Karniel Meir and Inbar, International Joint Conference on Neural Networks (IJCNN'99), Washington DC, July 10-16, 1999

Exploiting the Virtue of Redundancy

Amir Karniel, Ron Meir, and Gideon F. Inbar
Department of Electrical Engineering, Technion - IIT, Haifa, 32000, Israel
Email: karniel@tx.technion.ac.il, rmeir@dumbo.technion.ac.il, inbar@ee.technion.ac.il

Abstract

Bernstein suggests that redundancy is the main reason for the superb dexterity of human motor control. However, introducing redundancy, in an inversely controlled object, results in an ill-posed problem. We suggest learning all the possible solutions and choosing one of them in real time. In this paper we define redundancy and differentiate between finite, countable and uncountable redundancy. We introduce a general concept of multiple controller and describe a specific architecture, the polyhedral mixture of linear experts (PMLE). We extend some notions of learning theory to the case of multiple valued functions and stress the difference between estimated inverse and inverse estimation. Then we show that the multiple inverse PMLE is suitable in serving as a multiple controller.

I. Introduction

Two main features of the biological motor control system are adaptation and redundancy, e.g., [1] and references therein. Adaptation has been studied extensively, while redundancy has not received as much attention. The main asset of redundancy is the ability to perform the same task in more than one way, see [2].

Fig. 1 describes the controller and the plant in a feed-forward control. The control problem is: find *x*, such that *F(x)* will be close to a given desired goal, Y_d ?

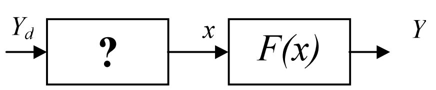

Fig. 1. Feed-forward control and the inverse controller problem.

The control of redundant systems involves an ill-posed problem of inverting a many-to-one mapping.

Most of the previous work addressed this issue using a pseudo-inverse or other criteria to find a single solution. See for example [3]. The recent progress in computer technology and in the field of neurocomputation enables us to consider an architecture that finds and retains all the solutions, and chooses one of them in real time according to a criterion that might be altered under different circumstances. This framework involves many new problems in control and learning theory. See [4] for a model to control redundant robot.

The purpose of this work is to describe a method for learning to control redundant systems, to develop the basic tools that are needed in order to build an intelligent multiple controller, and to analyze it.

In section II we define a redundant system and different types of redundant systems. In section III we introduce a general notion of multiple controller and describe an architecture for multiple controller, the Polyhedral Mixture of Linear Experts (PMLE). Then in section IV we shift to the language of learning theory and extend it to learning a multiple inverse controller. We stress the difference between the estimated inverse and inverse estimation, and justify the learning method of the PMLE. Finally we give an example in section V, and conclude in section VI.

II. Definition of Redundant System

In this section we define redundant systems as being many-to-one functions and suggest a set of definitions for different types of redundancy. These definitions provide the means to suggest an architecture for the control of redundant systems, and to discuss learning issues in the following sections.

A. Definitions

Definition-1: A <u>*system*</u> is defined by a function $f : X \rightarrow Y$.

In this definition we don't restrict the input and output domains, they can be scalars, vectors, continuous or discrete functions, or Laplace or Z transform domain functions.

Definition-2: A system $f : X \rightarrow Y$ is <u>*redundant*</u> if there exist $y \in Y$, $x_1 \in X$, $x_2 \in X$, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2) = y$.

It can be easily shown that according to this definition, a system is redundant if and only if it is not injective (injective being a one-to-one mathematical function). Let us further differentiate between three types of redundancy, finite, countable and uncountable.

Definition-3: A system is said to possess *finite redundancy* if it is redundant and if for each $y \in Y$ there is a finite number N and a finite set of input values $S = \{x_1, x_2, ..., x_N\}$ such that $f(x_i) = y$ for every $x_i \in S$; and $f(x) \neq f(x_j) = y$ for every $x \notin S$ $, x_j \in S$.

For example, $f(x) = |x|$ where $X = (-\infty, \infty)$ $Y = (0, \infty]$ is a redundant system with finite redundancy. (See Fig. 2a). A Physical example for such a case can be a tap with a handle that enables the water flow when the handle is rotated to the right or to the left.

Definition-4: A system is said to possess *countable redundancy* if it is redundant and if for some $y \in Y$ there is a countable set of input values $S = \{x_1, x_2, ...\}$ such that $f(x_i) = y$ for every $x_i \in S$; and $f(x) \neq f(x_j) = y$ for every $x \notin S$ $, x_j \in S$.

For example, $f(x) = \sin(x)$ where $X = (-\infty, \infty)$ $Y = [-1,1]$ is a redundant system with countable redundancy. (See Fig. 2b) A physical example can be a simple unconstrained rotation joint.

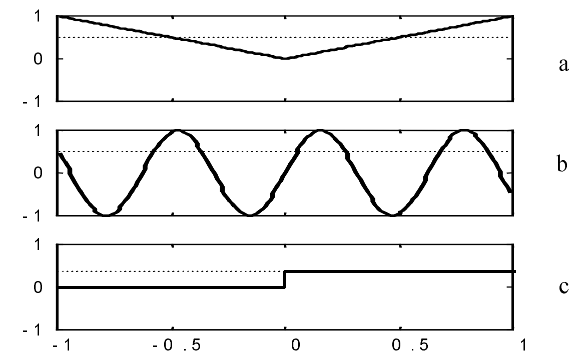

Fig. 2. Types of redundancy: (a) finite redundancy as $f(x) = |x|$, (b) countable redundancy as $f(x) = \sin(10 \cdot x)$, and (c) uncountable redundancy as $f(x) = 0.5 \cdot step(x)$.

Definition-5: A system is said to possess *uncountable redundancy* if it is redundant and if for each $y \in Y$ there is a set of input values S, such that $f(x) = y$ for every $x \in S$; and $f(x') \neq f(x) = y$ for every $x' \notin S$ $, x \in S$, and if there is at least one value of output for which the cardinality of S is greater than countable.

For example, $f(x) = 7$ where $X = (-\infty, \infty)$ $Y = 7$ is an uncountable redundant system. (See Fig. 2c). A physiological example can be two muscles that act on a single joint. There are uncountable number of combinations of forces in each muscle that will generate the same moment in the joint.

B. Remarks

1. In the literature, the finite and countable redundancies are not always considered as redundant systems (e.g., a manipulator without excess degrees of freedom can posses countable redundancy according to the definitions above, however it may not be considered as a *redundant manipulator*, see for example, [4]).
2. There is a crucial importance to the definition of the input and output domains, since different domain can imply different types of systems. E.g., $f(x) = \sin(x)$ where $X = (-\pi, \pi)$ $Y = [-1,1]$ is not redundant.
3. It is easy to see that any redundant system is either countable redundant or uncountable redundant. (Note that a finite redundant system is also countably redundant.) The proof is trivial, since any set S is either countable or not countable.
4. Note that a non-redundant system that is onto is invertible. The proof is by a theorem which states that a bijective (=one-to-one and onto) function is invertible.

III. Multiple Inverse Controller

From the definitions of the previous section, it is clear that for redundant system control, for some targets, there is more than one possible control signal. In this section we introduce the concept of multiple inverse controller, and concentrate on a specific architecture.

A. Multiple inverse controller

The main idea behind the concept of multiple controller is to learn all the possible control signals and choose one of them in real time according to a changeable criterion, see Fig. 3.

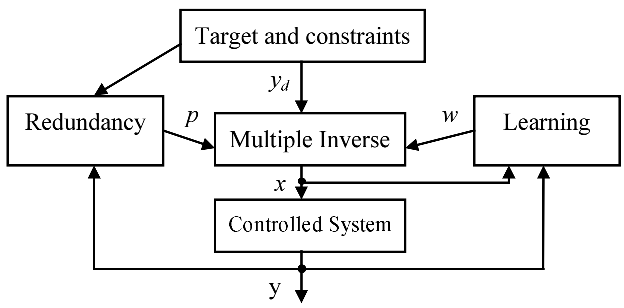

Fig. 3. The proposed architecture to exploit the virtue of redundancy by means of multiple controller that can be learned from examples.

We are facing here an ill posed problem and suggest trying to regulate the redundant system by expanding the output space Y to $Y \oplus P$, where $\oplus$ stands for a direct sum (i.e., each element in the expended space consist of one element from Y and one element from P). For a finite redundant system, P is a finite set; for a countable redundant system, P is a countable set; and for an uncountable redundant system, P is a vector space. In this line, let us state a formal definition of multiple inverse controller.

Definition-6: Let $f: X \to Y$ be a redundant system. The system $f_p^{MI}(Y): Y \oplus P \to X$ is called the <u>*multiple inverse system*</u> (or function or controller), if for every input value $x \in X$, there is a parameter $p \in P$, such that $f_p^{MI}(f(x)) = x$.

B. Polyhedral Mixture of Linear Experts (PMLE)

The PMLE architecture is suggested to serve as a multiple inverse controller. The PMLE learns a piecewise linear approximation of the system. Each area is governed by a linear function, which is called an expert, and one can invert each expert and get the multiple inverse. An illustration of this idea in one dimension is given in Fig. 4, where each expert governs an interval in the input space. In the general case where the dimension might be greater than one, each expert governs a polyhedral region in the input space and hence the name polyhedral mixture of linear experts.

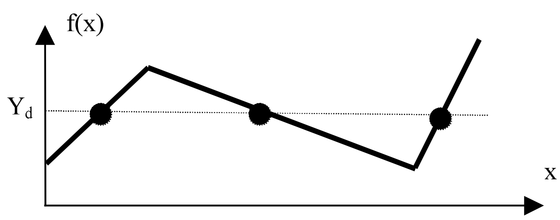

Fig. 4. Illustration of the PMLE in one dimension. The function is many-to-one and its piecewise linear approximation is drawn, there are three regions and therefore three experts. One can see that there are three different values of X that can drive the desired solution Y_d.

The advantage of the PMLE over other forward models is in the simplicity of the construction of the multiple inverse polyhedral mixture of linear experts (MI-PMLE).

This architecture was first presented in [5] where it was shown to be able to approximate inverse functions, however we didn't show its ability to learn multiple inverse functions. Now with the formal definition of redundancy, it is easy to be convinced that the PMLE is able to learn multiple inverse functions, of countable redundancy. However we have to slightly extend the MI-PMLE in order to deal with uncountable redundancy.

Proposition-1: The MI-PMLE architecture is able to represent the multiple inverse function of any piecewise linear system with polyhedral decision regions.

Comment: The MI-PMLE is able to *approximate* the multiple inverse of a broad class of function, since a broad class of functions can be approximated by a piecewise linear functions with polyhedral decision region.

Proof: By construction:

Any piecewise linear system with polyhedral decision regions can be accurately modeled by the PMLE, therefore our task is to construct the MI-PMLE. Each expert is linear and therefore the inverse of a specific output value can have one of the following three possibilities for each expert: (i) No solution at all. (ii) One solution. (iii) Infinite number of solutions (uncountable). Therefore in the extreme case, each expert will contribute an infinite (uncountable) number of solutions and will need its own regularization parameter. The space P will be a direct sum of natural number p_o and a set of real numbers p_i.

In order to find the parameterization of the infinite number of solutions (iii) above, we use a basic result from linear algebra about least squares and singular values (See Key Theorem (8.26) in [6]). For further details about the PMLE and the MI-PMLE, see [5,7]).

Practically the multiple inverse of a countably infinite redundancy cannot be implemented and also too many experts that contains an infinite number of solutions will encumber the architecture which is best suited for finite redundancies. The MI-PMLE construction method consists of two phases. First a piecewise linear approximation is learned from input-output examples and then the forward PMLE model is inverted in order to construct the MI-PMLE. In the next section we introduce some notions from learning theory in order to examine the issue of learning multiple inverse and then we justify the two phase method of constructing the MI-PMLE.

IV. Learning

The extensive research in the field of function approximation is naturally focused on the approximation of functions (i.e., single-valued functions). Many control problems requires the estimation of inverse functions, and in the case of redundant system, the inverse may be a "multivalued function", which is better termed multivalued relation.

In the first part of this section we introduce some concepts and notations from learning theory and extend them to the case of multiple inverse learning. Then we prove a basic theorem for learning multiple inverses and describe criteria for good multiple inverse learning algorithms. In the second part we stress the difference between the estimated-inverse and the inverse of the forward estimator, and finally we justify the MI-IPMLE learning algorithm.

A. Definitions notations and the basic result

Let us begin with the forward approximation. $\hat{f}_w(x) \in F$ is a candidate approximation of the function f, and is an element of a parameterized family of functions, F. In our case the family is the PMLE, that is, the approximation function is a piecewise linear function. The specific weight value, *w*, defines the approximation function and it can be learned from the set of samples $\{x_i, y_i\}_{i=1}^{m}$, where it is assumed that (x_i, y_i) are independently drawn from some unknown distribution *P(X,Y)*.

For the convenience of writing we will drop the chapeau and the subscript *w*. We will add a subscript *m* (i.e. f_m), to indicate that the function was chosen based on the *m* samples, and a star (i.e. f^*), to mention that the function is the best approximation by the MSE criterion, as defined next.

Given an approximation function, let us define the average discrepancy (mean square) as $D(f) = E\|y - f(x)\|^2$, and the empirical discrepancy as $D_m(f) = \frac{1}{m}\sum_{i=1}^{m}\|y_i - f(x_i)\|^2$, the best possible approximation from the family $f^* = \arg\min_{f \in F} D(f)$, and the best empirical approximation $f_m^* = \arg\min_{f \in F} D_m(f)$. We are interested in the following measure of the generalization capability:

$$D(f_m^*) - D(f^*), \tag{1}$$

which measures the performance difference between the empirical optimal model and the truly optimal model.

After the forward model is learned we build an inverse of the approximation function, that is the MI-PMLE.

$$\hat{f}_\theta^{MI}(y) = \{x | \hat{f}_w(x) = y\} \qquad \hat{f}_\theta^{MI} \in F^{MI}$$

here θ is a function of *w,* and again, for the convenience of notation we will drop the chapeau and the subscript θ. We will also assume that the redundancy is finite, therefore the set of solutions can be represented as a vector. $f^{MI}(y) = \underline{x}$

Let us define the discrepancies in this case as follows:

The average discrepancy:

$$D(f^{MI}) = E\left[\min_j \left\|x - \{f^{MI}(y)\}_j\right\|^2 \middle| y = f(x)\right],$$

and the empirical discrepancy,

$$D_m(f^{MI}) = \frac{1}{m}\sum_{i=1}^{m}\min_j \left\|x_i - \{f^{MI}(y_i)\}_j\right\|^2 .$$

Here we are interested in the following measure of the generalization capability:

$$D(f_m^{MI*}) - D(f^{MI*}) \tag{2}$$

where

$$f^{MI*} = \arg\min_{f^{MI} \in F^{MI}} D(f^{MI}) \quad \text{and} \quad f_m^{MI*} = \arg\min_{f^{MI} \in F^{MI}} D_m(f^{MI}).$$

A proper definition of the multiple inverse generalization capability should contain two parts.

(i) A measure of completeness, i.e., that the size of the multiple inverse vector is equal to the number of solutions to the problem. (ii) A measure of the accuracy of the estimation, such as (2).

As a start, we will assume that the number of the solutions is finite and known and concentrate on the second measure.

In learning theory, a common assumption is that the forward approximation, minimizes the sample error minimization (SEM), i.e., it computes f_m^*. Then the strive is to show that the measure (1) approaches zero as *m* grows. There are many results and bounds that show that this can happen in probability and even almost surely [8].

Since there are no results as to multiple inverse learning, we will formulate here a simple result for this case.

Proposition 2: If the algorithm for the MI-PMLE is a SEM algorithm, and if the range is bounded $X \in [0, A]$, and if the set F^{MI} is finite, then for any $\varepsilon > 0$

$$P\left\{\left\|D(f_m^{MI*}) - D(f^{MI*})\right\| > \varepsilon\right\} \xrightarrow[m \to \infty]{} 0 .$$

We will also get a bound on the rate of convergence.

Proof:

Following the literature, e.g., [8], let us first derive the following result

$$\begin{aligned} & D(f_m^{MI*}) - D(f^{MI*}) = \\ & = D(f_m^{MI*}) - D_m(f_m^{MI*}) + D_m(f_m^{MI*}) - D(f^{MI*}) \\ & \le D(f_m^{MI*}) - D_m(f_m^{MI*}) + D_m(f^{MI*}) - D(f^{MI*}), \\ & \le 2 \cdot \sup_{f^{MI} \in F^{MI}} \left|D_m(f^{MI}) - D(f^{MI})\right| \end{aligned} \tag{3}$$

where we have used $D_m(f_m^*) \le D_m(f)$. Now for a fixed function f^{MI}, one can use Hoeffding's inequality, since the map f^{MI} is into the region [0,*A*], then $\min_j \left\|x_i - \{f^{MI}(y_i)\}_j\right\|$ is bounded in [0,2*A*], and by Hoeffding's inequality, one can write:

$$P\left\{\left\|D_m(f^{MI}) - D(f^{MI})\right\| \ge \frac{\varepsilon}{2}\right\} \le 2 \cdot e^{-\varepsilon^2 m / 32A^4}$$

Since the set of possible functions is finite,

$$P\left\{\sup_{f \in F^{MI}}\left\|D_m(f) - D(f)\right\| \ge \frac{\varepsilon}{2}\right\} \le 2 \cdot \left|F^{MI}\right| \cdot e^{-\varepsilon^2 m / 32A^4} .$$

Now we can use (3) above and get our desired result:

$$P\left\{D(f_m^{MI*}) - D(f^{MI*}) \ge \frac{\varepsilon}{2}\right\} \le 4 \cdot \left|F^{MI}\right| \cdot e^{-\varepsilon^2 m / 32A^4}$$

We can make the bound tighter if we assume that the branches of the solutions are equally distributed, and then $\min_j \left\|x_i - \{f^{MI}(y_i)\}_j\right\|$ is bounded by A/k where *k* is the number of solutions, i.e. the size of the multiple inverse vector. With this assumption, the bound is:

$$P\left\{D(f_m^{MI*}) - D(f^{MI*}) \ge \frac{\varepsilon}{2}\right\} \le 4 \cdot \left|F^{MI}\right| \cdot e^{-\varepsilon^2 m k^4 / 32A^4}$$

Remarks

1. Proposition 1 is a preliminary result in learning multiple inverse, tighter bounds using fewer assumptions can be achieved for specific cases.

2. The assumption that the PMLE is a SEM algorithm does not hold for most of the learning algorithms, since they usually find a local minimum. As for the multiple inverse, this assumption is even more problematic. The definition of the empirical discrepancy (or sample error) is proposed here first, and therefore there are no algorithms that are proved to be SEM algorithm. Moreover, the MI-PMLE is *not* a SEM algorithm since it computes the inverse estimation rather then the estimated inverse as described in the next part of this section.

B. Estimated Inverse vs. Inverse Estimation

In many control applications there is a need to estimate an inverse model of the controlled system. Artificial neural networks (ANN) were suggested as a good candidate for this task, e.g., [9]. The simplest method to construct such an estimated-inverse is the *direct inverse learning*, i.e. feeding the ANN learning algorithm with the input-output samples in the reverse order, outputs as inputs and inputs as desired outputs. Another method is the *indirect learning*, i.e., first estimate a forward model and then construct the inverse of the estimated model.

We wish to stress that these two methods are not equivalent and the second method is generally superior for control application where there is an additive noise at the output of the plant.

Proposition-3: The best-estimated inverse model is not necessarily equal to the inverse of the best-estimated forward model.

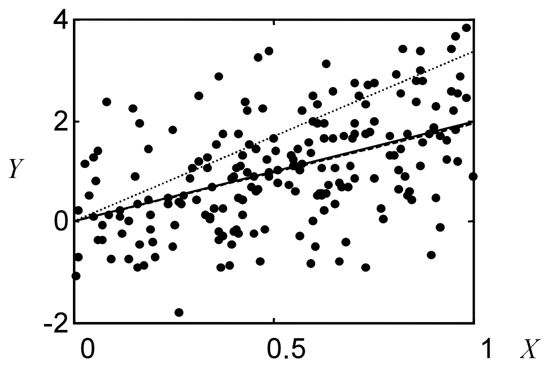

Fig. 5. The best forward model (dashed, almost overlap the solid line) versus the best inverse model (dotted), as estimated from the noisy data (dots) that was originated from the original system (solid). *X* is the input of the system and *Y* is the output of the system and of the forward model. *Y* is the input and *X* is the output of the inverse model.

An analysis of this observation is beyond the scope of this paper and is given in [10]. We actually show that in the presence of additive noise, with nonzero variance, the two methods differ asymptotically, and the gap increases with the variance of the noise.

The difference is apparent even in a simple simulation example in one dimension as illustrated in Fig. 5.

C. Justification of the IPMLE learning method

In this section we formalize the relevant discrepancy measure for the multiple inverse control system, and show that the learning algorithm minimizes this discrepancy measure.

The multiple inverse control system is described in Fig. 6.

Our goal is to construct the multiple inverse that will satisfy the two learning criteria, i.e., Accuracy and Completeness. Let us concentrate on the accuracy measure.

The accuracy measure should manifest our aim to get the output, y, as close as possible to the desired output, y_d, for all possible control inputs, that is for all values of the regularization parameter p.

We therefore suggest the following discrepancy measures for multiple controllers, the average and empirical discrepancies:

$$D^C\left(f^{MI}\right) = E\left[y_d - y\right]^2, \quad D^C_m\left(f^{MI}\right) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{k}\sum_{p=1}^{k}\left[y_{d_i} - y_i\right]^2$$

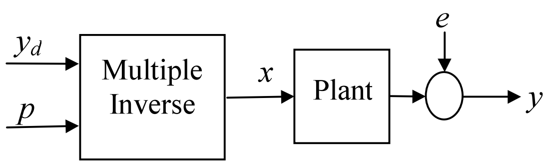

Fig. 6. The multiple-inverse control system configuration.

With the assumption of additive noise at the output as described in Fig. 6, one can write the output in terms of the noise, the controlled system, the multiple controller and the input as follows

$$y = e + f_s\left(\hat{f}_p^{MI}(y_d)\right)$$

Therefore the discrepancy measure is:

$$D^C\left(f^{MI}\right) = E\left[y_d - y\right]^2 = E\left[y_d - \left\{e + f_s\left(\hat{f}_p^{MI}(y_d)\right)\right\}\right]^2 =$$

$$= E\left[y_d - f_s\left(\hat{f}_p^{MI}(y_d)\right) - e\right]^2 = E\left[y_d - f_s\left(\hat{f}_p^{MI}(y_d)\right)\right]^2 + E[e]^2$$

Minimizing this measure implies the requirement $y_d = f_s\left(\hat{f}_p^{MI}(y_d)\right)$ that can be satisfied by constructing the multiple inverse of the plant estimation. Therefore combined with our previous observation (sub-section *B*), one can conclude that in the presence of additive noise at the output of the system, the best multiple inverse controller is the multiple inverse of the best forward estimation of the controlled system, which can be implemented by the MI-PMLE algorithm. Further details will appear in [7].

V. Example

The following example illustrates the method of learning to control a redundant system and demonstrates the virtue of redundancy in the presence of noise.

Let us consider a redundant system with three inputs and one output, that can be described by the following three functions, see Fig. 7.

$$y = f_1(x_1) + f_2(x_2) + f_3(x_3) \quad (4)$$

Such a redundancy is very common. For example, the force produced at a joint by all the motor units that act upon it, or the flow of fluid that is derived by several pumps in parallel.

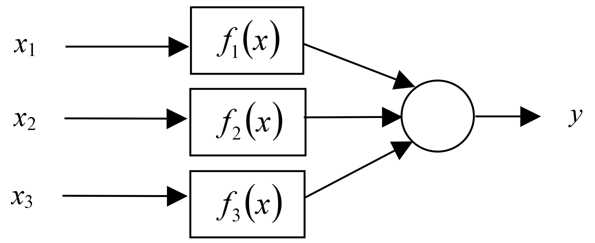

Fig. 7. A redundant system with uncountable redundancy.

First, we build the PMLE model, that is, in each polyhedral region we have a linear expert that approximate the system, such as: $y = w_0 + w_1x_1 + w_2x_2 + w_3x_3$. Then, we build the Inverse-PMLE that is we invert each expert and find the subspace of all the solutions which is parameterized by the parameter vector p:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} \frac{p_1}{w_1} & \frac{p_2}{w_2} & \frac{1 - p_2 - p_1}{w_3} \end{bmatrix} (y_d - w_0)$$

At this point the MI-PMLE is the multiple controller and we have to choose the parameter vector p.

Remarks:

1. There is another element in the vector p, that specifies which one of several possible experts should be selected, but we ignore this element at this point (or assume that the system has just one expert in each area).
2. The assumption regarding the structure of the system, (4) and Fig. 7, is not necessary for the results. One can take a general system and the linear approximation will yield the same analysis.

Let us concentrate on one possible criterion in the presence of noise or uncertainty. Let us assume that each branch of the system has its own noise or uncertainty modeled as an added value $n_i(x)$, therefore (4) will become

$$y = f_1(x_1) + n_1(x_1) + f_2(x_2) + n_2(x_2) + f_3(x_3) + n_3(x_3)$$

The output will not be exactly y_d. If the system is linear, or if we shift the nonlinear effects into the noise component n_i, the output will be

$$y = y_d + n_1\left(\frac{p_1}{w_1}\right) + n_2\left(\frac{p_2}{w_2}\right) + n_3\left(\frac{1 - p_2 - p_1}{w_3}\right).$$

A potential criterion in this case may be to minimize the deviation of y from y_d, that is:

$$[p_1, p_2] = \arg\min_{p_1, p_2} \left\{ n_1\left(\frac{p_1}{w_1}\right) + n_2\left(\frac{p_2}{w_2}\right) + n_3\left(\frac{1 - p_2 - p_1}{w_3}\right) \right\}$$

For independent white noise, if we assume that the noise is proportional to the input level, and if the weights are similar then the optimum solution is $p_1 = p_2 = 1/3$.

Another interesting result is that the redundancy can assist in noise rejection and robustness, since when the noises are independent of each other, the effect of the noise decreases as the redundancy increases. In this case, as the number of parallel elements in Fig. 7, grows to infinity then the sum of noises approximates zero.

VI. Conclusions

Redundancy can improve reliability and flexibility and is one of the main reasons for the superb dexterity of human movement control. The control of redundant systems is difficult, and presents many new challenges, theoretical and practical. The rapid growth in the available computation power enables us to consider complex intelligent controllers and even to introduce additional redundancy in controlled systems. This work presents a basic architecture for a multiple controller, and some mathematical tools for a new theory of learning to control redundant systems.

References

[1] Karniel A, and Inbar GF (in press) Human Motor Control: Learning to Control a Time-Varying Non-linear Many-to-One System, Accepted for publication in IEEE transactions on Systems, Man, and Cybernetics part C.

[2] Latash ML, and Turvey MT Eds. (1996) Dexterity and its development, Erlbaum, New Jersey.

[3] Jordan MI (1990) Motor learning and the degrees of freedom problem. In Attention and performance XIII, Jeannerod, Ed, pp. 796-836.

[4] DeMers DE (1993) Learning to Invert Many-To-One Mappings. Ph.D. dissertation, University of California, San Diego.

[5] Karniel A, Meir R, and Inbar GF (1998) Polyhedral mixture of linear experts for many-to-one mapping inversion. In Proc. ESANN98, M. Verleysen, Ed., pp. 155-160.

[6] Noble B and Daniel JW (1988) Applied linear algebra, third edition, Prentice-hall international INC.

[7] Karniel A (in preparation) Learning motor control of redundant systems, Ph.D. dissertation, Technion, Israel.

[8] Vidyasagar M (1997) A theory of learning and generalization. Springer-Verlag, London.

[9] Barto AG (1990) Connectionist learning for control. In Miller WT, Sutton RS and Werbos PJ (Eds) Neural Networks for control MIT press.

[10] Karniel A, Meir R and Inbar GF (1999) The best estimated inverse versus the inverse of the best estimation, EEPUB 1198, Department of Electrical Eng., Technion, Israel.