Nonlinear Force Fields: A Distributed System of Control Primitives for Representing and Learning Movements

F. A. Mussa-Ivaldi
Northwestern University Medical School

Abstract[1]

Electrophysiological studies have suggested the presence of a modular structure in the output stages of the motor system. In this structure, independent modules are connected to specific groups of muscles and generate nonlinear fields of force acting upon the controlled limbs. This paper explores the computational consequences of this structure in the framework of multivariate approximation. Movements are generated through the selection of independent modules and through the vectorial superposition of their output fields. It is shown that complex joint motions of a multi–segmental mechanism may be obtained by determining a set of time–independent parameters which scale the amplitude of each module's field. In addition, optimization results suggest that a system of such modules may evolve to improve the execution of smooth movements of the mechanism's endpoint across the whole workspace. The observed improvements generalize beyond the set of movements used to guide the optimization. These findings indicate that a rich repertoire of behaviors may be learned by adapting a system of force fields obtained from the combination of multiple viscoelastic actuators.

1 Introduction

An increasing number of investigations has focused on the role of impedance in the control of stable behaviors [7, 6, 11]. Impedance control has been proposed as a way to compensate for the lack of accurate prior information about the properties of a controlled system and its operating environment. While there are solid engineering motivations behind the choice of providing artificial systems with stable built–in passive behavior [2], this type of behavior is inherent to biological motor systems which are endowed with inherently viscoelastic actuators. Isolated skeletal muscles act like a non–linear springs whose length-tension properties are modulated by neuromuscular activation [12]. In addition to elasticity, muscles and connective tissue generate nonlinear viscous forces that are also under the influence of neuromuscular activity. The viscoelastic behavior of muscles is affected by descending commands generated in the brain as well as by reflex activities of the spinal cord. The final consequence of the combined viscoelastic properties of the skeletal muscles is that neural commands specify a mapping between state and force at the interface between the controlled limb and the environment. This mapping is described by a force field. In the highly nonlinear context of biological systems, the notion of local impedance has a very limited value. The mechanics of a limb are better described by the force field generated by the motor system across the whole workspace.

The mechanics of the motor apparatus are relevant not only to stability and control, but also to computation and, in particular, to the way in which the brain may represent a variety of motor behaviors. Recent electrophysiological investigations on vertebrate spinal cord [3] have suggested that the neural circuits in this part of the nervous system may be organized into a few distinct modules. Each module, when active, recruits a specific pattern of muscles. As a consequence the activity of a spinal module results in a field of forces acting on the ipsilateral limb. Different spinal modules generate different fields and the simultaneous activation of multiple modules leads to the vectorial summation of the corresponding fields [10]. These physiological findings gave rise to the hypothesis that the nervous system generates a very broad repertoire of force patterns by vectorial summation of few nonlinear fields [9]. This paper presents a computational framework in which the combination of few independent force fields such as those generated by stimulation of the spinal cord provides a general way for dealing with movement learning and control in both biological and artificial systems.

2 Force–field control

I consider the general problem of controlling a multiarticular mechanism that may be described as an open kinematic chain of rigid bodies. The workspace of this mechanism is an N–dimensional manifold. A point, q, in this manifold represents a configuration in generalized coordinates. I assume that the undriven passive dynamics of this system– that is the dynamics obtained when all control signals are set to zero– are described by a set of N second order differential equations that collectively may be represented as

$$D(\ddot{q}, \dot{q}, q) = 0.$$

The vector field, D, is the generalized force induced by inertial and (passive) viscoelastic components.

The net effect of the control signals operating on the actuators is a time–varying field of generalized forces, $C(q, \dot{q}, t)$, that depend on the state of the mechanism. By combining passive dynamics and control field, one obtains the basic equation of motor control:

$$D(q, \dot{q}, \ddot{q}) = C(q, \dot{q}, t). \tag{1}$$

This is a family of ordinary differential equations (ODE): each instance of a control field– that is each parametrization of $C(q, \dot{q}, t)$– corresponds to a particular action of the controller and to a particular ODE.

I will focus on a controller structure that is consistent with the experimental observations on spinal force fields [3]. This is a superposition of independent elementary fields, $\phi_i(q, \dot{q}, t)$, that, in analogy with basis vectors and basis functions, are also called "basis fields":

$$C(q, \dot{q}, t) = \sum_{i=1}^{K} c_i \phi_i(q, \dot{q}, t). \tag{2}$$

The scalar coefficients, c_i, act as selection and tuning parameters. A control module is selected when the corresponding c_i is non–zero and the overall magnitude of the force vectors (but not the spatiotemporal pattern) is scaled by the same parameter.

In the biological system, basis fields are the expression of the connections between neurons and muscles belonging to a control module. These connections may be considered as fixed, at least over the time scale of individual actions. The tuning coefficients, c_i, represent the intensity with which a module is recruited by the descending commands. There is no reason to limit the scope of this framework to the modeling of spinal/supraspinal interactions. The mathematical structure of Equation (1) may be applied recursively to higher order modules, involving neural connections between spinal cord and other brain structures.

The simple structure of Equation (2) poses a computational challenge when one deals with the problem of determining the tuning coefficients that may drive the mechanism along a desired trajectory, $\hat{q}(t)$. The tuning coefficients, c_i, are *time–independent* whereas the desired trajectory specifies the configuration of the mechanism as a function of time. Experimental observations of multi–joint arm movements in humans have suggested that a "building block" of arm kinematics is the smooth trajectory of the hand used in reaching movements [8]. This trajectory has a simple description in the Cartesian end–point coordinates of a limb: it is a rectilinear motion of the hand from start to end location with a "bell–shaped" velocity profile. Such a simple movement is achieved by rather complex and critically timed movements of the joints. This is because the transformation that relates hand to joint motions is non–linear. Accordingly a rectilinear movement of the hand is carried out by joint motions that may display changes in direction and that vary with the operating configuration of the arm. Given the non–linear nature of arm kinematics and dynamics it is not a surprise that past attempts at driving simulated hand movements with time–invariant linear controllers have failed [14, 4]. This paper explores how the same movements may be obtained by time–invariant control parameters that modulate the amplitude of non–linear time–varying force fields.

2.1 Simplifications

The following analysis makes some simplifying assumptions on the structure of the controller (2). One assumption has a general impact on the problem. This is the hypothesis that the basis fields may be factorized into the product of a state–dependent vector field and a time–dependent scalar function. This assumption reflects the experimental observation that when a spinal site is stimulated, the spatial pattern of force vectors across the limb's workspace remains invariant across time while the vector amplitudes are subject to uniform scaling [3]. This finding is consistent with the idea that spinal module is composed of a specific combination of limb muscles. It is this combination that dictates the geometry of the resulting force field. The time varying activities of the neurons that belong to a spinal module generate a signal that is distributed to the module's muscles without affecting their relative balance.

The other simplifications have an impact on the particular examples presented here but do not affect in a substantial way the general approach. The mechanism used in the examples is a two–joint planar limb whose inertial dynamics depend non–linearly upon the state (Figure 1). For convenience, it is assumed that the basis fields depend only upon the configuration, q, and time, but not upon velocity. A velocity dependent force is added in the model as a fixed component of the passive dynamics field, D. The functional form of the basis fields has a simple expression in generalized (joint) coordinates. This is to reflect the hypothesis that these fields are defined by the intrinsic geometry of the actuators and do not embody any prior knowledge of the transformation from actuator to end–point coordinates.

3 Pulse and step patterns

The neural signals recorded in different areas of the central nervous systems during a variety of goal directed motor actions may be partitioned in two general classes: steps (or transitions in "tonic activities") and pulses (or "phasic activities"). Models based on the combination of pulse and step signals have been successful in dealing with the control of saccadic eye movements [13]. Here, to describe the temporal components of the basis fields, I use two waveforms that belong to the same basic classes: a smooth step, $\sigma(t)$ ($= \frac{1}{T}\left[(t-t_0) - \frac{T}{2\pi}\sin(\frac{2\pi}{T}(t-t_0))\right]$ if $t_0 < t < t_0+T$; $= 0$ if $t \le t_0$; $= 1$ if $t \ge t_0+T$ for given parameters T and t_0) and a smooth pulse, $\beta(t) = \dot{\sigma}(t)$. The simulation examples are limited to these two functions but, of course, one may apply the same analysis to systems characterized by families of such time–functions.

Let us assume that the system's controller contains $2K$ modules generating two classes of basis fields: K step–fields

$$\phi_i(q,t) = \sigma(t)\chi_i(q)$$

and K pulse–fields:

$$\psi_i(q,t) = \beta(t)\chi_i(q).$$

obtained from a single set of K "kernel" fields, $\chi_i(q)$. The kernel fields represent the combined static forces generated by the ensemble of actuators connected to each module.

The kernel fields used here (Figure 1) are gradients of bivariate Gaussians centered at different locations in the workspace:

$$\chi_i(q) = -K_i(q-\eta_i)e^{-\frac{1}{2}(q-\eta_i)^T K_i(q-\eta_i)}. \qquad (3)$$

The characteristic parameters, η_i and K_i, represent, respectively, the "center" (ore equilibrium point) of the field and its (inverse) variance matrix. In this mechanical context, the inverse variance matrix is dimensionally equivalent to a local stiffness. Assuming symmetry of K_i, there is a total of five parameters for each kernel field. The main rationale for the choice of Gaussian fields is mathematical convenience. However, Gaussian fields are to some extent consistent with the mechanical behavior of biological muscles. In particular, with the known property that muscle force increase with stretch only up to a certain point. After the muscle reaches a critical length, the force reaches a maximum and beyond this length it may decrease.

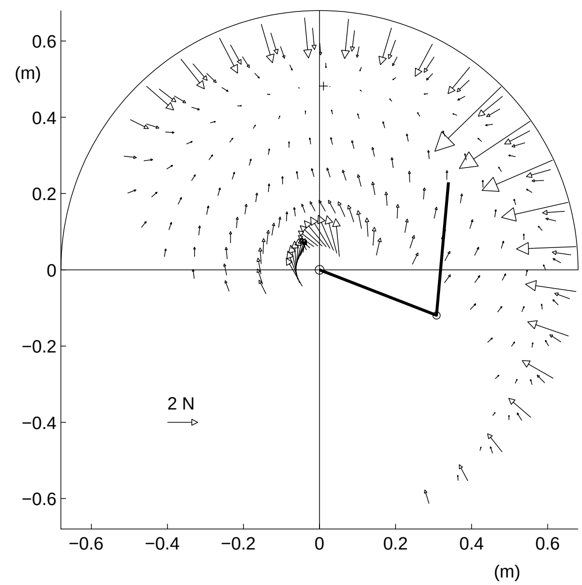

Figure 1: A Gaussian field. This is the gradient of a Gaussian potential function (Equation 3). The cross indicates the equilibrium point (where $F = 0$). The simulated controlled mechanism is a two–joint planar arm.

Under the above assumptions, the net force field generated by the entire control network is:

$$C(q,t) = \sum_{i=1}^{K} p_i\chi_i(q) + \sum_{i=1}^{K} c_i\phi_i(q,t) + \sum_{i=1}^{K} d_i\psi_i(q,t). \qquad (4)$$

The first sum on the right represents the baseline field generated by the control system as a consequence of previous movements (we may assume that initially $p_i = 0$). The computational task is to derive the coefficients (c_i, d_i) that drive the system's dynamics (1) along a desired trajectory $\hat{q}(t)$.

4 Movement by field–approximation

It may happen that there exist a choice of parameters, (c_i^*, d_i^*), that achieves *exactly* the desired trajectory. In this case, the equation

$$D(\hat{q}(t), \dot{\hat{q}}(t), \ddot{\hat{q}}(t)) = \sum_{i=1}^{K} p_i \chi_i(\hat{q}(t)) + \sum_{i=1}^{K} c_i^* \phi_i(\hat{q}(t), t) + \sum_{i=1}^{K} d_i^* \psi_i(\hat{q}(t), t) \quad (5)$$

is identically satisfied at each time, t. Most likely, however, such an ideal controller is not available and one must be content with parameters that provide an approximation of the desired trajectory. One may obtain such an approximation through some searching procedure. While searching procedures are theoretically feasible, they tend to be computationally expensive. An alternative to searching is offered by the fact that the tuning parameters appear linearly in the expression of the control field. Instead of minimizing the error between desired and actual *motions*, one may minimize the difference between the passive dynamic torque and the controller torque along the desired trajectory. This is the difference between the force fields on the right and left sides of Equation (5). Following this idea one is led to a least squares problem that admits a unique solution.

Transition of posture: the step. The step component of the controller is a transition from the initial to the final stable posture. This transition can be regarded as a concurrent switch–on of the final posture and switch–off of the initial posture. We may represent both the switch-on and the switch–off function using the step function $\sigma(t)$:

$$\begin{aligned} u^{ON}(t) &= \sigma(t) \\ u^{OFF}(t) &= 1 - \sigma(t) \end{aligned}$$

The initial posture is a static component of the control field:

$$P(q) = \sum_i p_i \chi_i(q) \quad (6)$$

which satisfies the algebraic equation:

$$D(q_I, 0, 0) = P(q_I).$$

At the end of the movement the controller generates a static field

$$F(q) = \sum_i f_i \chi_i(q) \quad (7)$$

that satisfies the algebraic equation

$$D(q_F, 0, 0) = F(q_F). \quad (8)$$

The transition between initial and final state of the controller is mediated by the switch–on and switch–off functions:

$$\begin{aligned} C(q,t) &= u^{OFF}(t)P(q) + u^{ON}(t)F(q) = \\ &= \sigma(t)P(q) + (1-\sigma(t))F(q). \end{aligned}$$

After substituting the expressions for $P(q)$ (6) and for $F(q)$ (7) one obtains:

$$C(q,t) = \sum_i p_i \chi_i(q) + \sum_i c_i \phi_i(q,t) \quad (9)$$

with $c_i = f_i - p_i$ and $\psi_i(q,t) = \sigma(t)\chi_i(q)$. This expression for the control field corresponds to the first two terms on the right of Equation (4).

The calculation of the coefficients, c_i, of the step–fields is carried out by solving Equation (8)

$$D(\hat{q}_F, 0, 0) = \sum_i f_i \chi_i(\hat{q}_F) \quad (10)$$

for f_i. With an N-dimensional configuration space and K kernel fields, Equation (10) is a system of N scalar equations in K unknowns. An additional, non–linear constraint is obtained by requiring that the final force field be statically stable. To this end, one must insure that the forces generated by the controller converge toward the final equilibrium point, $\hat{q}_F$. If each kernel field (with unit coefficient) is convergent toward its center, then a sufficient condition for the convergence of Equation (10) is that all tuning coefficients, f_i, be non–negative. Non negative minimum norm or least square solutions to Equation (10) are obtained by standard non-negative least–squares (NNLS) algorithms (Lawson and Hanson, 1974). The coefficients of the step–fields are then derived from

$$c_i = f_i - p_i. \quad (11)$$

The typical outcome of a step–field controller is shown in Figure (2). It is obvious that while this controller can enforce an arbitrary final position, the trajectory with which it gets there may be quite different than the intended movement[2]. The static component, $P(q)$, corresponding to the first term to the right of Equation (9) represents the baseline postural field before the onset of movement. Thus, the coefficients p_i need only to be updated at the end of each movement ($p_i \leftarrow p_i + c_i = f_i$).

[2] Earlier simulations have shown that properly tuned step–fields could generate succesful approximations of rectilinear end–point movements. However, an undesired (and unphysiological) side–effect of using only step–fields is that the net final force field depends upon both the final and the initial positions. This is not compatible with the avaliable observations on end–point impedance [11].

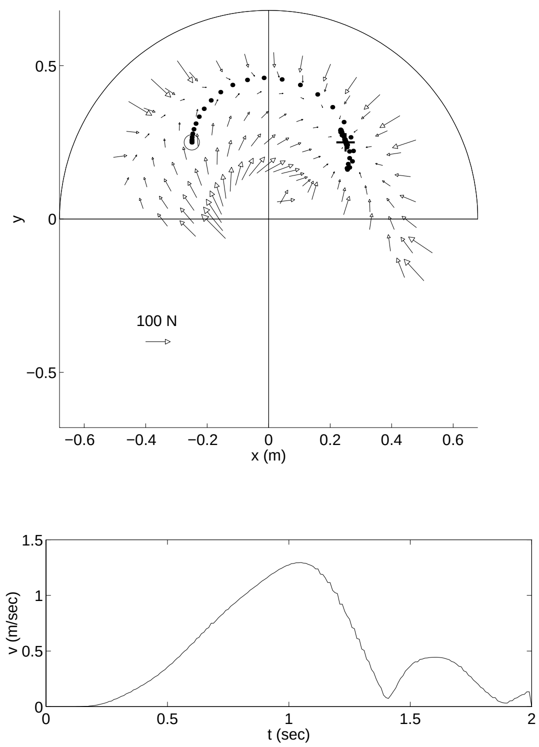

Figure 2: Movement toward a target generated by a combination of step fields. **Top**: The motion follows a curvilinear trajectory. Circle: initial position. Cross: Target location. **Bottom**: Tangential velocity profile.

Trajectory control: the pulse. The purpose of the pulse fields is to enforce the desired trajectory, $\hat{q}(t)$. The first step in the calculation of the pulse–control parameters, d_i is the definition of the "modified dynamics", $\tilde{D}$, arising from the combination of the passive dynamics term with the step–control field. This is a force field that depends upon state, acceleration and time:

$$\tilde{D}(q,\dot{q},\ddot{q},t) = D(q,\dot{q},\ddot{q}) - \sum_i p_i \chi_i(q) + \\ - \sum_i c_i \phi_i(q,t). \tag{12}$$

The pulse component of the controller must "drive" this modified dynamics along the desired trajectory. Ideally, we would like to find a set of parameters d_i such that the differential equation

$$\tilde{D}(q,\dot{q},\ddot{q},t) = \sum_i d_i \psi_i(q,t) \tag{13}$$

admit the desired trajectory as a solution. In this case one would obtain the algebraic identity

$$\tilde{D}(\hat{q}(t),\dot{\hat{q}}(t),\ddot{\hat{q}}(t),t) = \sum_i d_i \psi_i(\hat{q}(t),t) \tag{14}$$

as in Equation (5).

In the absence of an ideal solution, we look for the set of parameters that minimizes some measure of the difference between left and right sides of the above equation. This error measure requires the definition of an *inner product* between two fields, $F(q,\dot{q},\ddot{q},t)$ and $G(q,\dot{q},\ddot{q},t)$, along the trajectory $\hat{q}(t)$. To this end, we first define the *restriction*, $F[\hat{q}(t)]$, *of the field* $F(q,\dot{q},\ddot{q},t)$ *over the trajectory* $\hat{q}(t)$:

$$F[\hat{q}(t)] \equiv F(\hat{q}(t),\dot{\hat{q}}(t),\ddot{\hat{q}}(t),t)$$

This is an operation that maps a trajectory, $\hat{q}(t)$, in the temporal sequence of force vectors generated by the field over that trajectory. Next, we define the inner product of two fields, $F(q,\dot{q},\ddot{q},t)$ and $G(q,\dot{q},\ddot{q},t)$ along the trajectory $\hat{q}(t)$ as the integral

$$< F, G >_{\hat{q}(t)} \equiv \int_{t_I}^{t_F} F[\hat{q}(t)]^T G[\hat{q}(t)] dt. \tag{15}$$

Accordingly, the norm of a field along $\hat{q}(t)$ is:

$$\| F \|_{\hat{q}(t)} = < F, F >_{\hat{q}(t)}^{\frac{1}{2}} .$$

Following the above definitions, one readily obtains a formulation of the least–squares approximation whose goal is to minimize the square–error norm

$$\epsilon^2 \equiv \| \tilde{D} - \sum_i d_i \psi_i \|_{\hat{q}(t)}^2 \tag{16}$$

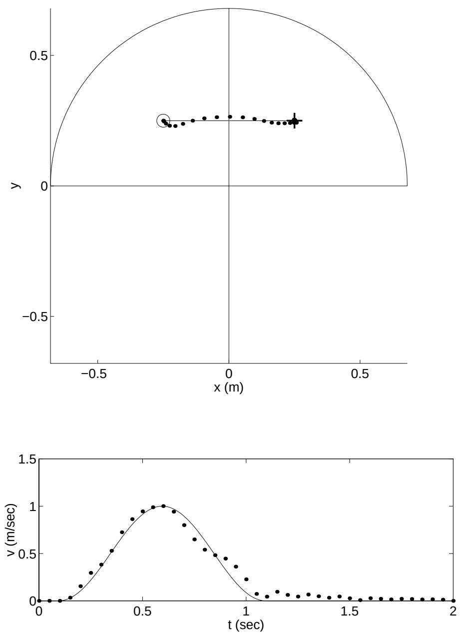

Figure 3: Movement toward a target generated by a combination of step and pulse fields. **Top**: Desired (solid line) and actual (dotted line) trajectories **Bottom**: Tangential velocity profiles of the desired (solid line) and actual (dotted line) movement.

This norm is the needed expression for the residual of Equation (13). Minimization is directly achieved by solving for d_i the system of linear equations:

$$\Psi_{j,i} c_i = \Lambda_j$$

$(\Psi_{j,i} = < \psi_j, \psi_i >_{\hat{q}(t)} \quad , \quad \Lambda_j = < \ \psi_j, \tilde{D} >_{\hat{q}(t)})$.

Time is "suppressed" by the projection operation (15) so that the final result is a set of fixed tuning parameters. The outcome of the least–squares is illustrated in Figure 3 which displays the trajectory obtained from the superposition of step and pulse fields. Note that there is a significant effect not only on the path but also on the tangential velocity profile that, after the pulse is added, is no longer determined by the passive impedance of the limb.

5 Virtual Trajectory

The observation of spring–like behavior in the motor system has led some investigators to suggest that a limb movement is planned and executed by the central nervous system as a temporal sequence of static attractor points for the limb [1, 5]. These are the equilibrium locations at which the static forces generated by the neuromuscular apparatus cancel out with each other. As the central commands directed to the muscles change in time, so does the attractor point. The temporal sequence of these points has been called a *virtual trajectory* [5]. The representation of movement control as a superposition of step and pulse fields is not only consistent with the notion of a virtual trajectory but it also provides a direct method for its computation (Figure 4).

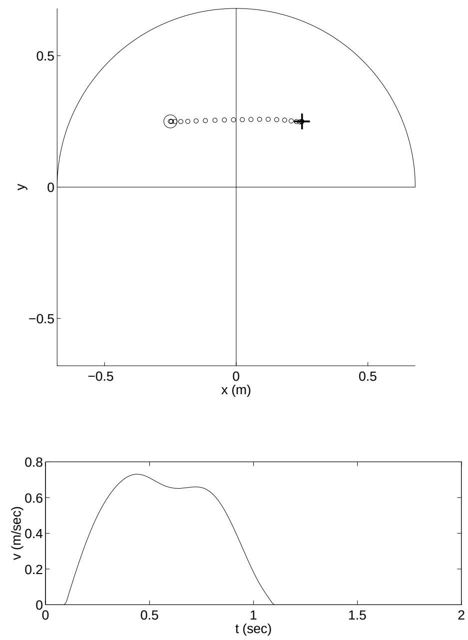

Figure 4: Virtual trajectry corresponding to the movement of Fig. 3. **Top**: Sequence of equilibrium points. **Bottom**: Velocity profile.

Once the parameters, c_i, d_i of the control field have been derived, one has obtained a full description of the basic differential equation (1) that generates the movement of the limb. The virtual trajectory is the sequence of configurations, $q_e(t)$ that is implicitly defined by the equation

$$M(q_e, t) \equiv D(q_e, 0, 0) - C(q_e, t) = 0. \qquad (17)$$

In this context, the virtual trajectory is merely a byproduct of force field approximation. It remains to be determined whether or not the virtual trajectory may play a role in computing the tuning parameters that achieve a desired movement.

6 Learning

By selecting the set of muscles connected to each spinal module, the motor system may improve its overall performance. If this process of adjustment is driven by an estimate of the error between actual and desired performance, we may call it learning. In the framework of field–approximation, the issue of learning may be addressed by considering how the set of basis fields can be modified to improve the execution of the desired movements. As an error measure one may take the difference between desired and actual movements or the same error norm (16) that was used to determine the tuning parameters. The second choice is less expensive in these simulations because it does not require integrating a set of differential equations. The error should be evaluated as a function of the parameters that are modified by the postulated learning process. With Gaussian fields, one may vary the coordinates of the centers and of the variance matrices. With a set of K 2–dimensional fields, this leads to a total of $5K$ parameters. These parameters, unlike the tuning coefficients, appear non–linearly in the expression of the controller field. Learning may then be simulated as a gradient descent over a complex error surface defined over the parameter space of the basis fields.

How adequate is a set of basis fields to learn a class of movements, such as straight–line motions of the hand with smooth velocity profile? To address this question, one may investigate how much does learning generalize across different movements of the same class. This is illustrated in Figure 5. Two sets of desired trajectories are shown at the top of the figure: training trajectories (forming a rectangular pattern) and test trajectories (forming a "Union Jack" pattern). In this case, the control system consisted of 9 kernel fields– giving rise to as many step and pulse fields– whose centers and variances were modified so as to reduce by a simplex method (Nelder and Mead, 1965) the net error measured after each repetition of the training set. The error on the test set was not involved in any way in this optimization. In the example shown in Figure 5 (solid line), the optimization resulted into an almost monotonic decrease of the error over the training set. The error on the test set displayed an interesting pattern. Initially, it followed the training error, then there was a sudden jump to high error values that lasted for a number of iterations. After this "burst" the test error returned to low values and stayed low for the remaining iterations The procedure lasted for more than 20,000 iterations before reaching what was likely a local minimum in the test error (only the first 2000 iterations are shown here). The final outcome on the two sets of trajectories is shown in Figure 6. The same tendency of the generalization error to reach a stable reduction after an optimization over the training set was observed in other simulations with different number of basis fields and different initial conditions. This finding indicates

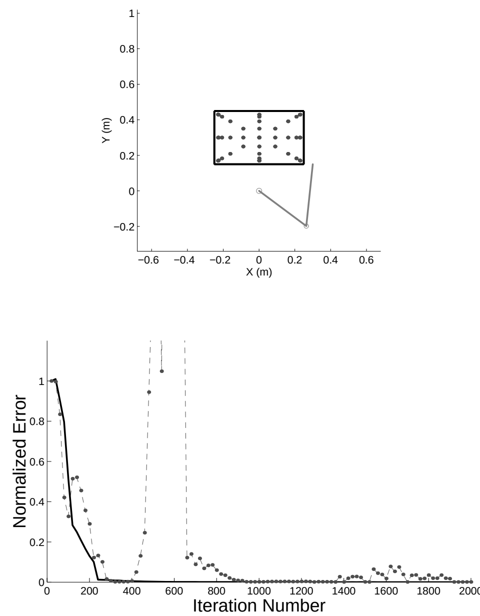

Figure 5: Learning **Top**: Training (solid lines) and test (dashed lines) trajectories. For each line there are two movements in opposite directions. **Bottom**: Optimization results. Vertical axis: percent error (initial error = 1). Solid line: Error over the training set. Dashed line: Error over the test set.

that basis fields such as the gradients of Gaussian functions constitute an appropriate control primitive for learning to control a repertoire of movement patterns across a substantial region of workspace. At the end of the optimization, one may say that the system has acquired a form of knowledge of the mapping that relates the kinematics of the joints and the dynamics of the limb to the geometry of the end–point space covered by the two sets of trajectories.

7 Concluding remarks

The task of generating coordinated movements in a multi–articular system may be carried out by combining the force fields generated by independent modules of control. This approach has emerged from experiments which suggested the presence of independent modules within the circuits of the spinal cord.

The work described here demonstrates that it is possible to generate a variety of coordinated movement by selecting the tuning parameters of a relatively small set of primitives. To derive these parameters I used a least–squares method that was made possible by the definition of an inner product between force fields. The method requires the evaluation of the dy-

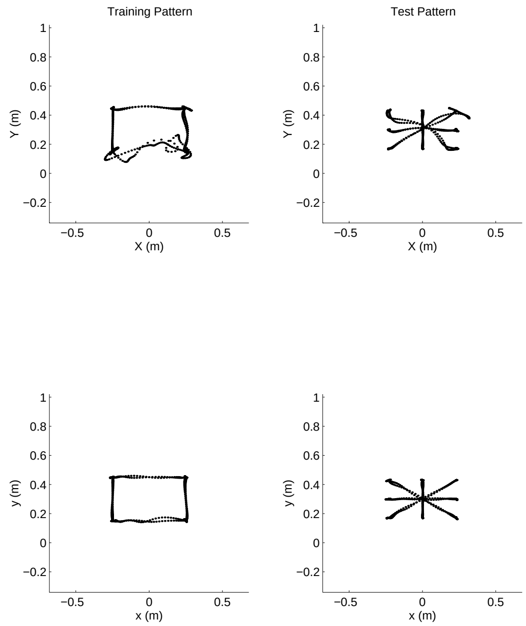

Figure 6: Optimization results over the training (right) and test (left) set. Initial (top) and final (bottom) performance.

namics and of the control field along the desired movements. While, this is a simple task for a computer simulation, it would be desirable to find alternatives that make use of available control and sensory signals rather than of some explicit evaluation of force fields. From this perspective, this work achieves a first step by demonstrating the existence of a simple association between entire trajectories and static patterns of control. An association that is established by representing the structure of each control module as a non–linear force field.

Acknowledgments

This research was supported by ONR grant N00014-95-1-0571, NIMH grant P50MH48185-0500 and NINDS grant NS-35673-01.

References

[1] Bizzi, E., N. Accornero, W. Chapple and N. Hogan (1984). Posture control and trajectory formation during arm movement. *Journal of Neuroscience* **4**, 2738 – 2744.

[2] Colgate, E. and N. Hogan (1989). An analysis of contact instability in terms of passive physical equivalents. In: *IEEE Proc International Conference on Robotics and Automation*. pp. 404 – 409.

[3] Giszter, S.F., F. A. Mussa-Ivaldi and E. Bizzi (1993). Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience* **13**, 467–491.

[4] Hatsopoulos, N. G. (1994). Is a virtual trajectory necessary in reaching movements?. *Biological Cybernetics* **70**, 541–551.

[5] Hogan, N. (1984). An organizing principle for a class of voluntary movements. *Journal of Neuroscience* **4**, 2745 – 2754.

[6] Hogan, N. (1985). Impedance control: an approach to manipulation: Parts i, ii, iii. *ASME Journal of Dynamic Systems, Measurement and Control* **107**, 1 – 24.

[7] Houk, J. C. and W. Z. Rymer (1981). Neural control of muscle length and tension.. In: *Handbook of Physiology. Section 1: The Nervous System, Vol. II, Part 2: Motor Control.* (V. B. Brooks, Ed.). American Physiological Society. Bethesda, MD.

[8] Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research* **42**, 223–227.

[9] Mussa-Ivaldi, F. A. and S. F. Giszter (1992). Vector field approximation: A computational paradigm for motor control and learning. *Biological Cybernetics* **67**, 491–500.

[10] Mussa-Ivaldi, F. A., S. F. Giszter and E. Bizzi (1994). Linear combinations of primitives in vertebrate motor control. *Proceedings of the National Academy of Sciences USA* **91**, 7534 – 7538.

[11] Mussa-Ivaldi, F.A., N. Hogan and E. Bizzi (1985). Neural, mechanical and geometrical factors subserving arm posture in humans. *Journal of Neuroscience* **5**, 2732 – 2743.

[12] Rack, P. M. H. and D. R. Westbury (1969). The effects of length and stimulus rate on tension in the isometric cat soleus muscle. *Journal of Physiology* **204**, 443 – 460.

[13] Robinson, D. A. (1981). The use of control system analysis in the neurophysiology of eye movements.. *Annual Review of Neuroscience* **4**, 463–503.

[14] Saltzman, E. L. and J. A. S. Kelso (1987). Skilled actions: A task–dynamic approach.. *Psychological Review* **94**, 84–106.