

Hit and Run and Stuff

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

March 29, 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 **2 Introduction**

²⁹ Optimal control of a musculoskeletal system is intrinsically related to mechanical con-
³⁰ straints. An endpoint's end effector forces are highly dependent upon tendon force
³¹ ranges, the leverage of each tendon insertion point across each joint, and the planes of
³² motion each degree of freedom (DOF), with these physical relationships defining the ca-
³³ pabilities of the system. In spite of the complexity of alpha-gamma neuromuscular drive
³⁴ models, every system exists under limitations intrinsic to physical mechanics, and as
³⁵ such, limbs have been modeled to behave under these constraints with stunning realism
³⁶ [cite]. With increasingly accurate and faceted models, a great body of research has been
³⁷ tasked with predicting kinetics, while being sensitive to subtle changes in muscle activa-
³⁸ tion [todorov's mujoco], skeletal weight distributions, neural synergies, and spatiotem-
³⁹ poral variables[Kornelius and FVC, Racz FVC]. While many of these models highlight
⁴⁰ their accuracy , and attribute it to nonlinear dynamic modeling, linear approximation
⁴¹ has long-remained a viable way to interpret the actions of physical limb systems, in the
⁴² context of a well-understood mathematical framework. As limbs exist under physical
⁴³ constraints, neuromuscular control must strategize within the generic Newtonian laws
⁴⁴ of physics, in the realm of linear statics and dynamics. While some would argue that
⁴⁵ linear approximation of a musculoskeletal system is a blunt instrument in researching
⁴⁶ what is considered a 'non-linear' system, linear approximation can offer a 'big picture
⁴⁷ view' of the system. Some attention has been given to the constraints that physical
⁴⁸ systems impart on control itself ['nice try' citations], with many placing emphasis on
⁴⁹ non-linear synergies between motor units, for instance, between the *vastus lateralis* and
⁵⁰ *vastus medialis* muscles of the leg. A breadth of modeling techniques have been applied
⁵¹ to physical systems to model and understand CNS control under the constants of a
⁵² given task, and many have been able to visualize some of the limitations animals must
⁵³ abide by in optimization.

⁵⁴ Optimal control theory must be implemented in a way such that it is computationally
⁵⁵ tractable. Control systems of designed (robotic) and evolved (neurophysiologic) origins
⁵⁶ can afford only a small measure of latency. Identifying how optimal control works within
⁵⁷ the framework of constraints could bring rise to more efficient algorithms, and this
⁵⁸ contextual understanding could introduce new ways to visualize how neuromuscular
⁵⁹ systems learn to improve over training. In dynamic systems we have seen jdo research
⁶⁰ on this;[cite].

⁶¹ In a static system, every possible combination of independent muscle activations
⁶² exists within the unit-n-cube, where N is set to however many independently-controlled
⁶³ muscles a system has. Prior work has highlighted the relationship between the feasible
⁶⁴ force space and the set of all activation solutions.[cite papers in the last 10 years] In
⁶⁵ effect, adding constraints on the FFS (e.g. requiring only force in a given plane) adds
⁶⁶ constraints to the FAS

⁶⁷ The effect of each muscle on each joint has been represented by the moment arm

68 matrix [citations], the relationship of each DOF on end-effector output directions . The
69 feasible force set (described in detail in [cite]) is an M-dimensional polytope containing
70 all possible force vectors an endpoint can output.

71 neurons do alot of stuff, and much work has been put into understanding how neural
72 drive results in force, motion, and kinetics. physical description of a musculoskeletal
73 system

74 Functional performance is defined by the ability for a system to identify optimal
75 solutions in a set of suboptimal solutions. {talk about local and global maxima and
76 minima in neuro optimization control theory}

77 The feasible force set represents every possible output force an end effector can impart
78 on an endpoint.

79 Described in a mathematical way the feasible activation set is expressed as follows.
80 For a given force vector $f \in \mathbb{R}^m$, which are the activations that satisfy

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

81 In our 7-dimensional example $m = 4$ and $n = 7$, typically n is much larger than m .
82 The constraint $\mathbf{a} \in [0, 1]^n$ describes that the feasible activation space lies in the n -
83 dimensional unit cube (also called the n -cube). Each row of the constraint $\mathbf{f} = A\mathbf{a}$ is a
84 $n - 1$ dimensional hyperplane. Assuming that the rows in A are linearly independent
85 (which is a safe assumption in the muscle system case), the intersection of all m equality
86 constraints constraints is a $(n-m)$ -dimensional hyperplane. Hence the feasible activation
87 set is the polytope given by the intersection of the n -cube and an $(n - m)$ -dimensional
88 hyperplane. Note that this intersection is empty in the case where the force f can not
89 be generated.

90 Issues with volume computations: As realistic musculoskeletal systems has many
91 more muscles, it's important for polytope calculation to be scalable to higher dimensions.

92 We first describe the stochastic method of hit-and-run, and illustrate its use on a
93 fabricated 3-muscle, 1-DOF system with a desired force output of 1N. We designed
94 this schematic (but mathematically viable) linear system of constraints to help readers
95 understand the mechanics of hit-and-run mathematics. Our index-finger model has too
96 many dimensions to show how the process works, so we hope this will help readers
97 understand what is going on in n dimensions (7 in the case of the index-finger model).
98 We also used this model to perform unit tests on our code in thoroughly validating our
99 hit-and-run implementation.

100 We investigated the distributions of the feasible activation set across each muscle.
101 State the purpose of the work in the form of the hypothesis, question, or problem you
102 investigated; and, Briefly explain your rationale and approach and, whenever possible,
103 the possible outcomes your study can reveal.

104 **3 Materials and Methods**

105 **3.1 Data and Samples**

106 We began with an index finger of a healthy human (male) right hand, which was taken
107 from []. The hand size, finger lengths, and weight was []. Dissected by []. Experimental
108 forces from []. All procedures were performed under protocol of the [] university stan-
109 dards, and the study was approved by []. The moment arm matrix, R , which contains
110 the leverage of each tendon's insertion points across each joint, was measured by doing
111 [], [], and then []. The Jacobian, J , which represents the effect of rotation at each DOF
112 on each component of endpoint wrench [cite jacobian usage], was measured with a [], by
113 [], with precision of $\pm x$. The force-naught vector, F_0 , which contains the tendon force
114 at maximal isometric contraction (MIC)[cite method used] for each muscle, was taken in
115 the [same or different] posture, with a [] measuring device, with precision of $\pm x$.

116 **3.2 Polytope representation of the feasible activation space**

117 Exact volume calculations for polygons can only be done in reasonable time in up to
118 10 dimensions [2, 6, 7]. We therefore use the so called Hit-and-Run approach, which
119 samples a series of points in a given polygon. Given the points for a feasibale activation
120 space, this method gives us a deeper understanding of its underlying structure.

121 **3.3 Hit-and-Run**

122 In this section we introduce the Hit-and-Run algorithm used for uniform sampling in a
123 convex body K , was introduced by Smith in 1984 [10]. The mixing time is known to be
124 $\mathcal{O}^*(n^2R^2/r^2)$, where R and r are the radii of the inscribed and circumscribed ball of K
125 respectively [1, 8]. I.e., after $\mathcal{O}^*(n^2R^2/r^2)$ steps of the Hit-and-Run algorithm we are at
126 a uniformly at random point in the convex body. In the case of the muscles of a limb,
127 we are interested in the polygon P that is given by the set of all possible activations
128 $\mathbf{a} \in \mathbb{R}^n$ 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.

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

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

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

- 132 1. Find a given starting point \mathbf{p} of P (Figure 2a) .

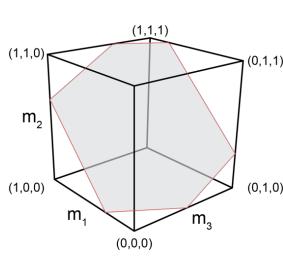


Figure 1: Feasible Activation

- 133 2. Generate a random direction through \mathbf{p} (uniformly at random over all directions)
- 134 (Figure 2b).
- 135 3. Find the intersection points of the random direction with the n -unit cube (Figure
- 136 2c).
- 137 4. Choose the next point of the sampling algorithm uniformly at random from the
- 138 segment of the line in P (Figure 2d).
- 139 5. Repeat from (b) the above steps with the new point as the starting point .

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

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

144 and hence

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

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

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

147 As shown by Marsaglia this can be done as follows [9].

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

151 A basis of a vectorspace V is a minimal set of vectors that generate V , and it is
 152 orthonormal if the vectors are pairwise orthogonal (perpendicular) and have unit length.
 153 Using basic linear algebra one can find a basis for $V = \{A\mathbf{q} = 0\}$ and orthogonalize it

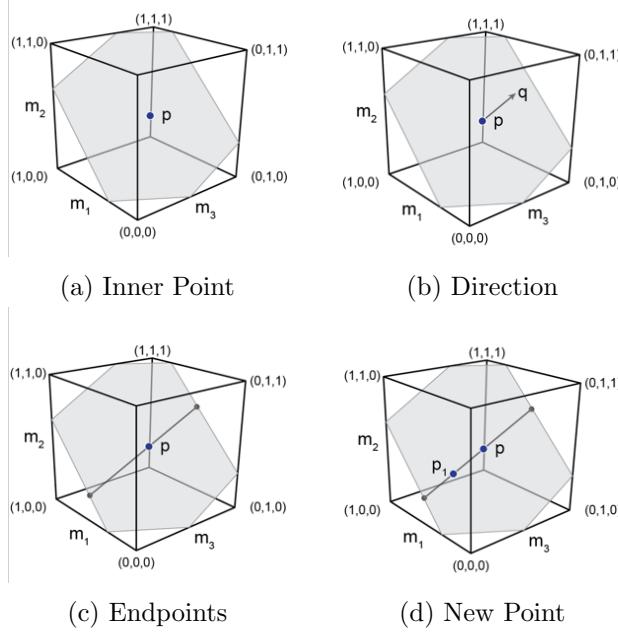


Figure 2: Hit-and-Run Step

154 with the well known Gram-Schmidt method (for details see e.g. [3]). Note that in order
 155 to get the desired u.a.r. sample the basis needs to be orthonormal. For the limb case we
 156 can safely assume that the rows of A are linearly independent and hence the number of
 157 basis vectors is $n - m$.

158 3.4 Mixing Time

159 How many steps are necessary to reach a uniformly at random point in the polytope?
 160 The theoretical bound $\mathcal{O}^*(n^2 R^2 / r^2)$ given in [8] has a very large hidden coefficient (10^{30})
 161 which makes the algorithm almost infeasible in lower dimensions.

162 These bounds hold for general convex sets. For convex polygons in higher dimensions,
 163 experimental results suggest that $\mathcal{O}(n)$ steps of the Hit-and-Run algorithm are sufficient.
 164 In particular Emiris and Fisikopoulos paper suggest that $(10 + 10\frac{n}{J})n$ steps are enough
 165 to have a close to uniform distribution [4]. In all cases tested, sampling more point did
 166 not make accuarcy significantly higher.

167 Ge et al. showed experimentally that up to about 40 dimensions, ??? random points
 168 seem to suffice to get a close to uniform discussion [5].

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

172 As a additional control, for each muscle we observe that the theoretical upper and
 173 lower bound of the feasible activation match the observed corresponding bounds (dif-

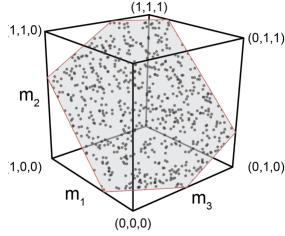


Figure 3: Uniform Distribution

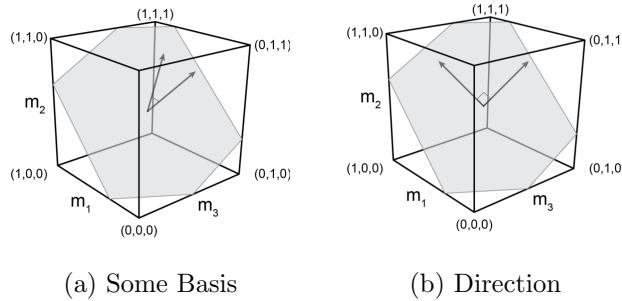


Figure 4: Find Orthonormal Basis

174 ference max ??). To find the theoretical upperbound (lowerbound) of a given muscle
175 activation we solve two linear programs maximizing (minimizing) a_i over the polytope.

176 3.5 Starting Point

177 To find a starting point in

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

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

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

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

186 **3.6 Parallel Coordinates: Visualization of the Feasible Activation Space**

187 Citation A common way to visualize higher dimensional data is using parallel coordinates[citations]. To show our sample set of points in the feasible activation space we
188 draw n parallel lines for each of the n muscles. With the axis labels of the line set
189 between 0 and 1, each point is then represented by connecting their coordinates by $n - 1$
190 lines.
191

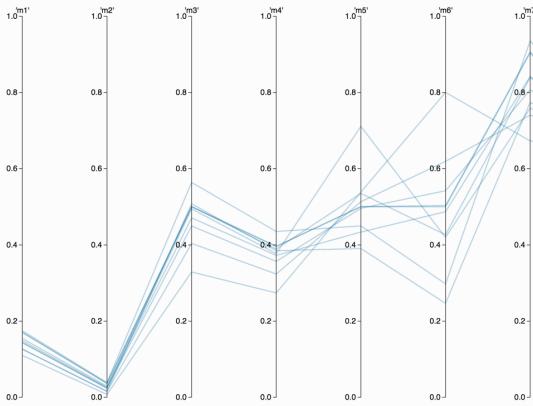


Figure 5: Feasible Activation

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

194 **NICE FIGURE OF RESTRICTED PARALLEL COORDINATES**

195 **3.7 Integration of muscle-metabolic and neural drive cost functions**

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

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

204 **NICE PICTURE WITH WEIGHTS INCLUDED**

205 **4 Results**

206 Many nice figures

207 1. Histograms

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	$\sum_{i=1}^n a_i F_{oi}$	REF
(weighted) l_2	$\sqrt{\sum_{i=1}^n (a_i F_{oi})^2}$	REF
(weighted) l_3	$\sqrt[3]{\sum_{i=1}^n (a_i F_{oi})^3}$	REF

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

208 2. Histograms 3 directions

209 3. PC

210 4.1 Activation Distribution on a Fixed Force Vector

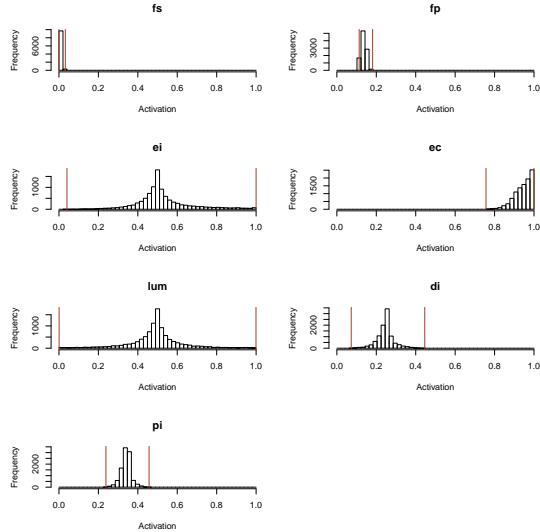


Figure 6: Histogram for Fixed Force

211 4.2 Changing Output Force in 3 Directions

212 We discuss different forces into three different directions, which are given by the palmar
 213 direction (x -direction), the distal direction (y -direction) and the sum of them. The
 214 maximal forces into each direction are given by ??, ?? and ?? respectively. For $\alpha =$

215 0.1, 0.2, ..., 0.9, we give the histograms where the force is $\alpha \cdot F_{\max}$, where F_{\max} is the
 216 maximum output force in the corresponding direction.

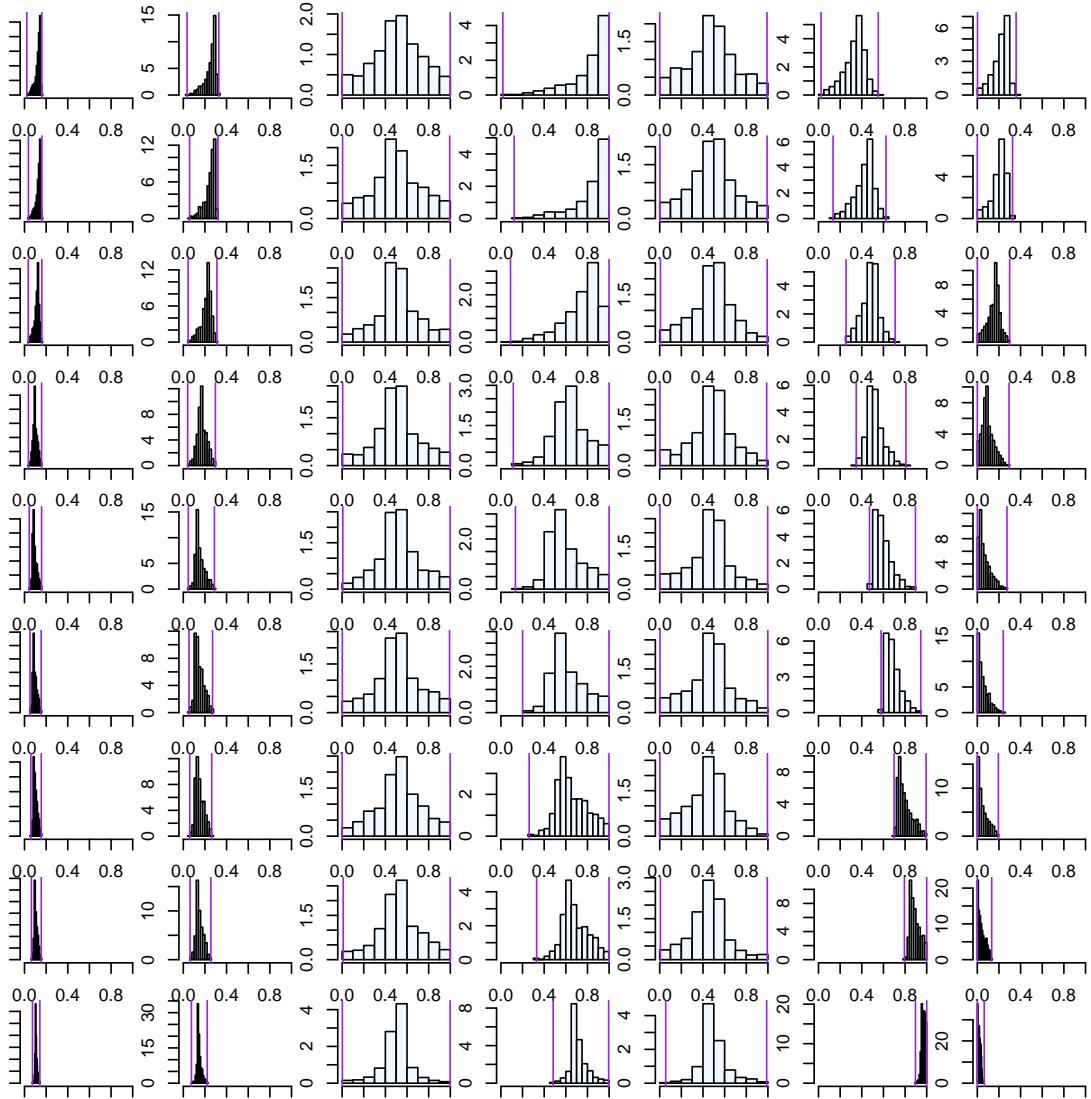


Figure 7: Histogram for x -Direction

217 4.3 Parallel Coordinates

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

220 **5 Discussion**

221 Mostly to be written by Brian

222 **5.1 Distributions**

- 223 • Bounding box away from 0 and 1 means muscle is really needed → Already known
224 from the bounding boxes
- 225 • High density → most solutions in that area

226 **5.2 Parallel Coordinates**

- 227 • Parallel lines in PC indicate opposite direction of muscles
- 228 • Crossing lines indicate similar direction

229 **5.3 Running Time**

230 The step of the algorithm which are time consuming are finding a starting point, which
231 solves a linear program and can take exponential running time in worst case. For each
232 fixed force vector we only have to find a starting point and an orthonormal basis once,
233 and are hence not of concern for the running time.

234 Running one loop of the hit and run algorithm only needs linear time, therefore the
235 method will extend to higher dimensions with only linear factor of additional running time
236 needed.

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

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

245 **6 Acknowledgments**

246 **References**

- 247 [1] M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for ap-
248 proximating the volume of convex bodies. *Proc. of the 21st annual ACM Symposium*
249 *of Theory of Computing*, pages 375–381, 1989.
- 250 [2] M. Dyer and M. Frieze. On the complexity of computing the volume of a polyhedron.
251 *SIAM Journal on Computing*, 17:967–974, 1988.

- 252 [3] T. S. Blyth E. F. Robertson. *Basic Linear Algebra*. Springer, 2002.
- 253 [4] I.Z. Emiris and V. Fisikopoulos. Efficient random-walk methods for approximating
254 polytope volume. *Preprint: arXiv:1312.2873v2*, 2014.
- 255 [5] C. Ge, F. Ma, and J. Zhang. A fast and practical method to estimate volumes of
256 convex polytopes. *Preprint: arXiv:1401.0120*, 2013.
- 257 [6] L.G. Khachiyan. On the complexity of computing the volume of a polytope. *Izvestia
258 Akad. Nauk SSSR Tekhn. Kibernet*, 216217, 1988.
- 259 [7] L.G. Khachiyan. The problem of computing the volume of polytopes is np-hard.
260 *Uspekhi Mat. Nauk* 44, 179180, 1989.
- 261 [8] L. Lovász. Hit-and-run mixes fast. *Math. Prog.*, 86:443–461, 1998.
- 262 [9] G. Marsaglia. Choosing a point from the surface of a sphere. *Ann. Math. Statist.*,
263 43:645–646, 1972.
- 264 [10] R.L. Smith. Efficient monte-carlo procedures for generating points uniformly dis-
265 tributed over bounded regions. *Operations Res.*, 32:1296–1308, 1984.

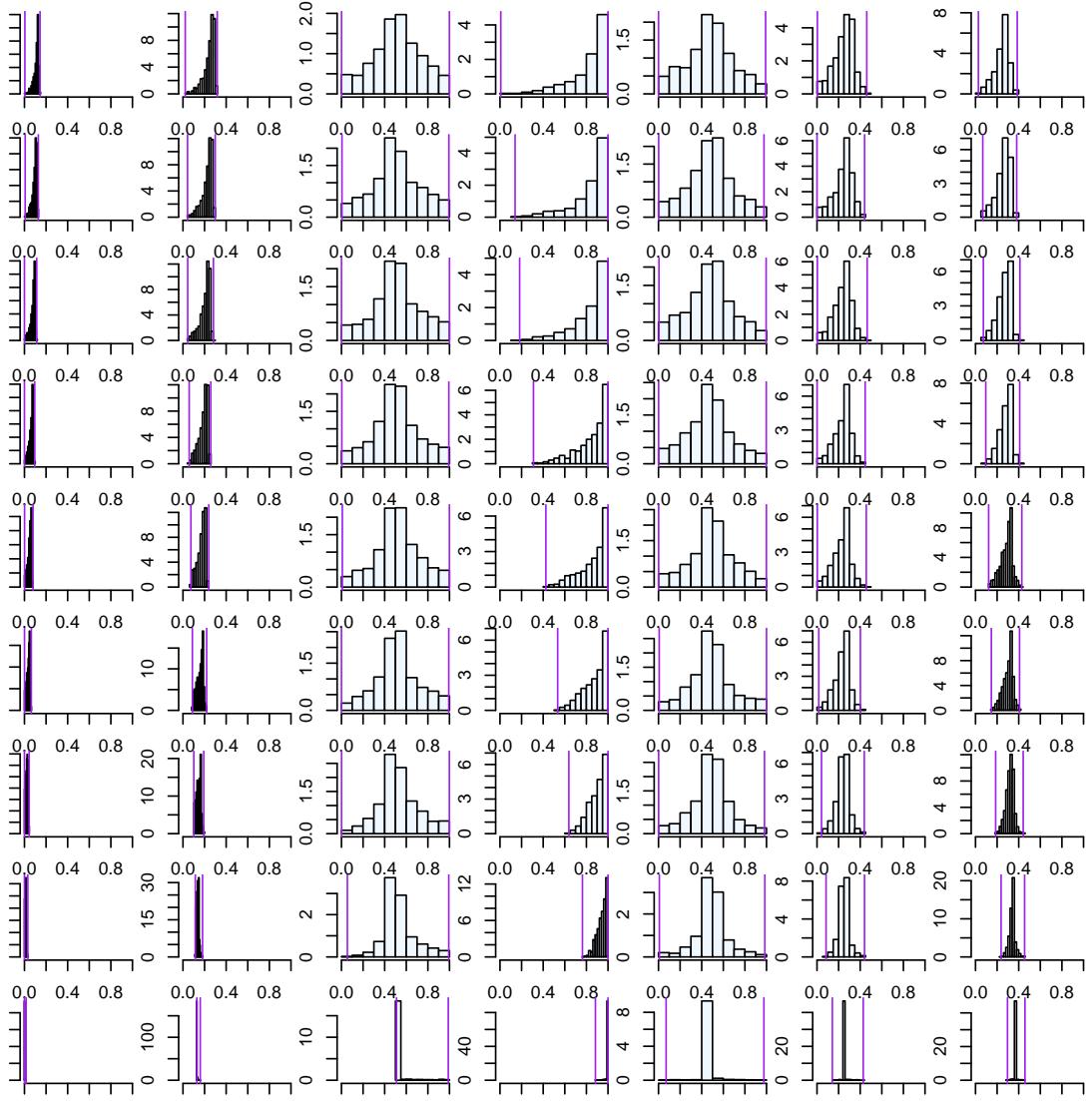


Figure 8: Histogram for xy -Direction

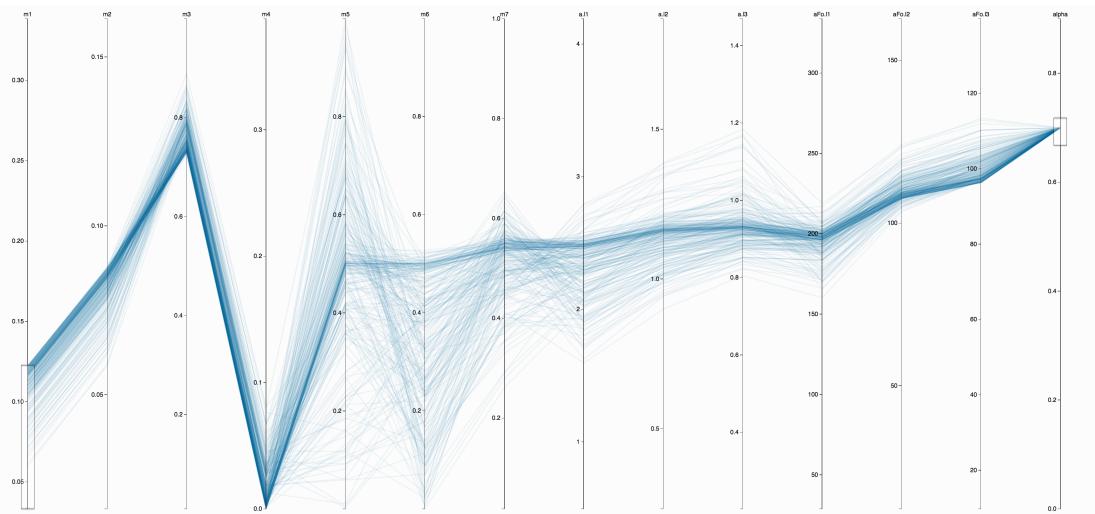


Figure 9: Low for Muscle 1

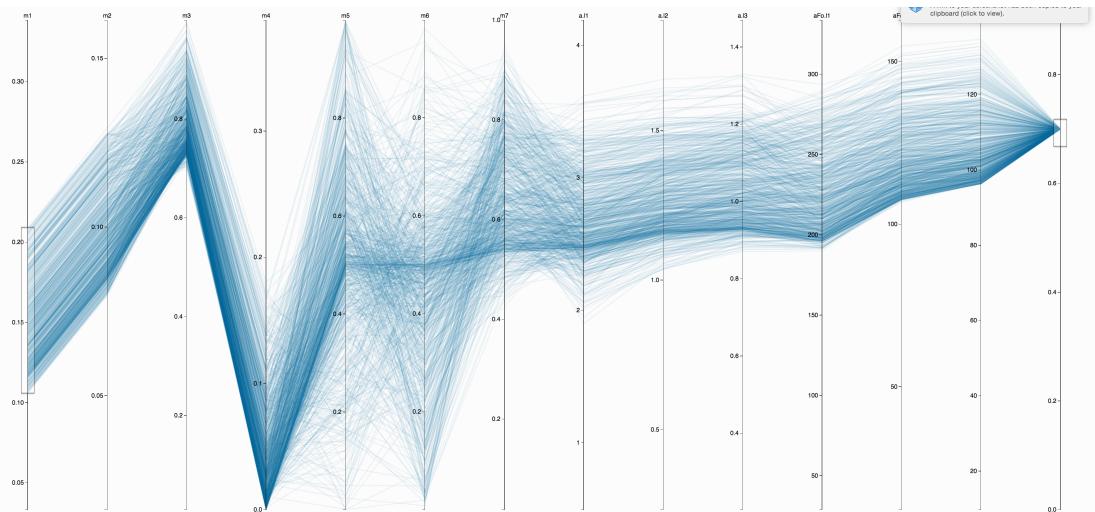


Figure 10: Middle for Muscle 1

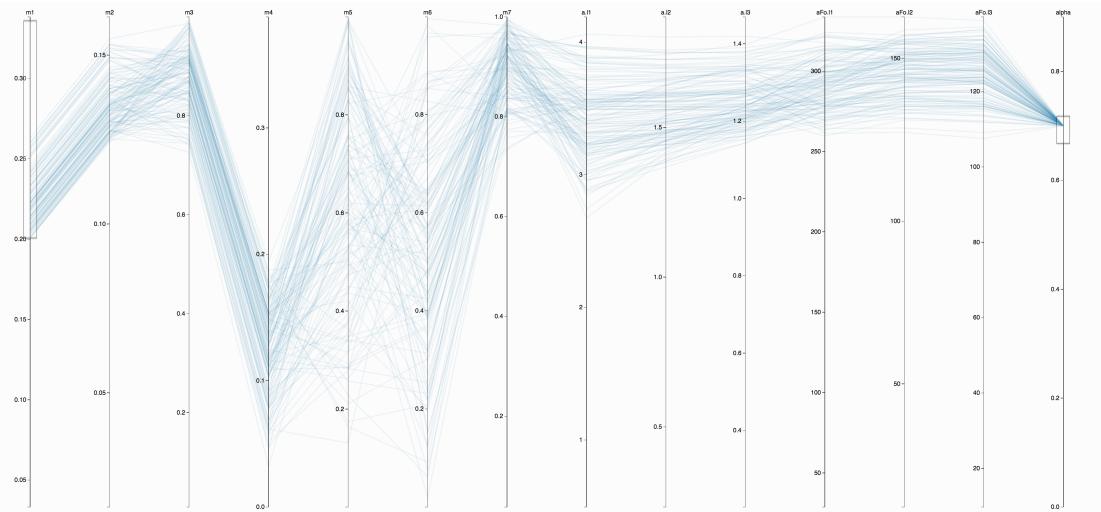


Figure 11: Upper for Muscle 1

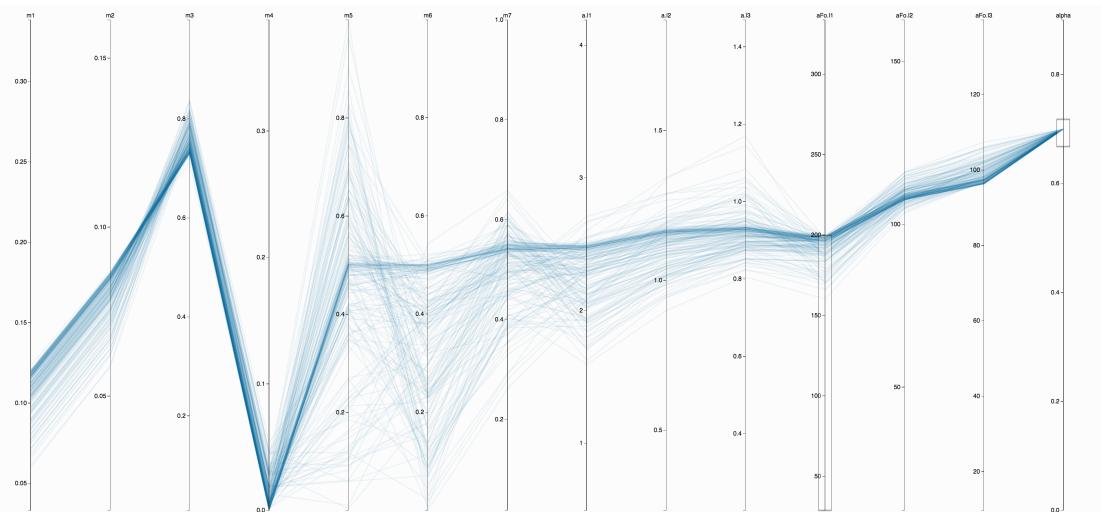


Figure 12: Weighted Cost