IFAC/IFORS/IIASA/TIMS

The International Federation of Automatic Control
The International Federation of Operational Research Societies
The International Institute for Applied Systems Analysis
The Institute of Management Sciences

SUPPORT SYSTEMS FOR DECISION AND NEGOTIATION PROCESSES

Preprints of the IFAC/IFORS/IIASA/TIMS Workshop

Warsaw, Poland

June 24-26, 1992

Editors:

Roman Kulikowski
Zbigniew Nahorski
Jan W.Owsiński
Andrzej Straszak

Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland

VOLUME 1:

Names of first authors: **A-K**

SYSTEMS RESEARCH INSTITUTE, POLISH ACADEMY OF SCIENCES

SUPPORT SYSTEMS FOR DECISION AND NEGOTIATION PROCESSES
Preprints, IFAC/IFORS/IIASA/TIMS Workshop, June 24-26, 1992, Warsaw, Poland

Optimization of Multiple Objectives in Control of Uncertain Systems

Allon Guez, Ziauddin Ahmad

Drexel University, Dept. of Electrical and Computer Engineering
32nd and Chestnut St., Philadelphia, PA 19104, USA.

ABSTRACT

We believe that an essential feature in machine learning is the real time satisfaction of multiple objectives such as identification, tracking etc. The machine learning problem may be viewed as a nonlinear adaptive control problem where the environment plays the role of the 'plant', while the learner is the controller. Multiobjective optimization (MOO) in the control problem typically deals with simultaneous optimization of more than one objective, where each objective is described via a cost functional. In such a situation there often exists a region of tradeoff wherein one cost may be improved at the expense of others. Such a region is called the Pareto optimal (PO) set. A parameterization of this set simplifies the attainment of the existing tradeoff. Working within the Pareto set guaranties optimum tradeoff. We present two examples for linear time invariant systems. These examples help illustrate different issues involved in this matter.

2. BACKGROUND

In this paper we are mainly concerned with the conflict between control and learning/identification that has long been identified, see for example ([Selinsky-89],[Guez-89b]). Previous work by different researchers include stable parameter updating laws only to accomplish tracking, while others such as ([Bayard-85],[Gerencsér-90],[Caines-84],[Gustavsson-77],[Mo-90]) also want to identify the parameters, but in both cases the data resulting through error in tracking the current trajectory is utilized for the purpose of updating the parameters. We present the approach where we utilize not only the current trajectory but also the repertoire of possible trajectories, termed as Exploratory Schedules (ES), in order to minimize the overall error over an extended period of time. In ([Guez-89],[Guez-],[Selinsky-89],[Guez-89b]) ES were generated off-line and used in an open loop fashion. Moreover, these ES were used inbetween actual control tasks therefore limiting the process of estimation during idle time. Here we attempt to generate ES in a closed loop manner. Such trajectories in general may not be the desired trajectories, resulting in larger tracking errors. However, ES offer faster convergence to the system parameters and therefore yield smaller long term tracking errors. The automation for the design of ES requires online modification of the desired trajectory to enhance learning at the expense of poorer initial tracking. Neurocomputing, exploratory schedules and multiobjective optimization form essential ingredients in the solution to this problem.

2.1. Learning in Neurocomputing

Learning in neurocomputing is the process of finding the 'correct' neural network architecture; e.g. searching for the best synaptic strengths and threshold values. For example, let the Neural Network (NN) input vector, at time t, be x(t) the NN output vector be y(t), the NN parameters describing its architecture (e.g. synaptic weights) be p(t), and let the input-output mapping defined by the NN for a given parameter vector value be:

$$y = N(p,x) \qquad (1)$$

Also let $\{x_{Di},y_{Di}\}$ be the i-th pair of samples of the desired input output map to be learned. Learning takes place by adjusting the NN architecture parameters p(t) according to a given learning algorithm:

$$\frac{dp(t)}{dt} = L(p(t),x_{Di},y_{Di}) \tag{2}$$

Learning (i.e. computing equation (2) for a sequence of data samples $\{x_{Di},y_{Di}\}_{i=1}^{S}$ may be done online or offline, depending upon time of availability of the data, computing power and other factors.

At present, research activities in learning for neural network focus on defining new learning algorithms (i.e. new learning operators $L(p(t),x_{Di},y_{Di})$), evaluating their performance in terms of what mappings are 'learnable', evaluating their computational complexity and cost and study their hardware implementation aspects. However, several additional important issues in NN and machine learning remain open and significant. We attempt to describe some of those issues below.

2.2. The Exploratory Schedule Design Problem

From equation (2) we see that the performance of the learning algorithm L clearly depends on the samples $\{x_{Di},y_{Di}\}$. Thus, important issues in NN learning are:

a) What is the 'best' sequence of learning examples (data samples) $\{x_{Di},y_{Di}\}$ for a given learning rule, L?
b) How do we define 'best' samples for NN learning?
c) How do we generate/construct the best learning samples? Can we and should we generate the samples 'on the go' i.e. in real time?

Although problem a) has been recognized in the past, no research results were found regarding its solution or regarding posing or solving problems b) and c). In our attempt to study the above issues we introduced the concept of Exploratory Schedules (ES). ES is a sequence of inputs to any learning algorithm (e.g. to $L(p(t),x_{Di},y_{Di})$ in (2)) whose purpose is to make that learning algorithm efficient. The ES found so far, are expressed as functions of time (open loop ES) [Selinsky-89],[Guez-89b]. We believe that ideally the best ES must be generated in closed loop (feedback form) and in real time, i.e. ES should be a function of the present performance or state of the NN. Put in other words "how one learns depends on what one knows". As we acquire more knowledge our strategy of best future learning may change in a way which depends heavily on our present state or knowledge. Thus we believe, best NN learning cannot be achieved by following a preplanned set of examples computed offline ahead of time, but rather, must be generated online depending upon the current state of the NN, that is in a feedback form (closed loop).

2.3. Pareto Optimality

A dynamic process can be expressed as

$$\dot{x} = f(x,u) \tag{3}$$

where $x \in A \subset R^n$ is the state vector with initial conditions given as $x(t_1) = x_1$ and final conditions given by $x(t_2) = x_2$, and $u \in U \subset X$, is the input vector. Here, X is a real linear topological space. In an optimal control problem we define a cost functional $J(u(\cdot))$ as

$$J(u(\cdot)) = \int_{t_1}^{t_2} C(x,u)\, dt \tag{4}$$

where $C(x,u) : A \times U \rightarrow B \subset R_+$ which is to be optimized resulting a local minimum value of $J(u(\cdot))$ corresponding to $u^*(\cdot)$. However, if we have more than one cost functional, $J_i(u(\cdot))$, $i = 1,\ldots, p$, given as

$$J_i(u(\cdot)) = \int_{t_1}^{t_2} C_i(x,u)\, dt \qquad (5)$$

then the minima for all $J_i(u(\cdot))$ may not occur for the same $u(\cdot)$. In this case we can talk about a region $U_1 \subset U$ such that the resulting $J_i(u(\cdot))$ are noninferior or unimprovable in the sense that for all $u(\cdot), \bar{u}(\cdot) \in U_1$ if $J_i(u(\cdot)) \leq J_i(\bar{u}(\cdot))$ then there exists $J_j(u(\cdot)) \geq J_j(\bar{u}(\cdot))$ for $j \neq i$. This region, U_1, is the Pareto optimal set. Formally, in the control problem, Pareto optimum is the set of controls $u^*(\cdot) \in U \subset X$, such that for the cost function $J(u(\cdot)) : X \rightarrow G \subset R^q$ and for all $u^*(\cdot)$-comparable $u(\cdot) \in U$, $J(u(\cdot)) \leq J(u^*(\cdot)) \Rightarrow J(u(\cdot)) = J(u^*(\cdot))$. Therefore, Pareto optimal solutions are also called unimprovable, and in general we obtain a set of such solutions.

A number of methods have been proposed to obtain the PO solution or the set (full or partial) of such solutions. Some of these methods are outlined below.

Weighting Function Method [Zadeh-63],[Cohon-78]
In this method the weighted sum of objectives is minimized:

$$P(w) = \min_{x \in X} \sum_{j=1}^{p} w_j f_j(x) \qquad \sum w_j = 1; \qquad w_j \geq 0$$

ε-Constraint Method [Haimes-73], [Marglin-67]
This method considers minimization of the kth objective while the remaining objective functions are constrained from above by some values ε_i. Conditions on these constraints are that these should be feasible and binding at the optimal solution.

Proper Equality Constraint Method [Lin-76]
Here the kth objective is minimized such that the remaining objectives keep a specified values ε_i. ε_is are varied to generate the entire Pareto set.

Hybrid Method [Corley-80], [Wendel-77]
In this method the weighted summation of objective functions is minimized while constraining from above all the objectives by some specified values.

The Best Compromise Method [Geoffrion-68], [Yu-73], [Zeleny-74], [Zeleny]
The best compromise solution is the solution that minimizes the following function

$$d_\alpha = \left\{ \sum_{k=1}^{p} w_j \, |f_k^* - f_k(x)|^\alpha \right\}^{1/\alpha}$$

where: $\quad f^*_j = \min_{x \in X} f_j(x)$

where $1 \leq \alpha \leq \infty$. For fixed w_j, $j \in \{1, \ldots, p\}$, The entire range of α results in a subset of the PO set.

Multiobjective Simplex Method [Cohon-78], [Chankong-84]
This is a direct approach, similar to the single objective simplex procedure, in obtaining the PO set for linear programming problems.

The following restrictions are associated with the weighting function method

(a) Solution obtained should be unique or weights should be positive. This is irrespective of the convexity of f and X.

(b) f, X convex then for x* a nondominated solution $\exists$ w such that P(w) results in x*. Such w is not unique in general.

In the weighting function method a change of preferences may be mimicked by an alternate set of weights.

In the above discussion we were concerned with 'global Pareto solutions'. However, there also exists the notion of local Pareto optimality. A point, x*, is locally Pareto optimal if it is Pareto optimal in some neighborhood, $N_\delta(x^*)$. Local Pareto optimality comes into play in the case of nonconvex cost functions. In this case not all weighted combinations of the objective functions result in a Pareto optimal solution (see example No. 2 below).

3. UNCERTAIN PLANT CONTROL VIA MULTIOBJECTIVE OPTIMIZATION

In this section we present an approach to the control of uncertain plant. In this respect we provide two examples that illustrate the issues that appear in the proposed approach concerning the control of uncertain plants.

The objective is to control a plant in the event when complete information about the plant is not known. The uncertainity in the knowledge of the plant is reflected through the uncertainty in the parameters of the plant. Therefore, the way of controlling the plant is via online identification of the parameters which in turn are employed by the controller in generating the control signal. However, as stated above the there exists a conflict between the problem of identification and the problem of control. We try to resolve this problem of identification vs. control by using the dichotomy of the cost functionals, to be minimized, one for the purpose of control and the other for the purpose of identification. The cost functional for control, J_C, takes into account only the current trajectory, therefore, enabling tracking of the task at hand. The cost functional for identification, J_I, takes into account the repertoire of possible trajectories for the plant via for example an average task. Minimizing such a cost functional would yield lower errors over the plant's life time. Therefore, we call J_I the cost functional related to identification. These cost functionals have the form as given below:

$$J_C = D_C(x,x_f) + \int_{t_0}^{t_f} C_C(x,x_d,u)\, dt \tag{6}$$

$$J_I = D_I(x,\bar{x}_f) + \int_{t_0}^{\infty} C_I(x,\bar{x}_d,u)\, dt \tag{7}$$

where x is the actual state of the system, u is the input, x_d is the desired current trajectory, $\bar{x}_d$ is the average desired trajectory. D_C and D_I are the terminal costs and C_C and C_I are the appropriate cost integrands for control and identification respectively. Simultaneous minimization of J_C and J_I to result in minimum values for both is in general not possible. Therefore, we apply the method of Pareto optimality to achieve an optimal solution.

Example No. 1: LTI System With Convex Costs

Consider a scalar linear time invariant system

$$\dot{x} = a\,x + b\,u \tag{8}$$

Further, for the purpose of illustration we assume that the system parameters, A, B, are known. We define the cost functionals J_C and J_I in (6) and (7) as follows:

$$J_C = (x-x_f)^T M_C (x-x_f) + \int_{t_0}^{t_f} [\ (x-x_d)^T\ Q_C\ (x-x_d) + u^T R_C u\ ]\ dt \qquad (9)$$

$$J_I = (x-\bar{x}_f)^T M_I (x-\bar{x}_f) + \int_{t_0}^{t_f} [\ (x-\bar{x}_d)^T\ Q_I\ (x-\bar{x}_d) + u^T R_I u\ ]\ dt \qquad (10)$$

Here x_d defines a trajectory to be tracked and $\bar{x}_d$ as the average trajectory. Minimizing J_C will result in optimal tracking of x_d , lets call it $x_{(x_d)}$, while minimizing $\bar{x}_d$ results in the optimal trajectory $x_{(\bar{x}_d)}$. Now if we form the following equation

$$x_{(\alpha)} = \alpha\ x_{(x_d)} + (1-\alpha)\ x_{(\bar{x}_d)} \qquad 0 \leq \alpha \leq 1 \qquad (11)$$

we get intermediate trajectories $x_{(\alpha)}$, as α varies between 0 and 1. With $\alpha = 0$ we get $x_{(\alpha=0)} = x_{(\bar{x}_d)}$ while with $\alpha = 1$ we get $x_{(\alpha=1)} = x_{(x_d)}$. J_I and J_C were minimized by dynamic programing of functionals [Kirk-70]. In our simulations $M_C = M_I = 10$, $Q_C = Q_I = 10$ and $R_C = R_I = 0.1$. The resulting trajectories and some of the intermediate trajectories $x_{(\alpha)}$ are shown in figure-1. For such a convex mixing of optimal trajectories we observe that $x_{(\alpha)}$ lies inbetween $x_{(x_d)}$ and $x_{(\bar{x}_d)}$. Therefore, the role of α in $x_{(\alpha)}$ is to define a new trajectory that takes into account both tracking as well as learning. Tracking such a trajectory in an optimal manner, i.e. minimizing

$$J_\alpha = (x-x_{(\alpha)})^T M_\alpha (x-x_{(\alpha)}) + \int_{t_0}^{t_f} C_\alpha(x, x_{(\alpha)}, u)\ dt \qquad (12)$$

where

$$C_\alpha = (x-x_{(\alpha)})^T\ Q_\alpha\ (x-x_{(\alpha)}) + u^T R_\alpha u \qquad (13)$$

will result in a deviation from both the trajectories, $x_{(x_d)}$ and $x_{(\bar{x}_d)}$. Therefore, we define two additional costs, the cost paid for tracking at the expense of learning as:

$$C_T = \int_{t_0}^{t_f} \|\ x_{(x_{(\alpha)})} - x_{(\bar{x}_d)}\ \|^2\ dt \qquad (14)$$

and the cost of learning at the expense of tracking as:

$$C_L = \int_{t_0}^{t_f} \|\ x_{(x_{(\alpha)})} - x_{(x_d)}\ \|^2\ dt \qquad (15)$$

These costs for the given problem relate to the area between $x_{(x_{(\alpha)})}$ and either $x_{(\bar{x}_d)}$ or $x_{(x_d)}$ of figure-1, and are plotted in figure-2 as a function of α, with $M_\alpha = 10$, $Q_\alpha = 10$ and $R_\alpha = 0.1$. Clearly, we observe that while one cost decreases the other cost increases. It has been shown [Da Cunha-67] that in the convex case minimization of such a composition results is a PO solution when the weights are positive and that the PO set is contained in the closure of this set. As stated earlier, weighted combinations of the costs can be made to obtain pareto optimal solutions if the costs are convex. Infact for the example given as above it can be shown that $J_\alpha = \alpha\ J_C + (1-\alpha)\ J_I$ and $u_\alpha = \alpha\ u_C + (1-\alpha)\ u_I$.

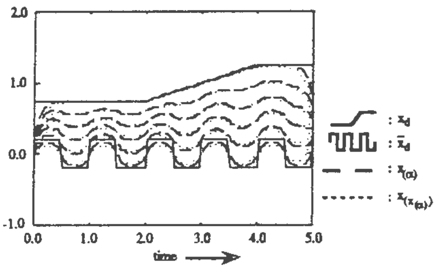

Figure 1: Optimal Trajectories due to J_C and J_I and their convex combinations.

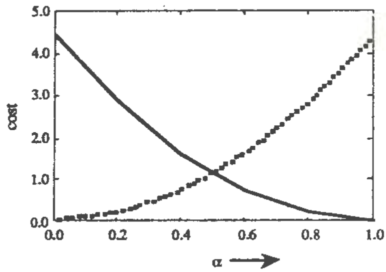

Figure 2: Control Cost (C_T: ——) and Identification Cost (C_I: ······) vs tradeoff factor "α".

The strategy to achieve good tracking requires the ability to choose α in a manner that the deviation from the desired trajectory remains bounded. A diagram illustrating this purpose is shown in figure-3.

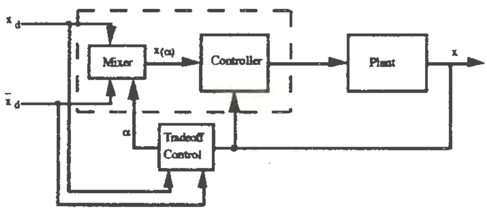

Figure 3: A scheme for the control of uncertain plant.

Example No. 2: LTI System With Nonconvex Costs:

Here we shall consider the control of a DC motor. The motor equation is given by

$$i(t) = \dot{v}(t) + \mu \tag{16}$$

where

i = the motor armature current

v = angular velocity of the motor in rads/sec

μ = acceleration due to friction in rads/sec^2, (constant).

$\dot{v} = \dfrac{dv}{dt}$

with

$v(t_1) = v_1$ (intial velocity) (17a)

$v(t_2) = v_2$ (final velocity) (17b)

Three cost functionals to be minimized are considered, as defined below

$$J_1 = \int_{t_1}^{t_2} dt \quad ; \quad J_2 = -\int_{t_1}^{t_2} v \, dt \quad ; \quad J_3 = \int_{t_1}^{t_2} f(i) \, dt \tag{18}$$

where f(i) is an arbitrary loss function of the armature current. It is seen that individual minimization of these functionals leads to minimizing the time (J_1), maximizing the angular displacement (J_2) and minimizing the energy losses (J_3). The problem statement that we shall consider is:

(P) *Minimize J_2 and J_3 subject to $J_1=t_f$, and equations (1) and (2)*

By employing the weighted convex combination method we form the new objective, J, to be minimized as:

$$J = \beta J_3 + (1-\beta) J_2$$

$$= \int_{t_1}^{t_2} (\beta f(i) - (1-\beta) v) dt \tag{19}$$

along with the isoperimetric constraint $J_1 = t_f$. Here $0 \le \beta \le 1$ is the parameter. For the purpose of illustration we shall consider f(i) in eq. (18) as given below

$$f(i) = (i-i_1)^2 (i-i_2)^2 + a (i-i_3)^2 \tag{20}$$

where i_1, i_2, i_3 are some specified constants. We shall consider positive values for a. Here we are interested in optimizing a nonconvex functional. The reason for the selection of the f(i) as given by (20) is that under some conditions on a, J_3 is nonconvex. For $i_1=-i_2=i_3$, with $i_1 = -1$ and a=0.2, f(i) is as shown in figure 4. These values were used in the simulation. Although, this loss function, (f(i)), is unrealistic it was considered as it results in a nonconvex losses functional, (J_3). Also for this example μ was taken to be zero therefore, from eqn. (16) we observe that the current is equal to the acceleration. Further, t_f was taken as 1 and $v_1 = v_2 = 0$ were considered.

For the problem (P) intermediate function is given as

$$H = \beta f(i) - (1-\beta) v + \lambda \tag{21}$$

The Euler equation

$$H_v - \frac{\partial H_{v'}}{\partial t} = 0 \tag{22}$$

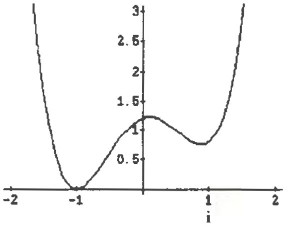

Figure 4: Losses cost integrand, f(i), as a function of current, i.

results in a cubic equation for the acceleration. Solving this equation and plugging the results in eqn. (1) and integrating we get the velocity profiles and eqn. (3) yields the costs J_2 and J_3. Figures (5) and (6) indicate these results graphically for different values of β ranging from 0.1 to 0.9. Figure (5a) shows the plots of the inputs (= current = acceleration). We observe that as the weighting of the losses functional (J_3) is increased by increasing β the maximum positive acceleration is decreased while the maximum negative acceleration increases. This increase in the negative acceleration is accompanied by a shift of the acceleration switching towards later in time. From figure (5b) it is seen that with the increase in beta the area under the velociy curve decreases therefore, a decreased amount of distance is covered. This is indicated by figure (6a) which shows the distance cost (J_2) as a function of β. As β increases J_2 increases. However, this monotonicity is not true for the losses cost (J_3) as given in figure (6b). As β increases from 0.1 to 0.3 the losses decrease while as β varies from 0.3 to 0.9 the losses increase. **Such a behavior is a result of J_3 being nonconvex** and is attributed to the fact that low values of the current result in higher values of the cost function integrand, f(i). From the figures (5a) and (5b) we make the following observation:

"As β varies from 0.1 to 0.3 J_2 increases while J_3 decreases. Whereas, when β varies from 0.3 to 0.9 both J_2 and J_3 increase."

Therefore, we observe that the region when β varies from 0.1 to 0.3 is locally Pareto optimal **while the region when β varies from 0.3 to 0.9 is not Pareto optimal.** Note that these ranges are only approximate, a finer resolution of β will generate an improved set of ranges. Further, this set in not a complete Pareto optimal set. The ranges $0 \leq \beta < 0.1$ and $0.9 < \beta \leq 1$, remain to be explored. Nevertheless, such a procedure can be applied to find the local Pareto optimal sets and therefore, the entire Pareto optimal set.

From this example we observe that in the nonconvex case not all convex combinations of the costs will result in a Pareto optimal solution. Therefore, one must be careful in making such a combination and a procedure similar to the one described above should be applied to ensure the local Pareto optimality of the result.

4. CONCLUSION

A Decision process involves satisfaction of a number of objectives. Control of uncertain system involves such a decisions process. Here, this problem has been delt with by considering different costs to be optimized. Resulting optimum solution, termed as the Pareto optimum, entails a tradeoff between different costs involved. The set of all such solutions is the Pareto optimal set.

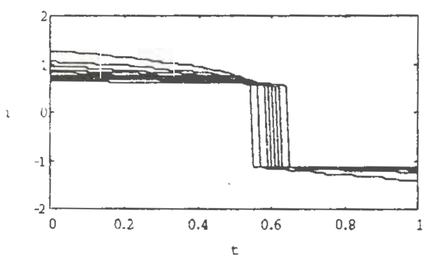

Figure 5(a): Current profiles for different values of β.

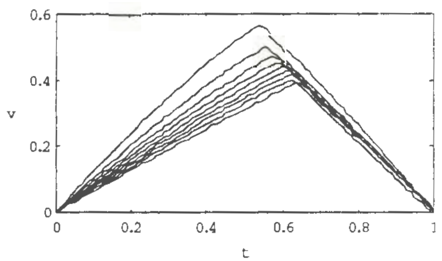

Figure 5(b): Velocity profiles for different values of β.

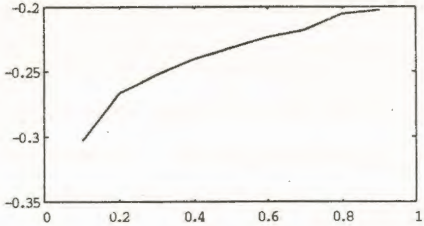

Figure 6(a): Distance cost (J_2) as a function of β

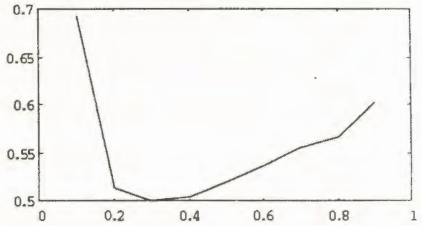

Figure 6(b): Losses cost (J_3) as a function of β

A procedure of generating the Pareto optimum set is applied to the optimal control of a linear time invariant system with quadratic costs defined for the purpose of tracking and identification. The results indicate that the convex mixing of the costs results in the same convex mixing of the controls. A similar procedure applied to the control of a D.C. motor with nonconvex cost functionals indicates that no general conclusion can be infered when the costs involved are nonconvex and a convex combination may not yield a Pareto optimal solution.

5. REFERENCES

[Bayard-85] David S. Bayard, Mansour Eslami, "Implicit Dual Control for General Stochastic Systems," Optimal Control Application & Methods, Vol. 6, 1985, pp. 265-279.

[Bryson-75] Bryson, A. E. Jr., Ho, Y., "Applied Optimal Control, optimization, estimation and control," Hemisphere Pub. Co., 1975, New York.

[Chankong-84] Chankong, V., and Haimes, Y. Y., Thadathil, J., and Zionts S., "Multiple Criteria Optimization: A State-of-the-Art Review," Decision Making with Multiple Objectives, Proceedings of the Sixth Int. Conf. on Multipl-Criteria Decision Making, Cleveland, OH, Y. Y. Haimes and V. Chankong (eds.), pp 36-90, Springer-Verlag, Berlin, June 1984.

[Cohon-78] Cohon, J. L. Multiobjective Programming and Planning, Academic Press, New York, 1978.

[Corley-80] Corley, N. W., "A New Scalar Equivalent for Pareto Optimization," IEEE Trans. on Automatic Control, Col. AC-25, pp 829-830, August, 1980.

[Da Cunha-67] Da Cunha, N.O. and Polak, E., "Constrained Minimization under Vector-Valued Criteria in Linear Topological Spaces," Mathematical Theory of Control, A. V. Balakrishnan and Lucien W. Neustadt (eds.), pp. 96-108, Academic Press, New York, 1967.

[Geoffrion-68] Geoffrion, A. M. "Proper Efficiency and the theory of vector maximisation," J. Math. Anal. Appl., vol. 22, pp 618-630, 1968.

[Guez-89] Allon Guez, John Selinsky, "A Learning/Adaptive Robot Controller," Proc. IEEE Int'l. Symp. on Intelligent Control, New York, 1989, pp. 576-581.

[Guez-89b] A. Guez, J. Selinsky, "Neurocontroller Design via supervised and unsupervised Learning," Journal of Intellignet and Robotic Sustems, Vol. 2, 1989, pp. 307-335.

[Guez-] A. Guez, J. Selinsky, "Design of Exploratory Schedules in Learning and Adaptive Robot Control," (To Apear) Journal of Adaptive Control and Signal Processing.

[Haimes-73] Haimes, Y. Y., "Integrated System Identification and Optimaization," Control and Dynamic Systems: Advances in Theory and Applications (C. Leondes, ed.), Academic Press, New York, Vol. 9, 435-518, 1973.

[Kirk-70] Donald E. Kirk, Optimal Control Theory, An Introduction, Prentice-Hall Inc. Englewood Cliffs, New Jersey, 1970.

[Lin-76] Lin, J. G., "Multiple-Objective Problems: Pareto-Optimal Solutions by Method of Proper Equality Constraints," IEEE Transactions on Automatic Control, Vol. AC-21, No. 5, October 1976.

[Marglin-67] Marglin, S., Public Investment Criteria, MIT Press, Cambridge, Massachusetts, 1967.

[Petrov-68] Petrov, I. P., "Variational Methods in Optimum Control Theory," Academic Press, 1968, New York.

[Salukvadze-79] M. E. Salukvadze, Vector-Valued Optimization Problems in Control Theory, Academic Press, Inc. New York, 1979.

[Selinsky-89] J. Selinsky, Closed Loop Learning of Rigid Robot Dynamics Via Design of Exploratory Schedules, MS Thesis, Drexel University, 1989.

[Stadler-88] Wolfram Stadler (editor), Multicriteria Optimization in Engineering and in the Sciences, Plenum Press, New York, NY, 1988.

[Wendel-77] Wendel, LR. E., and D. N. Lee "Efficiency in Multipleobjective Optimization Problems", Math. Programming, 12, pp 406-515, 1977.

[Yu-73] Yu, P. L., "Introduction to Domination Structures in Multi-Criteria Decision Problems," Multiple Criteria Decision Making, J. L. Cochrane and M. Zeleny (eds), pp 249-261, U. of South Carolina Press, Columbia, 1973.

[Zadeh-63] Zadeh, L.A., "Optimality and Non-Scalar-Valued Performance Criteria," IEEE Trans. on Automatic Control, Jan, pp 59-60, 1963.

[Zeleny-74] Zeleny, M., "Linear Multiobjective Programming," Springer-Verlag, 1974, New York, pp. 197-220.

[Zeleny] Zeleny, M., "A Concept of Compromise Solutions and the Method of the Displaced Ideal," Comput. Operations Res., vol. 1, pp479.