9781405141697_4_005.qxd 11/16/09 4:59 PM Page 161

In the equation of a critically damped mass–spring system that defines the gesture's trajectory, time is not explicitly represented but is instead a consequence of specifying two parameters: the stiffness (or the spring/the articulator) and the target (change from equilibrium position). Changing the stiffness increases peak velocity but does not affect the magnitude. When the target is changed, then the peak velocity and magnitude change proportionally.

/kn/ vs /kl/ clusters The interval between the tongue-dorsum and tongue-tip closures is greater for /kn/ than for /kl/. In addition, /kn/ was shown to have a greater acoustic voice-onset time as well as a bigger (and proportionately faster) tongue-dorsum closing gesture compared with /kl/.

5.7 Questions

A. This question is about exploring whether the data show a relationship between the extent of jaw lowering and the first formant frequency in the first [a] component of [aɪ] of *Kneipe* and *Kneipier*, or of [aʊ] of *Claudia* and *Klausur*. In general, a more open vocal tract can be expected to be associated with both ~~by~~ F1 raising and ~~by~~ a lower jaw position (Lindblom & Sundberg 1971).

A.1 Calculate the first two formants of this database (`ema5`) and store these in a directory of your choice. Modify the template file in the manner described in Chapter 2 so that they are visible to the database `ema5`. Since this is a female speaker, use a nominal F1 of 600 Hz.

A.2 Assuming the existence of the segment list `k.s` of word-initial /k/ segments as defined at the beginning of this chapter and repeated below:

```
k.s = emu.query("ema5", "*", "Segment=k & Start(Word,
Segment)=1")
```

how could you use `emu.requery()` to make a segment list, `vow`, containing the diphthongs in the same words, given that these are positioned three segments to the right in relation to these word-initial /k/ segments? Once you have made `vow`, make a trackdata object `vow.fm`, for this segment list containing the formants. (You will first need to calculate the formants in the manner described in Chapter 3. Use a nominal F1 of 600 Hz.)

A.3 Make a vector of word labels, `word.l`, either from `k.s` or from the segment list `vow` you created in A.2. A table of the words should look like this:

```
table(word.l)
word.l
Claudia  Klausur  Kneipe  Kneipier
   5        5       5        5
```

A.4 Make a trackdata object, `vow.jaw`, containing vertical jaw-movement data (in track `jw_posz`) for the segment list you made in A.2.

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 162

A.5 The jaw height should show a trough in these diphthongs somewhere in the first component as the jaw lowers and the mouth opens. Use `trapply()` and `peakfun()` given below (repeated from section 5.5.2) to find the time at which the jaw height is at its lowest point in these diphthongs.

```
peakfun <- function(fr, maxtime=T)
{
if(maxtime) num = which.max(fr)
else num = which.min(fr)
tracktimes(fr)[num]
}
```

A.6 Verify that the times you have found in A.5 are appropriate by making an ensemble plot of `vow.jaw` color-coded for the diphthong type and synchronized at time of maximum jaw lowering found in A.5.

A.7 Using `dcut()` or otherwise, extract (i) the first formant frequency and (ii) the jaw height at these times. Store the first of these as `f1` and the second as `jaw`.

A.8 Plot F1 as a function of the jaw-height minimum showing the word labels at the corresponding points. This can be done either with:

```
plot(f1, jaw, type="n", xlab="F1 (Hz)", ylab="Jaw position
(mm)")
text(f1, jaw, word.l)
```

or with:

```
eplot(cbind(f1, jaw), word.l, dopoints=T, doellipse=F,
xlab="F1 (Hz)", ylab="Jaw position (mm)")
```

where `word.l` is the vector of word labels you make in A.3. To what extent would you say that there is a relationship between F1 and jaw height?

B. This question is about lip aperture and tongue movement in the closure of [p] of *Kneipe* and *Kneipier*.

B.1 Make a segment list, `p.s`, of the acoustic [p] closure (`p` at the `Segment` tier) of *Kneipe* or *Kneipier*.

B.2 Make a vector of word labels `pword.l`, parallel to the segment list in B.1.

B.3 Make two trackdata objects from `p.s`: (i) `p.ll`, of the vertical position of the lower lip (track `ll_posz`) and (ii) `p.ul`, of the vertical position of the upper lip (track `ul_posz`).

B.4 One way to approximate the lip aperture using EMA data is by subtracting the vertical lower-lip position from the vertical upper-lip position. Create a new trackdata object `p.ap` consisting of this difference between upper- and lower-lip position.

B.5 Use `peakfun()` from A.5 to create a vector, `p.mintime`, of the time at which the lip aperture in `p.ap` is a minimum.

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 163

B.6 Make an ensemble plot of the position of the lip-aperture as a function of time from `p.ap` color-coded for *Kneipe* vs *Kneipier* and synchronized at the time of minimum lip aperture.

B.7 How could you work out the mean proportional time in the acoustic closure at which the lip-aperture minimum occurs separately for *Kneipe* and *Kneipier*? For example, if the acoustic [p] closure extends from 10 to 20 ms and the time of the minimum lip aperture is 12 ms, then the proportional time is (12 – 10) / (20 – 10) = 0.2. The task is to find two mean proportional times, one for *Kneipe* and the other for *Kneipier*.

B.8 How would you expect the vertical and horizontal position of the tongue-mid (Figure 5.4) sensor to differ between the words in the closure of [p], given that the segment following the closure is [ɐ] in *Kneipe* and [j] or [ɪ] in *Kneipier*? Check your predictions by producing two ensemble plots over the interval of the acoustic [p] closure and color-coded for these words (i) of the vertical tongue-mid position and (ii) of the horizontal tongue-mid position synchronized at the time of the lip-aperture minimum obtained in B.7. (NB: the horizontal movement of the tongue-mid sensor is in `tm_posy`; and lower values obtained from the horizontal movement of sensors denote more forward, anterior positions towards the lips.)

C. The following question is concerned with the production differences between the diphthongs [aʊ] and [aɪ] in the first syllables respectively of *Klausur/Claudia* and *Kneipe/Kneipier*.

C.1 Make a boxplot of F2 (second formant frequency) at the time of the jaw-height minimum (see A.5) separately for each diphthong (i.e., there should be one boxplot for [aʊ] and one for [aɪ]).

C.2 Why might either tongue backing or a decreased lip aperture contribute to the tendency for F2 to be lower in [aʊ] at the time point in C.1? Make ellipse plots separately for the two diphthong categories with the horizontal position of the tongue-mid sensor on the *x*-axis and the lip aperture (as defined in B.4) on the *y*-axis, and with both of these parameters extracted at the time of the jaw-height minimum identified in C.1. To what extent might these data explain the lower F2 in [aʊ]?

D. This question is about the relationship between jaw height and duration in the first syllable of the words *Kneipe* and *Kneipier*.

D.1 *Kneipe* has primary lexical stress on the first syllable, but *Kneipier* has it on the ~~second~~. It is possible that these lexical-stress differences are associated with a greater duration in the first syllable of *Kneipe* than that of *Kneipier*. Make a segment list of these words between the time of maximum tongue-tip raising in /n/ and the time of minimum lip aperture in /p/. (The way to do this is to make use of the segment list of the `lower` annotations for these words at the `TT` tier, and then to replace its third column, i.e., the end times, with `p.mintime` obtained in B.5.) Before you make this change, use `emu.requery()` to obtain a parallel vector of word labels (so that each segment can be identified as *Kneipe* or *Kneipier*).

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 164

D.2 Calculate the mean duration of the interval defined by the segment list in D.1 separately for *Kneipe* and *Kneipier*.

D.3 If there is less time available for a phonetic segment or for a syllable to be produced, then one possibility according to Lindblom (1963) is that the target is undershot, i.e., not attained. If this production strategy is characteristic of the shorter first syllable in *Kneipier*, then how would you expect the jaw position as a function of time over this interval to differ between these two words? Check your predictions by making an ensemble plot of the position of the jaw height color-coded according to these two words.

D.4 Derive by central differencing from D.3 a trackdata object `vz` of the velocity of jaw height over this interval.

D.5 Use `emu.track()` to make a trackdata object of the *horizontal* position of the jaw (`jw_posy`) over this interval and derive the velocity of horizontal jaw movement, `vy`, from this trackdata object.

D.6 The *tangential velocity* in some analyses of EMA data is the rate of change of the Euclidean distance in the plane of vertical and horizontal movement which can be defined by:

$$(7) \quad \sqrt{v_z^2 + v_y^2}$$

in which v_z is the velocity of vertical movement (i.e., the trackdata object in D.4 for this example) and v_y the velocity of horizontal movement (the trackdata object in D.5). Derive the tangential velocity for these jaw-movement data and make an ensemble plot of the tangential velocity averaged and color-coded for the two word categories (i.e., one tangential velocity trajectory as a function of time averaged across all tokens of *Kneipe* and another superimposed tangential velocity trajectory averaged across all tokens of *Kneipier*).

5.8 Answers

A.2

```
vow = emu.requery(k.s, "Segment", "Segment", seq=3)
vow.fm = emu.track(vow, "fm")
```

A.3

```
word.l = emu.requery(vow, "Segment", "Word", j=T)
```

A.4

```
vow.jaw = emu.track(vow, "jw_posz")
```

A.5

```
jawmin = trapply(vow.jaw, peakfun, F, simplify=T)
```

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 165

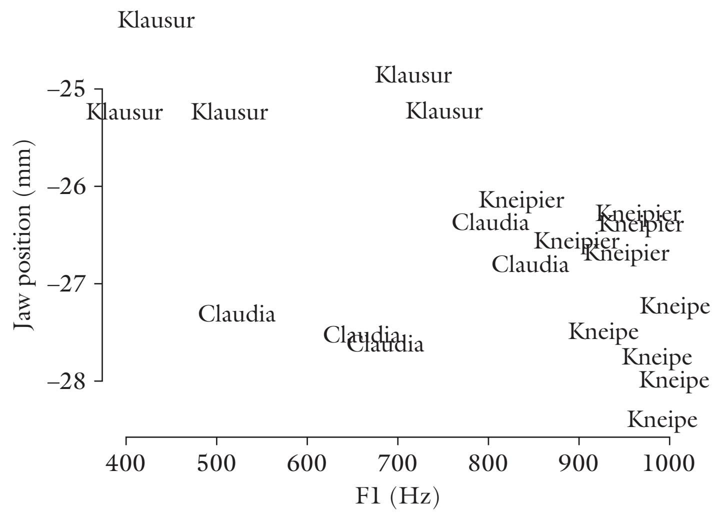

Figure 5.19 Jaw position as a function of the first formant frequency at the time of the lowest jaw position in two diphthongs showing the corresponding word label at the points.

A.6

```
dplot(vow.jaw, label(vow), offset=jawmin, prop=F)
```

A.7

```
f1 = dcut(vow.fm[,1], jawmin)
jaw = dcut(vow.jaw, jawmin)
```

A.8

Figure 5.19 shows that the variables are related: in very general terms, lower jaw positions are associated with higher F1 values. The (negative) correlation is, of course, far from perfect (in fact, −0.597 and significant, as given by `cor.test(f1, jaw)`).

B.1

```
p.s = emu.query("ema5", "*", "[Segment = p ^ Word=Kneipe |
Kneipier]")
```

B.2

```
pword.l = emu.requery(p.s, "Segment", "Word", j=T)
```

B.3

```
p.ll = emu.track(p.s, "ll_posz")
p.ul = emu.track(p.s, "ul_posz")
```

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 166

B.4

```
p.ap = p.ul - p.ll
```

B.5

```
p.mintime = trapply(p.ap, peakfun, F, simplify=T)
```

B.6

```
dplot(p.ap, pword.l, offset=p.mintime, prop=F)
```

B.7

```
prop = (p.mintime-start(p.s))/dur(p.s)
tapply(prop, pword.l, mean)
Kneipe  Kneipier
0.3429   0.2607
```

B.8 You would expect the tongue mid position to be higher and fronter in *Kneipier* due to the influence of the preceding and following palatal segments and this is supported by the evidence in Figure 5.20.[11]

```
p.tmvertical = emu.track(p.s, "tm_posz")
p.tmhorz = emu.track(p.s, "tm_posy")
par(mfrow=c(1,2))
dplot(p.tmvertical, pword.l, offset=p.mintime, prop=F,
ylab="Vertical position (mm)", xlab="Time (ms)", legend=F)
```

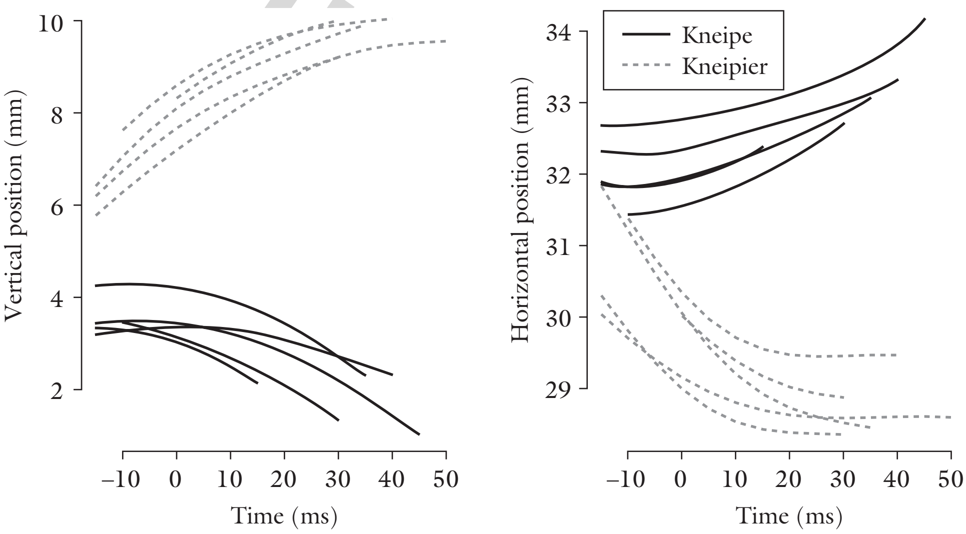

Figure 5.20 Vertical (left) and horizontal (right) position of the tongue-mid sensor over the interval of the acoustic closure of [p] synchronized at the time of the lip-aperture minimum in *Kneipe* (black) and *Kneipier* (dashed, gray).

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 167


```
dplot(p.tmhorz, pword.l, offset=p.mintime, prop=F,
ylab="Horizontal position (mm)", xlab="Time (ms)",
legend="topleft")
```

C.1

```
f2jaw = dcut(vow.fm[,2], jawmin)
boxplot(f2jaw ~ label(vow))
```

C.2

```
vow.tmhor = emu.track(vow, "tm_posy")
vow.ul = emu.track(vow, "ul_posz")
vow.ll = emu.track(vow, "ll_posz")
vow.ap = vow.ul - vow.ll
tongue = dcut(vow.tmhor, jawmin)
ap = dcut(vow.ap, jawmin)
d = cbind(tongue, ap)
eplot(d, label(vow), dopoints=T, xlab="Horizontal tongue
position (mm)", ylab="Lip aperture (mm)")
```

Overall, there is evidence from Figure 5.22 of a more retracted tongue position or decreased lip aperture at the jaw-height minimum in [aʊ], which could be due to the phonetically back and rounded second component of this diphthong. Either of these factors is likely to be associated with the observed lower F2 in Figure 5.21. In addition, Figure 5.22 shows that [aʊ] seems to cluster into two groups and these are probably tokens from the two words *Claudia* and *Klausur*. Thus, the data show either that the lip aperture in [aʊ] is less than in [aɪ] (for the cluster of points

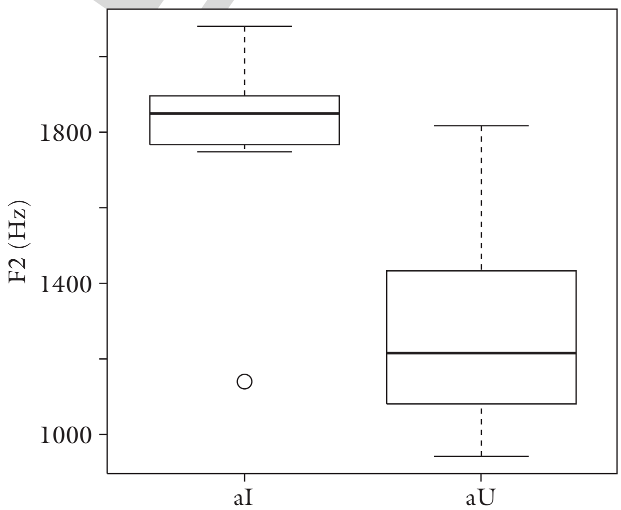

Figure 5.21 Boxplot of F2 in [aɪ] and [aʊ] at the time of the lowest position of the jaw in these diphthongs.

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 168

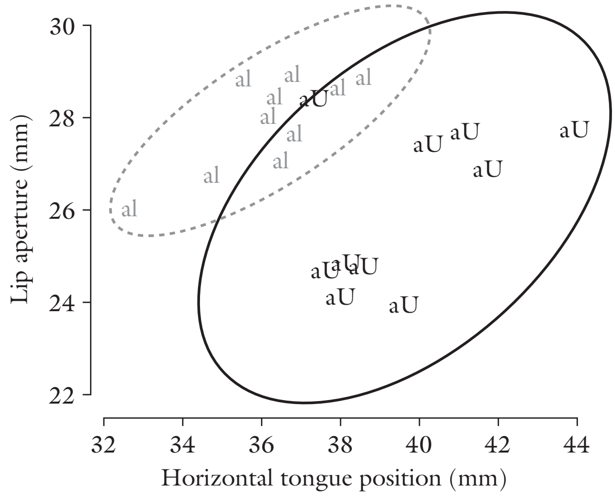

Figure 5.22 95% confidence ellipses for two diphthongs in the plane of the horizontal position of the tongue-mid sensor and lip aperture with both parameters extracted at the time of the lowest vertical jaw position in the diphthongs. Lower values on the *x*-axis correspond to positions nearer to the lips. The lip aperture is defined as the difference in position between the upper- and lower-lip sensors.

around 24 mm on the *y*-axis) or that the tongue is retracted (for the points around 27–8 mm on the *y*-axis) relative to [aI] (but not both).

D.1

```
syll.s = emu.query("ema5", "*", "[TT = lower ^ Word = Kneipe |
Kneipier]")
word.l = emu.requery(syll.s, "TT", "Word", j=T)
syll.s[,3] = p.mintime
```

D.2

```
tapply(dur(syll.s), word.l, mean)
 Kneipe   Kneipier
201.6592  161.0630
```

Yes: the first syllable of *Kneipier*, where syllable is defined as the interval between tongue-tip raising in /n/ and the point of minimum lip aperture in /p/, is some 40 ms less than that of *Kneipe*.

D.3

```
syll.jaw = emu.track(syll.s, "jw_posz")
dplot(syll.jaw, word.l, ylab="Position (mm)")
```

There does seem to be evidence for target undershoot of vertical jaw movement, as Figure 5.23 suggests.

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 169

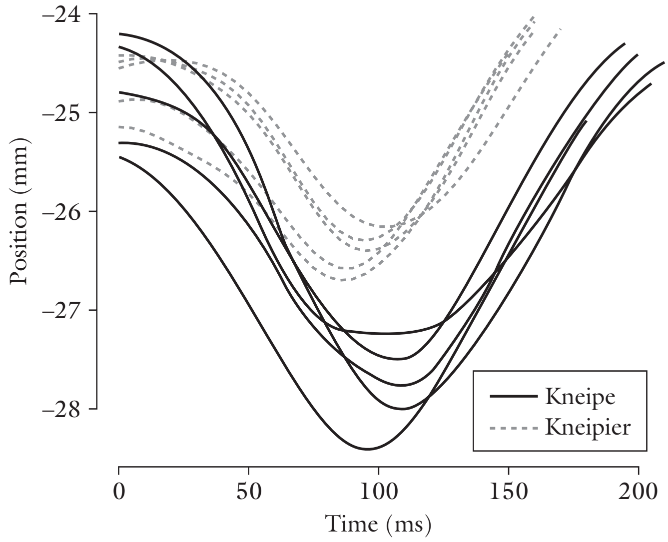

Figure 5.23 Jaw-height trajectories over the interval between the maximum point of tongue-tip raising in /n/ and the minimum jaw aperture in /p/ for *Kneipe* (solid) and *Kneipier* (gray, dashed).

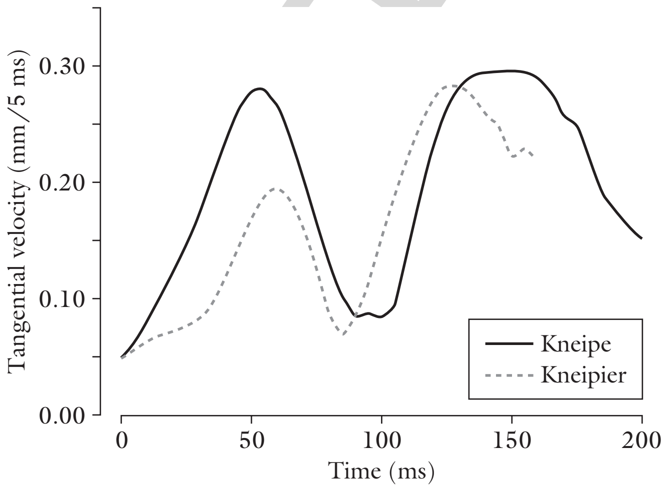

Figure 5.24 Tangential velocity of jaw movement between the time of maximum tongue-tip raising in /n/ and the lip-aperture minimum in /p/ averaged separately in *Kneipe* (solid) and *Kneipier* (dashed, gray).

D.4

```
vz = trapply(syll.jaw, cendiff, returntrack=T)
```

D.5

```
syll.jawx = emu.track(syll.s, "jw_posy")
vy = trapply(syll.jawx, cendiff, returntrack=T)
```

UNCORRECTED PROOFS

9781405141697_4_005.qxd 11/16/09 4:59 PM Page 170

D.6

```
tang = sqrt(vz^2 + vy^2)
dplot(tang, word.l, average=T, ylab="Tangential velocity
(mm/5 ms)", xlab="Time (ms)")
```

Notes

1 See www.phonetik.uni-muenchen.de/~hoole/5d_examples.html for some examples.

2 The script is `mat2ssff.m` and is available in the top directory of the downloadable `ema5` database.

3 The term speech frame will be used henceforth for these data to distinguish them from a type of object in R known as a data frame.

4 Both trackdata objects must be derived from the same segment list for `cbind()` to be used in this way.

5 For the sake of brevity, I will not always include the various options (see `help(par)`) that can be included in the plotting function and that were needed for camera-ready black-and-white images in this book. Thus Figure 5.10 was actually produced as follows:

```
par(mfrow=c(1,2)); lwd=lty=c(1,2); col=c(1, "slategray")
xlab = "Time (ms)"; ylab="Vertical tongue tip position (mm)"
dplot(tip.tt, son.lab, prop=F, offset=end(tbraise.s), bty="n",
ylab=ylab, xlab=xlab, col=col, lwd=lwd, lty=lty)
dplot(tip.tt, son.lab, prop=F, offset=end(tbraise.s), average=T,
bty="n", xlab=xlab, col=col, lwd=lwd, lty=lty, legend=F)
```

6 The velocity signals are also available in the same directory to which the `ema5` database was downloaded, although they have not been incorporated into the template file. They could be used to find peaks and troughs in the movement signals, as described in section 5.5.2.

7 This is because three-point central differencing is the average of the forward and backward difference. For example, suppose there is a signal `x = c(1, 3, 4)`. At time point 2, the forward difference is `x[2] - x[1]` and the backward difference is `x[3] - x[2]`. The average of these is `0.5 * (x[2] - x[1] + x[3] - x[2])` or `0.5 * (x[3]-x[1])` or 1.5. At time point 1, the three-point central difference would therefore be `0.5 * (x[2] - x[0])`. But this gives `numeric(0)` or `NA` because `x[0]` is undefined (there is no sample value preceding `x[1]`). At time point 3, the output of `0.5 * (x[4]-x[2])` is `NA` for the same reason that `x[4]` is undefined (the signal is of length 3). Consequently, `filter(x, c(0.5, 0, -0.5))` gives `NA 1.5 NA`.

8 If you want to convert this to cm/s, then divide by 5 to get to ms, multiply by 1,000 to get to seconds, and divide by 10 to get to cm: the combined effect of these operations is that the trackdata object has to be multiplied by 20, which can be done with `body.tbd = body.tbd * 20`.

9 My thanks to Elliot Saltzman for assistance in relating (3) to (4).

10 The units are not important for this example but, in fact, if the sampling frequency `fs` is defined, then the natural frequency is `w * fs/(2 * pi)` Hz (see e.g., Harrington & Cassidy 1999, p. 160). Thus, if the default of 100 points in the `critdamp()` function is assumed to take up 1 s (i.e., `fs = 100` Hz), then the default `w = 0.05` has a frequency in Hz of `0.05 * 100/(2 * pi)`, i.e., just under 0.8 Hz.

11 The following additional plotting parameters were used: `col=c(1, "slategray")`, `lwd=c(1,2)`, `lty=c(1,2)`, `bty="n"`.

UNCORRECTED PROOFS