Analysis of arm movement data

Jason Friedman

MACCS
Macquarie University

29th October 2010

Overview of kinematic measures and their uses

- This talk will describe how to extract several useful measures from movement trajectories, and arm movement data in particular
- The focus is on dependent variables that can be extracted from movements.
- The focus is *not* on modelling how the movements are produced (i.e., human motor control)

Segmenting movement data

- Movement data generally has variable starting times
- In order to compare across trials, we often want to trim the data in some consistent way
- Movement onset is usually defined by 5% of peak tangential velocity

Matlab: Load the data (refresher)

As with last time, we will load the data from a file, but this time we will load multiple files. Here we are just considering x and y.

```
for k=1:10
  data{k} = load(sprintf(...
   'optotrak_data%02d.csv',k));
  x1{k} = data{k}(:,2);
  % remove missing samples at start and end
  firstpoint = find(~isnan(x1{k}),1);
  lastpoint = find(~isnan(x1{k}),1,'last');
  rng = firstpoint:lastpoint;
  x1{k} = data{k}(rng,2);
  y1{k} = data{k}(rng,3);
  % The first column is the frame, convert to
  % seconds (sample rate was 200Hz)
  times{k} = (data{k}(rng,1)-data{k}(1,1))./200;
```

Matlab: Load the data (refresher)

Interpolate the missing samples

```
goodsamples = ~isnan(x1{k});
x1{k} = interp1(times{k}(goodsamples),...
 x1{k}(goodsamples),times{k});
y1{k} = interp1(times{k}(goodsamples),...
 y1{k}(goodsamples),times{k});
```

We filter the data with a two-way 4th order Butterworth filter, and calculate the velocity.

```
[B,A] = butter(2,0.2);
x1{k} = filtfilt(B,A,x1{k});
y1{k} = filtfilt(B,A,y1{k});
xvel1{k} = [0;diff(x1{k})./(1/200)];
yvel1{k} = [0;diff(y1{k})./(1/200)];
```

Matlab: Find movement onset (and end time)

We calculate the tangential velocity ($v_t = \sqrt{\dot{x}_1^2 + \dot{x}_2^2}$), and find the first time when 5% of the maximum is crossed

```
tangvel{k} = sqrt(xvel1{k}.^2 + yvel1{k}.^2);
startmovement(k) = find(tangvel{k} > ...
 0.05*max(tangvel{k}),1);
```

In this case, we will consider the end of the movement the first time the velocity goes below 5% of peak (after passing 50% of the peak velocity).

```
passedhalf = find(tangvel{k} > ...
   0.5 * max(tangvel{k}),1);
endmovement(k) = find(tangvel{k}...
 (passedhalf:end,:) < ...
 0.05*max(tangvel{k}),1) + passedhalf- 1;
end
```

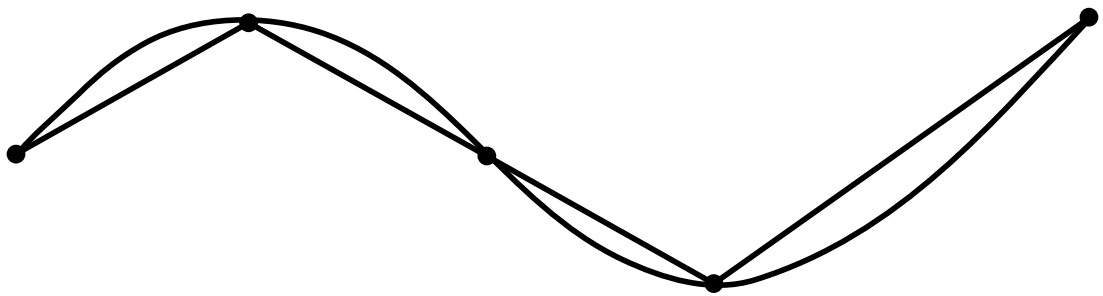
Matlab: Finding movement onset

We will now cut the data, and plot them all to see that they are reasonable.

```
for k=1:10
   x1_cut{k} = x1{k}(startmovement(k):...
                        endmovement(k));
   y1_cut{k} = y1{k}(startmovement(k):...
                        endmovement(k));
   plot(x1_cut{k});
   hold on;
end
```

Calculation of arc length

- Arc length is the length of the path, going along the path
- For curves with an analytic expression, it can be calculated exactly
- For experimentally measured trajectories, we approximate it by a sum of the length of straight lines joining the points



- Arc length can be used as a dependent measure (of how far the arm travelled), or for calculating other measures (e.g., what is the velocity half way along the path?)

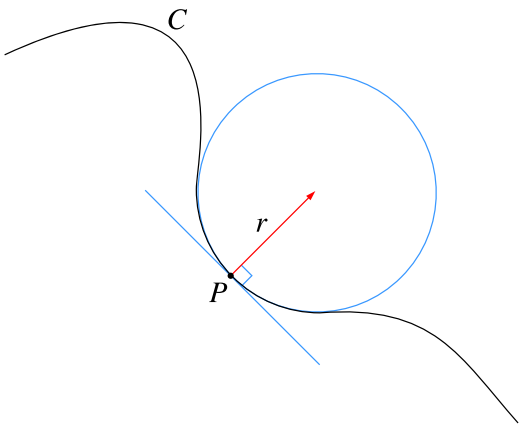
Matlab: calculation of arc length

We can compute the arc length by calculating the absolute distance between every two points, and summing

```
for k=1:10
   arclength{k} = cumsum(...
    sqrt(diff(x1_cut{k}).^2 + ...
         diff(y1_cut{k}).^2));
end
```

Calculation of path offset / "curvature"

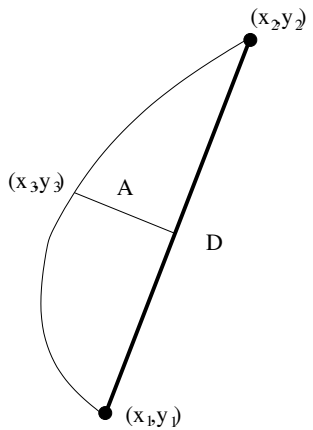
- Curvature is a measure of how "curved" something is.
- The mathematical definition of curvature for a continuous curve is that it is equal to the inverse of the radius of the osculating circle at that point (i.e., the circle that matches the curve at that point).



- The curvature of a straight line is 0.
- While in theory we could fit circles to every point, this is not done with real trajectory data because it is unstable.

Calculation of path offset / "curvature"

- Typically instead use a measure of how far the trajectory deviates from a straight line.
- Known as a measure of linearity (Atkeson & Hollerbach, 1985) or "path offset"
- $p = \frac{A}{D}$, where D is the distance of the straight line from start to end of the movement, A is the distance from the line to a point



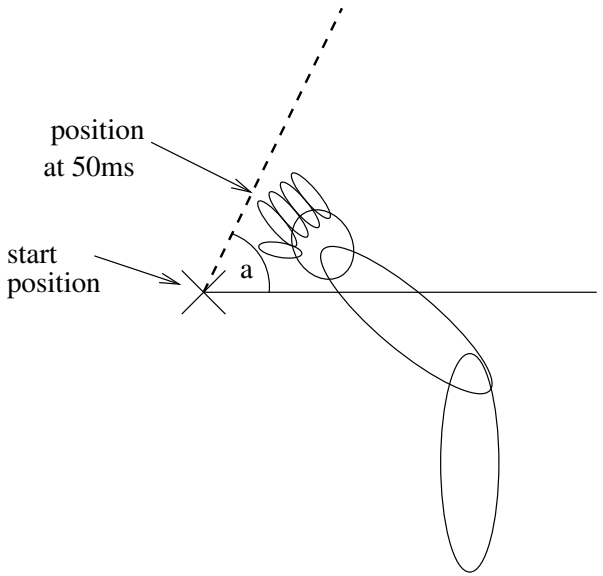
- Distance is given by $p = \frac{|(x_2 - x_1) \times (x_1 - x_0)|}{|x_2 - x_1|}$, where $\times$ is the cross product, and $||$ is the magnitude

Matlab: Calculation of path offset

```
start_end = ...
    [x1_cut{k}(end) - x1_cut{k}(1);
     y1_cut{k}(end) - y1_cut{k}(1);];
start_end_distance = sqrt(sum(start_end.^2));
for m=1:numel(x1_cut{k})
  thispoint_start = ...
    [x1_cut{k}(m) - x1_cut{k}(1);
     y1_cut{k}(m) - y1_cut{k}(1);];
     perp_distance(m) = sqrt(sum(cross...
       ([start_end;0],[thispoint_start;0])).^2) ...
        / sqrt(sum(start_end.^2));
end
pathoffset{k} = perp_distance ./ ...
  start_end_distance;
maxpathoffset(k) = max(pathoffset{k});
```

Calculation of initial angle

- Particularly when forced to move early, the "initial angle" can be a useful dependent variable to indicate the decision made at movement onset, which is not necessarily the final decision.
- As there is a spread of onset times, the initial angle can be also considered as a function of time (i.e., is there a trend in the initial angle early in the process?)
- Initial angle is calculated by taking a small time after movement onset (say 50ms), and calculating the angle of the line joining the starting position and the position at this time.



Matlab: Calculation of initial angle

We compare the position 50ms (10 samples) after the start of the movement with the position at the start of the movement. We calculate the angle using $tan^{-1}\left(\frac{y}{x}\right)$:

```
for k=1:10
   xdiff = x1_cut{k}(11) - x1_cut{k}(1);
   ydiff = y1_cut{k}(11) - y1_cut{k}(1);
   % calculate angle, convert to degrees
   angle(k) = atan2(ydiff,xdiff) * 180/pi;
end
```

In this example, we got negative values for 2 of the trials (i.e., they started going backwards). This should be dealt with somehow (e.g. ignore these trials, or choose a later starting point)

Submovements

- Arm movements are likely made up of *submovements* - discrete, stereotypical movements that are serially concatenated in time.
- Can observe evidence of submovements by bumps in the velocity profile
- This can be considered an extension of the heading direction.

Submovements

- Submovements are assumed to be (generally) approximately straight.
- The resultant movement may be curved due to the superposition of multiple submovements
- They are *discrete* rather than continuous at the planning stage, and planned in a feed-forward manner.
- This means that all the properties of a submovement are proscribed at the start of the movement (e.g. amplitude, direction, timing)

Submovements - minimum jerk (Flash & Hogan, 1985)

- Assumption: each submovement has a straight-line path, symmetrical bell-shaped velocity profile, that minimizes "mean squared jerk" cost:

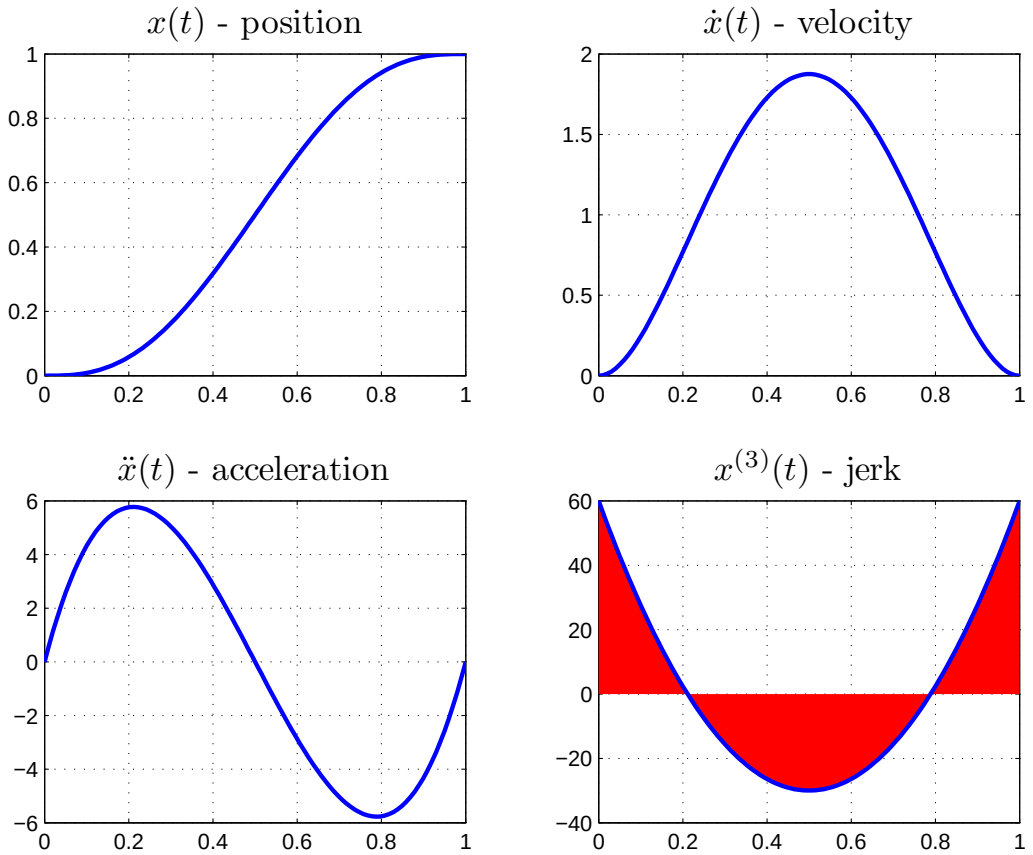
$$C = \int_0^1 \left(\frac{d^3 x}{d\tau^3} \right)^2 d\tau$$

- It can be shown that the analytic solution to this is:

$$\dot{x}(t) = D_x \left(30 \left(\frac{t - T_0}{D} \right)^4 - 60 \left(\frac{t - T_0}{D} \right)^3 + 30 \left(\frac{t - T_0}{D} \right)^2 \right)$$

where D_x is the amplitude, D is the duration, T_0 is the time of submovement onset, and $T_0 \leq t \leq T_0 + D$.

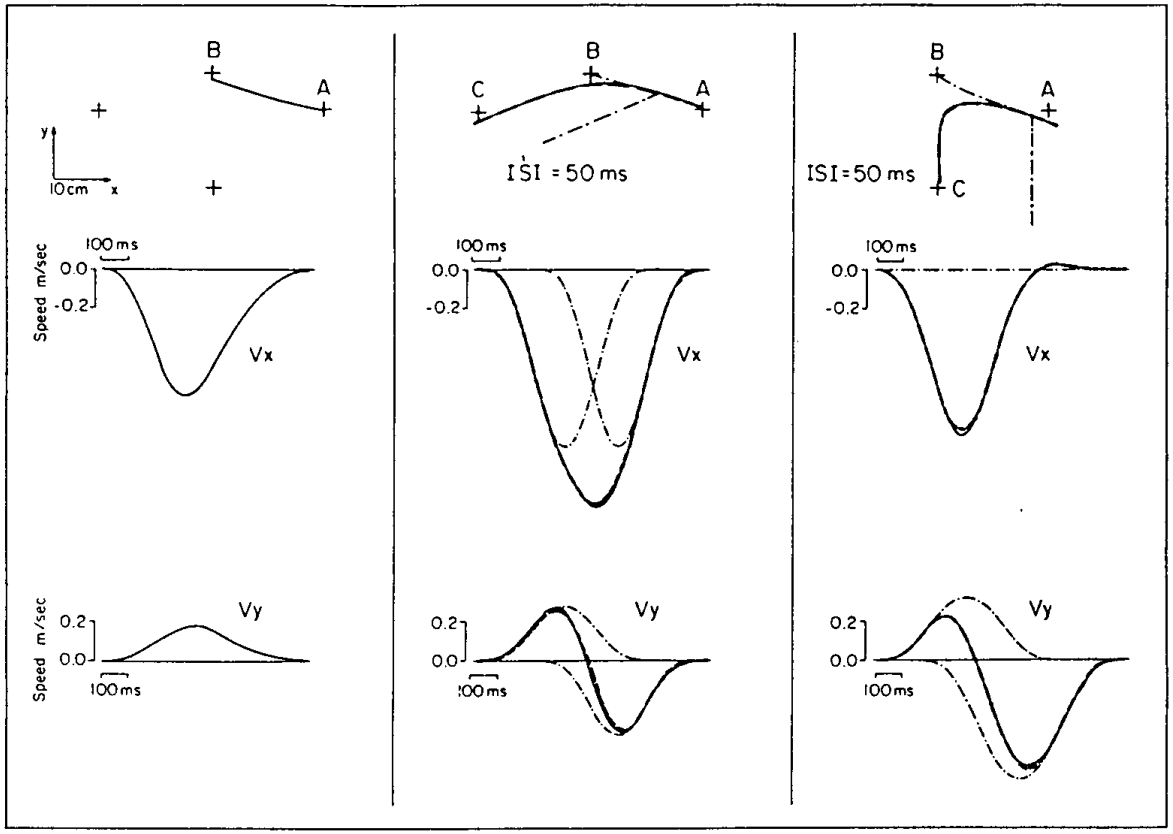
Submovements - minimum jerk (Flash & Hogan, 1985)



$$\dot{x}(t) = D_x \left(30 \left(\frac{t - T_0}{D} \right)^4 - 60 \left(\frac{t - T_0}{D} \right)^3 + 30 \left(\frac{t - T_0}{D} \right)^2 \right)$$

Submovement superposition - Flash & Henis (1991)

- Flash & Henis (1991) showed that when subjects had to change plans mid-flight (in a target-switch paradigm), the resultant movements were well modelled by a superposition of two movements: one from the start to the first target, and another from the first target to the second target.



Submovement decomposition

- This idea can be generalised to multiple submovements, with unknown start and end points
- Each movement is assumed to be the superposition of N submovements:

$$F(t) = \sum_{i=1}^{N} \begin{cases} 0 & t < T_{0i} \\ \dot{x}_i(t) & T_{0i} \leq t \leq T_{0i} + D_i \\ 0 & t > T_{0i} + D_i \end{cases}$$

- The aim is to find the "best" reconstruction of the movement using the superposition of one or more submovements

Submovement decomposition - example

- Usually we are interested in movements in a plane, so the submovements are fit in the two horizontal dimensions (x, y).
- This gives four parameters per submovement:
  - T_0: onset time
  - D: duration
  - D_x: amplitude (x)
  - D_y: amplitude (y)
- An error measure is used to define the quality of a reconstruction

$$\sum_t \frac{(F_x(t)-G_x(t))^2+(F_y(t)-G_y(t))^2+\left(F_v(t)-\sqrt{G_x(t)^2+G_y(t)^2}\right)^2}{2(G_x(t)^2+G_y(t)^2)}$$

where F_V is the tangential velocity, included to prevent unrealistic simultaneous positive and negative velocities.

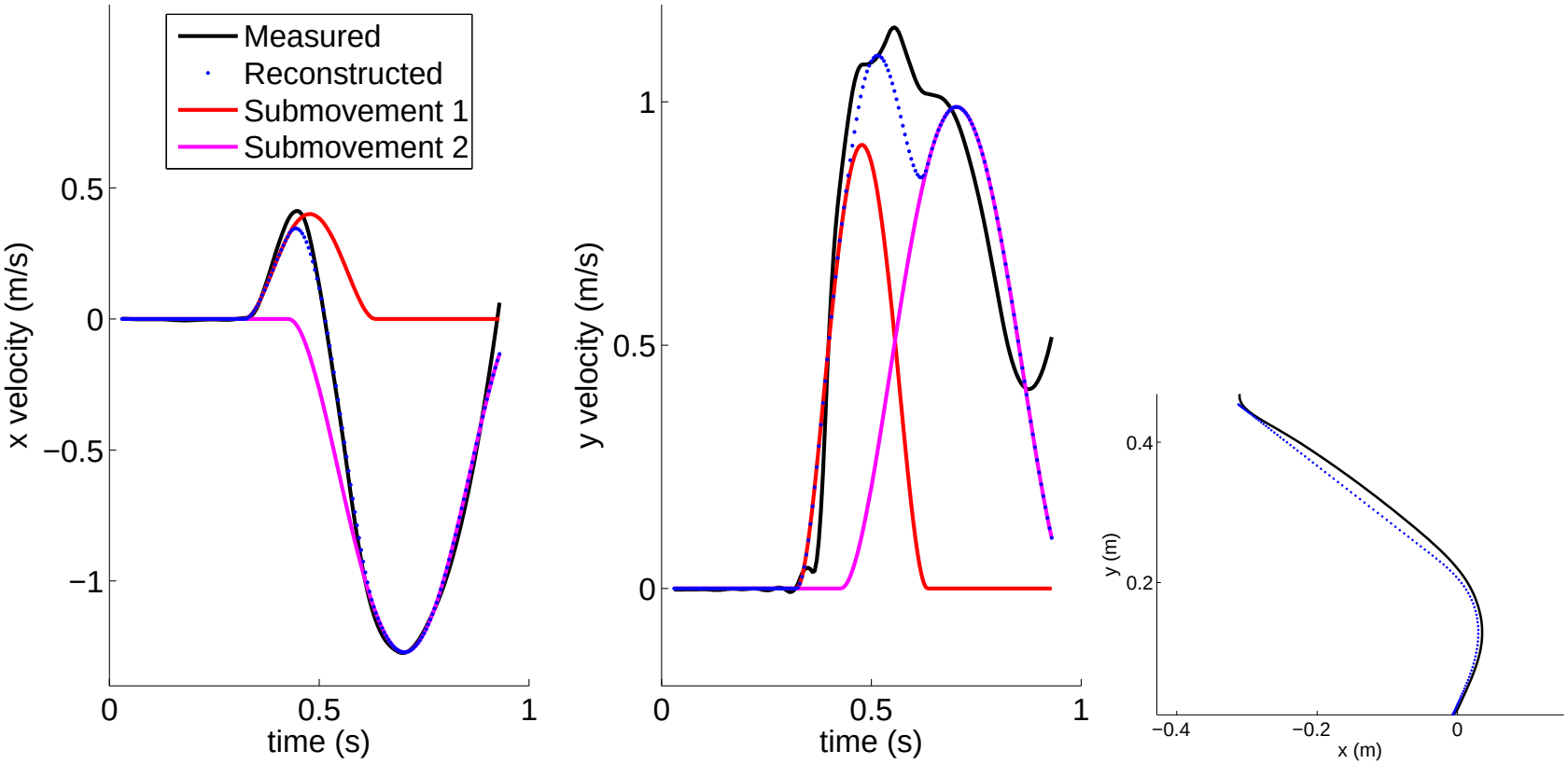
Submovement decomposition - example

- Use constrained nonlinear optimization (in Matlab) to find the parameters that minimize the reconstruction error. The constraints used (to make sure the submovements parameters are reasonable) are:

$$
\begin{aligned}
0 \leq T_0 \leq T_f - 0.167 \\
0.167 \leq D \leq 1.0 \\
-5 \leq D_x \leq 5 \\
0.1 \leq D_y \leq 5
\end{aligned}
$$

- Repeat from 10 different starting "guesses" to prevent local minima (Rohrer & Hogan, 2006)
- Repeat for 1 to 4 submovements (i.e. 4 to 16 parameters), select lowest number of submovement where the reconstruction error is < 0.03.

Submovement decomposition - example



- This method gives the onset times T_0 and amplitudes D_x of the submovements, which can be used in further analysis as a measure of *intent* at a particular time.

Summary

- Arm movements can provide us with a range of dependent measures useful in studies of cognitive processes:
- Arc length - how far the subject travelled (single value)
- Path offset - how curved the movement is (single value or time-varying)
- Initial angle / heading direction - measure of intent at start of movement (single value)
- Submovements - measure of intent throughout the movement (time-varying across trials)

More?

There are many other techniques that were not explored in these presentations. For example, the variation between trials can tell us:

- PCA / SVD / factor analysis: What is the structure of the variation between the variables?
- functional PCA: What variation is their between the trajectories when considered as a whole movement?
- synergies / redundancy measures: How do we structure variation to take advantage of redundancy?