Structure of the set of feasible neural commands for complex motor tasks

Cohn BA[1], Szedlák M[2], Fukuda K[2], Valero-Cuevas FJ[1], and Gärtner B[2‡]

May 5, 2015

Abstract

The brain must select its control strategies among an infinite set of possibilities, thereby researches believe that it must be solving an optimization problem. While this set of feasible solutions is infinite and lies in high dimensions, it is bounded by kinematic, neuromuscular, and anatomical constraints, within which the brain must select optimal solutions. That is, the seat of feasible activations is well structured. However, to date there is not method to describe and quantify the structure of these high-dimensional solution spaces, other than bounding boxes or dimensionality reduction algorithms that do not capture its full structure. We present a novel approach based on the well-known Hit-and-Run algorithm in computational geometry to extract the structure of the feasible activations that produce [briantodo pick percent] of maximal fingertip force. We use a realistic model of a human index finger with 7 muscles, 4DOF, and 4 output dimensions. For a given force vector at the endpoint, the feasible activation space is a 3D convex polytope, embedded in the 7D unit cube. It is known that explicitly computing the volume of this polytope can become too computationally complex in many instances. However, our algorithm was able to produce [briantodo number] random points in the feasible activation space, which converged to the uniform distribution. [briantodo: include activationprogressions, parcoords, & costs] The computed distribution of activation across each muscle shed light onto the structure of these solution spaces—rather than simply exploring their maximal and minimal values. Although this paper presents a 7 dimensional case of the index finger, our methods extend to systems with up to at least 40 muscles. This will allow our motor control community to understand the distributions of feasible muscle activations, which will provide important contextual information into the learning, optimization and adaptation of motor patterns in future research.

**This work was supported by NIH NIAMS R01AR050520 and R01AR052345 grants, and SNF Project 200021-150055-1.

†[1]Departments of Biomedical Engineering and Computer Science at the University of Southern California Viterbi School of Engineering, Los Angeles, CA 90089, USA brianaco@usc.edu

‡[2]Department of Computer Science, ETH Zurich,[maytodo add the umlaut and the official name] Switzerland

1 INTRODUCTION

Muscle redundancy is the term used to describe the underdetermined nature of neural control of musculature. The classical notion of muscle redundancy proposes that, faced with an infinite number of possible muscle activation patterns for a given task, the nervous system optimizes in some fashion to select one solution. Here, each of N muscles represents a dimension of control on an end effector, and at any moment of a task, a muscle activation pattern exists as a point in $[0,1]^N$,— the n-dimensional hypercube — where each muscle's maximal activation is normalized to 1 [18]. Thus researchers often seek to infer the optimization approach and the cost functions the nervous system utilizes to select effective points in activation space to produce natural behavior [2, 12, 13, 16, 4, 7].

Implicit in these optimization procedures is the notion that there exists a well structured set of feasible solutions. Thus several of us have focused on describing and understanding those high-dimensional subspaces embedded in $[0,1]^N$ [10, 11, 15, 18, 8].

For the case of submaximal and static force production with a limb, the muscle redundancy problem is phrased in computational geometry: find the structure of the set of all feasible muscle activations, given the limb mechanics and the task constraints [1, 18, 17, 8]. We aim to explore what the solution space looks like, and uncover the structure of the feasible activation space for a given static force task. If each muscle's maximal activation is normalized to 1, the constraint $\mathbf{a} \in [0,1]^n$ describes that the feasible activation space lies in the n-dimensional unit hypercube (also called the n-cube); the activation space for the index finger is within the unit 7-cube.

1.1 High dimensionality difficulties

Consider a model of a static fingertip force, with 7 muscles arculating the index finger's 4 degrees of freedom (DoF). Assuming independent control of each muscle (non-synergistic model), each muscle has a unique force vector at the endpoint; the end effector has 7 unique vectors it can linearly combine to generate any vector of static force. This yields a unit 7-cube in charge of producing a 4-dimensional output wrench. On order to uncover the structure and relationship of these spaces, we cannot visualize all dimensions simultaneously.

This convex polytope is called the *feasible activation set*. To date, the structure of this high-dimensional polytope is inferred by its bounding box [10, 15, 8]. But the bounding box of a convex polytope we lose the details of its shape and volume comparison is not possible, since the polytope is a lower dimensional object embedded into $\mathbb{R}^n$. Empirical dimensionality-reduction methods have also been used to calculate a basis vectors for such subspaces [3, 5, 9]. But those basis vectors only provide a description of the dimension, orientation, and aspect ratio of the polytope, but not of its boundaries or internal structure.

Here we present a novel application of the well-known Hit-and-Run algorithm

[14] to describe the internal structure of these high-dimensional feasible activation sets. The input to our hit-and-run procedure requires a task force, along with the system's endpoint Jacobian, maximal tendon forces, and a moment arm matrix.

We applied our approach to two separate musculoskeletal models:

1. A fabricated schematic system, which we designed to have three muscles articulating one DOF, and one dimension of output force.

2. A realistic model, with seven muscles articulating four DOFs, and four dimensional output force [18].

With this, below are the key ideas and findings we present with this paper:

- For some muscles, we found that the bounding box exceptionally misconstrues the actual shape of the feasible activation space.
- Hit-and-run sampling of the solution space is computationally tractable; fewer than ten thousand points uncovered the shape of the distributions.
- Our approach provides a more granular context to the space within which the central nervous system optimizes.
- We apply six different cost functions (post-hoc) to all solutions, thereby providing spatial context to where 'optimal' solutions lie within the space.
- We designed an interactive parallel coordinates platform for visualizing and manipulating constraints to the solution space, such as muscle dysfunction, muscle hyperactivity, as well as constraining the upper and lower bounds for six different cost functions. We can compare cost functions side-by-side and view subsets of the dataset after applying cost function constraints.

2 METHODS

In the case of a tendon-driven limb with n muscles, the feasible activation space is the unit n-hypercube (as muscles can only be activated positively from 0 to a maximal normalized value of 1). As explained in [17], when task constraints are introduced to the system, the feasible activation set is further reduced; in this context, a task is a static force vector produced at the endpoint of the limb, which is represented as a set of inequality and equality constraints. Thus if this simple limb meets all constraints, the feasible activation set of the polytope P contains all feasible activations $\mathbf{a} \in \mathbb{R}^n$ that satisfy

$$\mathbf{f} = A\mathbf{a}, \mathbf{a} \in [0,1]^n,$$

where $\mathbf{f} \in \mathbb{R}^m$ is a fixed force vector and $A = J^{-T}RF_o \in \mathbb{R}^{m \times n}$— where J, R and F_0 are the matrices of the Jacobian of the limb, the moment arms of the tendons, and the strengths of the muscles, respectively [18, 17]. P is bounded by the unit n-cube since all variables a_i, $i \in [n]$ are bounded by 0 and 1 from below, above

respectively. Consider the following 1×3 fabricated example, where the task is a 1N unidimensional force.

$$1 = \frac{10}{3}a_1 - \frac{53}{15}a_2 + 2a_3$$
$$a_1, a_2, a_3 \in [0,1],$$

the set of feasible activations is given by the shaded set in Figure 2.

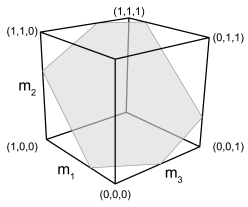

Figure 1: The feasible activation set for a three-muscle system meeting one functional constraint is a polygon in $\mathbb{R}^3$. Note that muscle activations are assumed to be bounded between 0 and 1.

2.1 Hit-and-Run algorithm

Hit-and-Run is a method used to uniformly sample a convex body [14]. The mixing time is known to be $\mathscr{O}^*(n^2R^2/r^2)$, where R and r are the radii of the inscribed and cicumscribed ball of K respectively [**?**, **?**]. I.e., after $\mathscr{O}^*(n^2R^2/r^2)$ steps of the Hit-and-Run algorithm has sampled a point uniformly at random in the convex body. Unfortunately the hidden constant is large, which makes the problem practically almost infeasible. However experimental results suggest that a number of points linear w.r.t. to the dimension suffices, which will be discussed in Section 2.3.

The Hit-and-Run walk on P is defined as follows (it works analogously for any convex body):

1. Find a starting point **p** of P (Figure 2a) .
2. Generate a random direction from **p** (uniformly at random over all directions) (Figure 2b).
3. Find the intersection points of the random direction with the edges of the polytope (Figure 2c).
4. Pick a random distance along the line formed by the endpoints (Figure 2d).
5. Repeat from (a) the above steps with the new point as the starting point .

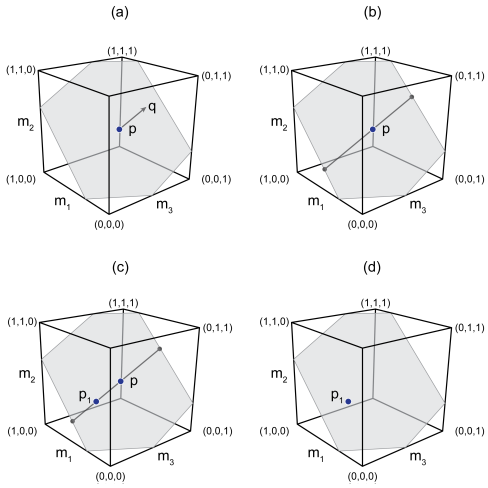

Figure 2: Graphical description of the Hit-and-Run algorithm.

2.2 Selecting a central seed point

To find a starting point in

$$\mathbf{f} = A\mathbf{a}, \mathbf{a} \in [0,1]^n,$$

we only need to find a feasible activation vector; that said, for the Hit-and-Run algorithm to mix faster we want a centrally located point within the polytope- that way, the early points will not be clustered in a corner [maytodo cite]. We use the following standard trick with slack variables ε_i to select a point where activations a_i for all muscles are far from 0 and 1, thereby finding a solution central within P [maytodo cite the use of slack variables to improve mixing time].

$$\begin{array}{lrcl} \text{maximize} & \sum_{i=1}^{n} \varepsilon_i & & \\ \text{subject to} & \mathbf{f} & = & A\mathbf{a} \\ & a_i & \in & [\varepsilon_i, 1-\varepsilon_i], \quad \forall i \in \{1,\ldots,n\} \\ & \varepsilon_i & \geq & 0, \quad \forall i \in \{1,\ldots,n\}. \end{array} \tag{1}$$

2.3 Removing inter-point dependence

As the function is recursive, any two successive points are not independent; therefore, we sample from our walk on P by selecting every m^{th} point to extract points which are distributed uniformly-at-random across P, and are independent to eachother. Some have said that it takes X Hit-and-Run steps before the first and last points gathered are independent of one another, so we chose to collect every [maytodo confirm 100th] point into an array of points $M = [p_0, p_1, p_2, ...p_{100,000}]$.

2.4 Determining target sample size

How many points do we need to record from Hit-and-Run to reach a representative distribution across the polytope? For convex polygons in higher dimensions c. 40, experimental results suggest that $\mathcal{O}(n)$ steps of the Hit-and-Run algorithm are sufficient. In particular Emiris and Fisikopoulos paper suggest that $(10+10\frac{n}{)}n$ steps are enough to converge upon the uniform distribution [6]. With the index finger model we collected a sample of [briantodo number] points.

2.5 Realistic index finger model

We used our published model in [18] to find matrix $A \in \mathbb{R}^{4\times 7}$, where $\mathbf{a} \in \mathbb{R}^7$; the four degrees of freedom were ad-abduction, flexion-extension at the metacarpophalangeal joint, and flexion-extension at the proximal and distal interphalangeal joints. The force directions we simulated are visible in Figure 3.

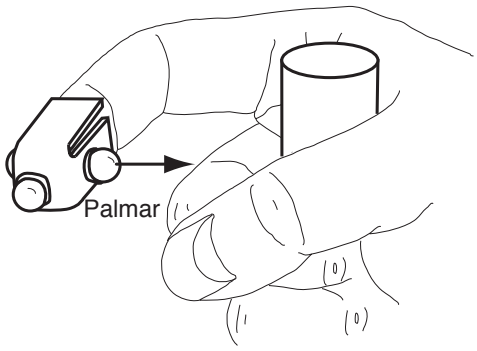

Figure 3: The index finger model simulated 50% of maximal force production in the palmar direction. Adapted from [18].

2.6 Solution projection histograms

The resulting array of points M can be thought of as a matrix, where each row is a point, and each column is a muscle's activation. With this, we can enter a column of this data into a histogram; importantly, this approach projects the density of points. This can be used to approximate the relative volume of different sections of P- by slicing (binning) from each muscle's dimension.

2.7 Parallel coordinates visualization

A common way to visualize higher dimensional data is using parallel coordinates[briantodo citations]. To show our sample set of points in the feasible activation space we draw n parallel lines for each of the n muscles. With the axis labels of the line set between 0 and 1, each point is then represented by connecting their coordinates by

$n-1$ lines. Using an interactive surface we restrict each muscle function to any desired interval- see, figures [maytodo link these to result figures].

2.8 Integration of muscle-metabolic and neural drive cost functions

For every solution collected, we used popularly-used cost functions: we computed activation l_1, l_2 and l_3 norms, and the tendon-force l_1, l_2 and l_3 norms. Six additional vertical lines were added to the parallel coordinates plot to represent each cost function. With the same parallel coordinates framework as developed with muscle activation, we can restrict and subset solutions which fall into desired cost-function ranges, thereby masking sub-optimal solutions and highlighting only those meeting the criteria.

Name	**Cost function**	**Reference**
l_1	$\sum_{i=1}^{n} a_i$	REF
l_2	$\sqrt{\sum_{i=1}^{n} a_i^2}$	REF
l_3	$\sqrt[3]{\sum_{i=1}^{n} a_i^3}$	REF
$(weighted)l_1$	$\sum_{i=1}^{n} a_i Fo_i$	REF
$(weighted)l_2$	$\sqrt{\sum_{i=1}^{n} (a_i Fo_i)^2}$	REF
$(weighted)l_3$	$\sqrt[3]{\sum_{i=1}^{n} (a_i Fo_i)^3}$	REF

Table 1: Cost functions and their usage, where a_i and Fo_i represent a muscle's activation in a given solution, and that muscles MIC, respectively. [briantodo: cite cost functions]

For a given point $\mathbf{a} \in \mathbb{R}^n$ we are interested in the associated cost of every solution collected through Hit and Run.

We developed and tested our code in Ubuntu 14.04, Windows 8.1, and OSX Yosemite, using Scala 2.11.6 [briantodo cite] for our implementation of Hit-and-Run, R 3.1.3 [briantodo cite] for histograms and plots, and using Sygmatic Parcoord[briantodo cite] and d3.js[briantodo cite] for our interactive parallel coordinate visualization. All code required to replicate our research is readily available at https://github.com/bcohn12/space.

3 RESULTS

3.0.1 Density projection upon one dimension

Using Hit-and-Run to sample feasible activation sets, Figure 4 shows the distributions of activation solutions for a palmar submaximal force resulting from [briantodo number] solutions computed with Hit-and-Run sampling. This is the first time (to our knowledge) that the internal structure of the feasible activation set has been visualized for a sub-maximal force. Notice also that the lower and upper

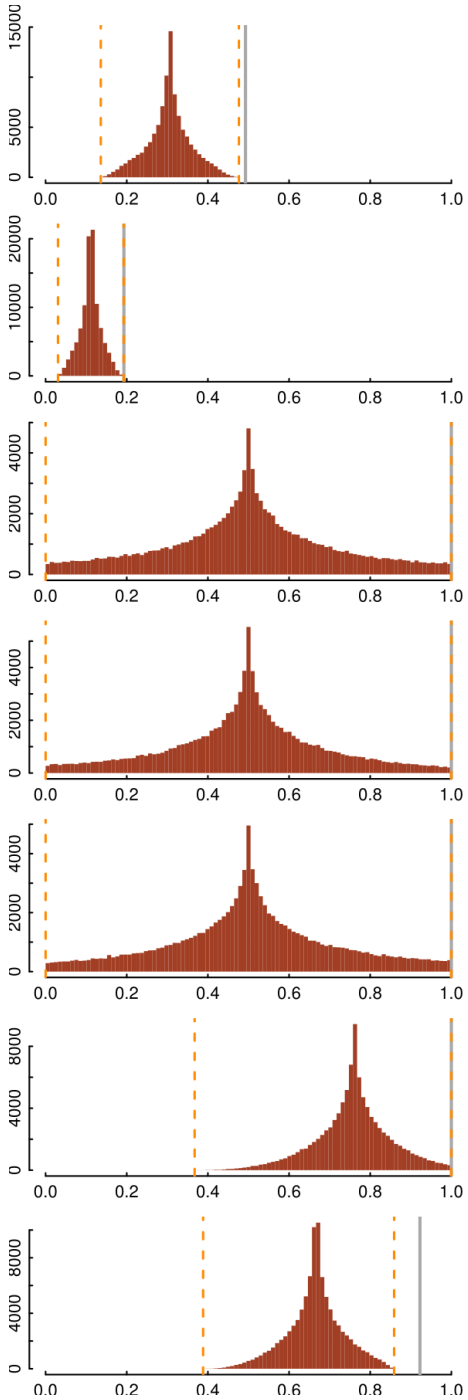

Figure 4: Distribution of feasible activations for [briantodo: select task percent]50% maximal force output in the palmar direction.

bounds of the activations (i.e., the dashed lines that indicate their bounding box), are uniquely uninformative of the actual density of distribution of feasible activations. Note also that the activation needed for the maximal force output (thick gray line) is very often not the mode of the activations at [briantodo select correct number]50%[maytodo: can we set this as a variable and use throughout?] of output. This figure shows ... [briantodo] talk about what the bounds mean with respect to the upper and lower bounds. Talk about the function of each muscle, with respect to the moment arm matrix and the relevant cell of the A matrix.

3.1 Activation spaces for increasing force

[1. briantodo generate pdfs as separate pages][2maytodo: insert a single pdf for palmar direction, and add other pages to the appendix] [briantodo write about how they are skewed, constrained, etc. Produce stats for each of the histograms and superimpose the data temporarily so you can write about it]

3.2 Parallel Coordinates

briantodo: add the following figures:

- parcoord Full
- parcoord muscle limited to 75%, then 50%, then 25% of its normalized maximal activation
- parcoord cost limited to 25% of costfn1, costfn2, costfn3

discuss what happens when you bring each of those muscles down, using the R produced stats. talk about how when you add X as a constraint, most of the solutions are distributed across the other muscles between X and X. Say which ones go up, which go down- which ones become clustered and which ones lose their peak/spike.

4 DISCUSSION

4.1 Parallel coordinate slopes

[maytodo: Talk about what it means to have slopes in Parallel, what a very positive slope means/what a very negative slope means, and what the crossing-slopes mean. Also put forward a couple suggestions of how these slopes could be more quantitatively interpreted/analyzed]

4.2 Cost distributions

Yes in further studies we could put activation constraints directly in the A matrix, instead of bounds between 0 and 1. But there are no advantages to adding activation constraints beforehand in the A matrix, as sampling is uniform- as long as

the resulting dataset is large enough for your purpose. You could also put l1 and weighted l1 cost bounds as constraints in the A matrix. Cannot put higher order cost functions such as l2,l3 or weighted l2,l3.

4.2.1 Concluding remarks

Our results clearly show:

- The Hit-and-Run algorithm can explore the feasible activation space for a realistic 7-muscle finger in a way that is computationally tractable.
- For some muscles, we find that the bounding box exceptionally misconstrues the internal structure of the feasible activation set.
- The Hit-and-Run algorithm is cost-agnostic in the sense that no cost function is needed to predict the distribution of muscle activation patterns. Therefore, we can provide spatial context to where 'optimal' solutions lie within the solution space; this approach can be used to explore the consequences of different cost functions.
- The distribution of muscle activations often show and strong modes that will critically affect the learning of motor tasks.

In comparison to traditional bounding-box representations, our application of hit-and-run in this context is decisively superior in capability for meaningful visualization, value in extracting associations between solutions, and computational tractability, in addition to being veritable of the true solution distributions within the FAS. Our bodies exist within a feasible activation space, and once we enter this space then optimization is possible. In this way, we can think of the solutions space as an effectie model for exporation-exploitation. This can help us in comparing the structure of the activation space as a set of high-dimensional bayesian priors which are narrowed/shifted over time to compensate for learning and skill-development. Essentially, once you enter into task-independent variation, it becomes a question of identifying the region of the activaiton space which both satisfieds the spatiotemporal constraints, but also approaches optimality under efficiency/speed demands. We want to 'close the loop' between the nervous system commands, and the mechanical output, thereby uncovering how the CNS collaborates with newtonian physics to select neural commands which effecitvey coordinate multi-link limbs, so we can act, play, and dance in the real world. Mechanical demands constrain the total space of musculoskeletal coordination options, thus, motile organisms first 'explore' coordination strategies conducive to the desired movement, and recursively redefine the more optimal subspaces. Once a desired task is mapped to an effective coordination strategy (as in, it gets the job done),

then training and experience (exploration-exploitation) can aid in finding the best coordination. As many tasks are similar (ie. they require the similar force generation or torque production over the course of a movement), the activation patterns for simlar actions must be similar as well. Optimal coordination strategies for one task may be near-optimal for a similar task. On the other hand, a suboptimal coordination strategy that achieves one task, may be furiously off-target for a very similar task.

[briantodo: address the following:] any given manuscript must satisfy the following criteria:

- Originality
- Innovation
- High importance to researchers in the field
- Significant biological and/or methodological insight
- Rigorous methodology
- Substantial evidence for its conclusions

References

[1] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(3):295–313, 1992.

[2] E Y Chao and K N An. Graphical interpretation of the solution to the redundant problem in biomechanics. *Journal of Biomechanical Engineering*, 100:159–67, 1978.

[3] R. H. Clewley, J. M. Guckenheimer, and F. J. Valero-Cuevas. Estimating effective degrees of freedom in motor systems. *IEEE Trans Biomed Eng*, 55:430–442, Feb 2008.

[4] R.D. Crowninshield and R.A. Brand. A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics*, 14(11):793–801, 1981.

[5] Andrea d'Avella and Emilio Bizzi. Shared and specific muscle synergies in natural motor behaviors. *Proceedings of the National Academy of Sciences of the United States of America*, 102(8):3076–3081, 2005.

[6] Ioannis Z Emiris and Vissarion Fisikopoulos. Efficient random-walk methods for approximating polytope volume. *arXiv preprint arXiv:1312.2873*, 2013.

[7] JS Higginson, RR Neptune, and FC Anderson. Simulated parallel annealing within a neighborhood for optimization of biomechanical systems. *Journal of biomechanics*, 38(9):1938–1942, 2005.

[8] Valero-Cuevas F. J., Cohn B. A., Yngvason H. F., and Lawrence E. L. Exploring the high-dimensional structure of muscle redundancy via subject-specific and generic musculoskeletal models. *J Biomech*, In press, 2015.

[9] Vijaya Krishnamoorthy, Simon Goodman, Vladimir Zatsiorsky, and Mark L Latash. Muscle synergies during shifts of the center of pressure by standing persons: identification of muscle modes. *Biological cybernetics*, 89(2):152–161, 2003.

[10] Jason J Kutch and Francisco J Valero-Cuevas. Muscle redundancy does not imply robustness to muscle dysfunction. *Journal of biomechanics*, 44(7):1264–1270, 2011.

[11] Jason J Kutch and Francisco J Valero-Cuevas. Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS computational biology*, 8(5):e1002434, 2012.

[12] B. I. Prilutsky. Muscle coordination: the discussion continues. *Motor Control*, 4(1):97–116, 2000. 0 1087-1640 Journal article.

[13] Stephen H Scott. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5(7):532–546, 2004.

[14] Robert L Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.

[15] M Hongchul Sohn, J Lucas McKay, and Lena H Ting. Defining feasible bounds on muscle activation in a redundant biomechanical task: practical implications of redundancy. *Journal of biomechanics*, 46(7):1363–1368, 2013.

[16] Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.

[17] F. J. Valero-Cuevas. A mathematical approach to the mechanical capabilities of limbs and fingers. *Adv. Exp. Med. Biol.*, 629:619–633, 2009.

[18] F. J. Valero-Cuevas, F. E. Zajac, and C. G. Burgar. Large index-fingertip forces are produced by subject-independent patterns of muscle excitation. *J Biomech*, 31:693–703, Aug 1998.