Biol. Cybern. 89, 190–199 (2003)
DOI 10.1007/s00422-003-0428-4

Biological Cybernetics

Avoiding spurious submovement decompositions: a globally optimal algorithm

Brandon Rohrer[1], **Neville Hogan**[2]

[1] Intelligent Systems and Robotics Center, Sandia National Laboratories, PO Box 5800, Albuquerque, NM 87185, USA
[2] Department of Mechanical Engineering and Department of Brain and Cognitive Science, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, USA

Received: 16 July 2002 / Accepted: 28 May 2003/ Published online: 9 July 2003

Abstract. Evidence for the existence of discrete submovements underlying continuous human movement has motivated many attempts to "extract" them. Although they produce visually convincing results, all of the methodologies that have been employed are prone to produce spurious decompositions. Examples of potential failures are given. A branch-and-bound algorithm for submovement extraction, capable of global nonlinear minimization (and hence capable of avoiding spurious decompositions), is developed and demonstrated.

1 Introduction

Several phenomena in human motor behavior suggest that human movement is composed of discrete units or *submovements*. Observations of slow movements (Vallbo and Wessberg 1993), eye saccades (Collewijn et al. 1988), cyclical movements (Woodworth 1899; Crossman and Goodeve 1983; Doeringer 1999) ballistic movements (Morasso 1981), and movements requiring high accuracy are all consistent with a theory of submovements. Observations such as these have motivated several attempts to produce a general methodology for isolating submovements (Morasso and Mussa-Ivaldi 1982; Flash and Henis 1991; Milner 1992; Berthier 1996; Lee et al. 1997; Burdet and Milner 1998). If successful, such decompositions of movement into their constituent discrete building blocks would provide a new window through which to observe the operation of the human motor control system. In this paper, we show that previous decomposition attempts can produce spurious results, and we present an algorithm that is *guaranteed* not to do so.

Submovements are theoretically attractive because they provide a compact language for concisely coding movement. Under the working hypothesis that these discrete units of movement exist, the ability to accurately isolate and characterize them would provide a description of human movement on a fundamental level that has not previously been available. As such, submovement analysis could provide new insights into studies of motor performance, rehabilitation, and the human motor control system.

However, since the posited underlying discrete commands are not directly available, there is no way to verify that a given decomposition is accurate. Accuracy can be inferred only by examination of the residual error. It is important to note that, although inaccurate decompositions may only have slightly higher residual error, the characteristics of the submovements that they employ may be completely spurious. Figure 1c and d shows an example of this phenomenon and will be discussed in more detail in the following section. Of course, even zero decomposition error does not prove that submovements actually exist. Nevertheless, in testing an empirically motivated theory of submovements, highly successful decompositions would certainly lend some degree of support to the theory.

Submovement decomposition is a nonlinear optimization problem: simultaneously maximizing goodness of fit and minimizing the number of submovements used, given a submovement shape [e.g., minimum-jerk (Hogan 1984) or Gaussian (Crossman and Goodeve 1983) (see Appendix)], and a summing modality [e.g., scalar summation (Morasso and Mussa-Ivaldi 1982) or vector summation (Flash and Henis 1991)]. As a nonlinear optimization problem, it may have multiple local minima. However, all the optimization methods that have been applied to it previously are sensitive to getting caught in local minima and cannot guarantee a globally optimal solution. Gradient descent (Berthier 1996) and Powell's Direction Set Method (Lee et al. 1997) have been used in this context, as well as manually adjusting ("eyeballing") submovement parameters (Morasso and Mussa-Ivaldi 1982; Milner 1992). The optimality of the solution for these methods depends heavily on the quality of the initial guess; unless the initial guess is in

Correspondence to: B. Rohrer
(e-mail: brrohre@sandia.gov, Tel.: +1-505-8446569,
Fax: +1-505-8448323)

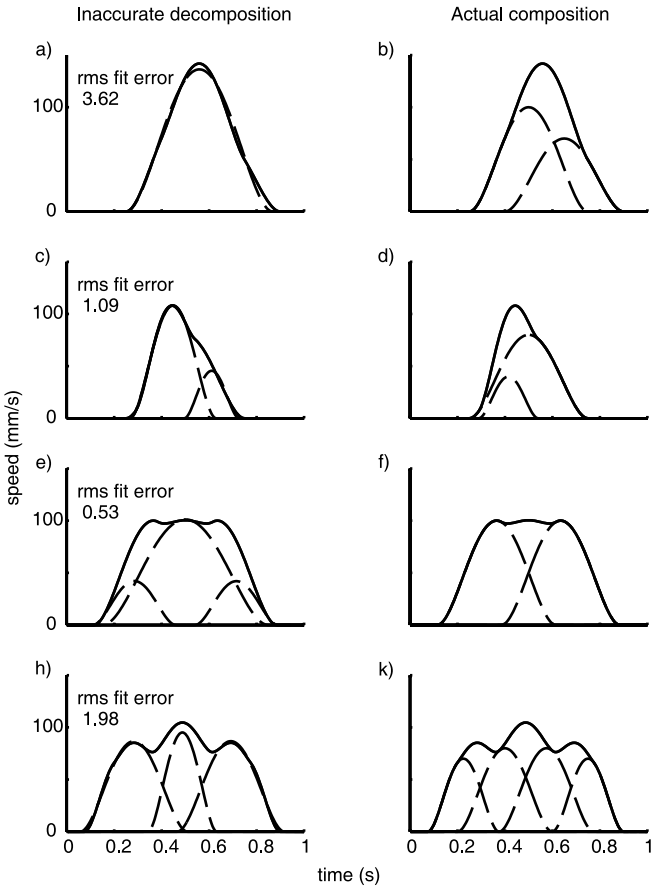

Fig. 1. Problematic decompositions for current algorithms. The panels on the right show the construction of the curves. The panels on the left show local minima in decomposition. In each case, the decomposition error is low, but the submovement characteristics do not resemble those used to construct the curve

the neighborhood of the global minimum, they will not find the best solution.

1.1 The difficulty of making a good initial guess

Making an initial guess that is in the neighborhood of the solution is not trivial. The right column of Fig. 1 shows several examples of speed profiles that pose problems for existing decomposition methods. Each speed profile is composed of minimum-jerk submovements that sum in a scalar fashion. Despite the fact that only a few simply parameterized submovements are used, the false decompositions are accurate to within as little as 0.5%. This illustrates the challenge that decomposition algorithms face in attempting to find the optimal solution.

Each speed profile shown has a different number of peaks than it has submovements. Any method that uses the number of peaks to estimate the number of submovements would fail to make an initial guess in the neighborhood of the global optimum. Examples of such failed decompositions are shown in Fig. 1a, e, and h.

The second movement (Fig. 1c and d) is of particular interest. It resembles the observed speed profiles of target-directed movements made by human subjects under moderate accuracy constraints (Milner and Ijaz 1990). Although decompositions similar to those shown in Fig. 1c have been assumed in previous analyses of such movements (Milner 1992), this example shows that other combinations of submovements can yield very similar shapes. This case illustrates the difficulty in making objective initial guesses and illustrates the high level of caution required when employing any technique that relies on subjective judgments.

Listed below are several methods for making initial guesses that have been previously applied to submovement decomposition. As originally implemented, some of the initial guessing methods below have not been used to initialize local minimization routines (such as gradient descent), but they could be used to do so in theory.

1.2 Initial guess methods

1.2.1 Subjective selection. Lee et al. (1997) have made initial guesses based on subjective estimation of submovement characteristics. This method is subject to the limitations illustrated in Fig. 1; submovement characteristics are difficult to intuit based on the speed profile, and therefore the initial guess is not guaranteed to be near the global optimum.

1.2.2 Matching pursuit. Matching Pursuit (Mallat and Zhang 1993) is a "greedy" algorithm; it finds the best fit for a single element at a time rather than for a set of elements. Matching Pursuit iteratively finds the submovement that, when subtracted from the function, minimizes the residual error. This repeats until some minimum error threshold is reached. The limitations of the Matching Pursuit algorithm are described in detail by Chen et al. (1998). The fact that Matching Pursuit fits a single submovement at a time does not allow it to optimize the fit for all submovements. Simple functions composed of as few as two submovements are incorrectly decomposed because of the greedy nature of the algorithm, as shown by Doeringer (1999).

1.2.3 Local fitting of the highest peak. The Irregular Sampling Radial Basis Function algorithm (Krebs 1997) and the method of Berthier (Berthier 1996) are also greedy algorithms. At each iteration both of these algorithms fit a submovement to the highest speed peak and its local neighborhood, and the fit function is subtracted from the original speed profile. As in Matching Pursuit, the process is repeated until either an error threshold is reached or a maximum number of submovements are fit. As illustrated in Fig. 1, aligning submovements with the highest speed peak provides no guarantee that the subfunctions chosen will actually coincide with those used to construct the original function.

1.2.4 Milner's method. Milner used zero velocity and maximum curvature points along individual axes to mark the onset of submovements in 3D movements (Milner 1992). As implemented, Milner's method is dependent on the choice of coordinate system and leads to a somewhat arbitrary division of the movement into submovements. Using maximum curvature points as submovement

delimiters suffers additionally from the fact that, although curvature tends to be maximum between submovements, it still does not need to be significant. Consecutive submovements in the same direction may have no clear peak in curvature by which to distinguish them.

1.2.5 High Resolution Pursuit. Another greedy algorithm, High Resolution Pursuit (HRP) (Jaggi et al. 1996) is similar to Matching Pursuit in that it minimizes the residual error but differs in that it emphasizes local fidelity of the fit and does not necessarily seek out the highest peak. Unfortunately, local fidelity of fit is not generally the best way to estimate submovement characteristics. Consider for example the movement depicted in Fig. 1k; HRP would likely fail to make an accurate initial guess for a speed profile since the chief characteristics of the speed profile do not resemble any of its component submovements.

The second movement of Fig. 1c and d provides an informative benchmark for gauging an algorithm's decomposition ability; it is both simple and resembles laboratory data. However, even assuming a priori knowledge of the correct number of submovements, the initial guesses of Milner's method and each of the greedy algorithms (Matching Pursuit, ISRBF, Berthier's method, and HRP) yield solutions that resemble Fig. 1c. Both the amplitude and the peak location of the decompositions would be spurious.

None of the initial guess methods listed above guarantees a guess that is in the neighborhood of the global optimum, and as shown in Fig. 1c and d, given plausible biological data they may yield inaccurate and misleading decompositions. Minimization based on these methods find, at best, locally optimal solutions to the submovement extraction problem. In order to reliably find the global minimum and solve the decomposition problem accurately, an algorithm capable of global nonlinear optimization is necessary.

2 Method

Branch-and-bound algorithms have been successfully applied to classical problems, such as the traveling salesman problem, as well as to economics problems (Little et al. 1963; Little 1966; Morin 1974) and are commonly used whenever it is necessary to find a *globally optimal* solution to a nonlinear problem rather than an approximate or a locally optimal solution.

Branch-and-bound actually describes a class of algorithms rather than a specific implementation. The idea underlying branch-and-bound algorithms is simple: to find the global minimum of a function over a bounded parameter space, repeatedly divide (branch) the parameter space into subspaces and bound the value of the function over each subspace. If the lower bound of the function over a subspace is higher than a known value of the function elsewhere, that subspace need not be searched further. This continues until the location of the solution is known sufficiently well. This algorithm requires that each parameter be bounded.

The application of the branch-and-bound algorithm to submovement extraction is straightforward. Consider a speed profile, g, and the current estimate of the speed profile $f(\boldsymbol{p})$, given by

$$f(\boldsymbol{p}) = \Sigma_{j=1}^{N} \lambda_j \tag{1}$$

where each λ_j is a submovement, completely described by m parameters. $\boldsymbol{p}$ is a vector containing the parameters of all the submovements. If N is the total number of submovements, then the total number of parameters in $\boldsymbol{p}$ is given by $M = N * m$. The formulation in Eq. 1 assumes scalar summation of submovements but could be generalized to other modes of summation.

The objective function to be minimized is the absolute error, $\mathscr{E}$, given by

$$\mathscr{E}(\boldsymbol{p}) = \int |g - f(\boldsymbol{p})| \tag{2}$$

The absolute error is chosen, rather than the rms error, because it simplifies the process of bounding error over solution subspaces (cf. Appendix).

2.1 Algorithm outline

The outline of the branch-and-bound algorithm as implemented in this work is as follows (refer to Fig. 2 for a step-by-step example in one dimension):

1. Bound the solution space. This requires finding upper and lower bounds for each element of $\boldsymbol{p}$. These parameter bounds can be thought of as describing an M-dimensional hyperbox that contains all permissible values of $\boldsymbol{p}$. Any given set of parameter values describes a point within the box and has a single value of $\mathscr{E}(\boldsymbol{p})$ associated with it. The goal of the algorithm is to find the point in the hyperbox for which $\mathscr{E}(\boldsymbol{p})$ is at a minimum.
2. Break the solution space into a number of subspaces.
3. Evaluate $\mathscr{E}(\boldsymbol{p}_c)$ (the value of $\mathscr{E}$ at the center of a subspace) for all subspaces.
4. Set $\mathscr{E}_{low} = \min(\mathscr{E}(\boldsymbol{p}_c))$ over all subspaces. $\mathscr{E}_{low}$ is the lowest known error in the solution space.
5. Calculate a lower bound L for $\mathscr{E}(\boldsymbol{p})$ over each subspace.
6. Eliminate subspaces for which $L > \mathscr{E}_{low}$ since they cannot contain the solution.
7. Break remaining subspaces down into yet smaller subspaces.
8. Return to step 3 and repeat until a termination criterion is met.

There are a few points in the algorithm that bear further explanation.

2.2 Bounding $\mathscr{E}(\boldsymbol{p})$ (step 5)

One method for calculating a lower bound on $\mathscr{E}(\boldsymbol{p})$ over a solution subspace is as follows:

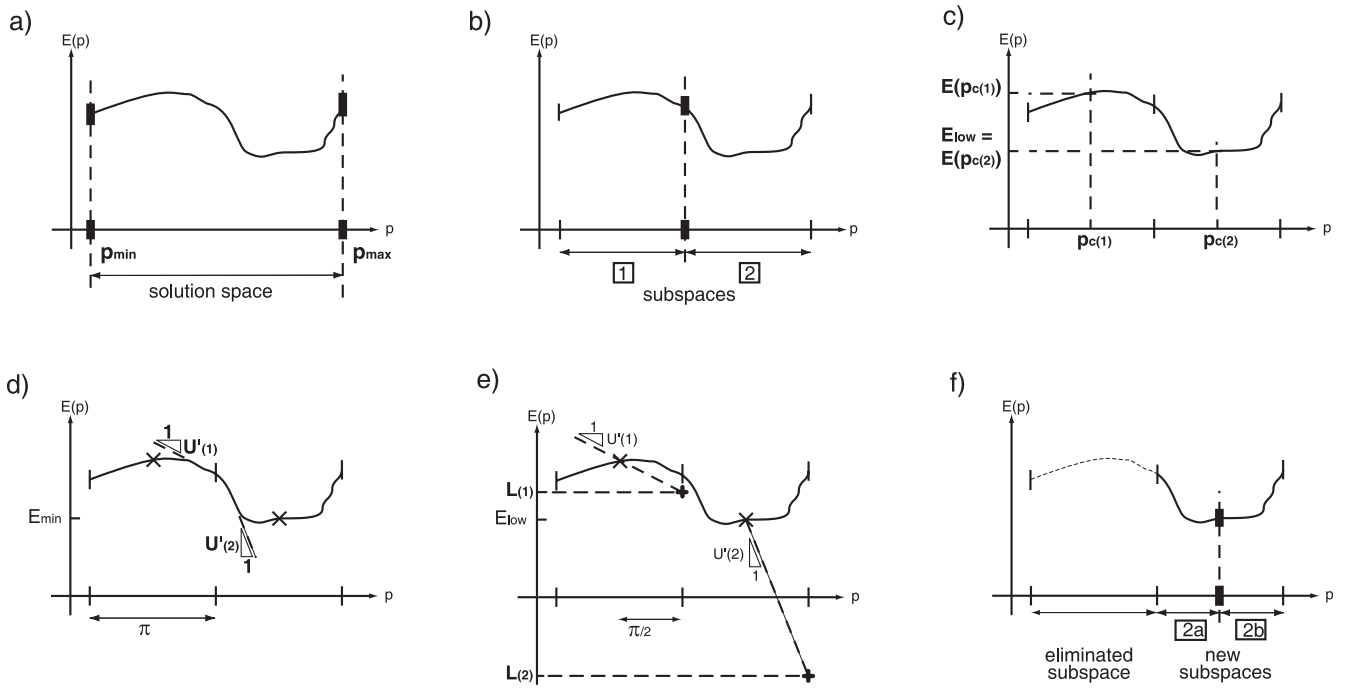

Fig. 2a-f. A step-by-step example of the branch-and-bound algorithm in one dimension: minimizing $\mathscr{E}(p)$ over a range of p. **a** Bound the solution space between p_{min} and p_{max}. **b** Break the solution space into subspaces 1 and 2. **c** Calculate the value of the objective function at the center of each subspace, $p_c(1)$ and $p_c(2)$. Retain the lowest value, $\mathscr{E}_{low}$, in memory. **d** Calculate an upper bound for the slope of the objective function over each subspace, $U'(1)$ and $U'(2)$. **e** Calculate a lower bound for the objective function over each subspace, $L_i = \mathscr{E}(p_{ci}) - \pi_i/2U_i'$, where π_i is the span (width) of each subspace. **f** Eliminate subspaces that cannot contain the solution. Break down remaining subspaces. Return to **c** and repeat until the error falls below a predetermined threshold, i.e., subspaces for which $L_i > \mathscr{E}_{low}$

1. Calculate an upper bound for $\left|\frac{\partial \mathscr{E}(\boldsymbol{p})}{\partial p_i}\right|$ (call it U_i') over the subspace. See the Appendix for methods of calculating U_i' for minimum-jerk, Gaussian, and lognormal speed curves.
2. Define π_i, the span of parameter i, as $\pi_i \equiv \max(p_i) - \min(p_i)$ over the subspace.
3. A lower bound, L, for $\mathscr{E}(\boldsymbol{p})$ over the subspace is given by

$$L = \mathscr{E}(\boldsymbol{p}_c) - \Sigma_i \frac{\pi_i}{2} U_i' \tag{3}$$

This guarantees that

$$\mathscr{E}(\boldsymbol{p}) \geq L \tag{4}$$

for all **p** in the current subspace.

2.3 Breaking down of solution subspaces (step 7)

There are many ways that the solution subspaces could be broken down into smaller subspaces. As implemented, the subspace was simply bisected along the parameter axis that had the greatest average value of $\pi_i/2U_i'$ on the previous iteration. This was done to shrink the error bounds of each subspace in as few iterations as possible.

Trisecting, rather than bisecting, may increase performance. When trisecting, values of p_c from previous iterations are also valid p_c's for future iterations. Because previously calculated values of $\mathscr{E}(\boldsymbol{p}_c)$ can be retained, subspaces can be divided more quickly with a very small increase in computational cost. Bisection does not have this advantage.

2.4 Terminating the search (step 8)

Iterations continue until every portion of the solution space has either (1) been eliminated or (2) had its error range bounded sufficiently narrowly. The error range of a subspace is calculated by taking the difference between the error at the center of the subspace, $\mathscr{E}(\boldsymbol{p}_c)$, and the lower bound of the error over the subspace, L. Once the error range falls below a certain threshold ($\mathscr{E}(\boldsymbol{p}_c) - L < \epsilon$), the subspace is not searched further.

This termination criterion does not ensure uniqueness of the solution. Uniqueness checks can be performed by checking whether the parameters for all the remaining solution spaces fall within a sufficiently small radius.

2.5 Minimizing the number of submovements

It should be noted that, to this point, the branch-and-bound algorithm has assumed that the number of submovements is given. Minimizing the number of submovements used can be achieved by starting with one submovement and iteratively incrementing the number of submovements fit to the objective curve until the error falls below a given threshold (see Fig. 3).

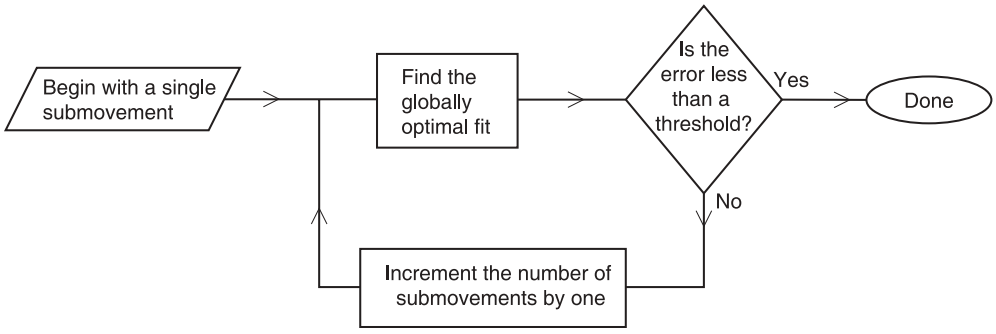

Fig. 3. Method for minimizing the number of submovements. An iterative error-checking algorithm provides a framework for minimizing the number of submovements while optimizing the fit of the submovements employed

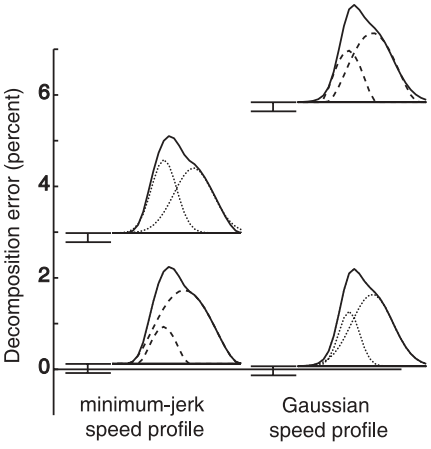

Fig. 4. Decomposition of simulated speed profiles composed of minimum-jerk and Gaussian submovements. *Solid lines* represent the original function in each case, *dashed lines* show the decompositions into minimum-jerk submovements, and *dotted lines* show the decompositions into Gaussian submovements. The plots show bounds on the decomposition error in each case

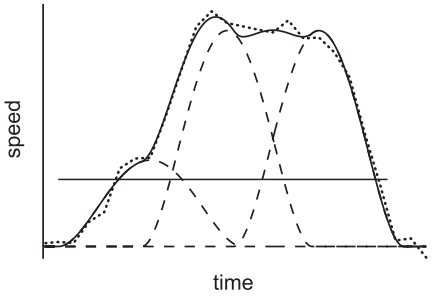

Fig. 5. Speed profile constructed for testing the noise sensitivity of the branch-and-bound algorithm. The *solid line* shows the composite speed profile, and *dashed lines* show the underlying submovements. A key characteristic of the curve is that peaks in the speed profile do not always correspond to the peaks of the underlying submovements. The *dotted line* shows the speed profile superimposed with Gaussian noise (from the 5% noise amplitude case)

3 Results

3.1 Solution-finding performance

In order to test the solution-finding performance of the branch-and-bound algorithm, two simulated speed profiles were created, one consisting of two minimum-jerk curves and one consisting of two Gaussian curves. Each was decomposed twice using two minimum-jerk curves in one attempt and two Gaussian curves in a second attempt. The results are summarized in Fig. 4.

A litmus test for any decomposition algorithm is how well it decomposes a synthetic speed profile where the submovement characteristics are known beforehand. Due to its likelihood of producing spurious solutions and to its similarity to patient data, the speed profile shown in Fig. 1c serves as a good initial benchmark. The simulated speed profiles in Fig. 4 were selected specifically to have the characteristics of the speed profile in Fig. 1c – a single peak followed by a clearly discernible "lobe" created by summing two submovements: a small one followed by a larger one. As can be seen in Fig. 4, both of these were successfully decomposed. Not only did the range of decomposition error include zero, but the submovement characteristics of the original function were captured as well.

Another test of a decomposition algorithm might be to reliably differentiate between speed profiles composed of different types of submovements. Figure 4 shows that the bounds on the decomposition error for each submovement function do not intersect. Nonintersection of the error bars shows the algorithm's ability to discriminate between the functions used to create each curve. It is also interesting to note that, because decomposition error cannot be negative, the fact that the error bounds extend below the x-axis in some cases illustrates the conservative nature of error bound estimation.

The Gaussian function's infinite tails did not distort this analysis; only the portion of the function that fell within the time window under consideration was used in error calculations.

3.2 Noise sensitivity

In order to test the sensitivity of the branch-and-bound algorithm to noise, a three-submovement speed profile was constructed (see Fig. 5). The movement contained several features that would prove problematic for existing decomposition algorithms: (1) one of the submovements had no corresponding peak in the composite speed profile, (2) the composite speed profile had a peak that did not correspond to an underlying submovement, and (3) the valleys between peaks were all shallow, which allows a small amount of signal noise to obscure existing peaks or create new ones.

In a series of tests, the speed profile was summed with varying amounts of Gaussian noise. Three tests were conducted with a noise signal amplitude of approximately 1%, 2%, and 5% of the total speed profile amplitude (see Fig. 5). Each noise-corrupted speed profile was decomposed using the branch-and-bound algorithm. In the case of 1% and 2% noise, submovement parameters extracted from the corrupted data were within 5% of the original submovement parameters. Even in the case of 5% noise, a significant distortion of the data, the submovement parameters did not depart far from nominal values but remained within 25% (seven of the nine submovement parameters remained within 10%). This is remarkable given the ease with which dramatically different submovements can produce similar composite speed profiles (see Fig. 1). The parameter error increased approximately in proportion with the amplitude of the noise signal. The modest and noise-dependent error in submovement parameters shows a graceful degradation of algorithm performance with signal corruption. The branch-and-bound algorithm is not overly sensitive to signal noise, even when it creates new peaks in the composite speed profile.

It should be noted, as illustrated in Fig. 1, that existing extraction algorithms, including those of Milner (1992), Berthier (1996), and Lee et al. (1997), cannot even be guaranteed to find a correct solution to the noise-free case. The absence of a peak corresponding to the first submovement in Fig. 5 and the presence of a peak between the second and third submovements pose a difficult, if not insurmountable, challenge for them. Although making use of acceleration to infer submovements would help identify the first, peakless submovement, it would not aid in identifying the second peak as extraneous.

In a second series of noise-sensitivity tests, submovements were extracted from the same speed profile shown in Fig. 5, but with noise added to the submovements, rather than to the overall movement from which they were extracted. The added noise was of the same form for each submovement – a sum of sines of frequency $n/2\tau$, where n is a small integer and τ is the duration of the submovement – such that the shape of each submovement was altered in the same way. The amplitude of the noise (and hence the extent of distortion of the underlying submovement shape) was varied at 1%, 2%, 5%, and 10% of the submovement amplitude. Due to the computational time required, the precision demanded of the algorithm remained low for these tests – 20% precision was guaranteed (however, it was observed that the solution produced differed from the true global optimum by only a few percent). Despite this low accuracy requirement, the submovement parameters did not vary greatly from nominal values and degraded gracefully as the noise amplitude increased. For example, in every case (1%, 2%, 5%, and 10% noise), the centers of the submovements were in error by no more than 5%.

As discussed in the Introduction, the ability of the branch-and-bound algorithm to accurately extract submovements independent of extrema in the kinematic data makes it unique among submovement extraction algorithms. The fact that the branch-and-bound algorithm can also locate submovements in the presence of noise is a further demonstration of its robustness.

3.3 Required computing power

The calculations required to do the decompositions in Fig. 4 took approximately 30 h on a 1.2-GHz Athlon processor. The calculations performed during the analysis for the "Noise sensitivity" section took over 660 h on a 1.7-GHz Intel processor. These are very modest decompositions compared to, say, decomposing the velocity profile produced while slowly tracing a circle, which may have ten or more submovements. As the number of submovements increases, the dimensionality of the solution space, M, increases. The number of subspaces to be evaluated at any given iteration is on the order of C^M, where C is a function of the specific problem parameters and the iteration number. As the solution space dimensionality increases, the computational requirements increase dramatically. Computational cost is the primary weakness of the branch-and-bound algorithm.

3.4 Convergence

Convergence (i.e., reaching the termination criterion) can be guaranteed because of the nature of the error bounds on each solution subspace. U_i' for each parameter is a scalar that is a function of the maximum parameter values in each subspace. Because each parameter is bounded individually, U_i' is bounded as well. Therefore, as the span π of each parameter decreases, that is, as the subspaces are divided and redivided, the difference between the known error at the center of the subspace, $\mathscr{E}(\boldsymbol{p}_c)$, and the lower bound of the error over the subspace L will necessarily decrease as well, eventually becoming arbitrarily small. It is guaranteed to fall below the chosen threshold ϵ at some point.

3.5 Extension to multidimensional movements

The branch-and-bound algorithm presented here was demonstrated using 1D movements, but there is no inherent limitation in the algorithm that precludes its application to multiple degrees of freedom. All that is required to do this is to define submovements with sufficient parameters and redefine $\mathscr{E}$ to reflect error in multiple directions. For example, 3D submovements may be represented by adding parameters with azimuth and elevation angles to define the direction of movement. The error function, $\mathscr{E}$, can be redefined as the sum of the errors in the projections along the x-, y-, and z-axes. Likewise, submovements of other than straight-line trajectories can also implemented. The only constraints are that the shape of the trajectories can be parameterized and that the effects on the error due to

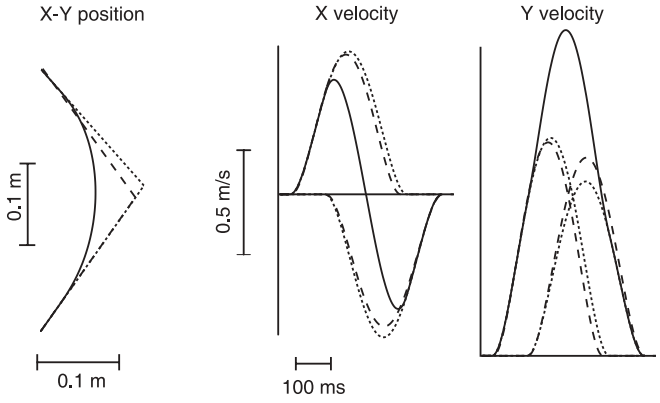

Fig. 6. Submovements extracted from a simulated planar (2D) movement. *Solid lines* represent the original movement, *dotted lines* represent the original submovements, and *dashed lines* represent the extracted submovements. The trajectory described by the extracted submovements differs from the original trajectory by approximately 1%. This is due to the fact that an excessively high guaranteed maximum of error of 10% was used in order to keep computation time reasonable. Applying greater computational resources will result in more accurate submovement extractions. In this case, submovements and the error were defined in terms of position rather than tangential velocity. See Appendix for a complete derivation of the 3D case

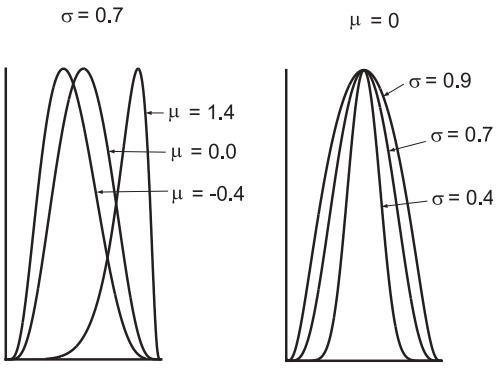

Fig. 7. An example of shape variations in a single submovement parameterization. The support-bounded lognormal function can take on a variety of shapes. Varying μ changes the symmetry, and changes in σ affect the kurtosis

changes in those parameters can be reasonably bounded. Aside from the additional computational expense incurred by increasing the dimensionality of the solution space, there is no barrier to extracting submovements from multidimensional movements. An example of multidimensional submovement extraction is shown in the decomposition of a simple 2D movement in Fig. 6. (See the Appendix for derivations of U_i' for 3D submovements.)

3.6 Fixed submovement parameterization does not imply a fixed submovement shape

The branch-and-bound algorithm requires that all submovements have a fixed parameterization, i.e., be describable through the same set of parameters throughout the decomposition process. In the case of minimum-jerk submovements, this implies that all submovements have the same shape; that is, that they can all be represented by scaled, dilated, and translated versions of a single "mother shape." This is not necessarily true for other types of submovements, however.

Support-bounded lognormal submovements (Plamondon 1992) for instance, are completely described by five parameters and can take on a variety of shapes (Fig. 7). Keeping fixed the amplitude and start and stop points, the shape can be varied considerably by varying the other two parameters. One modifies the symmetry and the other modifies the kurtosis or "fatness" of the curve.

4 Conclusion

The branch-and-bound algorithm described above is capable of finding the globally optimal decomposition for simulated speed data, making it unique among decomposition algorithms. This advance, if successfully applied to submovement decomposition in actual data, has the potential to clarify submovement structure in human movements. The insights gained from this process would provide a new window for observing the operation of the human motor control system and may allow for advances in measurement of motor performance, diagnostic procedures for motor disorders, and identification of motor control strategies.

Appendix: Calculating U′

What follows is an outline of the calculation of the upper bound on $\partial\mathscr{E}/\partial p_i$, that is, the change in error, $\mathscr{E}$, with respect to each parameter, p_i, over each subspace. This is also identified as U'. These derivations are graphically motivated; the optimization criterion has been chosen to be the absolute error ($\int |f(t) - g(t)|dt$) rather than the squared error ($\int (f(t) - g(t))^2 dt$) in order to facilitate this. The change in error with a change in each parameter ($\partial\mathscr{E}/\partial p_i$) can be visualized as the change in *area* between the candidate function and the objective function. The change in error can be no more than the sum of the area newly occupied and the area vacated by the function during the parameter change.

4.1 Minimum-jerk submovements

Minimum-jerk submovements can be uniquely described by three parameters – the amplitude of the peak A, the time at which the peak occurs t, and the duration of the movement w:

$$v(\tau) = \frac{A}{1.875}\left(30\left(\frac{\tau - t + \frac{w}{2}}{w}\right)^2 - 60\left(\frac{\tau - t + \frac{w}{2}}{w}\right)^3 + 30\left(\frac{\tau - t + \frac{w}{2}}{w}\right)^4\right), t - \frac{w}{2} \le \tau \le t - \frac{w}{2}$$
$$= 0\,, \quad \text{otherwise} \tag{5}$$

The area under a minimum-jerk curve is equal to $wA/1.875$.

4.1.1 Bounding $\partial\mathscr{E}/\partial t$. For all unimodal (single-peaked) submovements, changes in t will both occupy and vacate no more than a rectangle of width Δt and height equal to the height of the function.

$$\Delta\mathscr{E} \leq 2\Delta t A \tag{6}$$

$$\frac{\Delta\mathscr{E}}{\Delta t} \leq 2A \tag{7}$$

$$\frac{\partial\mathscr{E}}{\partial t} \leq 2A_{MAX} \tag{8}$$

where A_{MAX} is the maximum amplitude of the submovement over the current subspace.

4.1.2 Bounding $\partial\mathscr{E}/\partial w$. Changes in w will either vacate area or occupy new area, but not both. The change in area occupied is given by

$$\Delta\mathscr{E} = \frac{\Delta w}{w} \text{ (area under minimum-jerk curve)} \tag{9}$$

$$\Delta\mathscr{E} = \frac{\Delta w}{w}\left(\frac{wA}{1.875}\right) \tag{10}$$

$$\Delta\mathscr{E} = \frac{\Delta w A}{1.875} \tag{11}$$

$$\frac{\Delta\mathscr{E}}{\Delta w} = \frac{A}{1.875} \tag{12}$$

$$\frac{\partial\mathscr{E}}{\partial w} \leq \frac{A_{MAX}}{1.875} \tag{13}$$

4.1.3 Bounding $\partial\mathscr{E}/\partial A$. As with w, changes in A will either vacate area or occupy new area, but not both. The area scales linearly with amplitude.

$$\Delta\mathscr{E} = \frac{\Delta A}{A} \text{ (area under minimum-jerk curve)} \tag{14}$$

$$\Delta\mathscr{E} = \frac{\Delta A}{A}\left(\frac{wA}{1.875}\right) \tag{15}$$

$$\Delta\mathscr{E} = \frac{\Delta A w}{1.875} \tag{16}$$

$$\frac{\Delta\mathscr{E}}{\Delta A} = \frac{w}{1.875} \tag{17}$$

$$\frac{\partial\mathscr{E}}{\partial A} \leq \frac{w_{MAX}}{1.875} \tag{18}$$

4.2 Gaussian submovements

Gaussian speed curves can also be uniquely described by three parameters, the amplitude of the peak A, the time at which the peak occurs t, and the standard deviation of the curve σ: $v(\tau) = Ae^{(\tau-t)^2/2\sigma^2}$. The area under such a Gaussian curve is equal to $\sqrt{2\pi}\sigma A$.

4.2.1 Bounding $\partial\mathscr{E}/\partial t$. Because Gaussian curves are unimodal, changes in t will both occupy and vacate no more than a rectangle of width Δt and height equal to the height of the function.

$$\Delta\mathscr{E} \leq 2\Delta t A \tag{19}$$

$$\frac{\Delta\mathscr{E}}{\Delta t} \leq 2A \tag{20}$$

$$\frac{\partial\mathscr{E}}{\partial t} \leq 2A_{MAX} \tag{21}$$

where A_{MAX} is the maximum amplitude of the submovement over the current subspace.

4.2.2 Bounding $\partial\mathscr{E}/\partial\sigma$. Changes in σ will either vacate area or occupy new area, but not both. The change in area occupied is given by

$$\Delta\mathscr{E} = \frac{\Delta\sigma}{\sigma} \text{ (area under a Gaussian curve)} \tag{22}$$

$$\Delta\mathscr{E} = \frac{\Delta\sigma}{\sigma}\left(\sqrt{2\pi}\sigma A\right) \tag{23}$$

$$\frac{\Delta\mathscr{E}}{\Delta\sigma} = \sqrt{2\pi}A \tag{24}$$

$$\frac{\partial\mathscr{E}}{\partial\sigma} \leq \sqrt{2\pi}A_{MAX} \tag{25}$$

4.2.3 Bounding $\partial\mathscr{E}/\partial A$. As with σ, changes in A will either vacate area or occupy new area, but not both. The area scales linearly with amplitude.

$$\Delta\mathscr{E} = \frac{\Delta A}{A}\text{(area under Gaussian curve)} \tag{26}$$

$$\Delta\mathscr{E} = \frac{\Delta A}{A}\left(\sqrt{2\pi}\sigma A\right) \tag{27}$$

$$\Delta\mathscr{E} = \sqrt{2\pi}\sigma\Delta A \tag{28}$$

$$\frac{\Delta\mathscr{E}}{\Delta A} = \sqrt{2\pi}\sigma \tag{29}$$

$$\frac{\partial\mathscr{E}}{\partial A} \leq \sqrt{2\pi}\sigma_{MAX} \tag{30}$$

4.3 Lognormal submovements

Both minimum-jerk and Gaussian submovements are symmetric. Lognormal submovements are asymmetric and hence are capable of reproducing the asymmetries often encountered in measured submovements. A lognormal curve is created by plotting a Gaussian curve on a logarithmic x-axis, instead of on a linear x-axis. (Aitchison and Brown 1964) It can be expressed as follows:

$$\Lambda(\mu,\sigma,t) = \frac{1}{\sigma\sqrt{2\pi}(\tau-t)}e^{-\left\{\frac{[ln(\tau-t)-\mu]^2}{2\sigma^2}\right\}},\quad \tau > t \tag{31}$$

Four independent parameters control the shape and position of the lognormal curve. t is the offset along the x-axis and controls the starting position of the curve. μ

controls the extent of asymmetry, and σ controls the kurtosis or "fatness" of the curve. As μ gets larger and σ gets smaller, the curve moves from being left-skewed toward being symmetric. (A lognormal curve cannot be right-skewed.) The fourth parameter is a scaling parameter, A. When scaled by $\sigma\sqrt{2\pi}e^{(\mu-\sigma^2/2)}A$, a lognormal curve, Λ, has a peak amplitude of A. As the area under an unscaled lognormal curve (the distance traveled, the integral of the velocity) is 1, the area under a lognormal curve of amplitude A is $\sigma\sqrt{2\pi}e^{(\mu-\sigma^2/2)}A$.

4.3.1 Bounding $\partial\mathscr{E}/\partial t$. Because, like Gaussian and minimum-jerk curves, lognormal curves are unimodal, changes in t will both occupy and vacate no more than a rectangle of width Δt and height equal to the height of the function.

$$\Delta\mathscr{E} \leq 2\Delta t A \tag{32}$$

$$\frac{\Delta\mathscr{E}}{\Delta t} \leq 2A \tag{33}$$

$$\frac{\partial\mathscr{E}}{\partial t} \leq 2A_{MAX} \tag{34}$$

where A_{MAX} is the maximum amplitude of the submovement over the current subspace.

4.3.2 Bounding $\partial\mathscr{E}/\partial A$. Changes in A will either vacate area or occupy new area, but not both. The area scales linearly with amplitude.

$$\Delta\mathscr{E} = \frac{\Delta A}{A}(\text{area under a lognormal curve}) \tag{35}$$

$$\Delta\mathscr{E} = \frac{\Delta A}{A}\left(A\sigma\sqrt{2\pi}e^{(\mu-\sigma^2/2)}\right) \tag{36}$$

$$\Delta\mathscr{E} = \left(\sigma\sqrt{2\pi}e^{(\mu-\sigma^2/2)}\right)\Delta A \tag{37}$$

$$\frac{\Delta\mathscr{E}}{\Delta A} = \sigma\sqrt{2\pi}e^{(\mu-\sigma^2/2)} \tag{38}$$

$$\frac{\partial\mathscr{E}}{\partial A} \leq \sigma_{MAX}\sqrt{2\pi}e^{(\mu_{MAX}-\sigma_{MAX}^2/2)} \tag{39}$$

where σ_{MAX} and μ_{MAX} are the maximum σ and μ of the submovement over the current subspace, respectively. The area under the lognormal cure is at a maximum for $\sigma = 1$ and increases with μ. Since $\sigma < 1$ is capable of producing a full range of plausible submovement shapes, maximizing both σ and μ over the subspace maximizes the area under the curve.

4.3.3 Bounding $\partial\mathscr{E}/\partial\mu$. An analytical solution for $\partial\mathscr{E}/\partial\mu$ is not straightforward, and even if one is achieved, the expense involved in calculating it during algorithm operation is likely to make the algorithm inefficient. The fact that μ is bounded allows for numerical bounds on $\partial\mathscr{E}/\partial\mu$ to be calculated in a brute force method offline, before using the algorithm. By calculating $\Delta\mathscr{E}/\Delta\mu$ at a number of positions throughout the solution space, a numerical upper bound can be found. Due to the empirical nature of this method, it is sensitive to the bounds selected for μ and σ. In this analysis, $2 \leq \mu \leq 5$ and $0.02 \leq \sigma \leq 0.3$.

$$\frac{\Delta\mathscr{E}}{\Delta\mu} \leq 4Ae^{\mu} \tag{40}$$

$$\frac{\partial\mathscr{E}}{\partial\mu} \leq 4A_{MAX}e^{\mu_{MAX}} \tag{41}$$

where μ_{MAX} and A_{MAX} are the μ and maximum amplitude of the submovement over the current subspace, respectively.

4.3.4 Bounding $\partial\mathscr{E}/\partial\sigma$. A bound on $\frac{\partial\mathscr{E}}{\partial\sigma}$ can be calculated in a fashion similar to $\frac{\partial\mathscr{E}}{\partial\mu}$. It is also sensitive to the bounds selected for μ and σ ($2 \leq \mu \leq 5$ and $0.02 \leq \sigma \leq 0.3$).

$$\frac{\Delta\mathscr{E}}{\Delta\sigma} \leq 4.5Ae^{\mu} \tag{42}$$

$$\frac{\partial\mathscr{E}}{\partial\sigma} \leq 4.5A_{MAX}e^{\mu_{MAX}} \tag{43}$$

where μ_{MAX} and A_{MAX} are the maximum μ and amplitude of the submovement over the current subspace, respectively.

4.4 Three-dimensional minimum-jerk submovements

This section provides an example of how the branch-and-bound algorithm can be applied to higher-dimensional submovement analysis. The chief difference between 3D submovement extraction as implemented here and 1D submovement extraction is that positional errors at each time point are analyzed, as opposed to tangential velocity errors.

A 3D minimum-jerk submovement can be defined by five parameters: time of peak velocity, t, duration of movement, w, and amplitude of peak in the x-, y-, and z-directions, A_x, A_y, and A_z.

The derivation of the error bounds for 3D submovements are graphically motivated, just as those for 1D submovements were. In order to facilitate calculation of the error bounds, the error, $\mathscr{E}$, has again been chosen to be the absolute error (i.e., the l_1 norm or $\int(|f_x(t)-g_x(t)| + |f_y(t)-g_y(t)| + |f_z(t)-g_z(t)|)dt$, where $g_x(t)$ is x-position of the movement at t, and $f_x(t)$ is the position of the reconstructed movement at t, etc.), rather than the root mean square error (i.e., the l_2 norm or $\int\sqrt{(f_x(t)-g_x(t))^2 + (f_y(t)-g_y(t))^2 + (f_z(t)-g_z(t))^2}dt$). The change in error with a change in each parameter ($\partial\mathscr{E}/\partial p_i$) can be visualized as the change in *area* between the candidate function and the objective function, this time in terms of position rather than velocity, and scalar summed across the multiple axes. The change in error can be no more than the sum of the area newly occupied and the area vacated by the function during the parameter change.

The distance, D, covered by minimum-jerk movement is equal to $wA/1.875$.

4.4.1 Bounding $\partial\mathscr{E}/\partial t$

Graphically, the area between a submovement and its time-shifted twin is simply $D\Delta t$.

$$\Delta\mathscr{E} = \Delta t D \tag{44}$$

$$\frac{\Delta\mathscr{E}}{\Delta t} = D \tag{45}$$

$$\frac{\Delta\mathscr{E}}{\Delta t} = \frac{w(A_x + A_y + A_z)}{1.875} \tag{46}$$

$$\frac{\partial\mathscr{E}}{\partial t} \leq \frac{w_{MAX}(A_{xMAX} + A_{yMAX} + A_{zMAX})}{1.875} \tag{47}$$

4.4.2 Bounding $\partial\mathscr{E}/\partial w$. Changes in w will either vacate area or occupy new area, but not both, and will change the total distance covered by the submovement. If the submovement is allowed to proceed to completion, i.e., if the movement is not terminated before the submovement has a chance to play out, then the error incurred during the life of the submovement is less than or equal to $|D_{orig} - D_{new}| * w/2$, where w is the duration of either the new or old submovement, whichever is greater. In addition, the distance error is propagated throughout the remainder of the movement. This leads to a total error bound of $|D_{orig} - D_{new}| * (T - t)$, where t is the time at which peak submovement velocity is reached and T is the duration of the entire composite movement.

$$\Delta\mathscr{E} \leq \frac{\Delta w(A_x + A_y + A_z)(T - t)}{1.875} \tag{48}$$

$$\frac{\partial\mathscr{E}}{\partial w} \leq \frac{(A_{xMAX} + A_{yMAX} + A_{zMAX})(T - t_{MIN})}{1.875} \tag{49}$$

4.4.3 Bounding $\partial\mathscr{E}/\partial A$. As with w, changes in A will either vacate area or occupy new area, but not both. The area scales linearly with amplitude. The formulation is similar to the error for $\partial\mathscr{E}/\partial w$ and is identical for each of the three axes.

$$\Delta\mathscr{E} = \frac{\Delta A_x w(T - t)}{1.875} \tag{50}$$

$$\frac{\partial\mathscr{E}}{\partial A_x} \leq \frac{w_{MAX}(T - t_{MIN})}{1.875} \tag{51}$$

Acknowledgements. This work was supported by National Institutes of Health Grants R01-HD37397 and R01-HD36827 and by a National Science Foundation graduate fellowship (B.R.). Sandia is a multiprogram laboratory operated by Sandia Corp., a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. The work performed complies with the current laws of the United States of America.

References

Aitchison J, Brown JAC (1957) The lognormal distribution. Cambridge University Press, Cambridge, UK

Berthier NE (1996) Learning to reach: a mathematical model. Dev Psychol 32: 811–823

Burdet E, Milner TE (1998) Quantization of human motions and learning of accurate movements. Biol Cybern 78: 307–318

Chen SS, Donoho DL, Saunders MA (1998) Atomic decomposition by basis pursuit. SIAM J Sci Comput 20: 33–61

Collewijn H, Erkelens CJ, Steinman RM (1988) Binocular coordination of human horizontal saccadic eye movements. J Physiol 404: 157–182

Crossman ERFW, Goodeve PJ (1983) Feedback control of hand-movements and Fitt's law. Q J Exp Psychol A 35: 251–278

Doeringer JA (1999) An investigation into the discrete nature of human arm movements. PhD thesis, MIT, Cambridge, MA

Flash T, Henis E (1991) Arm trajectory modifications during reaching towards visual targets. J Cogn Neurosci 3: 220–230

Hogan N (1984) An organizing principle for a class of voluntary movements. J Neurosci 4: 2745–54

Jaggi S, Karl WC, Mallat S, Willsky AS (1996) High resolution pursuit for feature extraction. Technical Report LIDS-P-2371, Laboratory for Information and Decision Systems, MIT, Cambridge, MA

Krebs HI (1997) Robot-aided neurorehabilitation and functional imaging. PhD thesis, MIT, Cambridge, MA

Lee D, Port NL, Georgopoulos AP (1997) Manual interception of moving targets. II. On-line control of overlapping submovements. Exp Brain Res 116: 421–422

Little JDC (1966) Branch and bound methods for combinatorial problems. Technical Report 178–66, Working paper Sloan School of Management, MIT, Cambridge, MA

Little JDC, Murty KG, Sweeney DW, Karel C (1963) An algorithm for the traveling salesman problem. Oper Res 11: 972–989

Mallat S, Zhang Z (1993) Matching pursuit with time-frequency dictionaries. IEEE Trans Signal Process 41: 3397–3415

Milner TE (1992) A model for the generation of movements requiring endpoint precision. Neuroscience 49: 365–374

Milner TE, Ijaz MM (1990) The effect of accuracy constraints on three-dimensional movement kinematics. Neuroscience 35: 487–496

Morasso P (1981) Spatial control of arm movements. Exp Brain Res 42: 223–227

Morasso P, Mussa-Ivaldi FA (1982) Trajectory formation and handwriting: a computational model. Biol Cybern 45: 131–142

Morin TL (1974) Branch-and-bound strategies for dynamic programming. Technical Report 750-74, Working paper, Sloan School of Management, MIT, Cambridge, MA

Plamondon R (1992) Tutorials in Motor Behavior II. Chapter: a theory of rapid movements. Elsevier, Amsterdam, pp 55–69

Vallbo AB, Wessberg J (1993) Organization of motor output in slow finger movements in man. J Physiol 469: 673–691

Woodworth RS (1899) The accuracy of voluntary movement. Psychol Rev Monogr Suppl