

A cost-agnostic neuromechanical approach to muscle coordination

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

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

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

❸These authors contributed equally to this work.

†These authors also contributed equally to this work.

✉Ronald Tutor Hall, RTH-421 3710 S. McClintock Ave Los Angeles, CA 90089-2905, USA

* valero@usc.edu

Abstract

The study of muscle coordination emphasizes numerical optimization in finding unique solutions among the many that can accomplish a given motor task. This approach has been justified by the lack of computational methods to characterize the high-dimensional structure of the set of feasible muscle activations. We present a cost-agnostic and computationally efficient use of the Hit-and-Run algorithm to characterize the feasible activation set that emerges naturally from the interactions among the feasible neural commands, the anatomy of the limb, and the constraints of static force production. A toy problem describes the method for a three-muscle system, which we then apply to the characterize the convex polytope embedded in 7D that defines the feasible activation set for a realistic model of static fingertip force production by a 7-muscle human index finger with 4 kinematic DOFs. We first describe the structure of that feasible activation set by the histograms of valid activations for each muscle. But we find that an interactive parallel coordinates system entirely describes the nature and relative size of families of feasible solutions, across a variety of cost functions—and for fingertip force magnitudes ranging from 10% to 100% of maximal. We demonstrate that a bounding box approach (i.e., only considering the extreme upper and lower bounds of a muscle's activation) singularly misconstrues the families of feasible activations. We also note that the modes of the histograms at low force magnitudes do not necessarily reflect the muscle coordination patterns needed to achieve higher, or maximal, force magnitudes. Although we present a 7D index finger example, our methods are computationally applicable to systems with at least 40 muscles. Lastly, we explore the consequences of the structure of feasible activation sets to a probabilistic approach to muscle coordination, which has important implications to biological plausibility, muscle dysfunction, and motor learning/adaptation for neuromechanically realistic limbs producing real-world tasks.

Author Summary

Tendons, bones, joints and forces define our ability to move ourselves and manipulate objects. How does the brain and spinal cord control our bodies so efficiently, and how does mechanical control evolve over time? Before we can ask these overarching questions, we have to consider the remarkable set of possibilities a controller can use to achieve the same force or movement outcome. The challenge here is adding the physical constraints on the infinite set of possibilities, and enabling visualization of where evolution, learning, and optimization occurs. To this end, we combined approaches from computational geometry and biokinesiology to model endpoint force in a static human finger with seven muscles, and we revealed the true mathematical structure of all feasible solutions. In one of the two visualizations we use to view these structures, one can interactively disable a muscle and see exactly how the central nervous system must change its behavior. Our method was designed to work with an arbitrarily high number of muscles and could be used to predict the underlying altered pathways of learning after an injury.

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 movement or force production 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 unit hypercube (n -cube) — where each muscle's maximal activation is normalized to 1 [1]. Thus researchers often seek to infer the optimization approach and the cost functions the nervous system utilizes to select effective points in activation space to produce natural behavior [2–7].

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

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 [1, 11–13]. We aim to explore what the solution space looks like, and uncover the structure of the feasible activation space for given static force tasks.

High dimensionality difficulties

Consider a model of a static fingertip force, with 7 muscles articulating the index finger's 4 degrees of freedom (DOF), which will be further described in Section . Assuming independent control of each muscle (non-synergistic model), each muscle has a unique force vector at the endpoint (i.e. the fingertip 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. In 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, and is called the *feasible activation set* (see Section). To date, the structure of this high-dimensional polytope has been inferred by computing its bounding box, which is determined by the maximum and minimum activation of each muscle for a given output force across all

solutions [8, 10, 11]. Although useful in defining the ultimate requirements of a muscle, the bounding box of a convex polytope excludes the details of the polytope's shape, thereby precluding comparison, since the polytope is a lower dimensional object embedded into $[0, 1]^n$. Empirical dimensionality-reduction methods have been used to calculate basis vectors for such subspaces [14–16], but this approach only provides a description of the dimension, orientation, and aspect ratio of the polytope; basis vectors are uninformative of boundaries and internal structure.

Here we present a novel application of the well-known Hit-and-Run algorithm [17] to describe the internal structure of these high-dimensional feasible activation sets (see Section). The input to the Hit-and-Run procedure is a static end-effector force, along with the system's endpoint Jacobian, maximal tendon forces, and a moment arm matrix [13].

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 [1].

With this, below are the key observations we identified with our research:

- The Hit-and-Run sampling of the solution space is computationally tractable.
- 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.

With respect to the structure of the activation space, we set forth the following key ideas and findings:

- When the unique solution at maximal force production is scaled for submaximal wrenches, we find that the solution space is not centered (by the mode) on those scaled vectors.
- Our approach provides a more granular context to the space within which the central nervous system (CNS) optimizes.
- The bounding box exceptionally misconstrues the actual shape of the feasible activation space.

and most importantly,

- The high-dimensional activation space, which serves as the landscape upon which all neuromuscular learning transpires, is a remarkably shallow space within which to optimize. We provide substantial theoretical and experimental evidence suggesting that the 'dynamics' of neuromechanical evolution and learning are heavily constrained and defined by the simple statics.

Materials and Methods

For a tendon-driven limb with n muscles, the feasible activation space is the unit n -cube (As mentioned in the Introduction). As explained in [13], when task constraints are

91
92
93
94
95

introduced to the system, the feasible activation set is further reduced; in this context, a task is a static force vector produced at the end effector, which is imposed as a constraint on the limb's options in activating its muscles. Thus if this limb meets all constraints, the feasible activation set is given by the polytope P containing all $\mathbf{a} \in \mathbb{R}^n$, that satisfy

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

where $\mathbf{f} \in \mathbb{R}^m$ is a fixed output force vector and $A = J^{-T}RF_0 \in \mathbb{R}^{m \times n}$ — 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 [1, 13]. The dimensionality of output force m is typically at most 6-dimensional (with force and torque, each in 3 dimensions), and tends to be much smaller than n . P is bounded by the unit n -cube since all variables a_i , $i \in [n]$ are in the interval $[0, 1]$. Each constraint of $\mathbf{f} = A\mathbf{a}$, is a hyperplane, which has dimension $n - 1$. If \mathbf{f} is a feasible submaximal output force, the feasible activation set is a $(n - m)$ -dimensional space embedded into the n -cube. Note that in our applications, it is safe to assume that no linear dependencies exist. Consider the following 1×3 fabricated example (illustrated in Fig. ??), where the task is a 1 Newton unidimensional force in x . The set of feasible activations is given by the shaded set in Fig. ???. Since there are three muscles and one constraint the output space is of dimension $(3 - 1) = 2$. Although linearized synergies would be conducive to implement into this model, we do not explore this here.

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

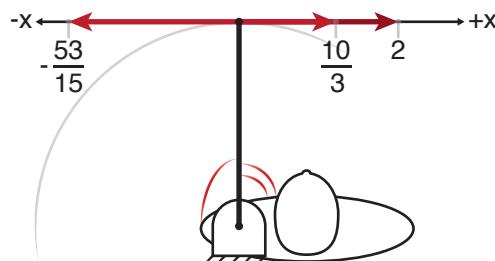


Figure 1. One imagined visualization of the fabricated tendon driven system, with 3 generators.

Difficulties of Volume Computation in Higher Dimension

96
97
98
99
100
101
102
103

We are interested in the volume and shape of P , but exact volume computations for polytopes is known to be $\#P$ -hard [18]. Several algorithms have been surveyed and implemented, but can only handle small dimensionality, 10 or slightly more [19]. Recent muscle system models we have used have been 31 dimensional [11], and other muscle models have over 40 muscles involved [9, 20–22], thereby limiting the feasibility of using direct volume computations. Instead, we chose to uniformly sample the continuous space, effectively approximating the shape of the polytope by calculating point densities.

Hit-and-Run algorithm

104
105
106

We chose to sample the activation space with the Hit-and-Run method that is known to converge to the uniform distribution across any convex body K [17]. It is a

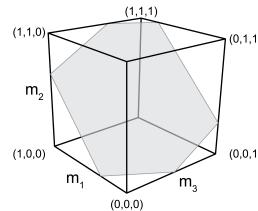


Figure 2. The feasible activation set for a three-muscle system meeting one functional constraint is a polygon in \mathbb{R}^3 .

generalization of a discrete Markov chain, which recursively samples a sequence of points in K as described below. After $\mathcal{O}^*(n^2R^2/r^2)$ steps, where r and R are the radii of the inscribed and circumscribed ball of K respectively [18, 23], the Hit-and-Run algorithm has sampled a point uniformly at random (u.a.r.) from the starting point in K . Unfortunately the hidden constant is large, which makes it practically infeasible to obtain the theoretical guarantee of a u.a.r. walk. Experimental results suggest that a number of points linear with respect to the dimension suffices (as discussed in Section). As the feasible activation space of the muscles are given by a convex polytope, this method can be directly applied for our problem. 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 [24].

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

1. Find a starting point \mathbf{p} of P .
2. Generate a random direction q from \mathbf{p} in P (u.a.r. over all directions) (Fig. 3a).
3. Find the two intersection points of the line given by the random direction q with the boundary of the polytope (Fig. 3b).
4. Choose a point u.a.r. on this line segment given by the intersection points (Fig. 3c).
5. Repeat from 2. the above steps with the new point as the starting point, for s iterations until the model is mixed.

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 one feasible activation vector in the space. For the Hit-and-Run algorithm to mix faster we don't want the starting point to be close to a vertex of the n -cube [23]. We use the following standard trick with slack variables ϵ_i , which for applications often gives a good starting point.

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

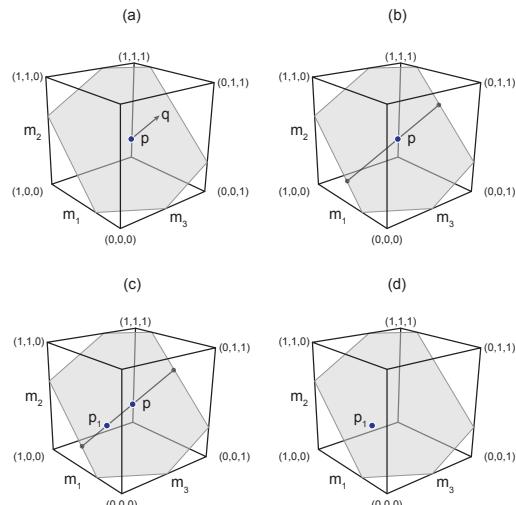


Figure 3. Graphical description of the Hit-and-Run algorithm.

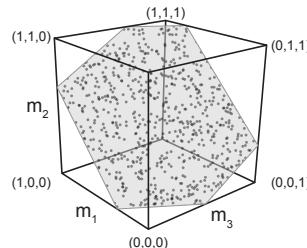


Figure 4. Uniform distribution across the feasible activation space. In the schematic arm example, the distribution is represented within a 2D plane.

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 $\mathbf{p} \in P$. 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

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

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

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

As shown by Marsaglia this can be done as follows [25].

1. Find an orthonormal basis $b_1, \dots, b_r \in \mathbb{R}^n$ of $\{\mathbf{q} \in \mathbb{R}^n \mid A\mathbf{q} = 0\}$.
2. Choose $(\lambda_1, \dots, \lambda_r) \in \mathcal{N}(0, 1)^r$ (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 $\{\mathbf{q} \in \mathbb{R}^n \mid A\mathbf{q} = 0\}$ and orthogonalize with the well known Gram-Schmidt method (for details see e.g. [26]). Note that in order to get the desired u.a.r. sample the basis needs to be orthonormal. 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 [27] by Emiris and Fisikopoulos suggests that 10($n + 1$) steps are enough to converge upon the uniform distribution [27], while in Ge et al.'s paper every single point of the Hit-and-Run algorithm is used in the sample [28].

Realistic index finger model

We used our published model in [1] 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 direction we simulated are visible in Fig. 5. In this model, for each input we ran 1,000,000 Hit-and-Run iterations and sampled every 100th point—thus setting mixing time to 100 iterations. To validate the Hit-and-Run algorithm, we computed the theoretical maximum and minimum activation for each muscle for the given force (Fig. 6 and 7); we found that the difference between the theoretical and observed bounds for all muscles was smaller than 0.001.

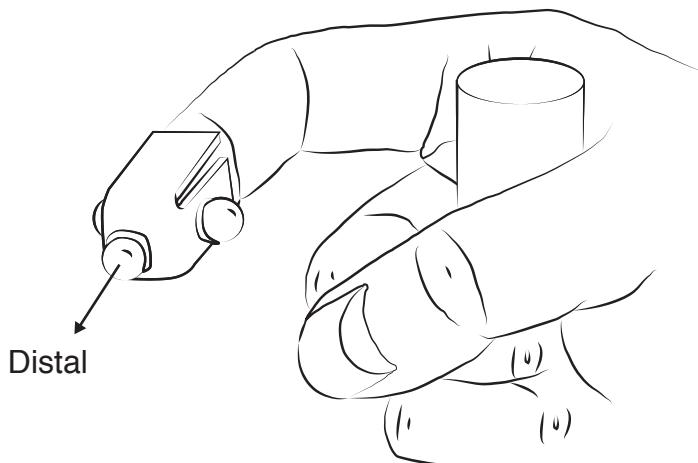


Figure 5. The index finger model simulated force production in the distal direction. Adapted from [1].

Solution projection histograms

Fig. 6 shows the activation distribution of each muscle, when activated with 50% of the maximal activation force. The diagram shows, for each muscle, the relative number of solutions that activate the muscle to some range in $[0, 1]$ (counted within each histogram break). We also show the observed activation upper and lower bounds for each muscle (vertical dotted lines). In Section we consider the distributions for different forces into distal direction.

Parallel coordinates visualization

A common way to visualize higher dimensional coordinate data is using parallel coordinates, and has been used in biomechanical studies [29, 30]. To show our sample set of points in the feasible activation space we draw n parallel lines for each of the n

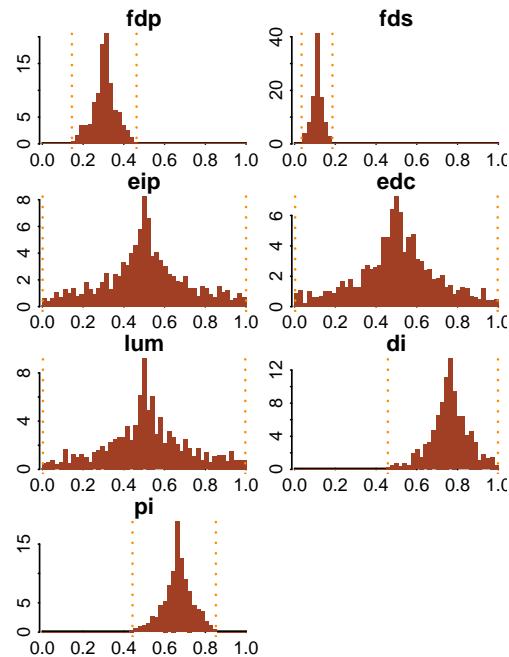


Figure 6. Distribution of feasible activations for $\alpha = 0.5$ (50% of the computed maximal force output in the distal direction). Dashed lines are the observed lower and upper bounds.

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 Fig. 8 and 9. We decided to simulate a 40% reduction in activation (feasible tendon force production) in the three index finger muscles innervated by the deep branch of the ulnar nerve- PI, DI, and FDP.

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 weighted norms.

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 the maximal musculotendon force, respectively.

We added six additional vertical lines, one for each cost function (Table 1), to the parallel coordinates plot to show the associated costs of each point. 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.

Results

Density projection upon one dimension

Using Hit-and-Run to sample feasible activation sets, Fig. 6 shows the distributions of activation solutions for a submaximal distal force resulting from 10,000 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 observed 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.

The density integrals perpendicular to each muscle are unimodal due to the convexity [31], and therefore it would be inadvisable to fit any bimodal distribution as the probability density function.

Activation spaces for increasing force

For maximal force into any direction, there is a unique activation vector satisfying all constraints. In Fig. 7 we give the distributions of the activations for increasing force, starting with $\alpha = 0.1$ (10% of the maximal force), and increasing α by 0.1 until maximal force is reached. This way, we can track the change in the distributions of a given muscle's activations by following its column from top to bottom.

The solution polytope converges as the difficulty of the task increases; 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 the distribution of all 10,000 points; when the distribution is compressed, the relative percentage of the bars will increase (as evident by the increasing y-axis limits), as we fixed break width (Δx) to remain constant to 2% of maximal activation [31].

The peaks seen in these figures is the perpendicular slice that has the largest relative volume; within the same muscle it does not have to be symmetric between the bounds, and can shift over differing tasks. As expected, the unique solution at $\alpha = 1.0$ appears as a single peak representing 100% of the sampled points; the bounds and the muscle's unique activation are superimposed.

The most simple finding is that the distributions cannot be inferred by their bounding boxes alone. 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. 7).

We find this holds not only for inter-muscle distribution comparisons, but intra-muscular 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 activation is

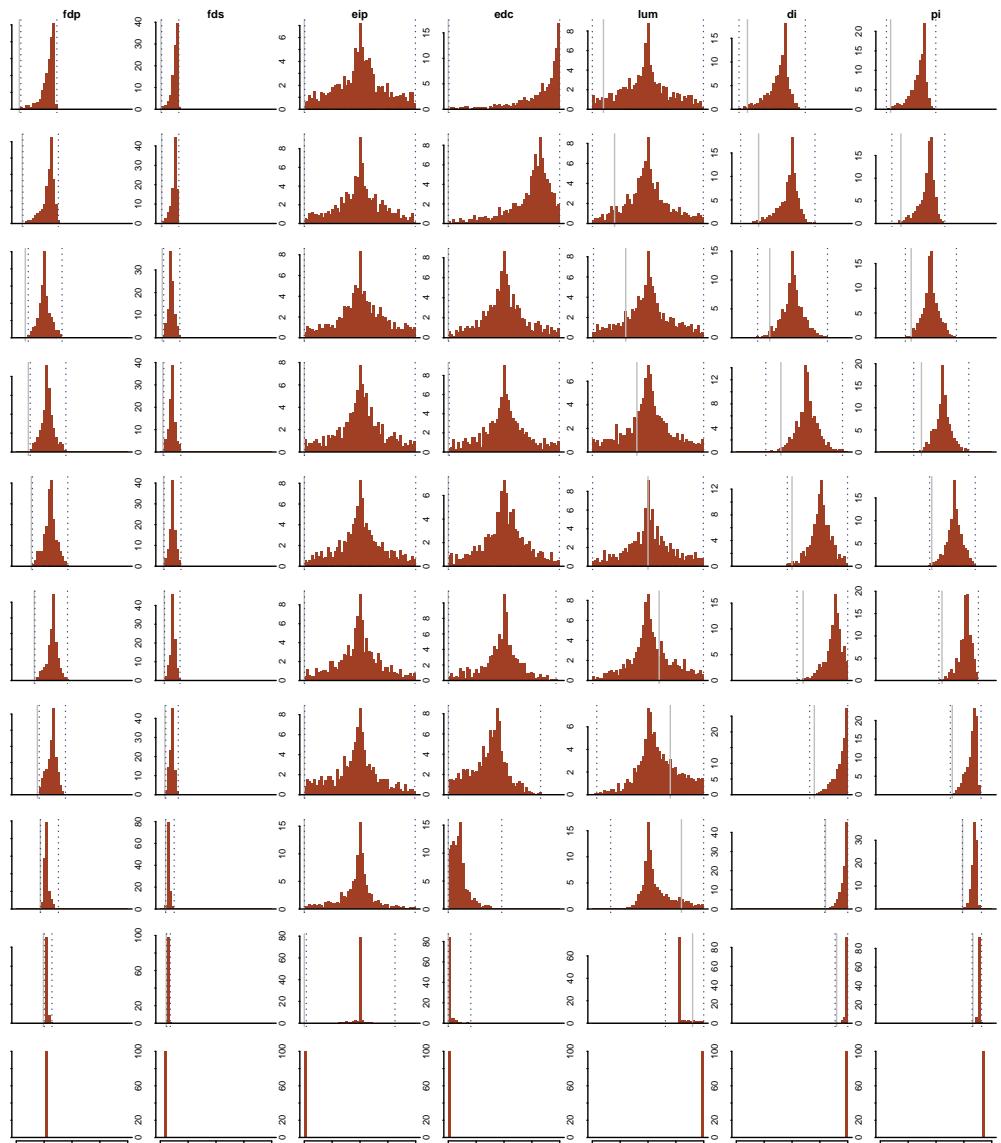


Figure 7. Distribution of activations in the distal direction and changing force. Each row of histograms uses a Hit-and-Run set. The height of each bar visualizes the percentage of 10,000 solutions found within a given 0.02 span of activation; the shape is more meaningful than the magnitude of the y -axis, as we expect convergence (and therefore peak increase within few bins) towards maximal contractions. We computed the upper and lower bounds of activation for each muscle (vertical dotted lines).

239
240
241
242
243
244
245
246
247
248
249

bounded between 0.1 and 0.81, approximately 79% of the solutions exist with EIP activation between just 0.49 and 0.51.

Next, we see that if one muscle had to be fixed throughout the entire force progression, DI and PI would fail; their bounding boxes of tasks below $\alpha = 0.4$ do not include the unique solution at $\alpha = 1.0$. We also placed a vertical grey line for the scaled unique solution at maximal force, denoted \mathbf{a}^* , (e.g. LUM converges to an activation of 1 at maximal distal force, so we put a grey line at 0.8 for $\alpha = 0.8$ of maximal distal force). Since $\alpha \mathbf{f}_{\max} = \alpha A \mathbf{a}^*$, $\alpha \mathbf{a}^*$ is a solution of the feasible activation set at α -fraction of the maximal force. However we observe that for some muscles, these points can lie arbitrarily in the distribution i.e. do not have to lie close to the corresponding peaks (e.g. muscle DI and EIP).

Parallel Coordinates Visualisation

Unlike the solution projections, which only show unidimensional information about a set of sampled solutions, we use parallel coordinate visualization to view how different parts of one muscle's distribution interact with the others. To maintain realtime interactivity of the plot, we used only the first 1000 points collected for each task, from $\alpha = 0.1$ to $\alpha = 1.0$.

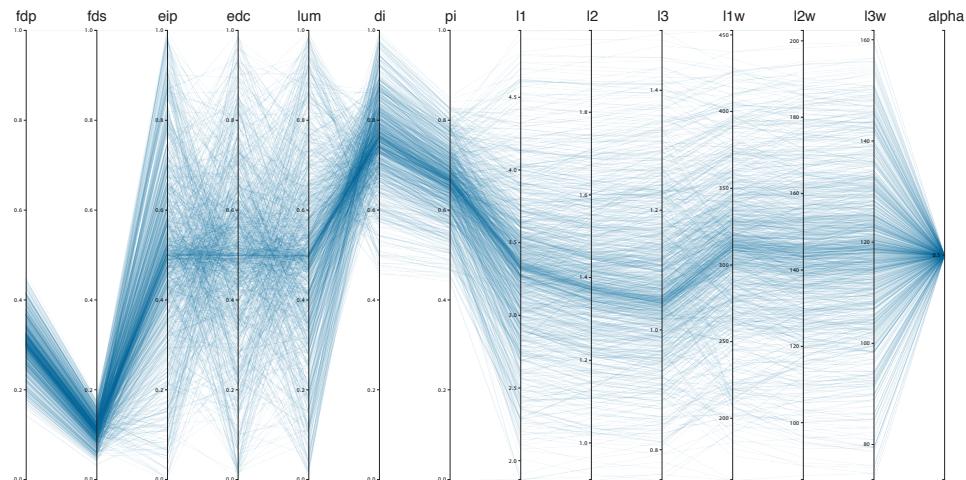


Figure 8. This figure is a snapshot of the interactive platform for visualizing all solutions. This parallel coordinates plot visualizes, where each feasible activation set is strewn across each dimension's axis as a line. While α could be set to view all tasks, we set α to a distal force of 50% of the computed maximal feasible force in this direction. Here we show 1000 points accrued from Hit-and-Run on a task of $\alpha = 0.5$.

On the histograms we can predict how many solutions would be lost if a constraint were imposed upon one muscle. Parallel coordinates allow us to visualize not only the amount of solutions lost, but the location of the remaining solutions across the other muscle dimensions. Furthermore one can constrain any number of muscles and still visualize the remaining solutions, whereas in the histograms one can not make any predictions.

250
251
252
253
254
255

256
257
258
259
260
261

262
263
264
265
266

Fig. 8 is a modified screen-shot of the platform, where we can constrain activation for a given muscle or cost function from above, below, or both. The resulting number of lines represents how the solution space changed across all dimensions recorded. Note how α can also be constrained- we can explore one or more levels of force output at a time.

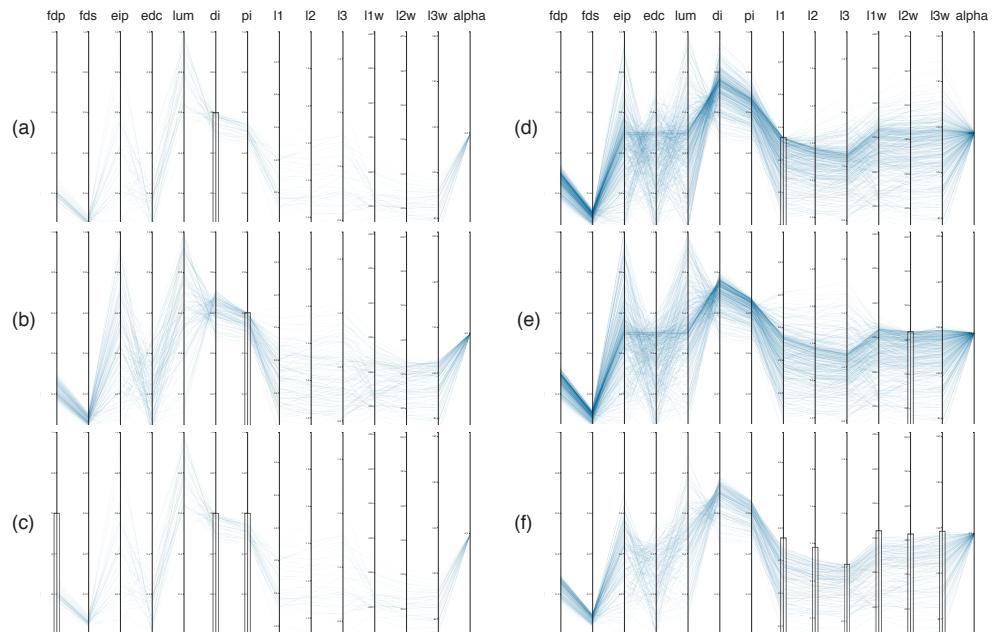


Figure 9. These snapshots show the use of the interactive parallel coordinate visualization of solutions across the activation space. For the task set to 50% of maximal in the distal direction, we show the (a) remaining 166 solutions when $PI < 60\%$ (b) remaining 57 solutions when $DI < 60\%$, and the (c) remaining 57 solutions when we constrain $PI < 60\%$ and $DI < 60\%$. We also show the (d) remaining 502 solutions when we select the lower 50% of l_1 costs, (e) remaining 498 solutions when we select the lower 50% of l_2^w costs, and the (f) remaining 220 solutions when we select the lower 50% across all cost functions in Table 1

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. 9 (b), (c). Constraining one muscles does have different effects on the other muscles as for example if eip 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. 10 and 11.

Discussion

Our approximations show accurate views of feasible activations in slices perpendicular to each axis, in both histogram and parallel coordinate visualizations, and are computationally tractable. Had we performed exact volume computations, we would

267
268
269
270
271
272
273
274
275
276

have had more accurate relative volumes; that said, the level of error generated through approximation is exceptionally small in comparison to error derived from measuring/predicting the musculoskeletal parameters to define the generators of A . Our code only solves one linear program to find the starting point, and the time-cost of each point thereafter is linear; therefore this method can be used for tendon driven models in very high-dimensional systems with at least 40 contributing muscles; the number of degrees of freedom and the relative strengths of the muscles (F_o) does not hinder the speed of Hit-and-Run.

If an end effector were completely unaffected by any activation of a given muscle (i.e. the muscle's linear endpoint force is the 0 vector), then we would see a uniform distribution across that muscle's activation. As such, muscles which are nearly uninvolved in the end effector's actions will form near-uniform distributions, as their involvement barely influences the activation space. In further studies one could put muscle activation constraints directly into the A ; that said, as long as there are enough remaining points after adding post-hoc constraints (i.e. the original dataset is large enough) there is no advantage to this. Importantly, if one muscle is fixed to an exact value of activation, the resulting polytope is reduced by 1 dimension; a fixed constraint must be added directly to A , prior to sampling.

Since l_1 and l_1^w are linear, one can also constrain A with these cost functions prior to Hit-and-Run, but our implementation does not support constraints based on functions of nonlinear degree (i. e. l_2 , l_3 , l_2^w and l_3^w). We note that the activation and metabolic classes of cost function are fundamentally different, and do not explore correlations between these two classes. We do, however, note that when all of the involved cost functions are 'minimized' to the bottom half of all solution costs, the union maintains a very high number of solutions (22%). With this we can note how all of these cost functions are similar in nature across the polytope (as one would expect).

In expanding upon the exciting research by Sohn et. al. we explored the space between the bounds of feasible activation [10]. While our research here looked to further constrain this space, the perspective of this space is fundamentally cost-agnostic; the central nervous system (CNS), especially in well-trained systems, likely explore only regions of the space which are more pragmatic in practice [5]. In comparison to bounding-box representations, our application of Hit-and-Run in this context represents a highly significant step forward in developing tools for meaningful visualization, value in extracting associations between solutions, and computational tractability, in addition to being veritable of the true solution distributions within the feasible activation set.

Our results provide evidence supporting the following:

- The Hit-and-Run algorithm can explore the feasible activation space for a realistic 7-muscle finger in a way that will remain computationally tractable in higher dimensions.
- 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 and the effects of muscle and cost constraints critically affect the space within which motor learning transpires.

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 (i.e. they require the similar force generation or torque production over the course of a movement), the activation patterns for similar actions must be similar as well. In this way, we can think of the solutions space as an effective model for exploration-exploitation, where the structure of the activation space contains high-dimensional Bayesian priors—these priors are narrowed/shifted over time to compensate for learning and skill-development, and must move within the space following significant changes in the CNS or musculoskeletal system.

Experiments into the ‘commonly-chosen’ coordination region over the course of a learned motor task could further elucidate how these task-irrelevant parameters are shifted and refined. It’s imperative to remember that the space is highly constrained by spatiotemporal demands; as a task changes slightly from moment to moment, applying the optimal coordination strategy for one task may only require slight modification to achieve the next. In this case, the region of activation space continues to be near-optimal in spite of changing circumstances. In a different situation, a near-optimal coordination strategy that achieves one task, may be furiously off-target for a similar task. With this, it’s important to consider how limbs optimize for a minimal total effort alongside a changing task [5], and what theoretical cost functions remain both mathematically and biologically reasonable across changing situations—as this may hold for any coordination strategy. While prior research has shown that submaximal force coordination is related to unique solutions at maximal force generation [32], the Hit-and-Run distributions offer a different view of the total set of solutions from which the CNS must select; along our march from $\alpha = 0.1$ to $\alpha = 1.0$, we observed no discernible clustering of solutions near the scaled unique solution. With this in mind, this does not preclude or disprove the use of scaling strategies.

Considering the limitations on muscle activation and deactivation speed, the set of feasible activation time-histories would be constrained to a relatively high metabolic cost for low forces on the way to maximal force production, especially for quick force ramp-ups.

This disconnect may let us infer that effective submaximal force generation requires a complex trajectory of muscle activations which have trajectory-dependent cost functions; while continuing to generate the intended wrench output, the system selects a time-history of activation for all muscles which minimize the cost of the entire movement, or discrete parts of the movement. Consider for example the effect of injury upon the selection of motor coordination patterns: if higher activation of a muscle induces pain, then the entire movement cost function must naturally incorporate some strategy of minimizing the levels of discomfort. Similarly, a stroke affecting an afferent motor neuron pool could drastically limit the feasible set of alpha-gamma feedback, thereby redefining the set of feasible coordination strategies, both in biomechanical movement, and its relevant muscle activation patterns. We understand that in the context of the entire closed loop, afferent activation is highly distant from neural decisionmaking; intention, alpha-gamma motor neuron excitation-inhibition, muscle excitation, activation dynamics and contractile properties represent the next logical additions to our set of constraints on the coordination space. Furthermore, the use of dynamical system modeling in addition to static force production could help us understand how changes in moment arms and muscle velocities affect the control situation for the CNS.

We look to ‘close the loop’ between nervous system commands and mechanical output, thereby uncovering how the CNS collaborates with Newtonian physics—with the intention of enlightening the community’s understanding of motor performance and

learning, it will require us to consider where we exist in activation space.

382

Supporting Information

383

1 APPENDIX

384

1.1 Finger model data

385

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

386

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

387

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

388

389

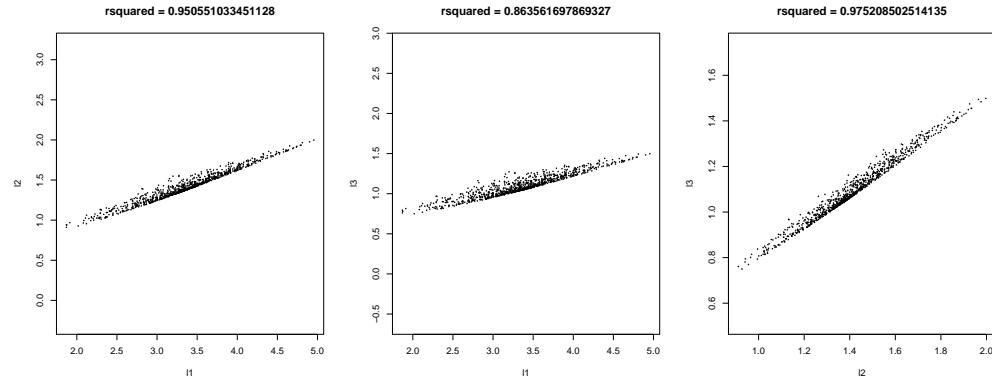


Figure 10. Non weighted cost functions

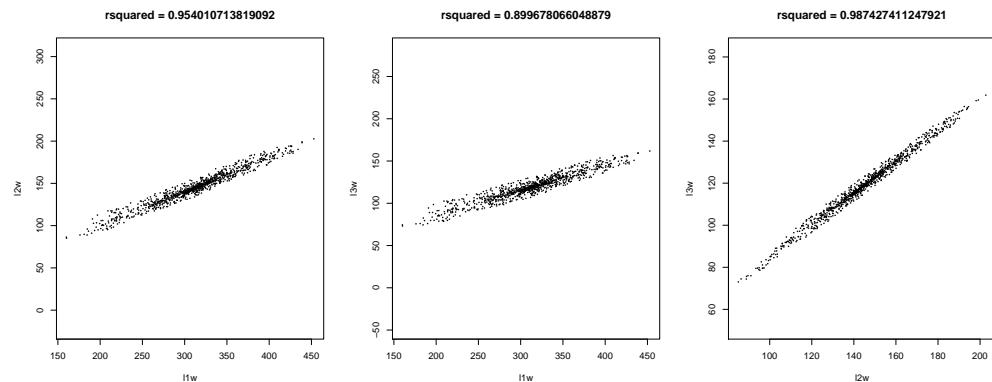


Figure 11. Weighted cost functions

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

390

391

```
# 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]
```

392

393

394

```
activations_of_interest <- c(0.49, 0.51) 395
ecdf(eip_solutions, activations_of_interest) 396
```

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 [33], 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

397
398
399
400
401

Acknowledgments

402
403
404

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

References

1. Valero-Cuevas FJ, Zajac FE, Burgar CG. Large index-fingertip forces are produced by subject-independent patterns of muscle excitation. *J Biomech.* 1998 Aug;31:693–703.
2. Chao EY, An KN. Graphical interpretation of the solution to the redundant problem in biomechanics. *Journal of Biomechanical Engineering.* 1978;100:159–67.
3. Prilutsky BI. Muscle coordination: the discussion continues. *Motor Control.* 2000;4(1):97–116. 0 1087-1640 Journal article.
4. Scott SH. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience.* 2004;5(7):532–546.
5. Todorov E, Jordan MI. Optimal feedback control as a theory of motor coordination. *Nature neuroscience.* 2002;5(11):1226–1235.
6. Crowninshield RD, Brand RA. A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics.* 1981;14(11):793–801.
7. 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.
8. Kutch JJ, Valero-Cuevas FJ. Muscle redundancy does not imply robustness to muscle dysfunction. *Journal of Biomechanics.* 2011;44(7):1264–1270.
9. Kutch JJ, Valero-Cuevas FJ. Challenges and new approaches to proving the existence of muscle synergies of neural origin. *PLoS computational biology.* 2012;8(5):e1002434.
10. 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.

¹<http://www.scala-lang.org/>

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

³<http://d3js.org/>

⁴<https://github.com/bcohn12/space>

11. J VCF, A CB, F YH, L LE. Exploring the high-dimensional structure of muscle redundancy via subject-specific and generic musculoskeletal models. *J Biomech.* 2015;In press.
12. 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.
13. Valero-Cuevas FJ. A mathematical approach to the mechanical capabilities of limbs and fingers. *Adv Exp Med Biol.* 2009;629:619–633.
14. Clewley RH, Guckenheimer JM, Valero-Cuevas FJ. Estimating effective degrees of freedom in motor systems. *IEEE Trans Biomed Eng.* 2008 Feb;55:430–442.
15. d'Avella A, Bizzi E. Shared and specific muscle synergies in natural motor behaviors. *Proceedings of the National Academy of Sciences of the United States of America.* 2005;102(8):3076–3081.
16. 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.
17. Smith RL. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research.* 1984;32(6):1296–1308.
18. 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.
19. Büeler B, Enge A, Fukuda K. Exact volume computation for polytopes: A practical study. *Polytopes: Combinatorics and Computation.* 2000;29:131–154.
20. 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.
21. Hamner SR, Seth A, Delp SL. Muscle contributions to propulsion and support during running. *Journal of biomechanics.* 2010;43(14):2709–2716.
22. 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.
23. Lovász L. Hit-and-Run Mixes Fast. *Math Prog.* 1998;86:443–461.
24. Vempala S. Geometric Random Walks: A Survey. *Combinatorial and Computational Geometry.* MSRI Publications. 2005;52:573–612.
25. Marsaglia G. Choosing a Point from the Surface of a Sphere. *Ann Math Statist.* 1972;43:645–646.
26. E F Robertson TSB. *Basic Linear Algebra.* Springer; 2002.
27. Emiris IZ, Fisikopoulos V. Efficient Random-Walk Methods for Approximating Polytope Volume. *arXiv preprint arXiv:13122873.* 2013;.
28. Ge C, Ma F, Zhang J. A Fast and Practical Method to Estimate Volumes of Convex Polytopes. *Preprint: arXiv:14010120.* 2013;.

29. 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.
30. 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.
31. Ball K. An elementary introduction to modern convex geometry. *Flavors of geometry*. 1997;31:1–58.
32. 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.
33. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2015. Available from: <http://www.R-project.org/>.