

The high-dimensional structure of feasible muscle activations reconciles alternative approaches to motor control

Brian A. Cohn^{1,❶}, May Szedlák^{2,❶}, Komei Fukuda^{2,‡}, Bernd Gärtner^{2,‡}, Francisco J. Valero-Cuevas^{3,4*,‡}

1 University of Southern California, Department of Computer Science, Los Angeles, CA, USA

2 Swiss Federal Institute of Technology, Department of Theoretical Computer Science, Zurich, Switzerland

3 University of Southern California, Department of Biomedical Engineering, Los Angeles, CA, USA

4 University of Southern California, Division of Biokinesiology and Physical Therapy, Los Angeles, CA, USA

❶These authors contributed equally to this work.

‡These authors also contributed equally to this work.

✉Current Address: Ronald Tutor Hall, RTH-404 3710 S. McClintock Ave Los Angeles, CA 90089-2905, USA

* valero@usc.edu

Abstract

How does the nervous system selects a particular muscle activation pattern from among multiple options? We present a computational and conceptual framework to unify the competing theories of (i) optimization of physiological cost functions, (ii) combination of low-dimensional synergies, and (iii) probabilistic learning of valid activations. The neuromechanical definition of a task also defines its feasible activation set (FAS), which is the landscape upon which neuromuscular learning, performance and optimization must occur. For example, the FAS to produce static force with a finger with seven muscles is a convex polytope embedded in 7-D activation space. We demonstrate, for the first time, how sampling such high-dimensional polytopes reveals the structure of the FAS as joint probability distributions, or Bayesian priors. This allows us to weigh the pros and cons of optimization approaches and dimensionality reduction techniques. This provably complete characterization of muscle redundancy—extendable to limbs with many muscles—provides an integrative perspective to motor learning and control.

Introduction

How the nervous system selects specific muscle activations for a given motor task is hotly debated. Some suggest the nervous system optimizes physiological cost functions [1–6], or combines low-dimensional synergies [7–14], or learns probabilistic representations of valid muscle activations [15–18]. At the core of this problem lies the computational challenge of describing and understanding the structure of high-dimensional ‘feasible activation sets’ (for a review see [19]). A feasible activation set is the family of valid solutions (i.e. muscle activation patterns) available to the nervous system to produce a given motor task. Figure 1 describes the neuromechanical interactions that define the feasible activation set for a particular task. The most the

nervous system can do is explore and exploit the mechanical wherewithal of the limb to meet the constraints of the task. In fact, it is the structure of the feasible activation set that defines the landscape upon which all neuromuscular learning, optimization and performance must occur. Therefore, research on the nature of muscle coordination must focus on how nervous system finds, explores, inhabits, and exploits those sets of feasible solutions [7–13].

But what defines a valid muscle activation pattern? As described in the Methods, a task is defined by its mechanical requirements (i.e., constraints) (Figure 1(f)), which must be satisfied by the biomechanical and muscular capabilities of the system (Figure 1(e)). Given the presence of muscle redundancy, the neuromechanical constraints for a particular motor task define the family of valid muscle activation patterns (i.e., the feasible muscle activation set) available to the nervous system to perform the task (Figure 1(d)). But the ‘curse of dimensionality’ [20–22] makes it conceptually and computationally challenging to calculate, describe and understand the nature and structure of those feasible solution sets [1, 11, 23–27]; even for time-invariant neuromechanical models of static force production [19, 28, 29].

We now present a novel approach to describe, in detail, the structure of feasible solution sets embedded in high-dimensional spaces. As an example, we describe the structure of the feasible activation set for the motor control problem of static fingertip force production [7, 30, 31]. This feasible activation set is a low-dimensional polytope embedded in 7-dimensional muscle activation space. Our approach hinges on the efficient sampling of high-dimensional polytopes to extract joint probability distributions of feasible activations across all muscles. These techniques can be used for larger-dimensional systems having c. 40 muscles. By providing a provably complete characterization of all muscle activation patterns for a given motor task, we are able to compare, contrast and reconcile today’s three dominant approaches to motor control.

Materials and Methods

The methods to obtain feasible activation sets for tendon-driven limbs are described in detail in the textbook *Fundamentals of Neuromechanics* [19]. We describe them briefly here.

A feasible muscle activation set contains all valid muscle activation patterns that can produce a motor task

Consider a tendon-driven limb, such as a finger, with n independently controllable muscles, where we define the neural command to each muscle as a positive value of activation between zero (no activation) and one (maximal activation). We can then visualize the set of all feasible neural commands (i.e., all possible muscle activation patterns) as the points contained in a positive n -dimensional cube with sides of length 1. A specific muscle activation pattern is a point (i.e., an n -dimensional vector \mathbf{a}) in this n -dimensional cube [1, 24, 25, 30].

Now consider a specific task, such as producing a given vector of static force with the fingertip to hold an object. Clearly, not all muscle activation patterns in the n -dimensional cube can produce that static force. The mechanical definition of the task is a set of constraint equations that specify the specific magnitude and direction of the fingertip force vector. As described in [1, 24, 25, 30] the equations for the mechanical constraints have the geometric equivalent of the equation of mechanical constraint is a plane, or manifold, in n -dimensional activation space. \mathbb{R}^n . a task for the tendon-driven limb means adding constraints upon the n -cube, such that only a subset of the points within the n -cube will produce the desired force. In the context of producing a vector of

static force with the endpoint of the limb in a given posture, the constraints that define that task (i.e., the direction and magnitude of the force vector at the endpoint) are linear equations. Each constraint equation defines a hyper-plane of dimension $n - 1$. The *feasible activation set* of the task, if it exists, is formed by the intersection of the n -cube and the constraint hyperplanes. The feasible activation set is given by the convex polytope P containing all $\mathbf{a} \in \mathbb{R}^n$, that satisfy

$$H\mathbf{a} = \mathbf{f}, \mathbf{a} \in [0, 1]^n \quad (1)$$

where H is the constraints matrix defined by the limb mechanics, \mathbf{a} is the vector of activations, $\mathbf{f} \in \mathbb{R}^m$ is the desired endpoint force vector. Muscles can only pull, so elements of \mathbf{a} cannot be negative, and is capped at 1 (where a muscle activation of 1 maps to 100% of maximal tendon force). The dimensionality of the output, m , is at most 6-dimensional (i.e., 3 forces and 3 torques) depending on the number of kinematic degrees of freedom of the limb, and usually $m < n$ because limbs have numerous muscles [19]. Thus, if \mathbf{f} is a feasible output, the feasible activation set P is a $(n - m)$ -dimensional convex polytope embedded into the n -cube; increasing task specificity reduces the size of the solution set [25, 29, 30].

Using the Hit-and-Run algorithm to uniformly sample from the feasible activation set

We are interested in characterizing the qualities that define all valid muscle activation patterns (i.e., n -dimensional vectors \mathbf{a}), that are points that make up P . This is equivalent to characterizing the structure of the convex polytope P . But calculating the geometric properties of convex polytopes in high dimensions is computationally challenging. Taking the generalized concept of a n -dimensional volume as an example of a geometric property of interest, the exact volume computations for n -dimensional polytopes is known to be tractable in a polynomial amount of time (i.e., $\#P$ -hard) [32]. Currently available volume algorithms can only handle polytopes embedded in small dimensions like 10 or slightly more [33]. Studying limbs in general, however, can require including several dozen muscles, such as our studies of a 17-muscle human arm and a 31-muscle cat hindlimb model [28]; and other limb models have over 40 muscles such as [7, 34–36].

Similar difficulties arise when computing other geometric properties such as the shape and aspect ratio of P in high dimensions. We and others have described polytopes P by their bounding box (i.e., the range of values in every dimension) [29, 37], but that singularly overestimates the shape and volume of the feasible activation set as discussed in [28]. Take Fig. ?? as an example, where the bounding box of the 2-dimensional polygon has a volume—even though a plane has zero volume—, and can be almost as large as the positive unit cube itself. Similar problems arise in the interpretation of the inscribed and circumscribed ball [38].

[note the precise wording of this next sentence... and propagate throughout] We propose a probabilistic method describing the structure of feasible activation sets P by the statistics, histograms, and point densities of numerous valid muscle activation patterns \mathbf{a} uniformly sampled from the polytope. To do so, we use the Hit-and-Run method. It is known to converge to a uniform sampling across any convex body, called K in the general case, up to about 40 dimensions [39]. The Hit-and-Run method is a generalization of a discrete Markov chain, which recursively samples a sequence of points in K as described below.

The application of the Hit-and-Run method to our feasible activation polytopes P is defined as follows (it works analogously for any convex body even if not a polytope) [40] and further discussion of the implementation is available in the Appendix:

1. Find a point \mathbf{p} in P to use as a starting point by using the method documented in the Appendix.
2. Generate a random direction q (uniformly-at-random over all directions) from \mathbf{p} in P (Fig. ??a).
3. Find the two intersection points of the line given by the random direction q with the boundary of the polytope (Fig. ??b).
4. Choose a new point uniformly-at-random on the line segment between the intersection points (Fig. ??c).
5. Repeat from 2. the above steps using the new point as the starting point, and generate a new random direction. Continue this process for s iterations until the model is mixed as shown in Fig. ???. Mixing time is the number of successive samples required to assert that the s th point is pulled u.a.r. from the polytope, and is also referred to as convergence to the uniform distribution.

We describe the mathematical basis and full notes of our implementation in the Appendix, including our application of slack variables to select a valid starting point within the feasible activation set.

Realistic index finger model

We applied this methodology to our published model of an index finger for static fingertip force production. The model of a right human index finger is described in detail elsewhere [30, 41, 42]. Briefly, the input to the model is a 7-D muscle activation pattern \mathbf{a} , and the output is a 4-D wrench (i.e., static forces and torques) at the fingertip \mathbf{w}

$$\mathbf{w} = H\mathbf{a} \quad (2)$$

$$H = J^{-T}RF_o \quad (3)$$

$$H \in \mathbb{R}^{4 \times 7} \quad (4)$$

where

$$\mathbf{a} = \begin{pmatrix} a_{FDP} \\ a_{FDS} \\ a_{EIP} \\ a_{EDC} \\ a_{LUM} \\ a_{DI} \\ a_{PI} \end{pmatrix} \quad (5)$$

In Cartesian coordinates, the output wrench is

$$\mathbf{w} = \begin{pmatrix} f_x \\ f_y \\ f_z \\ \tau_x \end{pmatrix} \quad (6)$$

which corresponds to these anatomical directions shown in Fig. ??

$$\mathbf{w} = \begin{pmatrix} f_{radial} \\ f_{distal} \\ f_{palmar} \\ \tau_{radial} \end{pmatrix} \quad (7)$$

The biomechanical model H includes three serial links articulated by four kinematic degrees of freedom (ad-abduction, flexion-extension at the metacarpophalangeal joint, and flexion-extension at the proximal and distal interphalangeal joints). The action of each of the seven muscles (FDP: *flexor digitorum profundus*, FDS: *flexor digitorum superficialis*, EIP: *extensor indicis proprius*, EDC: *extensor digitorum communis*, LUM: *lumbrical*, DI: *dorsal interosseous*, and PI: *palmar interosseous*) on each joint to produce torque is given by the moment arm matrix $R \in \mathbb{R}^{4 \times 7}$. Lastly, $J \in \mathbb{R}^{4 \times 4}$ and $F_0 \in \mathbb{R}^{7 \times 7}$ are the Jacobian of the finger with 4 kinematic degrees of freedom, and the diagonal matrix containing the maximal strengths of the seven muscles, respectively [19, 30, 41]. The finger posture was defined to be 0° ad-abduction and 45° flexion at the metacarpophalangeal joint, and 45° and 10° flexion, respectively, at the proximal and distal interphalangeal joints.

Feasible activation set for a static force production task

Our goal is to find the family of all feasible muscle activation patterns that can produce a given task. In particular, the task we explored is producing various magnitudes of a submaximal static force in the distal direction f_{distal} — in the absence of any τ_{radial} , shown in Fig. ???. Therefore the feasible activation set is a polytope P in 7-dimensional activation space that meets the following four linear constraints in \mathbf{a} [19, 30, 41]

$$f_{radial} = 0 \quad (8)$$

$$f_{distal} = \text{desired magnitude as \% of maximal} \quad (9)$$

$$f_{palmar} = 0 \quad (10)$$

$$\tau_{palmar} = 0 \quad (11)$$

These four constraints on the static output of the finger yield a 3-dimensional (i.e., $7 - 4 = 3$) polytope P embedded in 7-dimensional activation space. For details on how to create such models, apply task constraints and find such polytopes via vertex enumeration methods, see [19, 30].

Thus all valid output wrenches will have the form

$$\mathbf{w} = \begin{pmatrix} 0 \\ \text{desired magnitude as \% of maximal} \\ 0 \\ 0 \end{pmatrix} \quad (12)$$

For the index finger model used in this paper, the maximal feasible force in the distal direction is 28.81 Newtons. For the maximal force, there is only one solution, but with submaximal forces there are infinite solutions. Within the infinite space, one can use this model to find a single optimal muscle activation pattern—as opposed to the family of all feasible muscle activation patterns. This is done, for example, by phrasing the problem as a linear or quadratic program using linear or quadratic cost functions on muscle activation values, respectively [19, 30, 43].

Histograms of sampled points for each muscle

The goal of this paper is to uniformly sample valid muscle activation patterns \mathbf{a} from the polytope P to extract a detailed and provably complete probabilistic representation of the structure of P and the relationships among the valid \mathbf{a} muscle activation patterns. Muscle-by-muscle histograms are a straightforward way to visualize the many points sampled from the convex polytope. Histograms are particularly helpful because they are

approximations to probability density functions. This helps us visualize the relative number of solutions (i.e., density of solutions) that required a particular level of activation from a particular muscle within its native range of $[0, 1]$. In addition, the upper and lower bounds of the histograms show the length of the bounding box of the polytope in every dimension.

Parallel coordinates visualization shows the location of all points across all dimensions

Parallel coordinates are a common graphical way to visualize interactions among high-dimensional data, which has been used in biomechanical studies [44, 45] [pls show me these papers]. To demonstrate this visualization method, consider the results of the simple 3-dimensional example shown in Fig. ???. We begin by drawing n parallel vertical lines for each of the dimensions n (i.e., 3 muscles). With the axis limits of each line set between 0 and 1, each point (Fig. 4a) is then represented by connecting their coordinates by $n - 1$ lines as shown in Fig. 4b.

Neural and metabolic cost functions

As mentioned in the Introduction, the field of neuromuscular control has a long historical tradition of using optimization to find muscle activation patterns that minimize effort, which requires the (often contentious) definition of cost functions [2, 5]. Therefore, we used four representative cost functions to calculate the relative fitness of each of the 10,000 muscle activation points sampled. The cost functions are defined at the level of neural effort (L_1 , and L_2 norms); and at the level of metabolic cost, thought to be approximated by neural drive weighted by the strength of each muscle (L_1^w and L_2^w norms). To visualize the cost of each valid muscle coordination pattern, we simply added four vertical lines, one for each cost function (Table ??), to the parallel coordinates plot to show the associated costs of each point. Fig. 4c illustrates this mapping.

Dimensionality reduction

[[add PCA methods here]]

Results

We sampled from the feasible activation set for the task of producing nine different sub-maximal magnitudes of static fingertip force in the distal direction, evenly spaced between 10% and 90% of maximal static force. These are labeled as intensity values $\alpha = 0.1, 0.2, \dots, 0.9$ in the Figs. For each sub-maximal force magnitude, we ran 1,000,000 Hit-and-Run iterations and sampled every 100th point to produce 10,000 uncorrelated points uniformly distributed in P . Recall that for 100% of maximal force P shrinks to a single, unique solution [?].

Parallel Coordinates Visualization

While histograms provide detailed descriptions of the relative density of solutions at various levels of activation for each muscle, parallel coordinates visualization effectively shows the relatedness of activation levels across muscles for a given task, magnitude of desired fingertip force, and cost.

We applied parallel coordinate visualization to the 10,000 sampled points for each sub-maximal force level to view how different parts of one muscle's distribution interact with the others. To maintain real-time interactivity of the plot in our interface, and without perceptible loss of accuracy, we used only the first 1,000 points collected for each task, from $\alpha = 0.1$ to $\alpha = 1.0$. This interactive plot can be seen at <https://parcoord.firebaseio.com/>. Fig. ?? shows the parallel coordinate visualization for the same case as in Fig. ??.

An interactive interface for parallel coordinate visualization allows us to explore subsets of the valid muscle activation patterns, such as those setting bounds on the activation of specific muscle. For example, restricting the range of muscle activation of one or multiple muscles shows us the activation levels of the remaining muscles (Fig. ?? and ??). This can be used to, say, simulate a 40% reduction in activation (due to muscle dysfunction, for example) in the two index finger muscles innervated by the radial nerve (EIP and EDC). Similarly, we can restrict to any subset solutions which fall into desired ranges for a given cost function, thereby characterizing families of sub-optimal muscle activations patterns, or the one unique optimal muscle activations pattern.

Parallel coordinates allow us to visualize not only the proportion of solutions lost when limiting one or more muscle activation or cost function to a given range, but also the location and interrelatedness of the remaining solutions. The density of different solutions is additionally highlighted by the color density resulting from overlapping lines. While we invite you to explore the nature of the feasible activation set at the interactive website for yourself, Fig. ?? shows six such explorations.

Constraining $PI < 60\%$ still leaves a solution space about three times as large as $DI < 60\%$. Observe that constraining $DI < 60\%$ already implies that $PI < 60\%$ as seen in Fig. ?? (b), (c). Constraining one muscle does have different effects on the other muscles as for example if EI is constrained below 60% activation, we see that the bounding box of PI is significantly restricted; however, the same constraint has no effect on the bounding box of EIP .

As there are few steep crossings between L_1 , L_2 and L_3 , the parallel coordinates suggest correlation of those three cost functions. Similar correlation is observed for the respective weighted functions. We provide a scatter plot for their relationships in Appendix Fig. ?? and ??.

Changes in the structure of the feasible activation space with increasing force magnitude, or how muscle redundancy is lost

The maximal static fingertip force vector in a given direction (i.e., at $\alpha = 1$) is produced by a unique combination of muscle activations [1, 19, 24]. Therefore, the histograms in Fig. ?? show how the multiple solutions available for sub-maximal forces change and shrink for increasing force magnitude, from $\alpha = 0.1$ to $\alpha = 1$. You can track the change in the feasible distributions of a given muscle's activations by following its column from top to bottom ending, naturally, with the unique value needed for maximal force magnitude.

Fig. ?? shows a more fine-grained view of this phenomena by taking 1000 steps from 0 to maximal distal force production, and representing those histograms as a density heatmap.

The solution polytope converges to a point (a unique solution) as the magnitude of the force increases, but the rate of convergence is different across muscles. For some muscles the convergence only occurs after $\alpha = 0.6$ or $\alpha = 0.8$ (as in LUM and EIP), while others converge across the entire progression (e.g. DI and PI). Whereas for example FDS already has a small range of feasible activations at $\alpha = 0.1$, EIP has feasible activation $[0, 1]$ up to 80%.

It is imperative to keep in mind that every histogram (regardless of its convergence) is composed of all 10,000 points obtained at each force magnitude. Thus when the distribution is more compressed, the relative percentage of the bars will increase (as evident by the increasing y-axis limits) because we defined the width of each bar (Δx) to be 2% of the total activation range [46].

The peak (i.e., modes) of each histogram is the slice of the polytope that has the largest relative volume along that dimension. These modes need not be centered, and the histograms need not be symmetric or have the same shape as the magnitude of the output force increases. As expected, the unique solution at $\alpha = 1.0$ appears as a single peak that is also its upper and lower bounds.

A first consequence of these histograms is that their structure cannot be inferred by their bounding boxes alone (i.e., upper and lower bounds). Consider the activation distributions between $\alpha = 0.7$ and $\alpha = 0.8$ for LUM, where the median changed by less than 4% while the lower bound increased by nearly 13%. Notably, we observe that a meaningful cross-muscle comparison of point distributions cannot be ascertained by the bounding box. For example, at a task of 10% of maximal distal force production, EIP and EDC both have lower and upper bounds of 0, and 1, respectively, yet their distributions are thoroughly distinct; the shape of EIP is more symmetric (lower 25% = 0.36, median = 0.5029, upper 75% = 0.62), while 75% of the solutions sampled have EDC higher than 0.74 (see Fig. ??).

We find this holds when comparing inter- and intra-muscle distributions. Consider the significant change in the shape of the distributions across the progression for EDC until the 60% task; the lower and upper bounds change less than 1% and 4%, respectively, while the median shifted by nearly 40%. In the most extreme case, the median activation can be exceptionally narrow, while the bounds are wide. For example, EIP at a 90% task although its activation is bounded between 0.1 and 0.81. That is, approximately 79% of the solutions exist with EIP activation between just 0.49 and 0.51.

Importantly, the modes of the distributions do not correspond to the scaled-down value needed for maximal force magnitude. That is, we placed a vertical grey line corresponding to scaled-down value of the unique solution at maximal force, denoted \mathbf{a}^* . Thus for example, if LUM converges to an activation of 1 for maximal distal force, we put a grey line at 0.8 for a force that is 80% of maximal, or $\alpha = 0.8$. This allows us to see that, for some muscles like DI and EIP, these scaled-down optimal activations can lie arbitrarily in their distributions.

PCA's ability to describe the feasible activation set is highly task-dependent

A detailed explanation of PCA and its application to high-dimensional data is available in [19].

The variance explained by PCA changes as the task force increases (Fig. 7). PC3 is exactly equal to the remaining variance not explained by the first two components—this is a result of the activation space being three-dimensional by construction (as the 7-dimensional space was presented a 4-dimensional task). The impact of impoverishing the number of samples fed to PCA illustrates that low sampling obscures real changes in dimensionality reduction performance. The drop in PC1 and PC2 variance-explained at 19.2N is not captured in a sample size of 10, but is clearly visible in samples of 100 and 1000 points.

It would be interesting to explore the performance of dimensionality reduction algorithms as noise, bias, and error is introduced.

We also explored how the PCA loadings would change when the output force is increased.

PC loadings change dramatically for PC1, but do not change nearly as much for PC2.

Discussion

Our novel computational approach to understanding the high-dimensional structure of feasible activation sets has important consequences to our conceptual and practical approaches to the neural control of musculature. Our work provides an integrative neuromechanical perspective that places the currently dominant optimization and dimensionality reduction approaches in perspective.

Technical considerations

Our Hit-and-Run approach has ample opportunities for performance improvements with parallel computing; our functional programming approach is highly conducive to this.

We validate the Hit-and-Run algorithm by using vertex enumeration methods as in [30, 47] to find the exact maximal and minimal activation for each muscle (i.e., the individual sides of the bounding box of P) at each sub-maximal force level, Figs. ?? and ???. We found that the difference between the exact and sampled bounds for all muscles was smaller than 0.001, or < 0.1 % of the [0, 1] range of each muscle. We chose Hit-and-run because of its easy structure and mixing guarantee; however, it would be interesting to compare Hit-and-Run with the Grid Walk, Ball walk, or other sampling paradigms [48].

Piercing the curse of dimensionality

We begin by pointing out that, at its core, our approach is a direct response to the so-called ‘curse of dimensionality.’ This computational challenge comes up in the context of optimization [4] or combinatronics [19] when used to understand how the nervous system solves the muscle redundancy problem. The term curse of dimensionality was first coined by Richard E. Bellman in the context of dynamic programming [20]. It refers to the fact that the difficulty of finding optimal solutions, or characterizing the set of feasible solutions, often grows exponentially with dimensionality (i.e., the number of muscles).

The Hit-and-Run algorithm is easy to implement. Grid Walk, Ball walk, or other sampling paradigms would be interesting to compare in performance and applicability to musculoskeletal modeling problems [48]. Nevertheless, in the field of neural control, the Hit-and-Run algorithm is an example of how to pierce the curse of dimensionality for musculoskeletal models with up to 40 muscles, which have historically remained beyond easy reach of our community because only specialists can get optimization or combinatronics algorithms to work with high-dimensional systems [4, 28, 29, 42].

The value of a cost function

How the brain confronts the problem of muscle redundancy remains a mystery. While some advocate optimization as a metaphor for the neural basis of voluntary motor muscle coordination [3, 4], others criticize it as physiologically unrealistic analogy [49, 50]. This criticism comes from two directions. The first is the choice of the cost function, and the second the notion that the nervous system can find optimal solutions in spite of the curse of dimensionality. However, let us recall that one of the reasons that optimization is used is because it is a tried-and-true method for engineers and mathematicians to solve underdetermined problems [42]. Cost functions are a necessary means to gauge the

merits of individual solutions, and the pitfalls of exploring high-dimensional spaces to find global (as opposed to local) minima are well known—both of which are compatible with the evolutionary pressures to use trial-and-error to find adequate solutions to physical problems. Thus many of the debates about the merits of cost functions [2, 5] and reasonable means to find global minima [51] are to a certain extent driven by the very real need to inject physiological realism to the engineering methods available to us.

The results of our Hit-and-Run approach allow us to place optimization in perspective. Finding the feasible activation set for a task is in fact the mathematical dual, or counterpart, to optimization [52, 53]. If optimization is like a very efficient means to feel your way around the solution space in the dark—guided only by your cost-function and local gradient; then finding the structure of the feasible activation set is like a computationally means to turn on the lights to see all possible solutions and their individual merits [19].

As mentioned above, being able to ‘turn on the lights’ is no longer as computationally challenging or expensive as it used to be as we demonstrate it is quite feasible and accessible to non-specialists. Having obtained the structure of the feasible activation set now allows us to see every see and interpret families of solutions for a variety of cost functions. Clearly, a cost function will still allow you to select a particular solution. But you can be more flexible and classify solutions as generally valid (i.e., good enough or sub-optimal), instead of insisting on finding the one optimal solution which may not be global, easy to find, generalizable, or robust.

Figures ?? and ?? show examples of this. Having sampled the feasible activation set for the task of producing a submaximal force in the distal direction, the parallel coordinate plots show the families of solutions and their associated merits as per six different cost functions. This allows one to see the joint distributions of muscle activations in a cost-agnostic way; or enforce a particular cost function to see the diverse population of solutions that share similar or different costs. One can see the landscape of solutions, and the multiple fitness landscapes that can be associated with them. This allows one to determine whether individual solutions, or families of solutions, are global, easy to find, generalizable, or robust.

Synergy and structure

We found that a sample of only 100 activations was sufficient to identify the effect of a changing task on the activation space’s lower-dimensionality approximation. This is in alignment with theories that control strategies are highly task-relevant, and extends those theories with a geometric visualization.

[[where to add this?:

To overcome these difficulties, it is commonplace (e.g., [54, 55]) to apply dimensionality reduction techniques to approximate the high-dimensional kinematic or muscle activation spaces [27, 56, 57] (Figure 1(g)). This dimensionality reduction approach is now being applied to ever higher-dimensional spaces of neural activity representing the dynamics of populations of motor neurons (e.g., [58, 59]) (Figure 1(a, b)). At heart, however, dimensionality reduction is a means to *approximate* the latent structure of solution sets embedded in high-dimensional spaces [19]—with its inherent advantages and limitations [55].

]]

The families of feasible solutions in Figures ?? and ?? are informative in other ways. As discussed in [19], several authors have pointed out that understanding neural control from the perspective of synergies requires that we distinguish between synergies that are extracted descriptively from data vs. synergies that are implemented prescriptively by a controller or the nervous system [7, 10, 14, 27]. As shown in our results, feasible activation sets have a well-defined, low-dimensional structure that is given by the

biomechanics of the limb and the mechanical constraints defining the task. This explains why there have been repeated reports of experimental data exhibiting correlated activity and modularity that reflects a lower-dimensional structure embedded in the native higher-dimensional space of the measured variables [54]. Therefore, the lines shown in the parallel coordinates plots describe in detail the obligatory correlations in muscle activations that are bound to be detected by dimensionality reduction algorithms because the feasible activation set is a lower-dimensional entity to begin with.

The question then is, then, how can one infer prescriptive synergies (i.e., the existence of synergies of neural origin) from experimental data that naturally exhibit descriptive synergies? This is the heart of the debate in this area at the moment. It is a subtle, hard to prove, distinction whether the low dimensionality of the observed data arises naturally from the nervous system meeting the constraints of the task (by whatever means) vs. whether the nervous system is prescriptively implementing a low-dimensional basis to meet the constraints of the task (to simplify the muscle redundancy problem or otherwise) [7, 10, 60]. This distinction is not purely semantic. It is our hope that our work, and the parallel coordinates exploration of the results it allows, aid us to explore these issues. For example, you can consider the family of coordination patterns that (i) are associated with a particular narrow range of values of a particular cost function; or (ii) have several muscles active within particular ranges like low for one, medium for another, and high for a third, etc. By showing in detail the obligatory correlations in activation values required by the mechanics of the limb and task, parallel coordinates plots allow one to explore the latitude the nervous system has in the coordination of the other muscles—and whether it uses that latitude or not. This is a computation capability we argued was necessary to advance our understanding of muscle synergies of neural origin [7], and we are happy to report that this exploration of the structure of high-dimensional feasible activation sets is now possible. This is in line with an emerging approach to treat computational models as experimental subjects to see how well current experimental techniques can detect the underlying control strategy purposely implemented in the model by the researcher, or evaluate the tradeoffs of alternative synergies (e.g., [?, ?, ?, 7, 13, 14]).

Learning the landscape

As mentioned above, the most the nervous system can do to control a limb is to identify, learn, and exploit its mechanical wherewithal. The structure of the feasible activation set is an internal representation of that mechanical wherewithal at the level of the controller. Therefore, as we have written elsewhere [7, 19], the question is not *whether* experimentally observed neural commands will inhabit a low-dimensional subspace. Rather, the question is *how* the nervous system finds and inhabits that low-dimensional subspace, i.e., implements prescriptive synergies. To make progress, we should focus on understanding the processes by which the nervous system finds, inhabits, traverses and remembers the feasible activation set to meet the requirements of real-world tasks [?, 11, 12, 31, 61].

Thankfully our results in the form of histograms and parallel coordinate plots provide, to our knowledge, the first characterization of the statistical distribution of muscle activations for a given task. This probabilistic approach to muscle coordination then allows us to explore the likelihood that a trial-and-error strategy (based on gradient descent or other heuristic or formal reinforcement learning approaches) will find functional solutions (i.e., land in the feasible activation set). Clearly, there are coordination patterns that are much more likely to be found by others as can be seen by the density of lines in the parallel coordinate plots.

Thus the structure of the feasible activation set provides strong priors that help interpret the nature of motor learning as related to exploration-exploitation strategies

driven by trial-and-error to find and remember the feasible activation set for a task (i.e., a family of good-enough solutions). There are several such strategies that have been producing good results where strict single-line optimization methods are brittle or fail to converge. For example the work of Lipson, Bongard, and colleagues [?, 62, 63]; and Schaal and colleagues [?, ?, ?] has used trial-and-error learning combined with model-free or semi-model free approaches (i.e., they do not require precise models or the full equations of motion of the body being controlled) to evolve complex behavior, or design controllers for complex behavior. It then starts to become biologically feasible that similar memory, interpolation and extrapolation approaches may be at the root of biological control—and the success of particular trial-and-error strategy for a given task is very much tied to the density of solutions (i.e., statistical priors) provided by the structure of the feasible activation set.

Implications for physical rehabilitation and performance improvement

The notion of the feasible activation set as the setting for learning raises unique perspectives of rehabilitation for injuries such as stroke and spasticity; if we introduce additional constraints (such as reduced muscle function in a set of muscles), understanding how the central nervous system must adapt to that change can help us design coaching programs that teach the other optimal corners of the solution space. Similarly, the same could be applied to professional performers seeking alternative solutions that could lower their sport-specific cost functions (e.g. speed or metabolic cost).

This is compatible with the notion that motor control is likely more habitual than optimal [49]. That is, once a particular region of the feasible activation set has been found and remembered—with some subregions with higher densities being easier to find—that is likely to become a ‘good enough’ (i.e., suboptimal) but nevertheless recurrent motor habit. It is only those individuals with skill and motivation to continue to explore the feasible activation space into less dense regions that are able to optimize, and thus win medals and such. But for most of us, easy to find, generalizable, and robust families of suboptimal solutions are likely good enough. This is made apparent in Figure ?? that shows that the level of activation of each muscle for the optimal solution capable of producing maximal force is not usually the mode of their histograms. That is, optimal solutions are not necessarily the most common, and therefore require either training or good coaching to be found given the curse of dimensionality mentioned above.

[To be done]

Supporting Information

Lower-dimensionality case study

Consider the simple case of $A \in \mathbb{R}^{1 \times 3}$ illustrated in Fig. ??, where the task is to produce a 1 Newton force in the rightward x direction with the endpoint of a rigid link that is hinged to lie in the plane, and is actuated by 3 muscles. The set of feasible activations is given by the shaded convex polygon in Fig. ???. This is because there are three independently controlled muscles meeting one linear constraint. Thus P is of dimension $(3 - 1) = 2$. In this case, we define a sample constraint equation

$$+ 1 = \frac{10}{3}a_1 - \frac{53}{15}a_2 + 2a_3 \quad (13)$$

$$a_1, a_2, a_3 \in [0, 1],$$

that represents how the sum of forces that each muscle produces at the endpoint must add to +1. The coefficients indicate the magnitude and direction of endpoint force each muscle produces.

Implementation of Hit-and-Run

The polytope P is defined by the mechanics of the limb and the constraints of the task as

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

and to find a starting point \mathbf{p} we need only to find one feasible activation vector in the space. For the Hit-and-Run algorithm to mix faster we do not want the starting point to be close to a vertex of the n -cube [64]. Intuitively, this is because a random direction from that point will be able to ‘travel far’ from the corners in P . We use the following standard trick with slack variables ϵ_i , which in practice often provides a good starting point for sampling algorithms. In essence, we use the slack variables to shrink the extreme values of the variables to find a solution that is in the core of the n -cube. [state the value of ϵ]

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n \epsilon_i \\ & \text{subject to} && \mathbf{f} = A\mathbf{a} \\ & && a_i \in [\epsilon_i, 1 - \epsilon_i], \quad \forall i \in \{1, \dots, n\} \\ & && \epsilon_i \geq 0, \quad \forall i \in \{1, \dots, n\}. \end{aligned} \quad (14)$$

Thus running a linear program with these reduced ranges of the variables will produce a starting point that is far from the extremes of the actual n -cube.

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 ? Recall that all points in P are by definition valid solutions to a given task. Therefore you can think of P as the null space of the task. Thus moving from point $\mathbf{p} \in P$ by the displacement vector $\mathbf{q} \in P$ takes you to another valid point $\mathbf{p}' = \mathbf{p} + \mathbf{q}$ that in fact produces the *same output*. Thus, by the definition of P as the feasible activation set, the direction \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

$$\mathbf{f} = A(\mathbf{p} + \mathbf{q}) = \mathbf{f} + A\mathbf{q} \Rightarrow A\mathbf{q} = 0.$$

This essentially says that the direction \mathbf{q} must be in the null space of the task, which is defined as $A\mathbf{x} = A(\mathbf{x} + \delta\mathbf{x})$ if $\delta\mathbf{x} \in N(A)$.

Hence we need to choose directions uniformly at random across the vector space

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

As shown by Marsaglia this can be done as follows [65].

1. Find an orthonormal basis $b_1, \dots, b_r \in \mathbb{R}^n$ of $\{\mathbf{q} \in \mathbb{R}^n | A\mathbf{q} = 0\}$.
2. Choose $(\lambda_1, \dots, \lambda_r) \in \mathcal{N}(0, 1)^r$ (from the zero-mean unit-variance Gaussian distribution).
3. $\sum_{i=1}^r \lambda_i b_i$ is a u.a.r. direction.

A basis of a vector space 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 $\{q \in \mathbb{R}^n | Aq = 0\}$ and orthogonalize with the well known Gram-Schmidt method (for details see e.g. [66]). Note that in order to get the desired u.a.r. sample the basis needs to be orthonormal. [let's discuss this: The limb model is defined such that the rows of A are linearly independent and hence $r = n - m.$]

Mixing Time

From a given starting point, how many iterations of the Hit-and-Run method are necessary to reach a u.a.r. point. For convex polytopes in n dimensions up to 40, experimental results suggest that $\mathcal{O}(n)$ steps of the Hit-and-Run algorithm are sufficient. In particular, the paper [67] by Emiris and Fisikopoulos suggests that selecting every $10(n + 1)$ points is sufficient to get a uniform distribution [67], while in Ge et al.'s paper every single point of the Hit-and-Run algorithm is used in the sample [68].

The Hit-and-Run algorithm was specifically developed to address $\#P$ -hard problems such as approximating the volume of high-dimensional polytopes [32]. The Hit-and-Run algorithms can converge to the uniform sampling across any convex body [39] after $\mathcal{O}^*(n^2 R^2 / r^2)$ steps (r and R are the radii of the inscribed and circumscribed ball of the polytope as mentioned in the Introduction) [32, 64]. Experimental results suggest that a number of points linear with respect to N suffices for convex polytopes embedded in dimensions up to 40. Emiris and Fisikopoulos [67] suggest that $(10 + \frac{10}{N}) N$ iterations suffice to uniformly sample a convex polytope.

0.1 Finger model data

$$F_0 = (123.0, 219.0, 23.52, 91.74, 21.6, 124.8, 129.6)$$

$$JR = \begin{pmatrix} -0.08941 & -0.0447 & -0.009249 & 0.03669 & 0.1421 & 0.2087 & -0.2138 \\ -0.04689 & -0.1496 & 0.052 & 0.052 & 0.0248 & 0.0 & 0.0248 \\ 0.06472 & 0.001953 & -0.1518 & -0.1518 & 0.2919 & 0.0568 & 0.2067 \\ 0.003081 & -0.002352 & -0.0001649 & -0.0001649 & -0.0004483 & 0.0001578 & -0.000685 \end{pmatrix}$$

Distal force is $task_z = (1.0, 0.0, 0.0, 0.0, 0.0)$

The result of these computations (evaluated in R) shows the cumulative percentage of points up to each of those points.

```
# fixed_db database where each row is a point, each column is a muscle, and one
library(stats)
eip_solutions <- fixed_db[fixed_db['alpha']==0.9,] [,3]
activations_of_interest <- c(0.49,0.51)
ecdf(eip_solutions, activations_of_interest)
```

We developed and tested our code in Ubuntu 14.04, Windows 8.1, and OSX Yosemite 10.10.3; implemented Hit-and-Run with Scala 2.11.6¹, developed our histograms and descriptive statistics with R 3.1.3 [69], and designed our parallel coordinate visualization with multiple open-source JavaScript projects ²³. All code and documentation used to develop this publication is readily available on a Github repository ⁴

The coordinate frame definition for the right index finger is defined by the last phalanx. From the fingertip, $+x$ is the radial direction, $+y$ is distal direction, $+z$ is palmar.

¹<http://www.scala-lang.org/>

²<http://syntagmatic.github.io/parallel-coordinates/>

³<http://d3js.org/>

⁴<https://github.com/bcohn12/space>

Acknowledgments

Research reported in this publication was supported by the National Institute of Arthritis and Musculoskeletal and Skin Diseases of the National Institutes of Health (NIH) under Awards Number R01 AR-050520 and R01 AR-052345 to FVC, and Swiss National Science Foundation (SNSF) Project 200021-150055-1 to BG, KF and FVC. The content is solely the responsibility of the authors and does not necessarily represent the official views of the SNSF or the NIH. We thank J Pugliesi, T Kim, Crystal Lim, A Baugus, P Vachhani, and A Boling for extending the scientific rigor of this work through reviews, documentation, and code.

Author contributions

M.S. derived the mathematics of our Hit and Run implementations, B.C. implemented those algorithms and designed the simulations. F.V.C supported the neurobiological interpretations, while K.F. and B.G. supported work with computational geometry.

Competing Financial Interests

We do not have any competing interest to declare.

Materials & Correspondence

F.V.C. can be contacted via valero@usc.edu, and the original code used to produce all analyses is freely available at <https://github.com/bcohn12/space>.

References

1. Chao E, An K. Graphical interpretation of the solution to the redundant problem in biomechanics. *Journal of Biomechanical Engineering*. 1978;100:159–67.
2. Prilutsky BI. Muscle coordination: the discussion continues. *Motor Control*. 2000;4(1):97–116.
3. Scott SH. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*. 2004;5(7):532–546.
4. Todorov E, Jordan MI. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*. 2002;5(11):1226–1235.
5. Crowninshield RD, Brand RA. A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics*. 1981;14(11):793–801.
6. Higginson J, Neptune R, Anderson F. Simulated parallel annealing within a neighborhood for optimization of biomechanical systems. *Journal of biomechanics*. 2005;38(9):1938–1942.
7. Kutch J, Valero-Cuevas F. Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS Computational Biology*. 2012;8(5):e1002434.

8. Steele KM, Tresch MC, Perreault EJ. The number and choice of muscles impact the results of muscle synergy analyses. *Frontiers in computational neuroscience*. 2013;7.
9. Bizzi E, Cheung VC. The neural origin of muscle synergies. *Frontiers in computational neuroscience*. 2013;7.
10. Tresch MC, Jarc A. The case for and against muscle synergies. *Current opinion in neurobiology*. 2009;19(6):601–607.
11. Dingwell JB, John J, Cusumano JP. Do humans optimally exploit redundancy to control step variability in walking? *PLoS computational biology*. 2010;6(7):e1000856.
12. Rácz K, Valero-Cuevas FJ. Spatio-temporal analysis reveals active control of both task-relevant and task-irrelevant variables. *Frontiers in computational neuroscience*. 2013;7.
13. Steele KM, Tresch MC, Perreault EJ. Consequences of biomechanically constrained tasks in the design and interpretation of synergy analyses. *Journal of neurophysiology*. 2015;113(7):2102–2113.
14. Alessandro C, Delis I, Nori F, Panzeri S, Berret B. Muscle synergies in neuroscience and robotics: from input-space to task-space perspectives. *Frontiers in computational neuroscience*. 2013;7.
15. Kording KP, Wolpert DM. Bayesian integration in sensorimotor learning. *Nature*. 2004;427(6971):244–247.
16. Kording KP. Bayesian statistics: relevant for the brain? *Current Opinion in Neurobiology*. 2014;25:130 – 133.
doi:<http://dx.doi.org/10.1016/j.conb.2014.01.003>.
17. Berniker M, O'Brien MK, Kording KP, Ahmed AA. An examination of the generalizability of motor costs. *PloS one*. 2013;8(1):e53759.
18. Sanger TD. Distributed control of uncertain systems using superpositions of linear operators. *Neural computation*. 2011;23(8):1911–1934.
19. Valero-Cuevas FJ. Fundamentals of Neuromechanics. vol. 8 of Biosystems and Biorobotics. SpringerVerlag London; 2015.
20. Bellman RE, Osborn HA. Dynamic programming and the variation of Green's functions. 1957;
21. Bellman RE. Adaptive control processes: a guided tour. Princeton university press; 2015.
22. Avis D, Fukuda K. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*. 1992;8(3):295–313.
23. Valero-Cuevas FJ, Hoffmann H, Kurse MU, Kutch JJ, Theodorou EA. Computational models for neuromuscular function. *IEEE Reviews in Biomedical Engineering* (2) October. 2009; p. 110ñ135.
24. Spoor C. Balancing a force on the fingertip of a two-dimensional finger model without intrinsic muscles. *Journal of Biomechanics*. 1983;16(7):497–504.

25. Kuo A, Zajac F. Human standing posture: multi-joint movement strategies based on biomechanical constraints. *Progress in Brain Research*. 1993;97:349–358.
26. Theodorou E, Valero-Cuevas FJ. Optimality in neuromuscular systems. In: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE; 2010. p. 4510–4516.
27. Scholz JP, Schöner G. The uncontrolled manifold concept: identifying control variables for a functional task. *Exp Brain Res*. 1999;126:289–306.
28. Valero-Cuevas FJ, Cohn BA, Yngvason HF, Lawrence EL. Exploring the high-dimensional structure of muscle redundancy via subject-specific and generic musculoskeletal models. *Journal of Biomechanics*. 2015 In Press;doi:10.1016/j.jbiomech.2015.04.026.
29. Sohn MH, McKay JL, Ting LH. Defining feasible bounds on muscle activation in a redundant biomechanical task: practical implications of redundancy. *Journal of biomechanics*. 2013;46(7):1363–1368.
30. Valero-Cuevas FJ, Zajac FE, Burgar CG. Large index-fingertip forces are produced by subject-independent patterns of muscle excitation. *Journal of Biomechanics*. 1998;31:693–703.
31. Venkadesan M, Valero-Cuevas FJ. Neural control of motion-to-force transitions with the fingertip. *J Neurosci*. 2008;28:1366–1373.
32. Dyer M, Frieze A, Kannan R. A Random Polynomial Time Algorithm for Approximating the Volume of Convex Bodies. Proc of the 21st annual ACM Symposium of Theory of Computing. 1989; p. 375–381.
33. Büeler B, Enge A, Fukuda K. Exact volume computation for polytopes: A practical study. *Polytopes: Combinatorics and Computation*. 2000;29:131–154.
34. Arnold EM, Ward SR, Lieber RL, Delp SL. A model of the lower limb for analysis of human movement. *Annals of biomedical engineering*. 2010;38(2):269–279.
35. Hamner SR, Seth A, Delp SL. Muscle contributions to propulsion and support during running. *Journal of biomechanics*. 2010;43(14):2709–2716.
36. De Sario V, Earl D, Green R, Saul K. Human Factors Simulation Using Demographically Tuned Biomechanical Models. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. vol. 58. SAGE Publications; 2014. p. 944–948.
37. Kutch JJ, Valero-Cuevas FJ. Muscle redundancy does not imply robustness to muscle dysfunction. *Journal of Biomechanics*. 2011;44(7):1264–1270.
38. Inouye JM, Kutch JJ, Valero-Cuevas FJ. Optimizing the topology of tendon-driven fingers: Rationale, predictions and implementation. In: The Human Hand as an Inspiration for Robot Hand Development. Springer; 2014. p. 247–266.
39. Smith RL. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*. 1984;32(6):1296–1308.
40. Lovász L. Hit-and-run mixes fast. *Mathematical Programming*. 1999;86(3):443–461.

41. Valero-Cuevas FJ. Predictive modulation of muscle coordination pattern magnitude scales fingertip force magnitude over the voluntary range. *J Neurophysiol.* 2000;83(3):1469–1479.
42. Valero-Cuevas FJ, Hoffmann H, Kurse MU, Kutch JJ, Theodorou EA. Computational models for neuromuscular function. *Biomedical Engineering, IEEE Reviews in.* 2009;2:110–135.
43. Chvatal V. Linear Programming. Klee V, editor. W.H. Freeman and company; 1980.
44. Bachynskyi M, Oulasvirta A, Palmas G, Weinkauf T. Biomechanical simulation in the analysis of aimed movements. In: CHI'13 Extended Abstracts on Human Factors in Computing Systems. ACM; 2013. p. 277–282.
45. Krekel PR, Valstar ER, De Groot J, Post FH, Nelissen RG, Botha CP. Visual Analysis of Multi-Joint Kinematic Data. In: Computer Graphics Forum. vol. 29. Wiley Online Library; 2010. p. 1123–1132.
46. Ball K. An elementary introduction to modern convex geometry. *Flavors of geometry.* 1997;31:1–58.
47. Valero-Cuevas FJ. Predictive modulation of muscle coordination pattern magnitude scales fingertip force magnitude over the voluntary range. *J Neurophysiol.* 2000;83:1469–1479.
48. Vempala S. Geometric Random Walks: A Survey. *Combinatorial and Computational Geometry, MSRI Publications.* 2005;52:573–612.
49. De Rugy A, Loeb GE, Carroll TJ. Muscle coordination is habitual rather than optimal. *The Journal of Neuroscience.* 2012;32(21):7384–7391.
50. Loeb G. Optimal isn not good enough. *Biological Cybernetics.* 2012;106(11-12):757–765.
51. Shadmehr R, Mussa-Ivaldi S. Biological learning and control: how the brain builds representations, predicts events, and makes decisions. Mit Press; 2012.
52. De Berg M, Van Kreveld M, Overmars M, Schwarzkopf OC. Computational geometry. Springer; 2008.
53. Chvatal V. Linear Programming. W.H. Freeman and Company; 1983.
54. D'Avella A, Giese M, Ivanenko ST YP, Flash T. Modularity in motor control: from muscle synergies to cognitive action representation. In: *Frontiers in Computational Neuroscience Research Topics.* Frontiers Media SA; 2012.
55. Clewley RH, Guckenheimer JM, Valero-Cuevas FJ. Estimating effective degrees of freedom in motor systems. *IEEE Trans Biomed Eng.* 2008;55:430–442.
56. Krishnamoorthy V, Goodman S, Zatsiorsky V, Latash ML. Muscle synergies during shifts of the center of pressure by standing persons: identification of muscle modes. *Biological cybernetics.* 2003;89(2):152–161.
57. Santello M, Flanders M, Soechting JF. Postural hand synergies for tool use. *The Journal of Neuroscience.* 1998;18(23):10105–10115.

58. Churchland MM, Cunningham JP, Kaufman MT, Foster JD, Nuyujukian P, Ryu SI, et al. Neural population dynamics during reaching. *Nature*. 2012;487(7405):51–56.
59. Sadtler PT, Quick KM, Golub MD, Chase SM, Ryu SI, Tyler-Kabara EC, et al. Neural constraints on learning. *Nature*. 2014;512(7515):423–426.
60. de Rugy A, Loeb GE, Carroll TJ. Are muscle synergies useful for neural control? *Frontiers in computational neuroscience*. 2013;7.
61. Keenan KG, Santos VJ, Venkadesan M, Valero-Cuevas FJ. Maximal voluntary fingertip force production is not limited by movement speed in combined motion and force tasks. *Journal of Neuroscience*. 2009;29:8784–8789.
62. Bongard J, Zykov V, Lipson H. Resilient machines through continuous self-modeling; 2006.
63. Rieffel J, Valero-Cuevas F, Lipson H. Morphological Communication: Exploiting Coupled Dynamics in a Complex Mechanical Structure to Achieve Locomotion. *Journal of the Royal Society Interface*. 2009;In Press.
64. Lovász L. Hit-and-Run Mixes Fast. *Math Prog*. 1998;86:443–461.
65. Marsaglia G. Choosing a Point from the Surface of a Sphere. *Ann Math Statist*. 1972;43:645–646.
66. E F Robertson TSB. *Basic Linear Algebra*. Springer; 2002.
67. Emiris IZ, Fisikopoulos V. Efficient Random-Walk Methods for Approximating Polytope Volume. *arXiv preprint arXiv:13122873*. 2013;.
68. Ge C, Ma F, Zhang J. A Fast and Practical Method to Estimate Volumes of Convex Polytopes. *Preprint: arXiv:14010120*. 2013;.
69. R Core Team. R: A Language and Environment for Statistical Computing; 2015. Available from: <http://www.R-project.org/>.

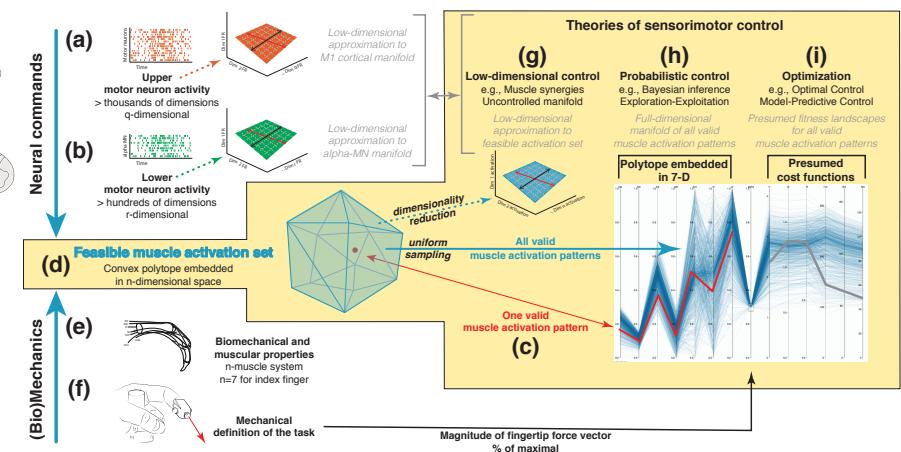


Fig 1. Feasible activation sets for a motor task. The descending command for a motor task is issued by upper motor neurons in primary motor cortex (a), which project onto the spinal cord via the cortico-spinal tract to drive *alpha*-motor neurons (b). The combined drive to all *alpha*-motor neurons of a muscle can be considered the level of activation to that muscle. If we consider that a motor command is sent to all relevant muscles simultaneously, then the resulting motor command is a multi-dimensional muscle activation pattern (i.e., a point) in a high-dimensional muscle activation space [1, 24, 25, 30] (c). For that muscle activation pattern to be a valid solution, it must satisfy the mechanical requirements of the task (f), given biomechanical capabilities of the limb (e). Given the large number of muscles in vertebrates, there is muscle redundancy in the sense that there is an infinite number of valid solutions. We now present what to our knowledge is the first detailed description of all valid solutions (the structure of the feasible activation set, (d)), which allows to us to compare, contrast and reconcile today's three dominant approaches to motor control (g, h, i).

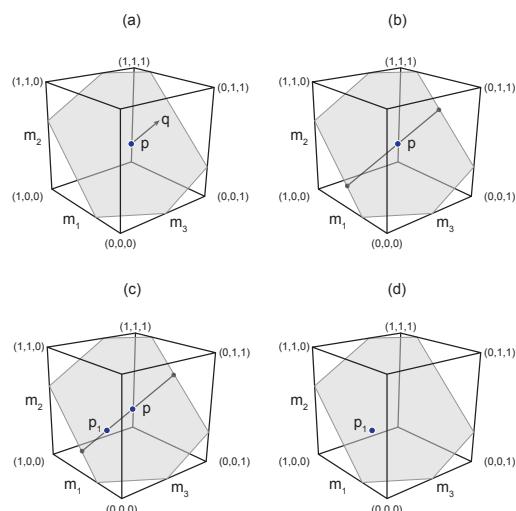


Fig 2. Graphical description of the Hit-and-Run algorithm.

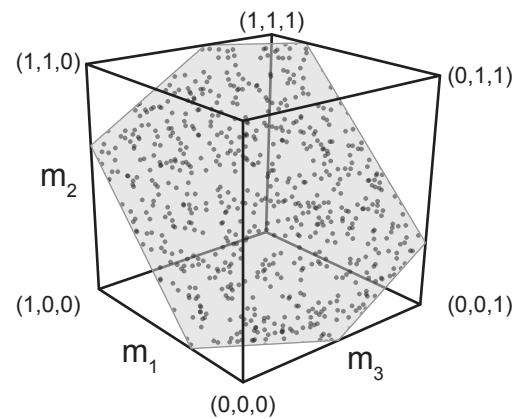


Fig 3. Points from the solution space.

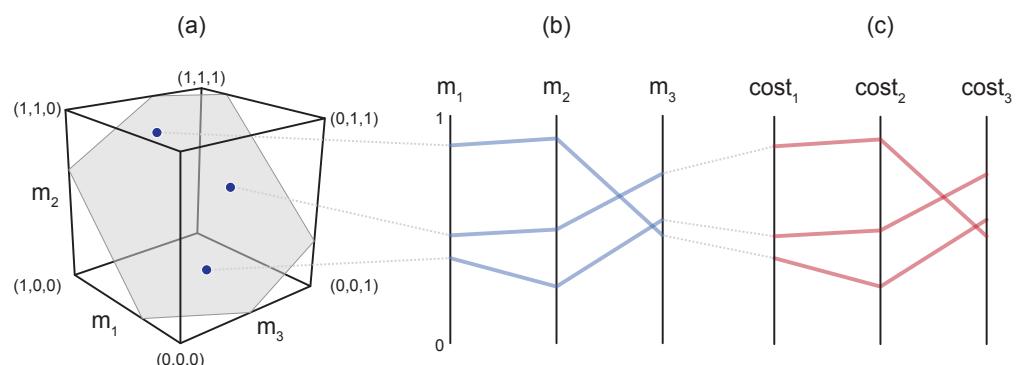
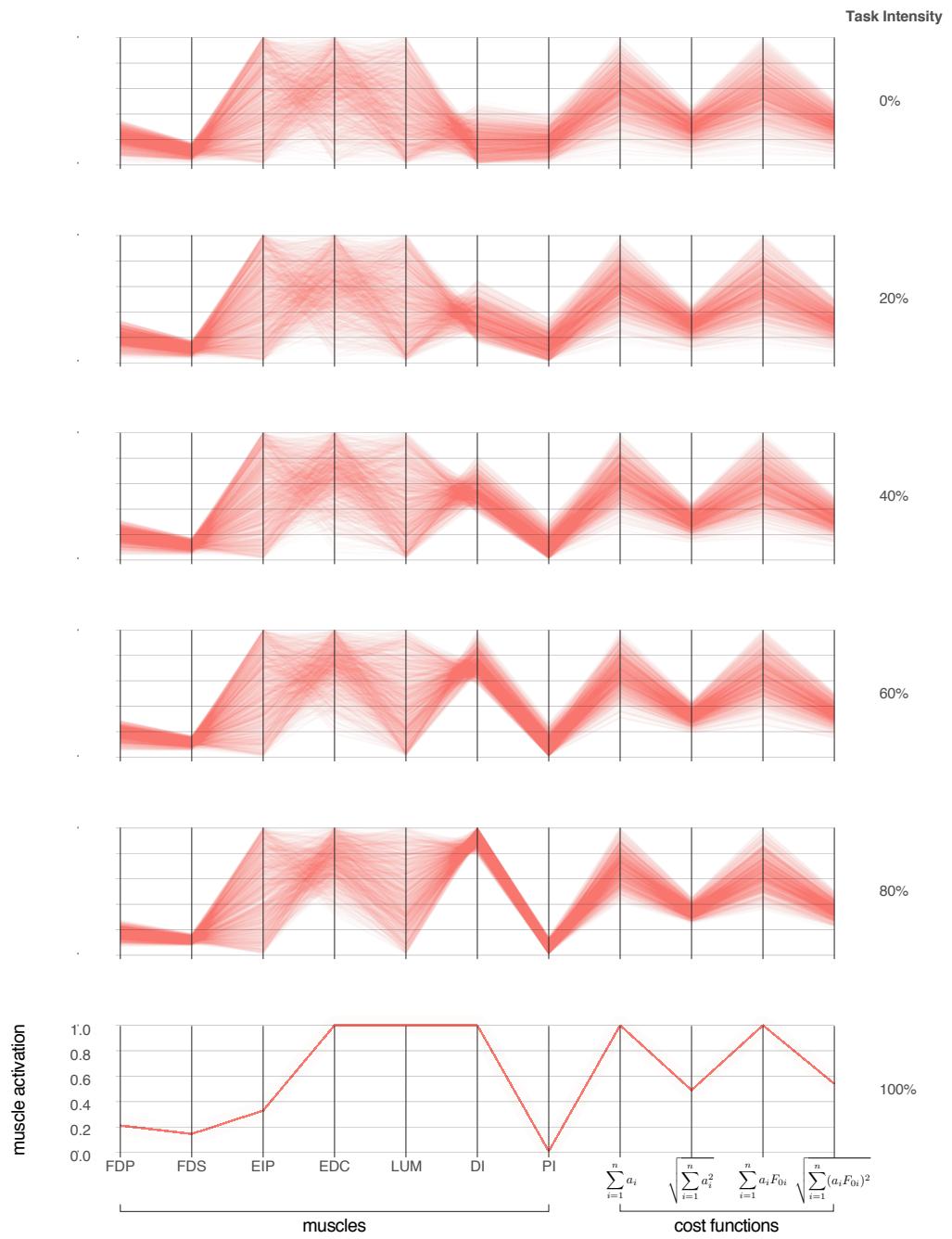


Fig 4. Parallel coordinates visualization of valid muscle activation patterns. (a) Consider three valid points (i.e., muscle activation patterns) belonging to the 2-D polytope of the feasible activation set embedded in 3-D. (b) Activation levels (i.e., coordinates) for each point sown in parallel coordinates format. (c) The associated cost of each muscle activation pattern as per three representative cost functions. The activation levels are bound between 0 and 1, and cost are normalized to the observed range of each cost.



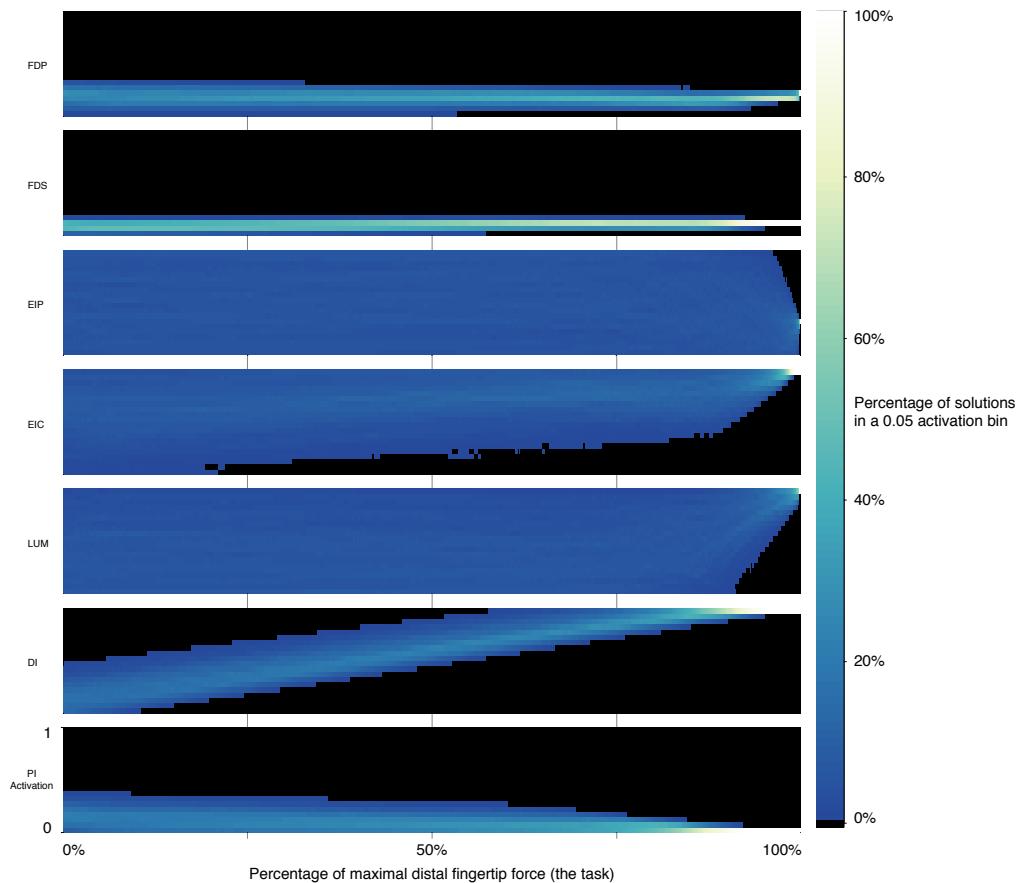


Fig 6. Density distributions converging as the output task approaches maximal distal force. Color indicates the percentage of solutions for a given task level that landed within a given 0.05-width activation bin, for each of the seven muscles (labeled on the left). The y axis indicates activation from 0 to 1 (full activation).

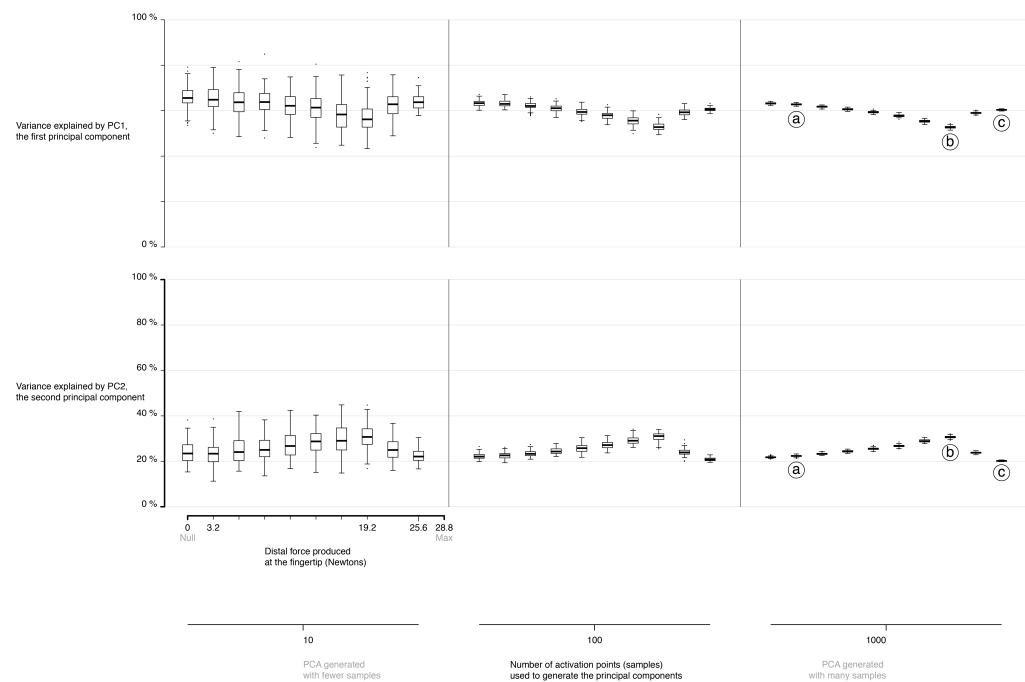


Fig 7. As more points are sampled from the activation space, the more clearly we can see a shift in the lower dimensional understanding of the space.

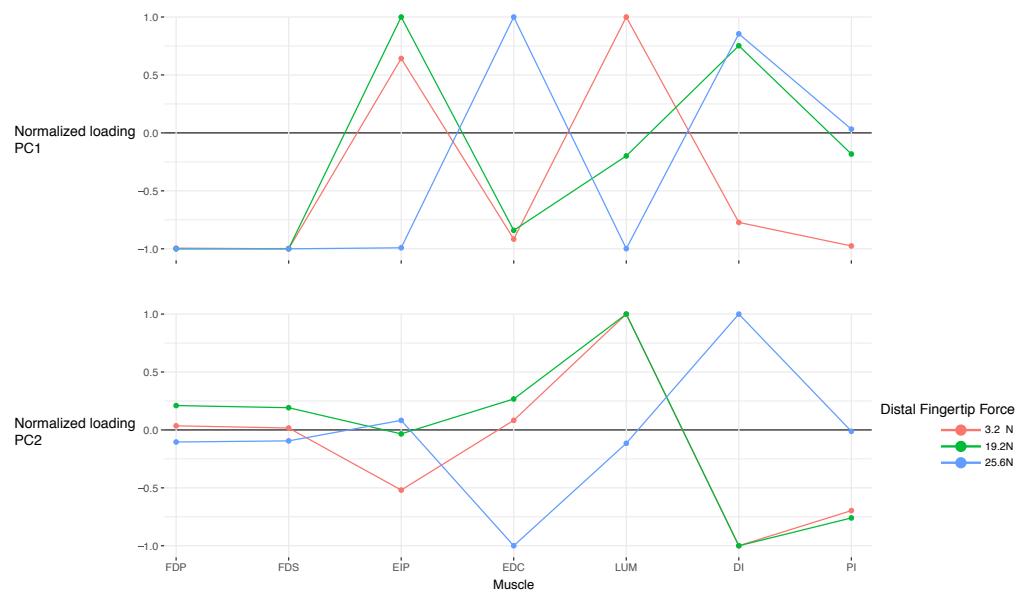


Fig 8. The loadings change as we increase force in the distal direction.

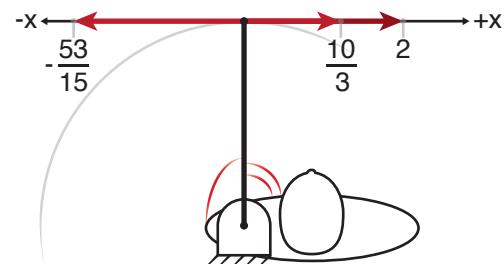


Fig 9. Supplemental Information Figure: One imagined visualization of the fabricated tendon driven system, with 3 generators.

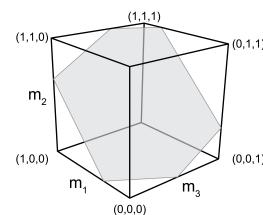


Fig 10. Supplemental Information Figure: The feasible activation set for a three-muscle system meeting one functional constraint is a polygon in \mathbb{R}^3 .