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 6, 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.

[†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 [27]. 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 [4, 21, 22, 25, 6, 14].

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 $[0, 1]^N$ [17, 18, 24, 27, 15].

For the case of 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 [2, 27, 26, 15]. We aim to explore what the solution space looks like, and uncover the structure of the feasible activation space for given static force tasks. 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).

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), which will be further discribed in Section 2.5. 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.

The solution of the above system is a convex polytope is called the *feasible activation set* (see Section 2). To date, the structure of this high-dimensional polytope is inferred by its bounding box [17, 24, 15]. But the bounding box of a convex polytope excludes the details of its shape, thereby precluding comparison, since the polytope is a lower dimensional object embedded into $[0, 1]^n$. Empirical dimensionality-reduction methods have also been used to calculate a basis vectors for such subspaces [5, 7, 16]. 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 [23] 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 [27].

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

- The high-dimensional space serves as a Bayesian posterior probability distribution, with a nonparametric, piecewise shape.

- 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.

- 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 [26], 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 is given by the polytope containing all $a \in \mathbb{R}^n$, 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 [27, 26]. P is bounded by the unit n-cube

since all variables a_i, $i \in [n]$ are in the interval $[0,1]$. Consider the following 1×3 fabricated example, where the task is a 1N unidimensional force.

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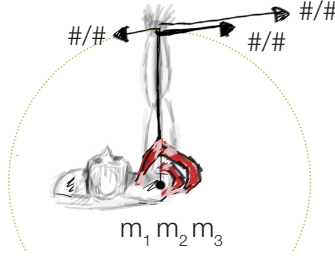
$$a_1, a_2, a_3 \in [0,1],$$

Figure 1: One imagined visualization of the fabricated tendon driven system, with 3 generators.

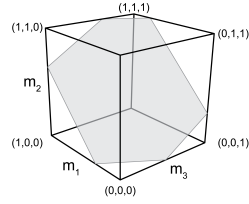The set of feasible activations is given by the shaded set in Figure 2.

Figure 2: The feasible activation set for a three-muscle system meeting one functional constraint is a polygon in $\mathbb{R}^3$.

2.1 Difficulty in multidimensional integration

Exact volume computations for polytopes is know to be *P*-hard [9]. Several algorithms have been suveyed and implemented, but can only handle up to 10 dimensions [3]. Recent models we have used have been 31 dimensions [15], and other muscle models have over 40 muscles involved [1, 18, 13, 8], thereby limiting the feasibility of using direct volume computations. Instead, we chose to sample the continuous space, effectively approximating the volume of the polytope by calculating point densities. [maytodo mention whether the volume computation is a closed-form solution]

2.2 Hit-and-Run algorithm

We chose to sample the activation space with the Hit-and-Run method because it is known to converge to the uniform distribution across a convex body [23]. The Hit-and-Run method is a generalization of the discrete Markov chains. The mixing time is known to be $\mathcal{O}^*(n^2 R^2/r^2)$, where R and r are the radii of the inscribed and cicumscribed ball of K respectively [9, 19]. I.e., after $\mathcal{O}^*(n^2 R^2/r^2)$ steps, the Hit-and-Run algorithm has sampled a point uniformly at random (u.a.r.) 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; this will be discussed in Section 2.4. As the feasible activation space of the muscles are given by a convex polytope, this method can be directly applied for our problem. Notably, there are other methods which sample uniformly; we chose Hit-and-Run because of [maytodo insert some other sampling paradigmns and our justification for Hit and run (include the Metropolis Hastings algorithm, Gibbs sampling?)].

[maytodo mention whether hit and run is a closed form solution]

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

2. Generate a random direction from **p** in P (uniformly at random over all directions) (Figure 3a).

3. Find the intersection points of the random direction with the edges of the polytope (Figure 3b).

4. Choose a point u.a.r. on this line segment given by the intersection points (Figure 3c).

5. Repeat from 1. the above steps with the new point as the starting point .

2.3 Implementation of Hit-and-Run

To find a starting point in P the polytope given by

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

we only need to find a feasible activation vector. For the Hit-and-Run algorithm to mix faster we want the starting point not close to a corner. We use the following standard trick with slack variables ε_i, which for applications often gives a good starting point.

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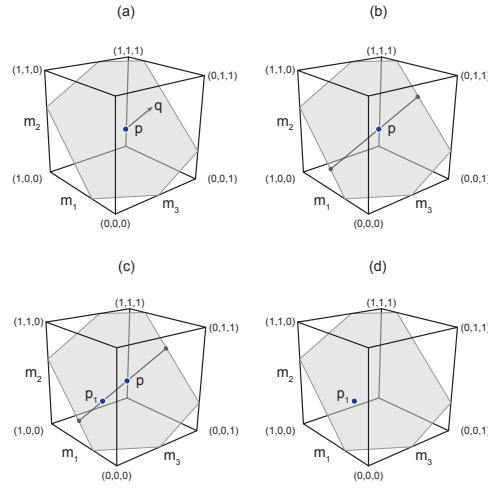\varepsilon_i &\geq 0, \quad \forall i \in \{1,\ldots,n\}.
\end{aligned}
\tag{1}
$$

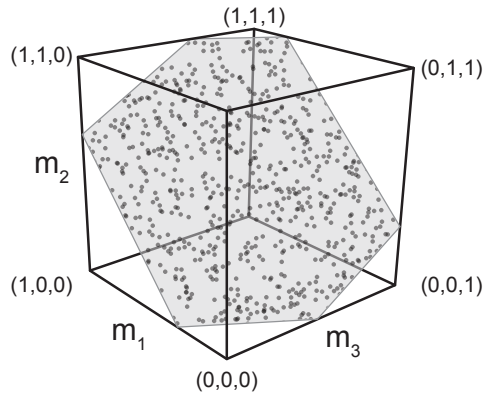Figure 3: Graphical description of the Hit-and-Run algorithm.

Figure 4: Uniform distribution aross the feasible activation space. In the schematic arm example, the distribution is represented within a 2D plane.

The rest of the implementation of the Hit-and-Run algorithm is straight forward except for the choice of the random direction. How do we sample u.a.r. from all directions in P? Suppose that $\mathbf{q}$ is a direction in P and $p \in P$. Then by definition of P, $\mathbf{q}$ must satisfy $\mathbf{f} = A(\mathbf{p} + \mathbf{q})$. Since $\mathbf{p} \in P$, we know that $\mathbf{f} = A\mathbf{p}$ and therefore

$$\mathbf{f} = A(\mathbf{p} + \mathbf{q}) = \mathbf{f} + A\mathbf{q}$$

and hence

$$A\mathbf{q} = 0.$$

We therefore need to choose directions uniformly at random from all directions in the vectorspace

$$V = \{\mathbf{q} \in \mathbb{R}^n | A\mathbf{q} = 0\}.$$

As shown by Marsaglia this can be done as follows [20].

1. Find an orthonormal basis $b_1, \ldots, b_r \in \mathbb{R}^n$ of $A\mathbf{q} = 0$.

2. Choose $(\lambda_1, \ldots, \lambda_r) \in \mathcal{N}(0, 1)^n$ (from the Gaussian distribution).

3. $\sum_{i=1}^{r} \lambda_i b_i$ is a u.a.r. direction.

A basis of a vectorspace V is a minimal set of vectors that generate V, and it is orthonormal if the vectors are pairwise orthogonal (perpendicular) and have unit length. Using basic linear algebra one can find a basis for $V = \{A\mathbf{q} = 0\}$ and orthogonalize it with the well known Gram-Schmidt method (for details see e.g. [10]). Note that in order to get the desired u.a.r. sample the basis needs to be orthonormal. For the limb case we can safely assume that the rows of A are linearly independent and hence the number of basis vectors is $n - m$.

2.4 Mixing Time

AFrom a given starting point, how many steps of the Hit-and-Run method are necessary to reach a u.a.r. point? For convex polygons in higher dimensions up to 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 $(10 + \frac{10}{n})n$ steps are enough to converge upon the uniform distribution [11], while in Ge et al.'s paper every point of the Hit-and-Run algorithm is used in the sample [12].

2.5 Realistic index finger model

We used our published model in [27] 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 5. In this model, for each input we collected 1.000.000 points and sampled every 100th point. Our code only solves one linear equation to find the starting point, each step only needs time linear in the dimension. Therefore it can be used for other tendon driven models in higher dimension. [briantodo: please rephrase this crappy sentence]
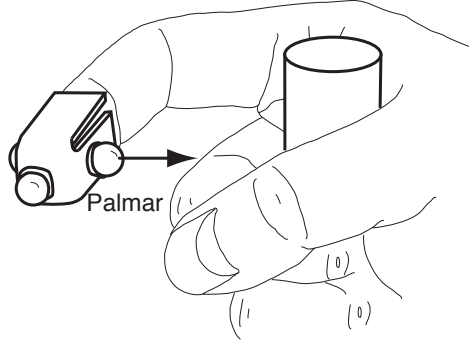
Figure 5: The index finger model force production in the palmar direction. Adapted from [27].

2.6 Solution projection histograms

The visualization for the distribution of solutions over each muscles is given in the histogramms in Figure6. Since the generated points are uniformly at random distributed, this gives a close approximation.

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 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^w, l_2^w and l_3^w 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.

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
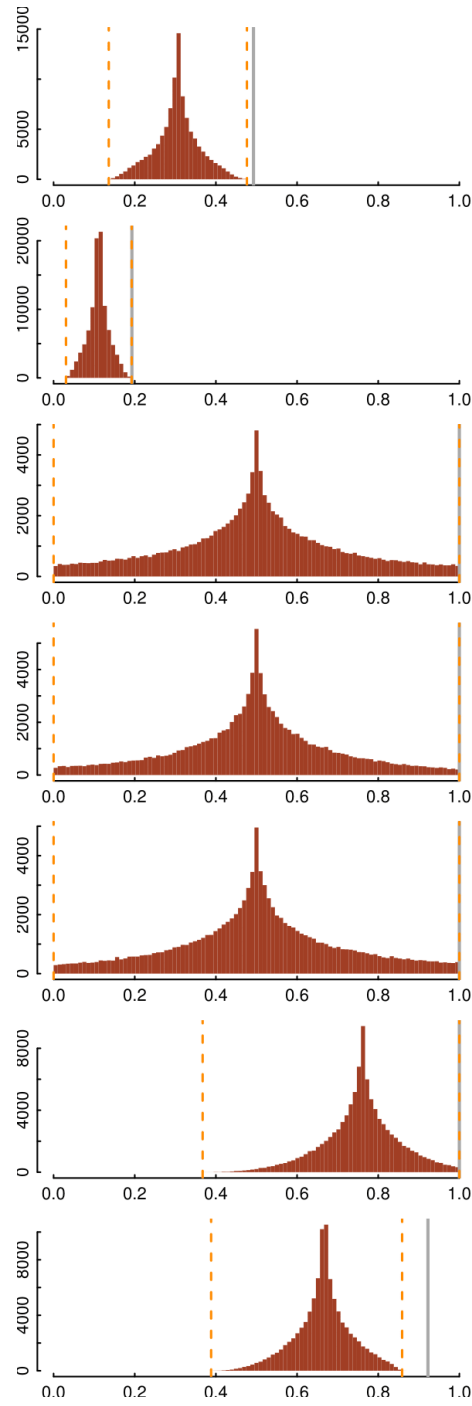
Figure 6: Distribution of feasible activations for 50% maximal force output in the palmar direction.

Name	Cost function
l_1	$\sum_{i=1}^{n} a_i$
l_2	$\sqrt{\sum_{i=1}^{n} a_i^2}$
l_3	$\sqrt[3]{\sum_{i=1}^{n} a_i^3}$
l_1^w	$\sum_{i=1}^{n} a_i F_{0i}$
l_2^w	$\sqrt{\sum_{i=1}^{n} (a_i F_{0i})^2}$
l_3^w	$\sqrt[3]{\sum_{i=1}^{n} (a_i F_{0i})^3}$

Table 1: Cost functions and their usage, where a_i and F_{0i} represent a muscle's activation in a given solution and that muscles MIC, respectively.

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 6 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 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.

The density integrals perpendicular to each muscle are provably unimodal due to the convexity [maytodo cite or add supporting evidence], and therefore it would be inadvisable to fit a normal distribution as the probability density function.

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]

The solution polytope converges as the difficulty of the task increases; the rate of convergence is different across muscles. For some muscles (such as X and Y) the convergence only occurs in the last [briantodo insert maximal feasible palmar force * 0.90], while others converge earlier-in lower forces of maximal (X and Y are examples of this)[briantodo fill out these descriptive statistics].

It is imperative to keep in mind that every histogram (regardless of its convergence) is composed of the distribution of all 10,000 points; when the distribution is compressed, the relative percentage of the bars will increase, as we made break width (Δx) remain constant to 2% of maximal contraction. [briantodo cite http://library.msri.org/books/Book31/files/ball.pdf]

The peaks seen in these figures refers to the perpendicular slice that has the highest number of points, and therefore has the highest volume (relative to the other parts).

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
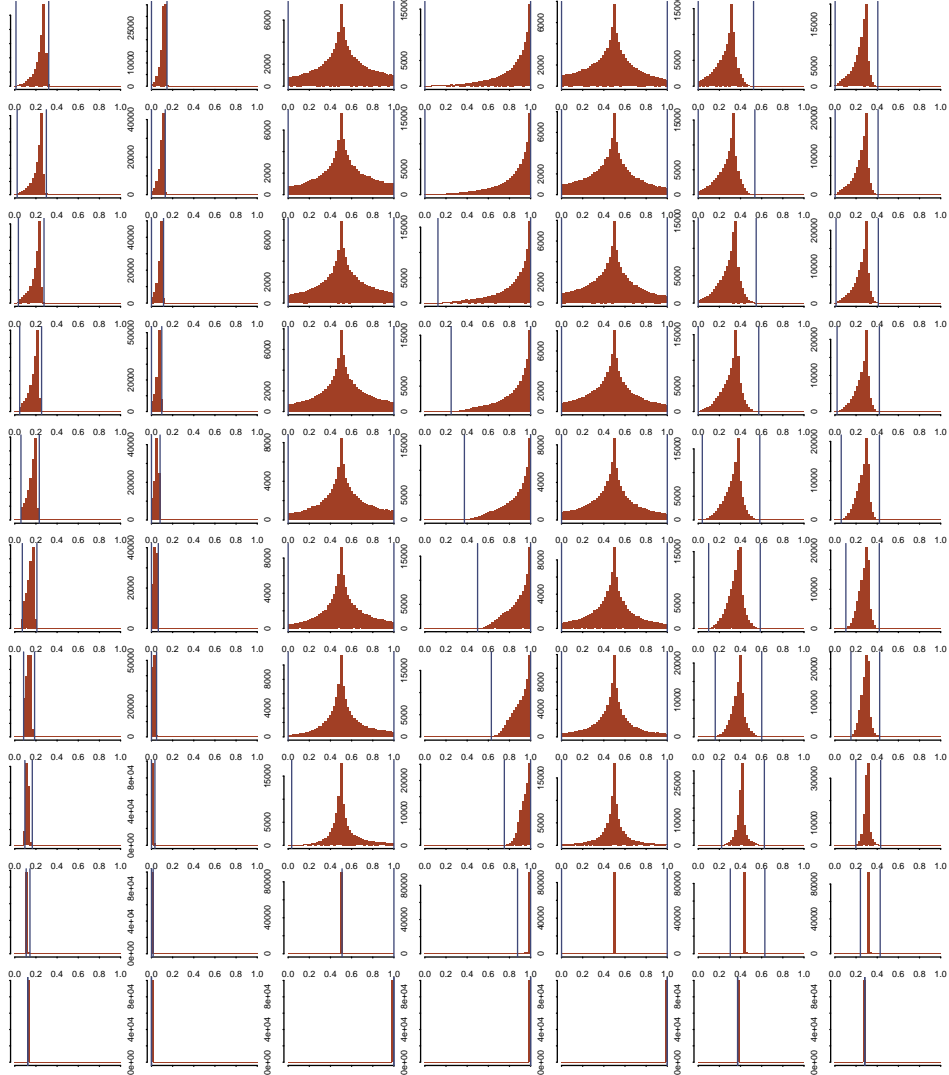
Figure 7: Distribution of activations in *XY*-direction and changing force. Each row of histograms uses a hit-and-run set. The height of each bar visualizes the percentage of 10,000 solutions found within a given 0.02 span of activation; the shape is more meaningful than the magnitude of the y axis, as we expect convergence (and therefore peak increase within few bins) towards maximal contractions.

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] Edith M Arnold, Samuel R Ward, Richard L Lieber, and Scott L Delp. A model of the lower limb for analysis of human movement. *Annals of biomedical engineering*, 38(2):269–279, 2010.

[2] 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.

[3] B. Büeler, A. Enge, and K. Fukuda. Exact volume computation for polytopes: A practical study. *Polytopes: Combinatorics and Computation*, 29:131–154, 2000.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Vincent De Sapio, Darren Earl, Rush Green, and Katherine Saul. Human factors simulation using demographically tuned biomechanical models. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 58, pages 944–948. SAGE Publications, 2014.

[9] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. *Proc. of the 21st annual ACM Symposium of Theory of Computing*, pages 375–381, 1989.

[10] T. S. Blyth E. F. Robertson. *Basic Linear Algebra*. Springer, 2002.

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

[12] C. Ge, F. Ma, and J. Zhang. A fast and practical method to estimate volumes of convex polytopes. *Preprint: arXiv:1401.0120*, 2013.

[13] Samuel R Hamner, Ajay Seth, and Scott L Delp. Muscle contributions to propulsion and support during running. *Journal of biomechanics*, 43(14):2709–2716, 2010.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] L. Lovász. Hit-and-run mixes fast. *Math. Prog.*, 86:443–461, 1998.

[20] G. Marsaglia. Choosing a point from the surface of a sphere. *Ann. Math. Statist.*, 43:645–646, 1972.

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

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

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

[24] 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.

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

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

[27] 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.