Estimation of Symmetric Positive Definite of Matrices using Evolution Strategy for Differential Games

Hector Vargas[1], Vittorio Zanella[1], Vladimir Alexandrov[2], Mónica López[3] and Lorna Rosas[1]

[1] Autonomous Popular University of Puebla State, Computer Engineering, Santiago 1103, Puebla,Pue. 72160, Mexico
{hectorsimon.varga, vittorio.zanella, lornaveronica.rosas}@upaep.mx
http://www.upaep.mx

[2] Moscow State University, Mechanics and Mathematics, Glavnoe Zdanie, Leninskiye Gory, Moscow, 119899, Russian Federation
valexv@mech.math.msu.su
http://www.msu.ru

[3] Autonomous University of Puebla, Computer Engineering, University City, Puebla,Pue. 72570, Mexico
moniarchundia@cs.buap.mx
http://www.cs.buap.mx

Abstract. Estimation of symmetric positive definite of matrices is required when solving a variety of control problems including differential game theory, robotic control, smart structure control, and intelligent control. In differential game theory, the minimization or maximization of a functional is carried out, which is provided by means of the criterion of a quadratic integral whose parameters are: the coordinates of the dynamic system, the control information and the symmetric positive definite class, and the accuracy of its estimate directly affect control performance. These symmetric positive definite of matrices are obtained based on the engineer's experience on one dynamic system in particular. It is clear that the *good* choice of matrices, in an enclosed set, should improve the criterion's objective. In order to automate the choice of symmetric positive definite of matrices, we propose an algorithm that uses evolution strategy. The important result is obtained by means of two propositions which show that the operations of recombination and mutation preserve the symmetric positive definite properties of the matrices, thereby improving the computational complexity of said algorithm.

1 Introduction

In the theory of differential games [1], the problem of computational testing can be built and solved [2] by applying the concepts of minimax and maximin, that is, by obtaining an algorithm that evaluates and corrects the control strategy in order to stabilize a controllable dynamic system. During the process of building the test algorithm, the expert engineer must choose, based on his experience, confidence and

© L. Sánchez, O. Espinosa (Eds.)
Control, Virtual Instrumentation and Digital Systems.
Research in Computing Science 24, 2006, pp. 189-197

intuition, the positive definite matrices that obey the necessities of the dynamic system, taking into consideration its physical nature. The present article presents a way of automating the experience of the human expert by using evolution strategy (ES) [3], [4], [6]. The important result is achieved through two proposals which refer to the fact that the recombination and mutation operations preserve the positive definite properties of the matrices, thereby improving the computational complexity of said algorithm.

2 Statement Problem

We consider the following ([2],[7],[8]) for the computer tests: we suppose that an unknown, control algorithm exists, from which we are only able to know its output $u_I = u(t,x)$, and we furthermore want to organize a computer testing system in order to know how good this algorithm is. In other words, if we wanted to evaluate the said algorithm, how could we obtain an excellent evaluation with which we could compare the obtained evaluation with the said algorithm? Under the same conditions as in the minimax case [2], the maximin problem can be proposed:

$$\inf_{u_1 \in U} \Im(x(t_0), p, u_1) \rightarrow \sup_{\substack{x(t_0) \in X_0 \\ p \in P}} \tag{1}$$

or its equivalent:

$$\inf_{u_1 \in U} \Im(x(t_0), p, u_1) \rightarrow \sup_{\substack{|x(t_0)| \le \upsilon \\ p \in P}} \tag{2}$$

We know that the following inequality is fulfilled [9]:

$$\sup_{\substack{|x(t_0)| \le \upsilon \\ p \in P}} \inf_{u_1 \in U} \Im(x(t_0), p, u_1) \le \inf_{u_1 \in U} \sup_{\substack{|x(t_0)| \le \upsilon \\ p \in P}} \Im(x(t_0), p, u_1) \tag{3}$$

We can therefore consider the following as an excellent grade or evaluation for the stabilization algorithm:

$$\wp = \sup_{\substack{|x(t_0)| \le \upsilon \\ p \in P}} \inf_{u_1 \in U} \Im(x(t_0), p, u_1) \tag{4}$$

where

$$\Im = \int_{t_0}^{t_1} \left[x^T G x + \Delta u^T N \Delta u \right] dt \tag{5}$$

$G > 0$, $N > 0$ - are with $x = A(p)x + B(p)u$, $x(t_0) = c$, $p \in P$.

These matrices G y N are obtained based on the engineer's experience, which will balance the relative importance of the input and state in the cost function that you are trying to optimize. Then, the algorithm will be able to obtain this excellent evaluation and sometimes it will not, depending on whether a seat point exists in the expression of the particular case being analyzed (3). Figure 1 shows how this maximin test is carried out. The control system behaves like a black box, in which we only know the input information $x(t)$ and the output, which would be the control expression u_I.

The inputs in the testing system are the control *u*, and the system deviations *x(t)*. This block executes the stabilization quality computer tests. It will then be possible to obtain an evaluation of the stabilization algorithm, information about the initial conditions $x(t_0)$ and the perturbations *P*.

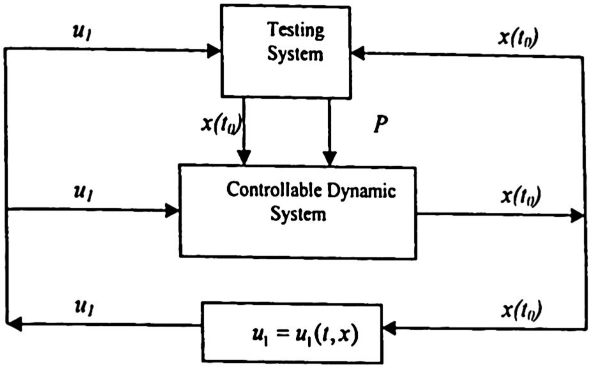

Fig. 1. Diagram of Computer Testing System (CTS).

These tests are organized into three stages:

Stage 1: Solution to the maximin problem: to obtain the excellent solution $\Im^0$; and the solution $x^0(t_0), p^0$ from the maximin problem, which will be used as a strategy in the second stage.

Stage 2: To obtain $\tilde{\Im}$, an estimate of the real evaluation obtained by the stabilization algorithm.

Stage 3: Comparison of $\Im^0$ with $\tilde{\Im}$. The closer the estimate comes to the excellent evaluation, the better the control algorithm will evaluate.

3 Problem Solution

This approach uses an Evolution Strategy (ES) to make Computer Testing System CTS parameters optimization. ES have the following characteristics, desired to parameters optimization of dynamic processes: (a) they are good on real values parameters optimization [5]; (b) usually they are used with self-adaptation of mutation parameters [5]. This is desired because the fitness surface is not known and it can change (in accordance with variations of testing data characteristics); (c) they have low takeover time [5]. This characteristic permit a fast adaptation to the moment characteristics.

This is good for on-line optimization because it is not desired that the population has not became adequate before other changes occur in data characteristics. The used ES utilizes gaussian perturbation mutation with self-adaptation of the standard deviations and one standard deviation for each variable representing an CTS parameter to be optimized. This kind of mutation is adequate for ordinal numeric parameters optimization.

Therefore, the definitions of the recombination and mutation operations will be given, and the fact that these operations preserve the properties of the positive definite matrices will be demonstrated. The section will conclude with the evolution algorithm.

Definition 1 (Representation and Fitness Evaluation): Search points in ES are symmetric defined positive matrices, that is:

$$\Xi = \{G \mid G \subseteq \Re^n \times \Re^n,\ x^T G x > 0\ \forall x \neq 0\ x \in \Re,\ G = G^T\} \tag{6}$$

Given the objective function $\Im : \Xi \to \Re$ (note that $\Im$ is defined as (5), the fitness function Φ is in principle identical to $\Im$, i.e. given an individual $\vec{a} \in I$, we have

$$\Phi(\vec{a}) = \Im(G) \tag{7}$$

Here G is the object variable component of $\vec{a} = (G, \vec{\sigma}, \vec{\alpha}) \in I = \Xi \times A_s$, where

$$\begin{aligned} A_s &= \Re_+^{n_\sigma} \times [-\pi, \pi]^{n_\alpha} \\ n_\sigma &\in \{1, \ldots, n\} \\ n_\alpha &\in \{0, (2n - n_\sigma)(n_\sigma - 1)/2\} \end{aligned} \tag{8}$$

Besides representing the object variable G, each individual may additionally include one up to n different standard deviations σ_i as well as up to $n \cdot (n-1)/2$ rotation angles $\alpha_{ij} \in [-\pi, \pi]$ $(i \in \{1, \ldots, n-1\}, j \in \{i+1, \ldots, n\})$, such that the maximum number of strategy parameters amounts to $\omega = n \cdot (n+1)/2$. For the case $1 < n_\sigma < n$, the standard deviations $\sigma_1, \ldots, \sigma_{n_\sigma - 1}$ are coupled with object variables $G_{1,1}, \ldots, G_{n_\alpha - 1, n_\alpha - 1}$ and σ_{n_σ} is used for the remaining variables $G_{n_\sigma, n_\sigma}, \ldots, G_{n,n}$.

Definition 2 (Mutation): An individual space $I = \Xi \times \Re^n \times \Re^{n \cdot (n-1)/2}$ is assumed. Mutation $m_{\{\tau, \tau', \beta\}} : I^\lambda \to I^\lambda$ is an asexual operator, which yields a triple $\vec{a}' = (G', \vec{\sigma}', \vec{\alpha}')$ when applied to a particular individual $\vec{a} = (G, \vec{\sigma}, \vec{\alpha})$. Then, mutation is formalized as follows

$$\begin{aligned} \sigma_i' &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ \alpha_j' &= \alpha_j + \beta \cdot N_j(0,1) \\ G' &= G + \vec{N}(\vec{0}, C(\vec{\sigma}', \vec{\alpha}')) \end{aligned} \tag{9}$$

The factors τ, τ', and β in equation (9) are rather robust parameters, which Schwefel suggests to set as follows [11]:

$$\begin{aligned} \tau &\propto \left(\sqrt{2\sqrt{n}}\right)^{-1} \\ \tau' &\propto \left(\sqrt{2n}\right)^{-1} \\ \beta &\approx 0.0873 \end{aligned} \tag{10}$$

Usually, the proportionality constants for τ and τ' have the value one, and the value suggested for β (in radians) equals 5°. The $C(\vec{\sigma}',\vec{\alpha}')$ is the covariance matrix with diagonals elements $c_{ii} = \sigma_i^2$, i.e. the variances.

Finally, for mutation of the object variable G the resulting vector $\vec{\sigma}'$ and $\vec{\alpha}'$ are used to create the random object for modifying G, i.e. for its properties of G, it decompose in $G = U\Lambda U^T$, so the mutation operation: $\forall \lambda_i \in \Lambda$ (eigenvalues) with $i = 1,2,\ldots,n$, it is carried out $\vec{\lambda}' = \vec{\lambda} + \vec{N}(\vec{0}, C(\vec{\sigma}',\vec{\alpha}'))$ such that $\vec{\lambda}' > 0$, this way $G' = U\Lambda' U^T$ where Λ' contains them λ_i for $i = 1,2,\ldots,n$, and the new matrix $G' = G + \vec{N}(\vec{0}, C(\vec{\sigma}',\vec{\alpha}'))$ is the result of the mutation that suffer its eigenvalues of the original matrix G.

Definition 3 (Recombination) An individual space $I = \Xi \times \Re^n \times \Re^{n\cdot(n-1)/2}$ is assumed. Recombination $r_{r_x,r_\sigma,r_\alpha} : I^\mu \to I^\lambda$ is an sexual operator. Then, recombination have sexual form, act on two individuals randomly chosen from the parent population, where choosing the same individual twice for creation of one offspring individual is not suppressed. So, we know that G decomposed in $G = U\Lambda U^T$ and $G' = U'\Lambda' U'^T$ is other object variable, then we can express the recombination operator as: $r_{r_x,r_\sigma,r_\alpha}(G_1,G_2) = U_j \Lambda_i U_j^T$ where Λ_i represents an exchange of a subset of n selected elements of a group of $2n$ elements – these elements are $|\Lambda_1 \cup \Lambda_2| = 2n$, this way $i = 1,2,\ldots,\frac{(2n)!}{(2n-n)!}$ and j is alternated by each exchange that is carried out in the group $\{1,2\}$.

Now, when there is obtained a new population of his predecessors, a lot of care must be had if the operations of recombination or mutation did not alter the genetics of the individuals, an option is to verify in each step of the algorithm, that it generates a new population, if there were alterations, another option, which is described in this work, is to demonstrate that these operations do not degenerate the new populations, avoiding the checking in all step of the algorithm and improving the yield of the same one, this is, we improve its computational complexity.

Proposition 1: The operation of recombination preserves the properties of the positive defined matrix.

Proof: Let $X \subset \Xi$ be, for everything $A \in X$, you can express $A = U\Lambda U^T$, and then the rotation $y = U^T x$ produces the sum of squares

$$\begin{aligned} x^T A x &= x^T U \Lambda U^T x = y^T \Lambda y \\ &= \lambda_1 y_1^2 + \lambda_2 y_2^2 + \ldots + \lambda_n y_n^2 \end{aligned} \tag{11}$$

Now, of the equation (8) those λ_i are substituted for $\lambda_i' > 0 \;\; \forall i = 1,2,\ldots n$

$$\lambda_1' y_1^2 + \lambda_2' y_2^2 + \ldots + \lambda_n' y_n^2 = y^T \Lambda' y = \\ = x^T U \Lambda' U^T x = x^T A' x \tag{12}$$

Where $\lambda_i' > 0$ they are real, then A' it is defined positive. Without loss of generality, one can think that the new values $\lambda_i' > 0$ are the exchanges that are carried out in the operation of crossover to generate to the new matrix C_j for $j = 1,2,\ldots,\frac{(2n)!}{(2n-n)!}$. Reason why, we concludes that the operation of recombination preserve the properties of defined positive symmetrical. ♣

Proposition 2: The operation mutation conserves the properties of the positive defined matrices.

Proof: Let $X \subset \Xi$ be, for everything $A \in X$, then all the values characteristic of the matrix A they are positive, this is $\lambda_i > 0$, for $\forall i = 1,2,\ldots n$, let us suppose that for each one $\lambda_i > 0$ has the corresponding unitary own vector, then one has that

$$Ax_i = \lambda_i x_i \text{ so that} \\ x_i^T A x_i = x_i^T \lambda_i x_i = \lambda_i \tag{13}$$

Since, $x_i^T x_i = 1$ and $x^T A x > 0$ $\forall x \neq 0$, it will be valid in particular for the own vector x_i and the quantity $x_i^T A x_i = \lambda_i$. Now, of the equation (10) we added to those λ_i, $\beta_i \in N(0,1)$ this is $\gamma_i = \lambda_i + \beta_i$, such that $\gamma_i > 0$, then one has that

$$0 < \lambda_i + \beta_i = x_i^T(\lambda_i + \beta_i)x_i = x_i^T A' x_i \tag{14}$$

This way, the sum of a normalized aleatory number $\beta_i \in N(0,1)$, doesn't affect the property of defined positive of the matrix A original, or the same thing that the new matrix is defined positive. ♣

A (μ,λ)-strategy is required in order to facilitate extinction of maladapted individuals. Selective pressure may not become too strong, i.e., μ is required to be clearly larger than one. Recombination on strategy parameters is necessary (usually, intermediate recombination gives best results).

Algorithm

$t := 0$

Initialize $P(0) := \{\vec{a}_1(0),\ldots,\vec{a}_\mu(0)\} \in \Xi \times \Re^n \times \Re^{n\cdot(n-1)/2}$

Evaluate $P(0) : \{\Phi(\vec{a}_1(0)),\ldots,\Phi(\vec{a}_\mu(0))\}$ where $\Phi(\vec{a}) = \Im(G)$

While $(\iota(P(t)) \neq true)$ **do**

Recombine: $\vec{a}_k'(t) := r'(P(t))$ $\forall k \in \{1,\ldots,\lambda\}$

Mutate: $\vec{a}_k''(t) := m'_{\{\tau,\tau',\beta\}}(\vec{a}_k'(t))$ $\forall k \in \{1,\ldots,\lambda\}$

Evaluate: $P''(t) : \{\Phi(\vec{a}_1''(0)),\ldots,\Phi(\vec{a}_\lambda''(0))\}$

Select: $s_{(\mu,\lambda)}(P''(t))$

$t := t + 1$

od

4 Results

We consider the following mathematical model:

$$\ddot{x} = u_1 \qquad u_1 = -k_1 x_1 - k_2 \dot{x}_1 \tag{15}$$

We wish to solve the problem

$$\max_{|x(t_0)| \le \upsilon} \int_0^{\infty} \left(x_1^2 + \dot{x}_1^2 + \ddot{x}_1^2 \right) dt \to \min_{u_1 \in U} \tag{16}$$

whose physical sense is the following: given the worst initial conditions, we want to minimize the system's deviation, as well as the velocity and acceleration of these deviations.

In this case

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad K^T = (k_1, k_2) \quad x^T = (x_1, \dot{x}_1) \tag{17}$$

The expert specialist chooses the following positive, defined matrix

$$G^T = G = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix} \quad N = 1 \tag{18}$$

In this example, the seat point exists, with is fulfilled such that

$$\min_{K \in Q} \max_{|x(t_0)| \le \upsilon} \Im = \max_{|x(t_0)| \le \upsilon} \min_{K \in Q} \Im \quad \text{where}$$
$$\Im = \int_0^{\infty} \left(x_1^2 + \dot{x}_1^2 + \ddot{x}_1^2 \right) dt \tag{19}$$

After verifying that System (9) is completely controllable, we proceed to the calculation of internal Problem [2], that is:

$$\min_{K \in Q} \Im = \min_{K \in Q} \int_0^{\infty} \left(x_1^2 + \dot{x}_1^2 + \ddot{x}_1^2 \right) dt = x^{oT}(t_0) L_0 x^0(t_0) \tag{20}$$

Therefore, the external problem $\max_{|x(t_0)| \le \upsilon} x^{oT}(t_0) L_0 x^0(t_0) = \mu_{\max}$ is solved. Thus, an optimum value for the functional $1+\sqrt{3}$ was obtained for the worst initial conditions. The graph is shown below.

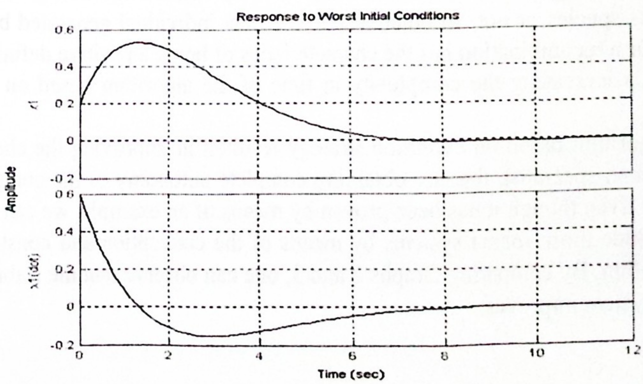

Fig. 2. Asymptotic behavior of the solution when the positive, defined matrix G has been fixed, chosen by a human specialist.

Now we present the result of applying the algorithm based on evolution strategy, in order to automatically choose the best, positive definite matrix G without the intervention of a human expert.

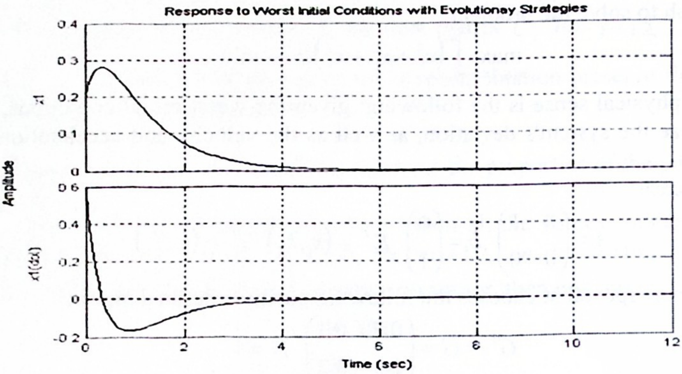

Fig 3: Results of the behavior of the system's coordinates, for the best choice of the pondering matrix, given the worst initial condition.

When the functional does not change in the next generations, this means that we have reached the best individual, that is, the best, positive definite matrix *G*, which, when applied to the functional, reduces our stabilization time, as is shown in the following graph.

5 Conclusions

The two proposals demonstrate that the recombination and mutation operations preserve the structures of the positive definite matrices. This means that within each iteration of the algorithm, there is no need to verify whether the new generation belongs to its species or not. In other words, the new individual generated by either a mutation or a recombination has the characteristics of being a positive definite matrix. This avoids increasing the complexity in time of the algorithm based on evolution strategies.

The algorithm based on evolution strategy resulted in improving the choice made by the human specialist, thereby obtaining complete autonomy of the computer tests algorithm. Even though it has been proven by means of an example, we can extend its use to include more robust systems by means of the conception and construction of said algorithm. By comparing Graphs 2 and 3, one can observe that the stability of the dynamic system improves.

References

1. N.N. Krasovskii and A.I. Subbotin, Game- Theoretical Control Problems, Springer-Verlag, 1988
2. Alexandrov V. V., Guerra L., Kaleonova I., Trifonova A., Minimax Stabilization and Testing Maximin. University Bulletin, Moscu No.5, 1999
3. Agapie, J. K. Hao, E. Ronald, M. Shoenauer, and D. Snyers, Genetic algorithms: Minimal conditions for convergence. In, Artificial Evolution: Third European Conference; selected papers, pp. 183 - 1943. Berlin Springer 1998
4. Coello, C.A.C.,Christiansen, A.D., and Aguirre, A.H., Using a New GA-Based Multiobjective Optimization Technique for the Design of Robot Arms, Robotica, 16(4), pp. 401-414,1998
5. Eiben, A.E., Smith, J. E. *Introduction to Evolution Computing.* Berlin: Springer, 2003.
6. Coello, C.A.C. and Christiansen, A.D., Multiobjetive optimization of trusses using genetic algorithms, Computers and Structures, 75(6), pp. 647-660, 2000
7. Alexandrov V. V., Guerra L., Sobolevskaya I., Trifonova A., Vargas H., Control Algorithms in Mechatronics, Mathematical Modeling of complex Information Processing Systems, Moscow University Press, 2001
8. Alexandrov V. V., Guerra L., Vargas H., Optimization and Computer Aided Testing of Stabilization Precision, II International Congress of Numerical Methods in Engineering and Applied Sciences, pp. 389-399, January 2002
9. Demyanov V. F. and Malozemov V.M., Introduction to Minimax, Nauka Editorial, Moscu, 1972
10. Eiben, A.E., Smith, J. E. *Introduction to Evolution Computing.* Berlin: Springer, 2003
11. H. P. Schwefel, Numerische Optimierung von Computer-modellen mittels der Evolutionsstrategie, volume 26 of Interdisciplinary Systems Research. Birkhäuser, Basel, 1977