, x(t)) = x_{t+N/t}^T P x_{t+N/t} + \sum_{k=0}^{N-1} x_{t+k/t}^T Q x_{t+k/t} + u_{t+k}^T R u_{t+k}\} \quad (2)$$

s.t. $y_{\min} \leq y_{t+k/t} \leq y_{\max}$, $k = 1, \ldots, N$
$u_{\min} \leq u_{t+k} \leq u_{max}$, $k = 0, \ldots, M-1$
$x_{t/t} = x(t)$
$x_{t+k+1/t} = Ax_{t+k/t} + Bu_{t+k}$, $k \geq 0$
$y_{t+k/t} = Cx_{t+k/t}$, $k \geq 0$

With respect to $U = [u_t^T, .., u_{t+M-1}^T]^T$ where $y_{min} \leq 0 \leq y_{max}$, $u_{min} \leq u_{t+k} \leq u_{max}$,
$R > 0$, $Q > 0$, $P > 0$, $x_{k+t/t}$ is the prediction of x_{t+k} at time t, and M is the control input horizon. The final cost matrix P is usually calculated from the algebraic Riccati equation with the assumption that the constraints are not active for $k \geq N$ [3].

3 THE META HEURISTICS

The gravitational search algorithm and the particle swarm optimization are classified as evolutionary (population) approaches which work simultaneously with a set of solutions that evolves gradually. The use of several solutions permits to improve the exploration of the search space simultaneously. In the following we give a brief overview of these Meta heuristics.

3.1 Gravitational search algorithm GSA [5]

In this section, we introduce the optimization algorithm based on the law of gravity. In the gravitational search algorithm, GSA, objects attract each other by the force of gravity which causes a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. The heavy masses, corresponding to good solutions, move more slowly than lighter ones. In GSA, each mass has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined using a fitness function. The algorithm is evolves by properly adjusting the gravitational and inertia masses. Eventually, masses will be attracted by the heaviest mass which presents an optimum solution in the search space.
To summarize, masses obey the following laws:
Law of gravity: each mass attracts every other particle. The gravitational force between two mass is proportional to the product of their masses and inversely proportional to the distance between them [5].
Law of motion: the current velocity of any mass equals the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

3.2 Particles swarm optimization PSO [6]

Particle swarm optimization is an evolutionary computation technique developed by Kennedy and Eberhart in 1995.The particle swarm concept originated as a simulation of a simplified social system. PSO is similar to genetic algorithm in that the system is initialized with a population of random solutions. It is unlike a GA, however, in that each potential solution is also assigned a randomized velocity, and the potential solutions, called particles, are then "flown" through the problem space. Each particle keeps track of its coordinates in the problem space which are associated with the best solution, *pbest*, (fitness) it has achieved so far. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This location is called *gbest.*

At each step, the PSO concept consists of changing the velocity of each particle toward its *pbest* and *gbest* locations. The velocity is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *gbest* locations.

The original process for implementing the global version of PSO is as follows:

1- Initialize a population of particles with random positions and velocities in the problem space
2- For each particle, evaluate the desired optimization fitness function.
3- Compare particle's fitness evaluation with particle's *pbest*. If current value is better than *pbest*, then set *pbest* value equal to the current value and *pbest* location equal to the current location.
4- Compare fitness evaluation with the population's overall previous best. If current value is better than *gbest*, then reset *gbest* to the current particle's array index value.
5- Change the velocity and position of the particle according to the equations (3) and (4), respectively:

$$v_{k+1}^{i} = v_{k}^{i} + c_1 r_1 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i) \quad (3)$$

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i} \quad (4)$$

6- Loop to step 2 until a criterion is met.

Where:

- x_k^i : Particle position
- v_k^i : Particle velocity
- p_k^i : Local best position
- p_k^g : Global best position
- c_1 , c_2 : Constants
- r_1 , r_2 : Random numbers between 0 and 1

3.3 Hybrid PSO-GSA optimization [11]

The hybrid PSO-GSA algorithm is a new approach for problem optimization that combines the ability of social thinking (*gbest*) in particle swarm optimization with the local search capability of gravitational search algorithm. In this algorithm, firstly, each agent (candidate solution) is randomly initialized. Then, in each iteration, all agents takes new positions obeying laws presented in section 3.1, and the best solution so far should be updated. The process of updating velocities and positions will be stopped by meeting an end criterion. The agents near good solutions try to attract the other agents which are exploring the search space. When all agents are near a good solution, they move very slowly. In this case, the *gbest* help them to exploit the global best. PSO-GSA use a memory (*gbest*) to save the best solution has found so far, so it is accessible anytime and agents can observe the best solution so far and tend toward it.

4 RESULTS OF SIMULATION

In this section, we present the application of the above meta heuristics for the online solution of the MPC problem in two examples. Our goal is to investigate when to obtain the control signal with a computation time less than the sampling period, satisfying all constraints posed by the problem. Algorithm that checks this constraint can be implemented ***online*** to control the process. All the computation times are achieved on" intel® Core™ 2 Duo CPU a 2.93 GHz 2.93GHz, RAM 2Go", running Matlab 7.7.

4.1 Example 1

Consider the constrained linear system described by the state space representation [9]:

$$x(t+1) = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x(t) + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u(t)$$
$$y(t) = [0 \quad 1.4142] x(t) \qquad (5)$$

s.t $-2 \leq u(t) \leq 2$

With $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, R = 0.01, the final cost matrix P is calculated from the algebraic Riccati equation with the assumption that the constraints are not active for $k \geq N$. The sampling time $T=0.1s$, and the initial conditions are $x(0) = [1\ 1]'$.

The task is to regulate the system to the origin. To this aim, we design controllers based on the optimization problem (2), using the approaches proposed above. Table1 represents necessary time to deliver the best control value with different approaches parameters.

The results of the simulation are shown in Figures 1 and 2. The control signal and the states are quite similar for the three heuristics. Table 1 gives the computational time where one can see the PSO algorithm is the fastest.

Table 1 - Computation times and parameters of the algorithms for linear mono variable system.

	PSO	GSA	PSO-GSA
Approach parameters	Swarm size:10 Iteration:10 Inertia weight:0.8 C1=C2=1.2	agent-size:5 Iterations:50 Alfa= 20,g0=1	agent-size:5 Iterations:20 Inertia weight:1 C1=0.5;C2=1.5
Computation time in ms	< 1.1	< 3.2	< 2
Remark (*Ts=100ms)*	*Tc ≤ Ts*	*Tc ≤ Ts*	*Tc ≤ Ts*

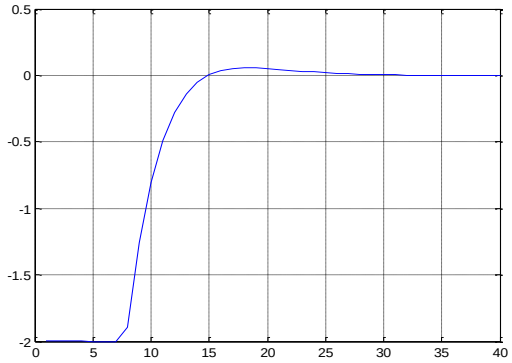

Figure 1: PSO, GSA, PSO-GSA control signal for linear mono variable system

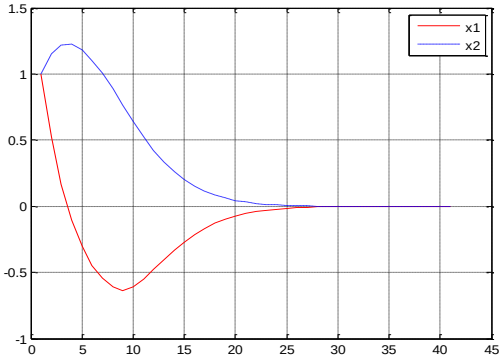

Figure 2: PSO, GSA, PSO-GSA states for linear mono variable system

4.2 Example 2

The laboratory model helicopter (Quanser 3-DOF Helicopter) sampled with $T = 0.01$ s [7], and the following state space representation is obtained [10]:

$$A = \begin{bmatrix} 1 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0.01 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0 & 0 \\ 0.0001 & -0.0001 \\ 0.0019 & 0.0019 \\ 0.0132 & -0.0132 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6}$$

States of the system are:
x1: elevation,
x2: pitch angle,
x3: elevation rate,
x4: pitch angle rate,
x5: integral of elevation error,
x6: integral of pitch angle error.
Inputs to the system are:
u1: front rotor power, *u2*: rear rotor power.
The system is to be regulated to the origin with the following constraints on the inputs and pitch and elevation rates:
$-1 \leq u1 \leq 3$; $-1 \leq u2 \leq 3$; $-0.44 \leq x3 \leq 0.44$; $-0.6 \leq x4 \leq 0.6$;
The cost function is given by :
$Q = diag$ (100; 100; 10; 10; 400; 200); R = I2×2, and P is given by the solution to algebraic Riccati equation.
The results of the simulation are shown in Figures 3 to 6 where it can be seen that the control signal and the states are quite similar for the three heuristics. Table 2 gives the computational time where one can see the PSO algorithm is the fastest.

Table 2 - Computation times and parameters of the algorithms for linear multi variable system

	PSO	GSA	PSO-GSA
Approach parameters	Swarm size:15 Iteration:25 Inertia weight:1.1 C1=C2=1.2	agent-size:6 Iterations:400 Alfa= 20,g0=1	agent-size:5 Iterations:190 Inertia weight:1 C1=0.5;C2=1.5
Computation time in ms	< 7.5	< 55	< 12
Remark *(Ts=10ms)*	*Tc* ≤ *Ts*	*Tc* ≥ *Ts*	*Tc* ≥ *Ts*

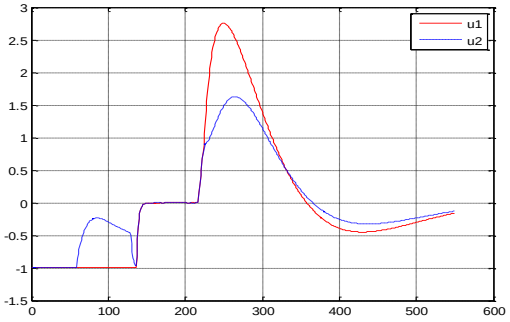

Figure 3: PSO, PSO-GSA control signals for linear multi variable system

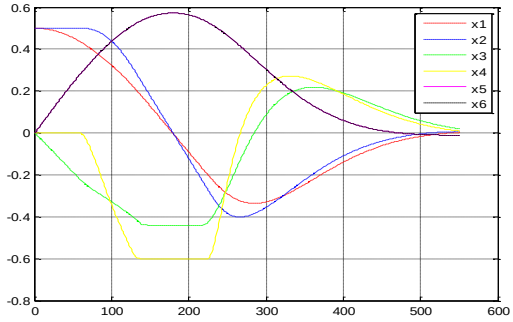

Figure 4: PSO, PSO-GSA states for linear multi variable system

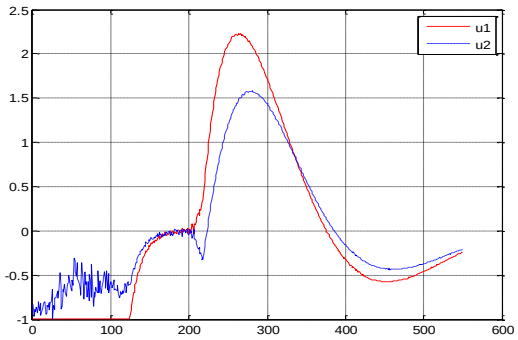

Figure 5: GSA control signals for linear multi variable system

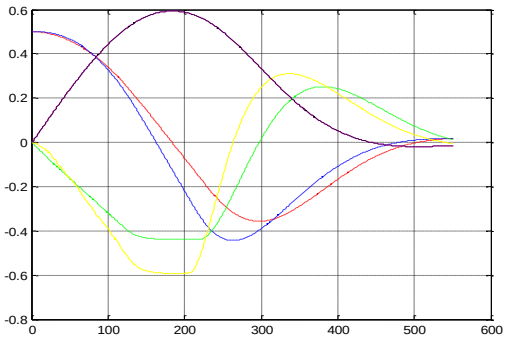

Figure 6: GSA states for linear multi variable system

5 CONCLUSION

In recent years, various heuristic optimization algorithms have been developed. Some of these algorithms are inspired by swarm behaviors in nature In order to compare the algorithms; we have examined it on a set of various standard benchmark functions.

References

[1] J.M. Maciejowski, Predictive control with constraints, Prentice Hall, 2001
[2] J. A. Rossiter, Model Based Predictive Control A Practical Approach, CRC PRES London , 2004
[3] A. Bemporad, M. Morari, V. Dua and E.N. Pistokopoulos, "The explicit linear quadractic regulator for constrained systems," *Automatica*, 38(1), 3-20, 2002.
[4] A. Bemporad, A. Oliveri, T. Poggi, and M. Storace, "Ultra-Fast Stabilizing Model Predictive Control via Canonical Piecewise Affine Approximations," *IEEE Transactions on automatic control*, vol. 56, N° 12, 2011pp. 2883-2897.
[5] E. Rashedi, H. Nezamabadi-pour , S. Saryazdi," GSA: A Gravitational Search Algorithm" *Information Sciences* , no. 179, 2009,pp. 2232–2248.
[6] Eberhart, R. C., and Shi, Y., "Particle swarm optimization: developments, applications and resources,"*Proc. Congress on Evolutionary Computation 2001*,Seoul, Korea.
[7] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization artificial ants as a computational intelligence technique", *IEEE Computational intelligence Magazine*, 2006
[8] K. Socha , M. Dorigo, "Ant colony optimization for continuous domains," *European Journal of Operational Research*, no. 185, 2008, pp. 1155–1173
[9] Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E. N., "The explicit linear quadratic regulator for constrained systems, " In *Proceedings of the American Contr. Conference* Chicago, IL., 2006, pp. 872–876.
[10] P. Thondel, T. Johansen, A. Bemporad, " An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica* no. 39, 2003, pp. 489 – 497