

Hit and Run and Stuff

² Brian Cohn May Szedlák Francisco Valero-Cuevas Bernd Gärtner
³ Komei Fukuda

May 3, 2015

Abstract

The brain must select its control strategies among an infinite set of possibilities, thereby solving an optimization problem. While this set is infinite and lies in high dimensions, it is bounded by kinematic, neuromuscular, and anatomical constraints, within which the brain must select optimal solutions. We use data from a human index finger with 7 muscles, 4DOF, and 4 output dimensions. For a given force vector at the endpoint, the feasible activation space is a 3D convex polytope, embedded in the 7D unit cube. It is known that explicitly computing the volume of this polytope can become too computationally complex in many instances. We generated random points in the feasible activation space using the Hit-and-Run method, which converged to the uniform distribution. After generating enough points, we computed the distribution of activation across each muscle, shedding light onto the structure of these solution spaces- rather than simply exploring their maximal and minimal values. We also visualize the change in these activation distributions as we march toward maximal feasible force production in a given direction. Using the parallel coordinates method, we visualize the connection between the muscle activations. Once can then explore the feasible activation space, while constraining certain muscles. Although this paper presents a 7 dimensional case of the index finger, our methods extend to systems with up to at least 40 muscles. We challenge the community to map the shapes distributions of each variable in the solution space, thereby providing important contextual information into optimization of motor cortical function in future research.

²⁷ **1 Author Summary**

²⁸ brian is a phd student
²⁹ may is a phd student
³⁰ bernt, komei, and francisco are professors

³¹ **2 Introduction**

³² The problem the brain solves is: Which set of muscle activations will proficiently achieve
³³ the desired kinetics?[citations] With a multi-link joint system, an end effector can pro-
³⁴ duce force and torque each in 3 dimensions, but the question of how muscle redundancy
³⁵ affects this output is still puzzling in neuroscience. In effect, the brain chooses an acti-
³⁶ vation ‘solution’, from a the set of all possible coordination strategies; we refer to this
³⁷ space as referred to as the Feasible Activation Space (FFS) [FVC citations].

³⁸
³⁹ Consider a model of a human index finger, with 7 muscles articulating 4 Degrees of
⁴⁰ Freedom (DOFs), resulting in 3 forces and one torque in the xY[??Fco?] plane. When
⁴¹ in a static position, if we define a task as a 4D wrench at the fingertip, we can constrain
⁴² the solution space. We aim to explore what the solution space looks like, and uncover
⁴³ the structure of the feasible activation space for a given static force task.

⁴⁴ We express the feasible activation set as follows. For a given force vector $f \in \mathbb{R}^m$,
⁴⁵ which are the activations that satisfy

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

⁴⁶ In our 7-dimensional example $m = 4$ and $n = 7$, typically n is much larger than m .
⁴⁷ The constraint $\mathbf{a} \in [0, 1]^n$ describes that the feasible activation space lies in the n -
⁴⁸ dimensional unit cube (also called the n -cube). Each row of the constraint $\mathbf{f} = A\mathbf{a}$ is a
⁴⁹ $n - 1$ dimensional hyperplane. Assuming that the rows in A are linearly independent
⁵⁰ (which is a safe assumption in the muscle system case), the intersection of all m equality
⁵¹ constraints constraints is a $(n - m)$ -dimensional hyperplane. Hence the feasible activation
⁵² set is the polytope given by the intersection of the n -cube and an $(n - m)$ -dimensional
⁵³ hyperplane. Note that this intersection is empty in the case where the force f can not
⁵⁴ be generated.

⁵⁵ for the previous section,

⁵⁶ MAYTODO: define what a hyperplane

⁵⁷ MAYTODO: define what a unit cube is in the context of muscles

⁵⁸

⁵⁹ A convex shape has been used to represent the set of all feasible forces the end ef-
⁶⁰ fector can generate, and this has been visualized in many different ways[cite JOB paper,
⁶¹ cite mckay and ting, cite josh paper].

⁶² Optimal control of a musculoskeletal system is intrinsically related to mechanical
⁶³ constraints of the system and the task. An endpoint’s end effector forces are highly
⁶⁴ dependent upon tendon force ranges, the leverage of each tendon insertion point across

65 each joint, and the planes of motion each degree of freedom (DOF), with these physical
66 relationships defining the capabilities of the system. In spite of the nonlinearity
67 of alpha-gamma neuromuscular drive, every system exists under limitations intrinsic to
68 physical mechanics, and as such, limbs have been modeled to behave under these constraints
69 with stunning realism [cite]. With increasingly accurate and faceted models, a
70 great body of research has been tasked with predicting kinetics, while being sensitive
71 to subtle changes in muscle activation [todorov's mujoco], skeletal weight distributions,
72 neural synergies, and spatiotemporal variables[Kornelius and FVC, Racz FVC]. While
73 many of these models highlight their accuracy, and attribute it to nonlinear dynamic
74 modeling, linear approximation has long-remained a viable way to interpret the actions
75 of physical limb systems, in the context of a well-understood mathematical framework.
76

77 As limbs exist under physical constraints, neuromuscular control must strategize
78 within the generic Newtonian laws of physics, in the realm of linear statics and dynamics.
79 While some would argue that linear approximation of a musculoskeletal system is a blunt
80 instrument in researching what is considered a 'non-linear' system, linear approximation
81 can offer a different view, in showing the bounds of feasible system behavior. Some
82 attention has been given to the constraints that physical systems impart on control
83 itself ['nice try' citations], with many placing emphasis on non-linear synergies between
84 motor units, for instance, between the *vastus lateralis* and *vastus medialis* muscles of
85 the leg[cite]. A breadth of modeling techniques have been applied to physical systems to
86 model and understand CNS control under the constraints of a given task, and many have
87 been able to visualize some of the limitations animals must abide by in optimization.

88 Optimal control theory must be implemented in a way such that it is computationally
89 tractable. Control systems of designed (robotic) and evolved (neurophysiologic) origins
90 can afford only a small measure of latency. Identifying how optimal control works within
91 the framework of constraints could bring rise to more efficient algorithms, and this
92 contextual understanding could introduce new ways to visualize how neuromuscular
93 systems learn to improve over training. In dynamic systems we have seen *do research*
94 *on this*[cite].

95 In a static system, every possible combination of independent muscle activations
96 exists within the unit-n-cube, where N is set to however many independently-controlled
97 muscles a system has. Prior work has highlighted the relationship between the feasible
98 force space and the set of all activation solutions.[cite papers in the last 10 years] In
99 effect, adding constraints on the FFS (e.g. requiring only force in a given plane) adds
100 constraints to the FAS

101 The effect of each muscle on each joint has been represented by the moment arm
102 matrix [citations], the relationship of each DOF on end-effector output directions . The
103 feasible force set (described in detail in [cite]) is an M-dimensional polytope containing
104 all possible force vectors an endpoint can output.

105 Functional performance is defined by the ability for a system to identify optimal
106 solutions in a set of suboptimal solutions.

107 We applied our approach to two separate musculoskeletal models: 1. A fabricated

108 schematic system, with three muscles articulating one DOF, and one dimension of output
109 force. 2. A literature model from [JOB1998/dissertation], with seven muscles articulat-
110 ing four DOFs, and 4 dimensions of output force.

111 We designed this schematic (but mathematically viable) linear system of constraints
112 to help readers understand the mechanics of hit-and-run mathematics. Our index-finger
113 model has too many dimensions to show how the process works, so we hope this will
114 help readers understand what is going on in n dimensions. We also used this model to
115 perform unit tests on our code in thoroughly validating our hit-and-run implementation.

116 Below are the key ideas and findings we present with this paper:

117

118 -We found that a hit-and-run algorithm to explore the solution space is computa-
119 tionally tractable. (Section X)

120 -Projecting feasible activation solutions upon each muscle dimension provides a more
121 granular context to the space within which the central nervous system must operate and
122 optimize. (Section X)

123 -Relatively few hit-and-run points were required to uncover the distributions. (Section
124 X)

125 -For some muscles, we found that the bounding box exceptionally misconstrues the ac-
126 tual shape of the feasible activation space. (Section X)

127 -The hit and run algorithm is cost-agnostic. Therefore, we can provide spatial context to
128 where 'optimal' solutions lie within the space; this approach can be used to view where
129 local optima exist. (Section X)

130 -As our approach is cost-agnostic, we can compare cost functions side-by-side and identify
131 the subspace when COST is a constraint. We designed an interactive parallel coordinates
132 plot for visualizing and manipulating constraints to the solution space, such as muscle
133 dysfunction, muscle hyperactivity, as well as constraining the upper and lower bounds
134 for six different cost functions. (Section X)

135

136 In comparison to traditional bounding-box representations, our application of hit-
137 and-run in this context is decisively superior in capability for meaningful visualization,
138 value in extracting associations between solutions, and computational tractability, in
139 addition to being veritable of the true solution distributions within the FAS. n

140 **3 Materials and Methods**

141 First, we first describe the stochastic method of hit-and-run, and illustrate how it func-
142 tions on our fabricated 3-muscle, 1-DOF system with a desired force output of 1N.

143 **3.1 Data and Samples**

144 We began with an index finger of a healthy human (male) right hand, which was taken
145 from [JOB 1998]. The hand size, finger lengths, and weight was []. Experimental forces
146 from []. The moment arm matrix, R , which contains the leverage of each tendon's

147 insertion points across each joint, was measured by doing [], [], and then []. The Jacobian,
 148 J , which represents the effect of rotation at each DOF on each component of endpoint
 149 wrench [cite jacobian usage], was measured with a [], by [], with precision of $\pm x$. The
 150 force-naught vector, F_0 , which contains the tendon force at maximal isometric contraction
 151 (MIC)[cite method used] for each muscle, was taken in the [same or different] posture,
 152 with a [] measuring device, with precision of $\pm x$.

153 3.2 Polytope representation of the feasible activation space

154 Exact volume calculations for polygons can only be done in reasonable time in up to
 155 10 dimensions [?, ?, ?]. We therefore use the so called Hit-and-Run approach, which
 156 samples a series of points in a given polygon. Given the points for a feasible activation
 157 space, this method gives us a deeper understanding of its underlying structure.

158 3.3 Hit-and-Run

159 In this section we introduce the Hit-and-Run algorithm used for uniform sampling in a
 160 convex body K , was introduced by Smith in 1984 [?]. The mixing time is known to be
 161 $\mathcal{O}^*(n^2R^2/r^2)$, where R and r are the radii of the inscribed and circumscribed ball of K
 162 respectively [?, ?]. I.e., after $\mathcal{O}^*(n^2R^2/r^2)$ steps of the Hit-and-Run algorithm we are at
 163 a uniformly at random point in the convex body. In the case of the muscles of a limb,
 164 we are interested in the polygon P that is given by the set of all possible activations
 165 $\mathbf{a} \in \mathbb{R}^n$ that satisfy

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

where $\mathbf{f} \in \mathbb{R}^m$ is a fixed force vector and $A = J^{-T}RF_m \in \mathbb{R}^{m \times n}$. P is bounded by the
 unit n -cube since all variables $a_i, i \in [n]$ are bounded by 0 and 1 from below, above
 respectively. Consider the following 1×3 fabricated example.

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

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

166 the set of feasible activations is given by the shaded set in Figure 1.

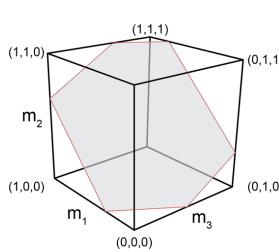


Figure 1: Feasible Activation

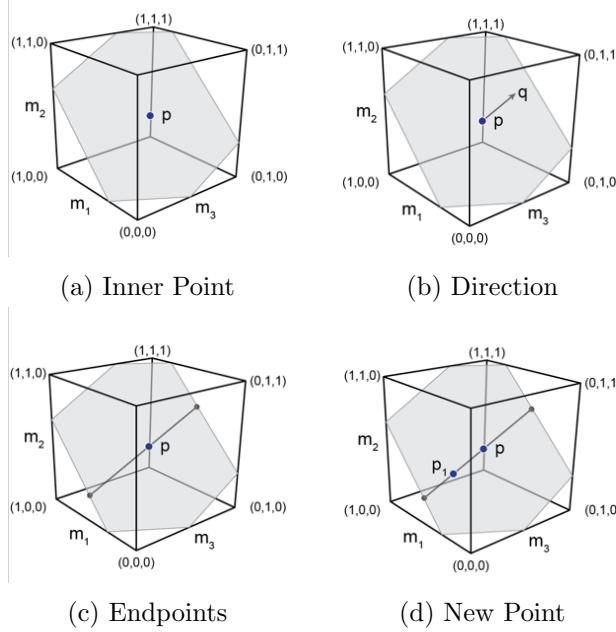


Figure 2: Hit-and-Run Step

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

- 169 1. Find a given starting point \mathbf{p} of P (Figure 2a) .
- 170 2. Generate a random direction through \mathbf{p} (uniformly at random over all directions)
171 (Figure 2b).
- 172 3. Find the intersection points of the random direction with the n -unit cube (Figure
173 2c).
- 174 4. Choose the next point of the sampling algorithm uniformly at random from the
175 segment of the line in P (Figure 2d).
- 176 5. Repeat from (b) the above steps with the new point as the starting point .

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

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

181 and hence

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

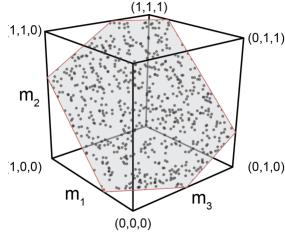


Figure 3: Uniform Distribution

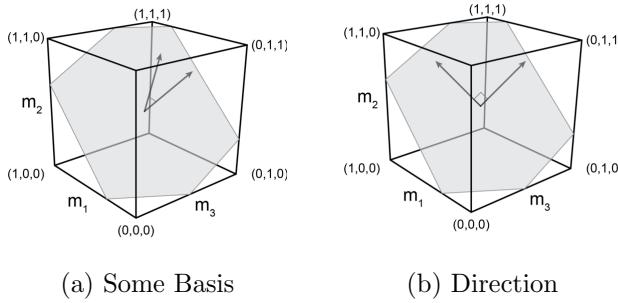


Figure 4: Find Orthonormal Basis

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

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

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

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

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

195 **3.4 Mixing Time**

196 How many steps are necessary to reach a uniformly at random point in the polytope?
 197 The theoretical bound $\mathcal{O}^*(n^2 R^2/r^2)$ given in [?] has a very large hidden coefficient (10^{30})
 198 which makes the algorithm almost infeasible in lower dimensions.

199 These bounds hold for general convex sets. For convex polygons in higher dimensions,
 200 experimental results suggest that $\mathcal{O}(n)$ steps of the Hit-and-Run algorithm are sufficient.
 201 In particular Emiris and Fisikopoulos paper suggest that $(10 + 10\frac{n}{r})n$ steps are enough
 202 to have a close to uniform distribution [?]. In all cases tested, sampling more point did
 203 not make accuracy significantly higher.

204 Ge et al. showed experimentally that up to about 40 dimensions, ??? random points
 205 seem to suffice to get a close to uniform distribution [?].

206 Therfore for given output force we execute the Hit-and-Run algorithm 1000 times
 207 on 100 points. The experimental results propose that those 1000 points are uniformly
 208 distributed on the polygon.

209 As a additional control, for each muscle we observe that the theoretical upper and
 210 lower bound of the feasible activation match the observed corresponding bounds (dif-
 211 ference max ??). To find the theoretical upperbound (lowerbound) of a given muscle
 212 activation we solve two linear programs maximizing (minimizing) a_i over the polytope.

213 **3.5 Starting Point**

214 To find a starting point in

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

215 we only need to find a feasible activation vector. For the hit and run algorithm to mix
 216 faster, we do not want the starting point to be in a vertex of the activation space. We
 217 use the following standard trick using slack variables ϵ_i .

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

218 This approach can still fail in theory, but this method has the choose $\epsilon_i > 0$ and
 219 therefore $a_i \neq 0$ or 1 . Since for all vertices of the feasible activation space lie on the
 220 boundary of the n -cube, at least $n - m$ muscles must have activation 0 or 1 . Documen-
 221 tation is included in our supplementary information, and all code is available at [Journal
 222 Link].

223 **3.6 Parallel Coordinates: Visualization of the Feasible Activation Space**

224 Citation A common way to visualize higher dimensional data is using parallel coordi-
 225 nates[citations]. To show our sample set of points in the feasible activation space we
 226 draw n parallel lines for each of the n muscles. With the axis labels of the line set

227 between 0 and 1, each point is then represented by connecting their coordinates by $n - 1$
 228 lines.

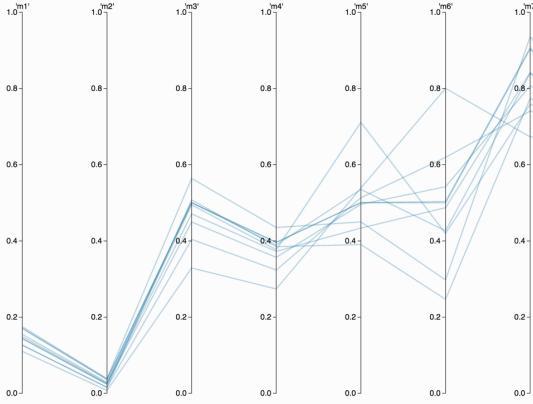


Figure 5: Feasible Activation

229 Using an interactive surface one can now restrict each muscle function to any desired
 230 interval, e.g., figure ??.

231 NICE FIGURE OF RESTRICTED PARALLEL COORDINATES

232 3.7 Integration of muscle-metabolic and neural drive cost functions

233 For every solution collected, we used popularly-used cost functions: we computed ac-
 234 tivation l_1 , l_2 and l_3 norms, and the tendon-force l_1 , l_2 and l_3 norms. Six additional
 235 vertical lines were added to the parallel coordinates plot to represent each cost function.
 236 With the same framework as developed with muscle activation coordinates, we can re-
 237 strict and subset solutions which fall into desired cost-function ranges, thereby masking
 238 sub-optimal solutions, and highlighting only those that meet the custom query's criteria.

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

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

239 For a given point $\mathbf{a} \in \mathbb{R}^n$ we are interested in the associated cost of every solution

240 collected through Hit and Run.

241 **NICE PICTURE WITH WEIGHTS INCLUDED**

242 **4 Results**

243 Many nice figures

244 1. Histograms

245 2. Histograms 3 directions

246 3. PC

247 **4.1 Activation Distribution on a Fixed Force Vector**

248 Feasible activations for palmar submaximal force production (at X) suggest very similar
249 solution distributions across muscles 3,4,5. Also, they are not bounded by the task in
250 activation; for each of these muscles, their activation can be maximal or zero, and the
251 model finger can still generate the force.

252 All of the muscles have a central peak between the bounds, suggesting that the
253 activation space is not skewed, but is rather a broad shape. The peak refers to the slice
254 of the activation space where the highest number of solutions exist. Said another way,
255 the integral of that slice represents the solutionspace volume.

256 **4.2 Changing Output Force in 3 Directions**

257 We discuss different forces into three different directions, which are given by the palmar
258 direction (x -direction), the distal direction (y -direction) and the sum of them. The
259 maximal forces into each direction are given by ??, ?? and ?? respectively. For $\alpha =$
260 $0.1, 0.2, \dots, 0.9$, we give the histograms where the force is $\alpha \cdot F_{\max}$, where F_{\max} is the
261 maximum output force in the corresponding direction.

262 **4.3 Parallel Coordinates**

263 The relationship between the histograms shown above, and the parallel coordinate inter-
264 active plots, is the measure of density across each muscle. Peak density areas of the
265 histogram will appear as dark blue sections of the muscle histograms, as those regions
266 are host to more solutions.

267 Although we computed XXX solutions for each level of alpha, we generate 1000
268 feasible solution points. Although using higher numbers of points works in the parallel
269 coordinates system, we elected to reduce sample size to provide realtime response rates
270 for interactive constraining of muscles and cost functions.

271 Typically, selecting solutions where one muscle is constrained to be lower activation
272 results in one or multiple compensatory increases in activation for the other muscles;
273 this is apparent when the bulk of the solutions moves up while constraining one muscle

274 down. When we see many solution lines in parallel across two muscles, it suggests a
275 consistent relationship between the two. Consistent slopes between the solutions suggest
276 an antagonist relationship. When the slope appears close to zero, the two muscles are
277 equal in effect in their antagonistic relationship; a negative slope suggests the first muscle
278 is less influential in task generation, while a positive slope suggests higher influence.

279 When two muscles have highly mixed slopes, it could suggest that these muscles
280 do not have a geometric interdependency, and could have orthogonal contributions to
281 output force.

When muscle1 was constrained to strictly equal to or lower than x percent of maximal contraction, that constraint reduced the number of solutions by X percent (Fig. fig_low). Along with this reduction, we saw a shift in the density peak for muscles X, X and X, while muscles X and

Muscle 5 and 6 same direction and same strength \Rightarrow Does not matter which one we activate for low cost

284 5 Discussion

285 Mostly to be written by Brian

286 5.1 Distributions

- 287 • Bounding box away from 0 and 1 means muscle is really needed \rightarrow Already known
288 from the bounding boxes
- 289 • High density \rightarrow most solutions in that area

290 5.2 Parallel Coordinates

- 291 • Parallel lines in PC indicate opposite direction of muscles
- 292 • Crossing lines indicate similar direction

293 MAYTODOS: (try to answer each one with 2-4 brief sentences, and include citations/notes and questions for me to answer.)
294 1. defend why we believe hit-and-run is a good way of viewing densities of distributions.
295 2. defend why we did the 'every hundredth' point instead of more points//
296 3. discuss other approaches to exploring multidimensional spaces//
297 4. discuss other ways that people have tried to visualize high dimensional spaces (projections onto n-1 dimensions,
298 etc)//
299 explore and discuss which variables could affect the hit-and-run distributions the most if they were stochastically permuted (parametrically) the Jacobian J^{-T} , Moment
300 arm matrix R , maximal tendon forces (F_o).//
301 5. briefly explain the bounds of how much the bounding box could feasibly be overestimating the volume of the solution-space polytope. e.g. when would the bounding box be a good approximation? when would the
302 bounding box be misrepresentative of the space?//
303 6. fill out the section below on issues with volume computations. Essentially, describe how volume computations have been
304 used in the past, and why we did not use them in our analysis this time. Discuss how,

307 if volume computations were quicker, we could have forseeably used volume computa-
308 tions. Mention how real-life systems are often way more than 7 dimensions (make a note
309 to cite my vectormap paper), and highlight how the math field feels about the level of
310 computational complexity that hit-and-run computations (ie. would other people agree
311 with us that hit and run is a good approach? if so, cite away!) Issues with volume
312 computations:

313 As realistic musculoskeletal systems has many more muscles, it's important for poly-
314 tope calculation to be scalable to higher dimensions.

315 **5.3 Running Time**

316 However, for each fixed force vector we only have to find a starting point and an or-
317 thonormal basis once, and are hence not of concern for the running time. Running
318 one loop of the hit and run algorithm only needs linear time, therefore the method will
319 extend to higher dimesions with only linear factor of additional running time needed.

320 TODO: add part on how we didn't deal with a dynamical system, but the feasible
321 force space still exists, just with constraints on muscle activations instantaneously, and
322 the momentums present in each of the limbs about their joints.

323 TODO: discuss how the FAS is an overestimate of the total number of possible
324 activation solutions, because of synergies. Our mathematical approach considers 100
325 percent independent control of every muscle, with no correlations or inverse correlations
326 between muscle activations. Then slightly discuss how synergies can limit activation
327 capabilities.

328 **6 Acknowledgments**

329 we thank people here

330 **References**

- 331 [1] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for ap-
332 proximating the volume of convex bodies. *Proc. of the 21st annual ACM Symposium*
333 *of Theory of Computing*, pages 375–381, 1989.
- 334 [2] M. Dyer and M. Frieze. On the complexity of computing the volume of a polyhedron.
335 *SIAM Journal on Computing*, 17:967–974, 1988.
- 336 [3] T. S. Blyth E. F. Robertson. *Basic Linear Algebra*. Springer, 2002.
- 337 [4] I.Z. Emiris and V. Fisikopoulos. Efficient random-walk methods for approximating
338 polytope volume. *Preprint: arXiv:1312.2873v2*, 2014.
- 339 [5] C. Ge, F. Ma, and J. Zhang. A fast and practical method to estimate volumes of
340 convex polytopes. *Preprint: arXiv:1401.0120*, 2013.

- 341 [6] L.G. Khachiyan. On the complexity of computing the volume of a polytope. *Izvestia*
342 *Akad. Nauk SSSR Tekhn. Kibernet*, 216217, 1988.
- 343 [7] L.G. Khachiyan. The problem of computing the volume of polytopes is np-hard.
344 *Uspekhi Mat. Nauk* 44, 179180, 1989.
- 345 [8] L. Lovász. Hit-and-run mixes fast. *Math. Prog.*, 86:443–461, 1998.
- 346 [9] G. Marsaglia. Choosing a point from the surface of a sphere. *Ann. Math. Statist.*,
347 43:645–646, 1972.
- 348 [10] R.L. Smith. Efficient monte-carlo procedures for generating points uniformly dis-
349 tributed over bounded regions. *Operations Res.*, 32:1296–1308, 1984.

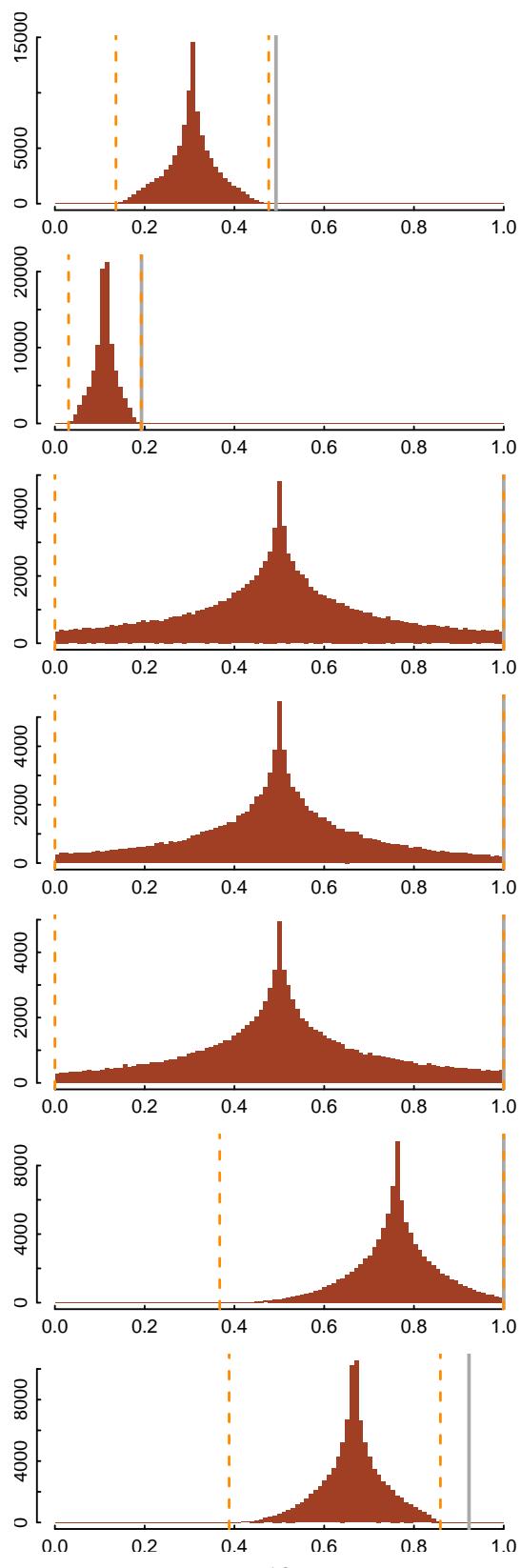


Figure 6: Histogram for Fixed Force

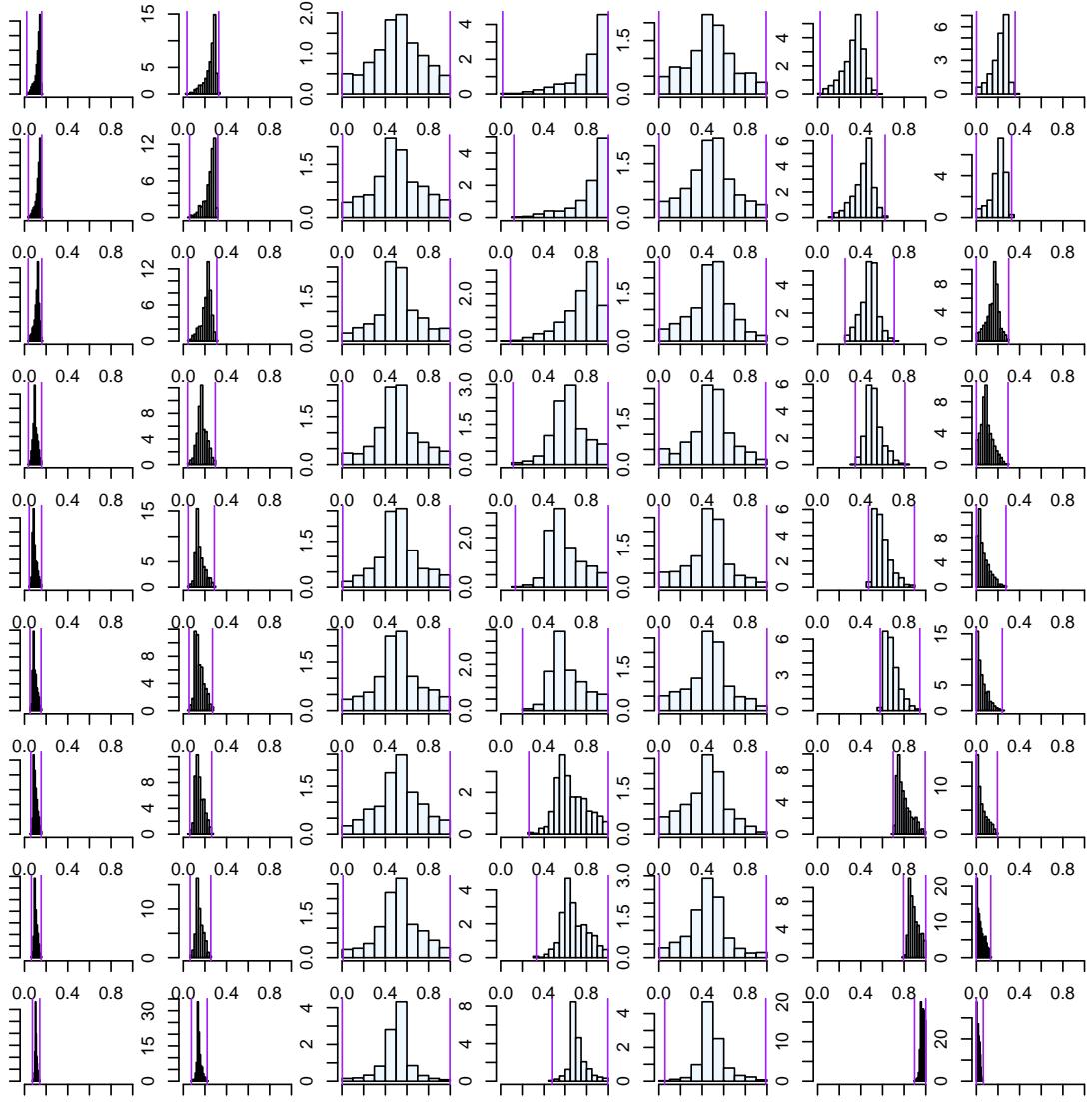


Figure 7: Histogram for x -Direction

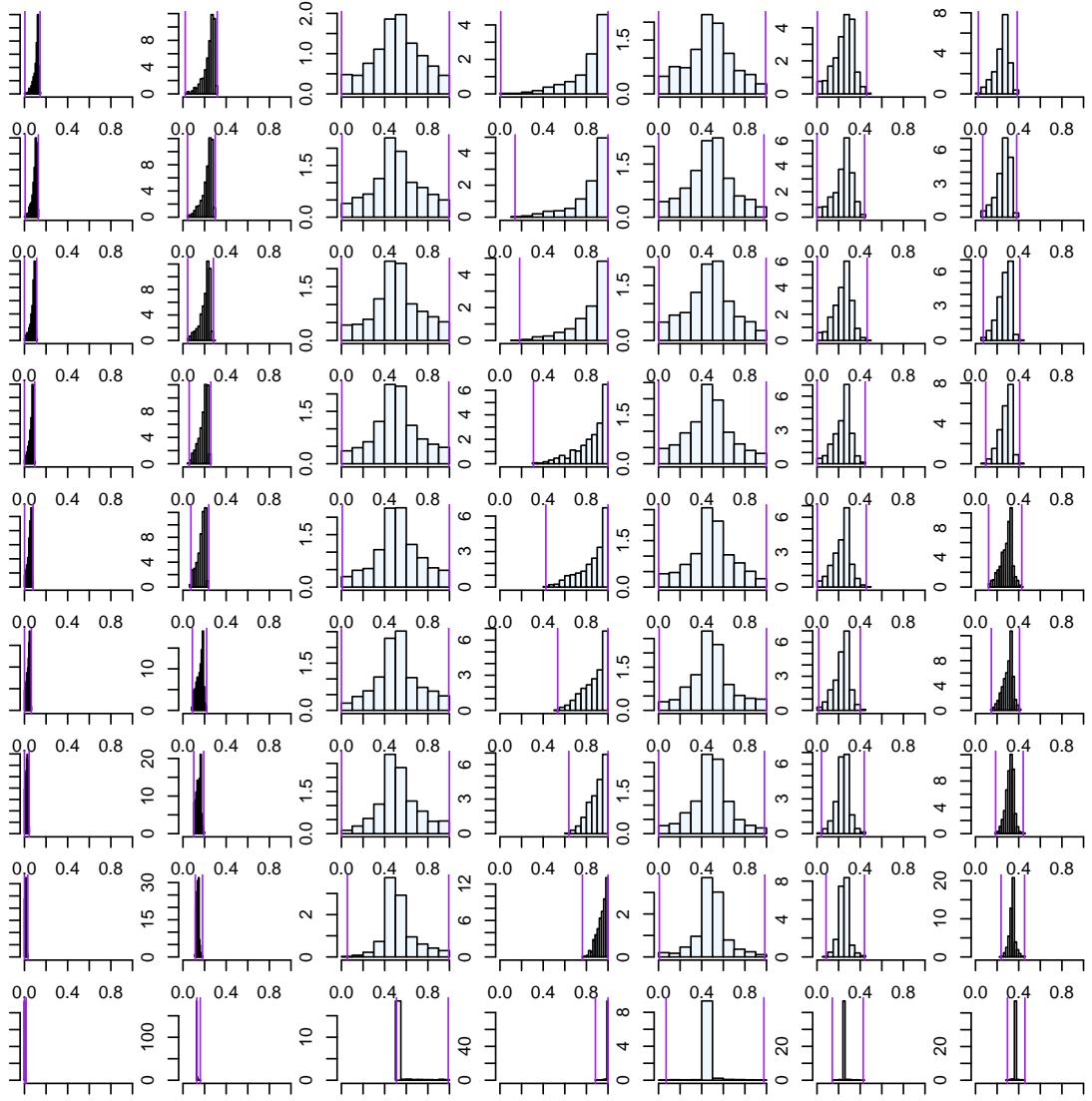


Figure 8: Histogram for xy -Direction

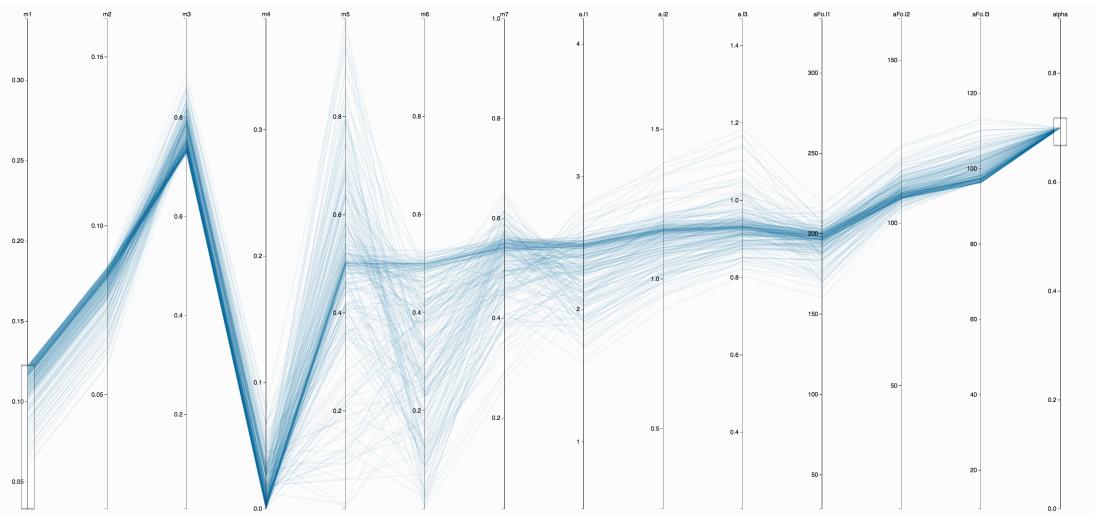


Figure 9: Low for Muscle 1

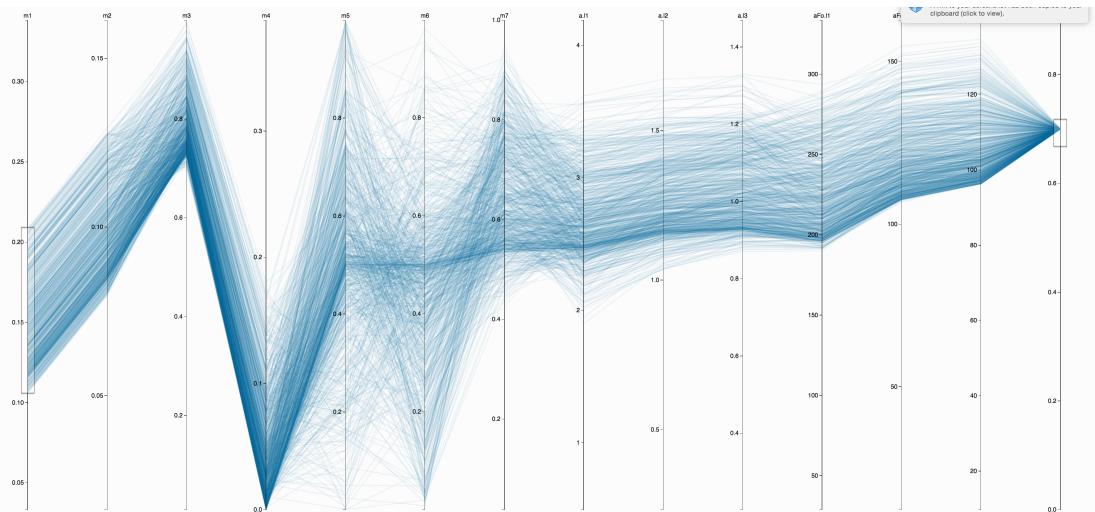


Figure 10: Middle for Muscle 1

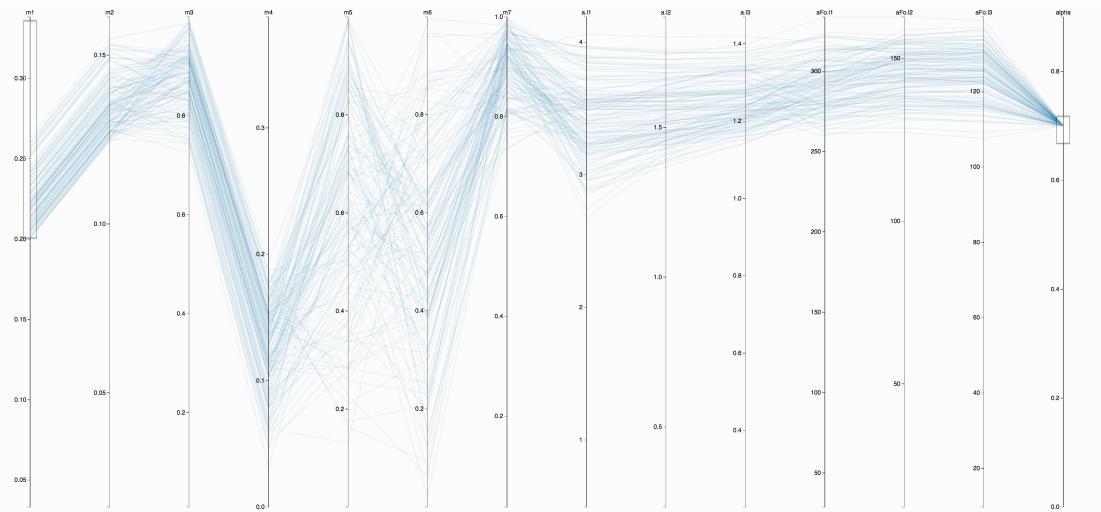


Figure 11: Upper for Muscle 1

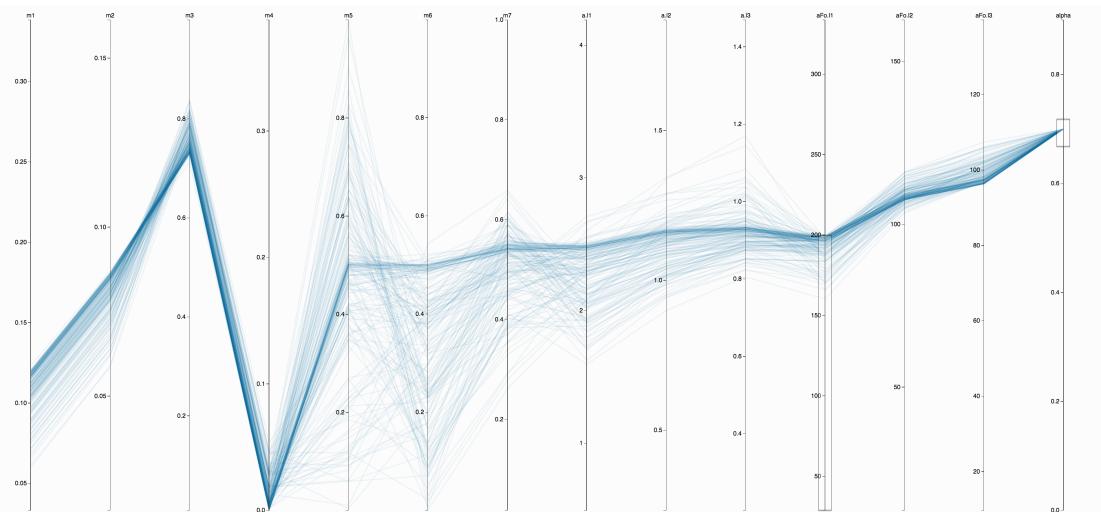


Figure 12: Weighted Cost