Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Possibilities of the Cross-Entropy Method Usage in the Control Theory

A. Tahirovic, B. Lacevic *,

* *Politecnico di Milano, Italy*
(Tel: 38761299231; e-mail: tahirovic@elet.polimi.it, lacevic@elet.polimi.it)

,

Abstract: This paper deals with the Cross-Entropy method application in the control theory. The method is a the combinatorial optimization technique that is mostly used in the networks theory and could be used in deterministic optimization problems as well. The paper shows the possibiliy of the Cross-Entropy usage in the control parameter tuning. Similar to genetics algorithms, this method minimizes a given performance function in order to find optimal parameters. The appropirate conclusions about relability and the convergence rate of the method were experimentaly supported. The first two experiments used PI controller with different performance functions, where the optimal controller values have been obtained through the simulation. The third experiment used LQR controller to control a complex system. The tunining of four parameters of a LQR control matrix and obtained values were compared with the ones generated by LQR algorithm.

1. INTRODUCTION

The idea of the Cross-Entropy method, as a relatively new approach for combinatorial optimization problems, was first explored in complex stochastic networks (Rubinstein [1997]). It was used for probability estimations of rare events. In Rubinstein [1999] and Rubinstein [2001], the possiblity for solving combinatorial optimization problems was presented as well. The idea was to adapt the deterministic optimization problem to a related stochastic one and to use the technicques for rare events estimation, explored in Rubinstein [1997]. One of the main advantages of the Cross-Entropy method is presence of relativly large learning rate, based on simulations. There are already some combinatorial optimization techniques such are simulated annealing (Aarts et al. [1989]), tabu search (Glover et al. [1993]) and genetics algorithms (Goldberg [1989]). Genetic algorithms are widely explored in the control theory as a technique for optimization of the given index of performance. The idea of this paper is to show the simple ability of using the Cross-Entropy method in the control theory.

In the second section, overview of the maximum entropy principle is given. The choice of the probability density function that is used for exploration of controller parameter space is based on this principle. The Gaussian multivariate distribution is the one that maximizes the differential entropy and it is used in the paper.

In the third section short overview of the originated Cross-Entropy method was presented. In order to understand the final algorithm of the method, minimization of Kullback-Leibler distance was described as well as maximization of likelihood function. These techniques are already well known in the probability theory.

Finaly, the different simulations where used in order to present the Cross-Entropy approach in the control theory. These simulations used not only different systems and controllers but also various performance functions that should have been minimized.

2. MAXIMUM ENTROPY PRINCIPLE

Suppose we have a set of values of a random variable X generated with an unknown probability distribution with its constraints derived (learned) from the data. These constraints could be any order moments calculated from these data. The goal is to choose the most appropriate probability model that is the optimum in some sense and that satisfies these constraints. There are infinite solutions of possible models to be chosen from. Maximum entropy principle (Jaynes [1957]) gives the possibility to solve this problem. The correctness of the maximum entropy principle was proved in Shore et al. [1980].

The maximization of the differential entropy could be described as a constrained optimization problem. The objective function is

$$h(x) = -\int_{-\infty}^{\infty} f_X(x) log f_X(x) dx \tag{1}$$

that is the differential entropy over all probability density functions $f_X(x)$ of a random variable X. The constraints are:

1. $f_X(x) \geq 0$
2. $\int_{-\infty}^{\infty} f_X(x) dx = 1$
3. $\int_{-\infty}^{\infty} f_X(x) g_i(x) dx = \alpha_i$,

where $g(x)$ is any order function and α is obtained value from the data.

The third constraint gives the prior knowledge of this random variable X. If these constraints are moments of the first and the second order then the optimal probability distribution is the Gaussian one. This means that for the given mean μ and variance σ^2 the Gaussian random variable has the largest differential entropy and this entropy is given by:

$$h(x) = \frac{1}{2}[1 + log(2\pi\sigma^2)] \tag{2}$$

Similar to single variable Gaussian distribution, if there are N random variables, given in vector $\mathbf{X}$, then the multidimensional Gaussian probability distribution is the one with maximum entropy. This entropy is given by:

$$h(\mathbf{X}) = \frac{1}{2}[N + log(2\pi) + log|det(\Sigma)|] \tag{3}$$

where Σ is the second order statistic of the vector $\mathbf{X}$.

In order to optimize the given objective function of the system to be controlled, all values of considered random variables, given in the vector $\mathbf{X}$, are generated by multivariate Gaussian probability distribution. The optimal parameters of the Gaussian probability distribution has been derived by Cross-Entropy method.

3. CROSS-ENTROPY METHOD

This method was motivated by an adaptive algorithm for estimating the probabilities of rare events in complex stochastic networks (Rubinstein [1997]). By converting a deterministic optimization problem into an appropriate stohastic one, it was also shown that this method could be used in order to solve combinatorial optimization problems.

Let $S(\mathbf{X})$ be the performance function where

$$\mathbf{X} = [\mathbf{X}_1\mathbf{X}_2 \cdots \mathbf{X}_m]^T$$

is the vector of parameters of m dimensional performance function $S(\mathbf{X})$, where each parameters of the vector $\mathbf{X}$ will be considered as a random variable, where $X_i \sim f(X_i; \mathbf{u}_i)$ and $\mathbf{u}_i$ $(i = \overline{1, m})$ is the parameter vector of related distribution.

Similar to a variant of genetic algorithm, this method is also based on elitism. This means that, in every iteration step (generation), method generates N samples that are consited of a fixed number of parameter space points (chromosoms) giving related values of performance function $S(\mathbf{X})$. In the second phase of every iteration the best parameter space points are involved in providing the major part of information for the next iteration. Unlike genetic algorithms, in the Cross-Entropy method, every parameter space points are generated using multivariate probability density function. This probability density function has its own vector of distribution parameters (for Gaussian probability distribution, the appropriate distribution parameters are mean and variance). The only way for improoving the next generation of parameter space points is to adapt these distribution parameters. The adaptation algorithm of these probability distribution parameters is the main goal of the Cross-Entropy technique.

In order to use the Cross-Entropy technique as a combinatorial optimization method, apropriate stohastic optimization problem should be defined (Boer et al. [2005]). Let l be the probability that the object function $S(\mathbf{X})$ will be greater than the given level γ, $l = P(S(\mathbf{X}) \geq \gamma)$. The main goal of the stohastic optimization problem is to estimate this probability. The reason why this estimation is realted to maximization problem of $S(\mathbf{X})$ will be given in subsection 3.2.

If a set of indication functions is given $I_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}}$, then l could be estimated with Crude Monte Carlo method using equation:

$$\widehat{l} \approx \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}} \tag{4}$$

because l could be considered as the excpected value

$$l = EI_{\{S_{(X)} \geq \gamma\}} = \int_{-\infty}^{\infty} I_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}} f_X(x)dx \tag{5}$$

of the indication function with respect to distribution $X \sim f(X; \mathbf{u})$. This means that N samples of $\mathbf{X}_j = (x_{1j}x_{2j} \ldots x_{mj})$, where $j = \overline{1, N}$, are generated using different probability density functions for each random variable $\mathbf{X}_i$.

For large γ the probabiliy l will be very small and, in order to estimate it, N should be very large (indication function is eqal to zero for almost every sample $\mathbf{X}_j$). This means that the main disadvantage of Crude Monte Carlo method is very large simulation time.

3.1 Importance Sampling

In order to decrease simulation time, *Importance Sampling* (Smith et al. [1997]) is used. Importance Sampling could be considered as an adaptation, making an adequate model that speed up generation of rare events for the case when γ is large enough.

If $X \sim g(X; \boldsymbol{v})$ is another probability density function, where $\boldsymbol{v}$ is its distribution parameter vector, then l could be rewritten as:

$$l = \int_{-\infty}^{\infty} I_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}} \frac{f_X(x)}{g_X(x)} g_X(x)dx = EI_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}} \tag{6}$$

if holds $g(x) = 0 \Rightarrow I_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}} f_X(x) = 0$. Now, l could be considered as expected value of indication function with respect to the distribution $X \sim g(X; \mathbf{u})$. This means that l could be estimated using following equation:

$$\widehat{l} \approx \frac{1}{N}\sum_{i=1}^{N} I_{\{S(\mathbf{X}_{(i)}) \geq \gamma\}} W(\mathbf{X}_i), \tag{7}$$

where $W(X; \mathbf{u}, \boldsymbol{v}) = \frac{f(X;\mathbf{u})}{g(X);\boldsymbol{v})}$ and X is now a random variable generated with distribution $g(X; \boldsymbol{v})$, that is $X \sim g(X; \boldsymbol{v})$. Function g is called the *importance sampling density*, function W *the likelihood ratio* (LR) *function* , while $\widehat{l}$ is called *the likelihood ratio* (LR) *estimator*.

The main goal of introducing *importance sampling* was to decrease simulation time of the probabilty l estimation

comparing to Crude Monte Carlo method. This pulls the question: 'What is the best vector parameter $\boldsymbol{v}$, of the distribution $g_X(X;\boldsymbol{v})$, that gives the most accurate estimation of l for a given simulation effort?'

Let us suppose that only one sample of the variable $\mathbf{X}$ is nedded in order to obtain the probability l, that is

$$l = I_{\{S(\mathbf{X}_{(i)})\geq\gamma\}}\frac{f(\mathbf{X}_i;\mathbf{u})}{g^*(\mathbf{X}_i;\boldsymbol{v})}\ \forall i, \tag{8}$$

where g^* is the optimal distribution given for the chosen parameter vector $\boldsymbol{v} = \boldsymbol{v}^*$. This is the case when the variance of the random variable X is equal to zero because every sample X_j produces the same value of the probability l. In that case, function g could be found as:

$$g^*(X,\boldsymbol{v}) = I_{\{S(X)\geq\gamma\}}\frac{f(X;\mathbf{u})}{l}, \tag{9}$$

The problem is that this case is not possible because the probability density g depends on the uknown value l.

The best vector $\boldsymbol{v}^*$, that could be used for simulation given with (7) , could be found such that the chosen distance between considered optimal density function g^* and the density function $g(X;\boldsymbol{v})$ is minimal. The possible measure of this distance could be *Kullback-Leibler* distance.

3.2 Kullback-Leibler Distance (Cross Entropy)

Cross-entropy or Kullback-Leibler distance (divergence) between two probability functions g and h (Kullback [1968], Gray [1990] and Cover et al. [1991]) is defined by

$$D(g,h) = E_g ln\frac{g(\mathbf{X})}{h(\mathbf{X})} = \int ln\frac{g(x)}{h(x)}g(x)dx \tag{10}$$

In order to minimize $D(g^*,g)$, supposing that the chosen h probability function is taken from the same family as the probability function f (For our case, this is the Guassian one), that is $g(.;\boldsymbol{v}) = f(.;\boldsymbol{v})$, the expression of the interest becomes:

$$D(g^*,f) = \int g^*(x)lng^*(x)dx - \int g^*(x)lnf(x)dx \tag{11}$$

This means that the expression $\int g^*(x)lnf(x)dx$ has to be maximized, that is to find the solution of the problem:

$$\max_{\boldsymbol{v}} \int g^*(x)lnf(x)dx \tag{12}$$

If (9) is used, the optimization problem becomes:

$$\max_{\boldsymbol{v}} \int I_{\{S(\mathbf{X})\geq\gamma\}}\frac{f(x;\mathbf{u})}{l}lnf(x;\boldsymbol{v})dx \tag{13}$$

that is equivalent to:

$$\max_{\boldsymbol{v}} D(\boldsymbol{v}) = \max_{\boldsymbol{v}} EI_{\{S(\mathbf{X})\geq\gamma\}}lnf(\mathbf{X};\boldsymbol{v}) \tag{14}$$

If the distribution of the random variables belongs to a natural exponential family, then the solution of the given optimization problem (14) could be derived analitically.

When Kullback-Leibler distance is minimized, the most accurate estimation of l will be obtained for the given simulation effort. Suppose that γ has been chosen to be close to the maximum of the function $S(\mathbf{X})$. The most accurate estimation of l will be for the case when function $f(.;\boldsymbol{v})$ assigns the most of its probability mass arround the optimal point where the function $S(\mathbf{X})$ has its maximum, that is, when $S(\mathbf{X}\geq\gamma)$. This means that this information could be used in order to approximate the optimal solution.

3.3 Maximum Likelihood Estimation

In this subsection, the relation between optimization problem (14) and the well known Maximum Likelihood Estimation, Fisher [1925], will be illustrated. Maximum Likelihood Estimation is a statistical method making an optimal model according to the given data.

Let x_1, x_2, ..., x_n be the sample data, drawn from the known distribution $f(.,\boldsymbol{v})$, where $\boldsymbol{v}$ is the parameter vector to be tuned in order to find the best fitting model for this data. The likelihood function is defined as:

$$\mathfrak{L}(\boldsymbol{v}) = f_\upsilon(x_1,x_2,...,x_n;\boldsymbol{v}). \tag{15}$$

The solution to this optimization problem is given as:

$$\widehat{\boldsymbol{v}} = arg\max_{\boldsymbol{v}} \mathfrak{L}(\upsilon) \tag{16}$$

The method of maximum likelihood estimation gives υ that maximize the probability density function (15). Assuming that all data are the outcomes from the independent, identically distributed random variables $\mathbf{X}_1,\mathbf{X}_2,...,\mathbf{X}_n$, then multivariate density function could be described as a product of n univariate probability density functions (17).

$$\mathfrak{L}(\boldsymbol{v}) = \prod_{i=1}^{n} f_\upsilon(x_i;\boldsymbol{v}). \tag{17}$$

Furthermore, the optimization problem is invariant to monotonic transformation. If the natural logarithm, that is incereasing function, is used, then (17) could be rewritten into (18)

$$\mathfrak{L}^*(\boldsymbol{v}) = \sum_{i=1}^{n} lnf_\upsilon(x_i;\boldsymbol{v}). \tag{18}$$

The maximization of (18) is similar to the maximization problem given in (14). The only difference is that in (14) there is an indication function. This means that only data that satisfies given conditions (14) will be exclusively involved in the caluculation of the maximum.

The optimal vector parameter for the Gaussian density function, $\upsilon = [\mu^*,\sigma^{*2}]$, could be analitically derived (see Rubinstein et al. [2004]) and the solution is:

$$\mu^* = \frac{1}{n}\sum_{i=1}^{n} x_i$$
$$\sigma^{*2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu^*)^2 \quad (19)$$

Because each random values related to specific control parameter are independant and identically distributed, these formulas could be used for every control parameter.

4. SIMULATION

The algorithm used in the simulation was based on adaptive update of γ. This algorithm was nicely described in Boer et al. [2005].

The posibility of the algorithm usage in a control parameters optimization is shown in three different examples. The first two examples use the same system that is controlled by PI algorithm but the performance functions to be optimized are different. In the third example, the more complex system of invertum pendulum is used.

Let $G(s) = \frac{1}{s+1}e^{-0.2s}$ be the transfer function of the system. This system will be controlled with PI control algorithm given in the form $G(s) = k_p + k_i\frac{1}{s}$. The optimum value of the vector of control parameters $k_u^* = [k_p\ k_i]^T$, obtained by the brutal force search algorithm, that minimizes the performance function

$$J = \int_0^\infty |e(t)|dt, \quad (20)$$

is $k_u^* = [2.99\ 2.94]^T$. The related value of the performance function is $J^* = 13.326$.

The initial parameters used in the Cross-Entropy algorithm are: $N = 1000, \rho = 0.2, \boldsymbol{v}_{k_p} = [\mu_{k_p}\ \sigma^2_{k_p}]^T = [1\ 10]^T$ and $\boldsymbol{v}_{k_i} = [\mu_{k_i}\ \sigma^2_{k_i}]^T = [1\ 10]^T$, where N is the number of samples, ρ the number correlated with adaptation of γ ($N\rho$ is equal to the number of cases when $S(x) \geq \gamma$) and v_{k_p}, v_{k_i} are the vectors of distribution parameters related to control parameters k_p and k_p, respectively.

Fig.1 shows the best performance function convergence through the number of iterations. Fig.2 presents the average history of the best values of the performance function, obtained by 30 different Algorithm runs, through iterations. Further, in order to stress the uncertainties of the algorithm, standard deviations, caluclated from the history of the performance function using 30 different simulations, are given in Fig.3.

In the second experiment, the same system was used but the performance function was:

$$J = \int_0^\infty e^2(t)dt, \quad (21)$$

The optimum value of the vector of control parameters $k_u^* = [k_p\ k_i]^T$, obtained by the brutal force search algorithm, that minimizes the given performance function J is $k_u^* = [4.01\ 2.65]^T$. The related value of the performance function is $J^* = 15.41$.

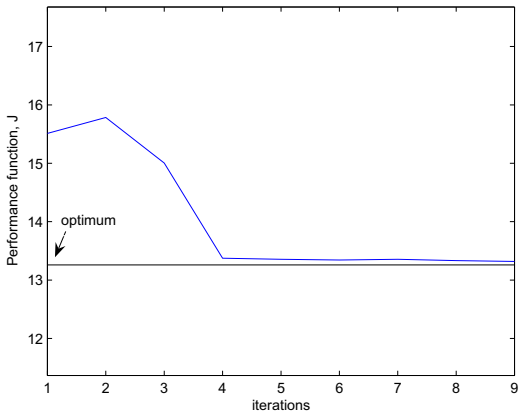

Fig. 1. Performance function through iterations

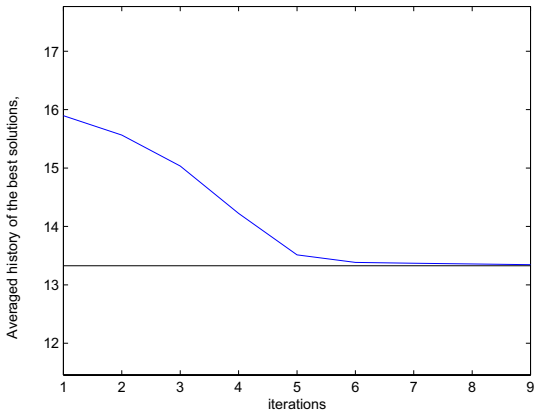

Fig. 2. The averaged history of the best values of the performance function through iterations

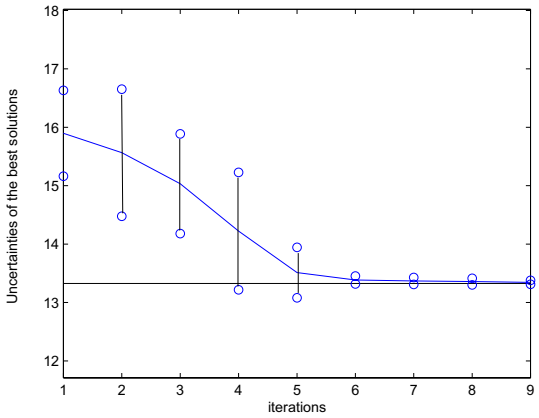

Fig. 3. The uncertaincies of the best values of the performance function through iterations

As in the first experiment, the same diagrams were obtained and are given in Fig. 4 and 5. It could be conlcuded that the convergence rate and the relability of the algorithm are satisfactory.

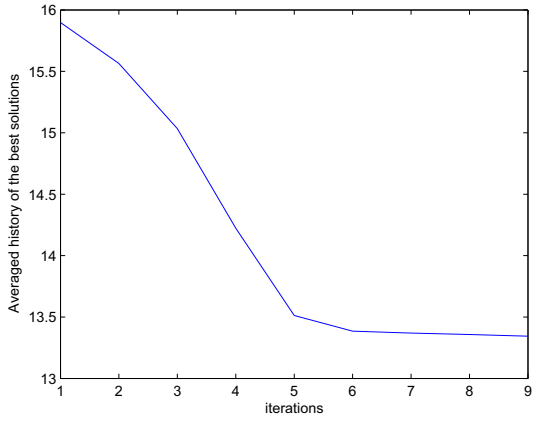

Fig. 4. The averaged history of the best values of the performance function through iterations

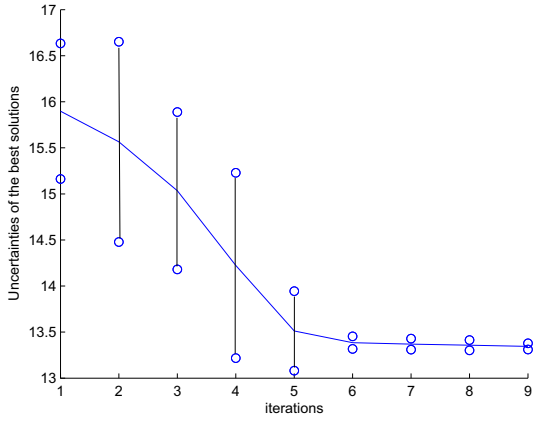

Fig. 5. The uncertaincies of the best values of the performance function through iterations

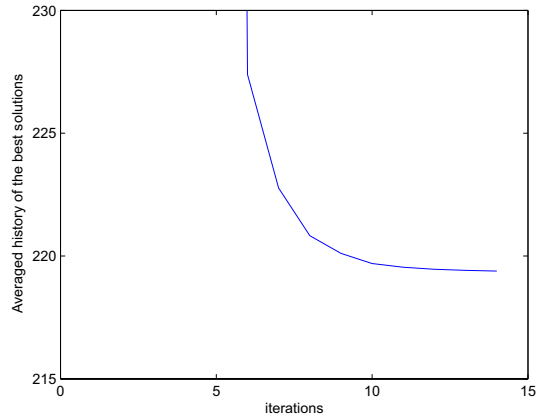

Fig. 6. The averaged history of the best values of the performance function through iterations

In the third experiment, a complex system given in the state space model was used (Fig. 6 and Fig. 5). The system was linearized and the state space matricies are:

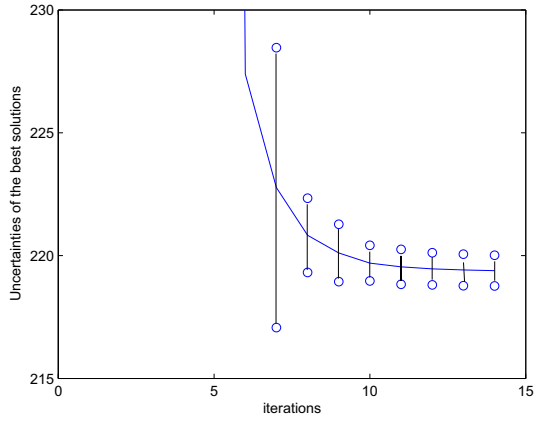

Fig. 7. The uncertaincies of the best values of the performance function through iterations

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -4.905 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 29.43 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0.5 \\ 0 \\ -1 \end{pmatrix} \quad (22)$$

$C = (\,0\ 1\ 0\ 0\,), D = (\,0\,),$

The system was controlled by LQR algorithm using four parameters in the control gain $\mathbf{K}$. The optimal value of the control matrix, for the given case, is:

$$\mathbf{K}^* = [1\ 2.0416\ \ -1.6482\ \ -1.3343].$$

The LQR algorithm optimizes the following performance function:

$$J = \int_0^\infty x^T \mathbf{Q} x + u^T \mathbf{R} u dt, \quad (23)$$

where $\mathbf{x}$ is the state vector, u the control variable, and $\mathbf{Q}$, $\mathbf{R}$ are matrices that penalize the states and the control, respectively. For this experiment, the chosen values of these matrices are: $\mathbf{Q} = \mathbf{I}_{4x4}$ and $\mathbf{R} = \mathbf{I}_{1x1}$, where $\mathbf{I}$ is the identity matrix.

In order to use the Cross-Entropy algorithm, the same LQR control structure and performanse function were used in the simulation. The goal was not to find the optimal control gain K with LQR algorithm to approach the minimum of the performance function (23) but using the Cross-Entropy method. As in the previous experiments, the same diagrams were obtained and given in Fig. 6 and 7. Even for the case where the control algorithm has large number of parameters, the Cross-Entropy method gives satisfactory results.

5. CONCLUSION

The paper presents the possibility of using the Cross-Entropy algorithm in a control system design. This possiblity was shown through systems using controllers with various number of parameters. Also, different performanse functions to be minimized were used. For this purpose, the system of the first order with a time delay was chosen. This system was controlled with PI control algorithm. The method has shown good results for the integrate absolute error performance function (IAE) as well as for

the integrate square error (ISE). The results were compared with the values obtained by the brutal force search algorithm. The rate of the algorithm convergence and the method relability were also concluded in the paper. This conclusion was derived from the data obtained from 30 different algorithm runs. The main advantage of this algorithm comparing to other techniques of combinatorial optimization is relatively large learning rate. Also, the same conclusion has been derived from the more complex case where a multivariable system was controlled by LQR controller. This controller had four parameters whoos optimal values could be easly obtained by LQR algorithm. The performance function that the LQR algorithm uses is also an integrate square error criteria. This performance function was used for the Cross-Entropy method as well. The conclusion about convergence and the reliability of the method, considering 30 different algorithm runs, was also presented for this case. Furthermore, it should be emphesises that the convergence rate is rather independent on the number of parameters.

REFERENCES

R.Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operation Research*, volume 99, pages 89–112, 1997.

R.Y. Rubinstein. The simulated entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, volume 2, pages 127–190, 1999.

R.Y. Rubinstein. Combinatorial optimization, cross-entropy, ants and rare events. In S. Uryiasev and P. M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications* , 304–358. Kluwer, 2001.

R.Y. Rubinstein and D.P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer-Verlag, New York, 2004.

E.H.L. Aarts and J.H.M. Korst. *Simulated Annealing and Boltzman Machines*. John Wiley & Sons,1989.

F. Glover and M. Laguna. *Modern Heuristic Techniques for Combinatorial Optimization*, chapter 3: Tabu search. Blackwell Scientific Publications, 1993.

D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Lerning*. Addison Wesley, 1989.

E.T. Jaynes. Information Theory and Statistical Mechanics. *Phisical Review*, volume 106, pages 620–630, 1957.

J.E. Shore and R.W. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions*, volume IT-26, pages 26–37, 1980.

S. Kullback. *Information Theory and Statistics*. Gloucester, MA:Peter Smith, 1968.

R.M. Gray. *Entropy and Information Theory*. New York:Springer-Verlag, 1990.

T.M. Cover and J.A. Thomas. *Elements of Information Theory*. New York:Wiley, 1991.

R.A. Fisher. Theory of Statistical Estimation. *Proceedings of the Cambridge Philosophical Society*, volume 22, pages 700–725, 1925.

P.T. Boer, D.P. Kroese, S. Mannor and R.Y. Rubinastein. *A Tutorial on the Cross-Entropy Method*.

P.J. Smith, M. Shafi and H. Gao. Quick Simulation: A review of importance sampling techniques in communication systems. *IEEE J.Select.Areas Commun.*, volume 15, pages 597–613, 1997.