Chapter 8
Registration: Aligning Features for Samples of Curves

This chapter presents two methods for separating phase variation from amplitude variation in functional data: landmark and continuous registration. We mentioned this problem in Section 1.1.1. We saw in the height acceleration curves in Figure 1.2 that the age of the pubertal growth spurt varies from girl to girl; this is *phase variation*. In addition, the intensity of the pubertal growth spurt also varies; this is *amplitude variation*. Landmark registration aligns features that are visible in all curves by estimating a strictly increasing nonlinear transformation of time that takes all the times of a given feature into a common value. Continuous registration uses the entire curve rather than specified features and can provide a more complete curve alignment. The chapter also describes a decomposition technique that permits the expression of the amount of phase variation in a sample of functional variation as a proportion of total variation.

8.1 Amplitude and Phase Variation

Figure 1.2 presented the problem that curve registration is designed to solve. This figure is reproduced in the top panel of Figure 8.1 along with a solution in the bottom panel. In both panels, the dashed line indicates the mean of these ten growth acceleration curves. In the top panel, this mean curve is unlike any of the individual curves in that the duration of the mean pubertal growth is longer than it should be and the drop in acceleration is not nearly as steep as even the shallowest of the individual curves. These aberrations are due to the ten girls not being in the same phase of growth at around 10 to 12 years of age. We see from the figure that peak growth acceleration occurs around age 10.5 for many girls, but this occurred before age 8 for one girl and after age 13 for another. Similarly, the maximum pubertal growth rate occurs where the acceleration drops to zero following the maximum pubertal acceleration. This occurs before age 10 for two girls and around age 14 for another, averaging around 11.7 years of age. If we average the growth accelerations at that age, one girl has not yet begun her pubertal growth spurt, three others are

J.O. Ramsay et al., *Functional Data Analysis with R and MATLAB*, Use R,
DOI: 10.1007/978-0-387-98185-7_8,

at or just past their peak acceleration, and the rest are beyond their peak pubertal growth rate with negative acceleration. This analysis should make it fairly easy to understand why the average of these acceleration curves displays an image that is very different from any of the individual curves.

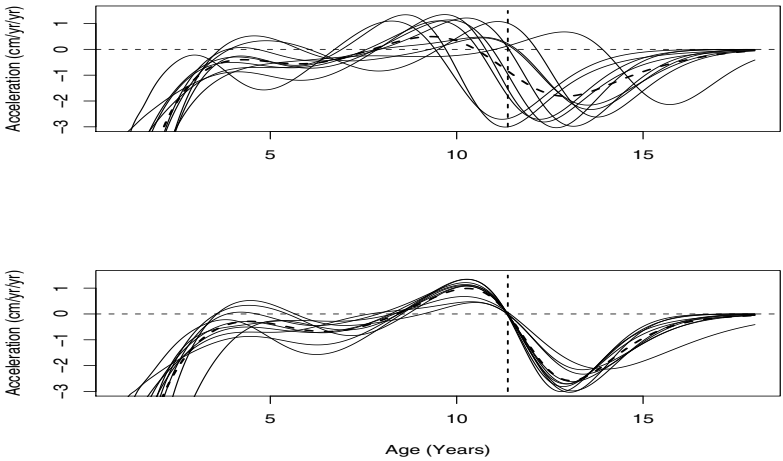

Fig. 8.1 The top panel reproduces the second derivatives of the growth curves shown in Figure 1.2. The landmark–registered curves corresponding to these are shown in the bottom panel, where the single landmark was the crossing of zero in the middle of the mean pubertal growth spurt. The dashed line in each panel indicates the mean curve for the curves in that panel.

The bottom panel in Figure 8.1 uses landmark registration to align these curves so the post–spurt accelerations for all girls cross zero at the same time. Then when we average the curves, we get a much more realistic representation of the typical pubertal growth spurt, at least among the girls in this study.

Functions can vary in both phase and amplitude, as illustrated schematically in Figure 8.2. *Phase variation* is illustrated in the top panel as a variation in the location of curve features along the horizontal axis, as opposed to *amplitude variation*, shown in the bottom panel as the size of these curves. The mean curve in the top panel, shown as a dashed curve, does not resemble any curve; it has less amplitude variation, but its horizontal extent is greater than that of any single curve. The mean has, effectively, borrowed from amplitude to accommodate phase. Moreover, if we carry out a functional principal components analysis of the curves in each panel, we find in the top panel that the first three principal components account for 55%, 39% and 5%, of the variation. On the other hand, the same analysis of the amplitude-varying curves requires a single principal component to account for 100% of the variation. Like the mean and principal components, most statistical methods when

translated into the functional domain are designed to model purely amplitude variation.

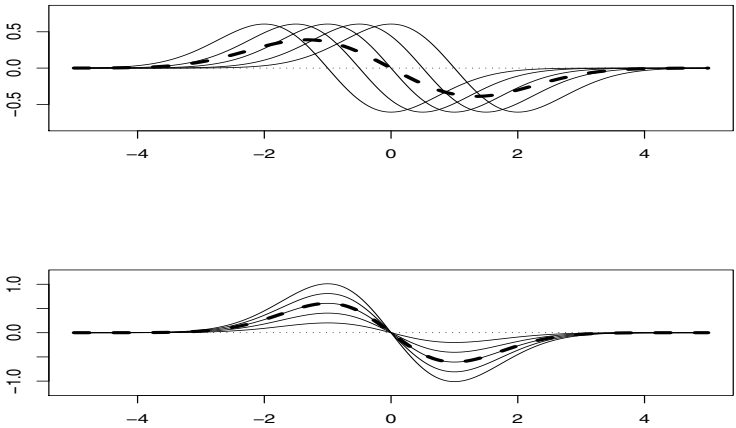

Fig. 8.2 The top panel shows five curves varying only in phase. The bottom panel shows five curves varying only in amplitude. The dashed line in each panel indicates the mean of the five curves. This curve in the bottom panel is superimposed exactly on the central curve.

There is physiological *growth time* that unrolls at different rates from child to child relative to *clock time*. In terms of growth time, all girls experience puberty at the same age, with the peak growth rate (zero acceleration) occurring at about 11.7 years of age for the Berkeley sample. If we want a reasonable sense of amplitude variation, we must consider it with this *growth time* frame of reference. Growth time itself is an elastic medium that can vary randomly from girl to girl when viewed relative to clock time, and functional variation has the potential to be *bivariate*, with variation in both the *range* and *domain* of a function.

8.2 Time-Warping Functions and Registration

We can remove phase variation from the growth data if we can estimate a *time-warping function* $h_i(t)$ that transforms growth time t to clock time for child i. For example, we can require that $h_i(11.7) = t_i$ for all girls, where 11.7 years is the average time at which the Berkeley girls reached their midpubertal spurt (PGS) and t_i is the clock age at which the ith girl reached this event. If, at any time t, $h_i(t) < t$, we may say that the girl is growing faster than average at that clock time but slower than average if $h_i(t) > t$. This is illustrated in Figure 8.3, where the growth

acceleration curves for the earliest and latest of the first ten girls are shown in the left panels and their corresponding time-warping functions in the right panels.

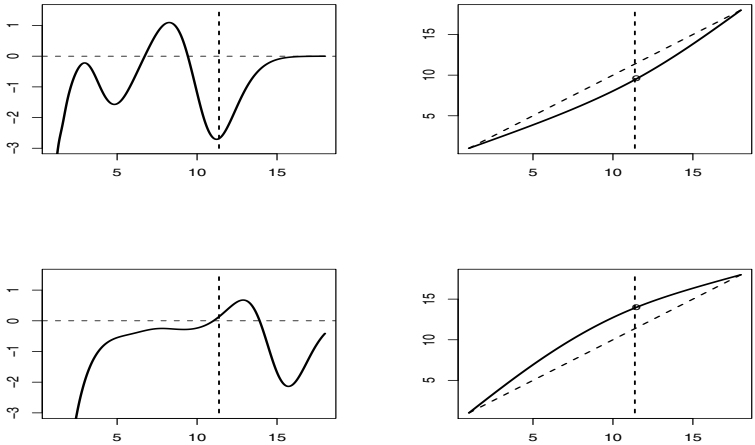

Fig. 8.3 The top panels show the growth acceleration curve on the left and the corresponding time-warping function $h(t)$ on the right for the girl among the first ten in the Berkeley growth study with the earliest pubertal growth spurt. The corresponding plots for the girl with the latest growth spurt are in the bottom two panels. The middle of the growth spurt is shown as the vertical dashed line in all panels.

Time-warping functions must, of course, be strictly increasing; we cannot allow time to go backwards in either frame of reference. Time-warping functions must also be smooth in the sense of being differentiable up to at least what applies to the curves being registered. If the curves are observed over a common interval $[0,T]$, the time-warping functions must often satisfy the constraints $h(0) = 0$ and $h(T) = T$, but it may be that varying intervals $[0,T_i]$ may each be transformed to a common interval $[0,T]$. In the special case of periodic curves, such as average temperature and precipitation profiles, we may also allow a constant shift $h_i(t) = t_i + \delta_i$.

The *registered* height functions are $x_i^*(t) = x_i[h_i^{-1}(t)]$, where the *aligning function* $h^{-1}(t)$ satisfies the following equation:

$$h^{-1}[h(t)] = t. \tag{8.1}$$

This is the *functional inverse* of $h(t)$.

For example, since at time $h_i(t_0)$ girl i is in the middle of her pubertal growth spurt, and since in her registered time $h_i^{-1}[h_i(t_0)] = t_0$, she and all the other children will experience puberty at time t_0 in terms of registered or "growth" time. In particular, if $h_i(t_0) < t_0$ for a girl i reaching puberty early, then aligning function $h_i^{-1}(t)$

effectively slows down or stretches out her clock time so as to conform with growth time.

8.3 Landmark Registration with Function landmarkreg

The simplest curve alignment procedure is *landmark* registration. A landmark is a feature with a location that is clearly identifiable in all curves. Landmarks may be the locations of minima, maxima or crossings of zero, and we see three such landmarks in each curve Figure 8.2. We align the curves by transforming t for each curve so that landmark locations are the same for all curves.

For the bottom panel in Figure 1.2, we used a single landmark t_i being the age for girl i at which her acceleration curve crossed 0 with a negative slope during the pubertal growth spurt. Also, let us define t_0 as a time specified for the middle of the *average* pubertal growth spurt, such as 11.7 years of age for the Berkeley growth study girls. Then we specify time-warping functions h_i by fitting a smooth function to the three points $(1,1),(t_0,t_i)$, and $(18,18)$. This function should be as differentiable as the curves themselves, and in this case could be simply the unique parabola passing through the three points $(1,1),(t_0,t_i)$ and $(18,18)$, which is what is shown in the right panels of Figure 8.3.

The code for smoothing the Berkeley female growth data is found in Section 5.4.2.2. From these smooths, we can compute the unregistered accelerations functions and their mean function by the commands

```
accelfdUN     = deriv.fd(hgtfhatfd, 2)
accelmeanfdUN = mean(accelfdUN)
```

This code allows you to select the age of the center of the pubertal growth spurt for each girl, applying the R function `locator()` to plots of a functional data object `accfd` that contains estimated acceleration curves.

```
PGSctr  = rep(0,10)
agefine = seq(1,18,len=101)
par(mfrow=c(1,1), ask=TRUE)
for (icase in 1:54) {
    accveci = predict(accelfdUN[icase], agefine)
    plot(agefine,accveci,"l", ylim=c(-6,4),
         xlab="Year", ylab="Height Accel.",
         main=paste("Case",icase))
    lines(c(1,18),c(0,0),lty=2)
    PGSctr[icase] = locator(1)$x
}
PGSctrmean = mean(PGSctr)
```

We don't need much flexibility in the function fitting the three points defining each warping function, so we define four order three spline basis functions and apply a very light level of smoothing in these commands:

```
wbasisLM = create.bspline.basis(c(1,18), 4, 3,
                                c(1,PGSctrmean,18))
WfdLM    = fd(matrix(0,4,1),wbasisLM)
WfdParLM = fdPar(WfdLM,1,1e-12)
```

The landmark registration using function `landmarkreg` along with the extraction of the registered acceleration functions, warping function and w-functions is achieved by the commands

```
regListLM = landmarkreg(accelfdUN, PGSctr,
                        PGSctrmean, WfdParLM, TRUE)
accelfdLM     = regListLM$regfd
accelmeanfdLM = mean(accelfdLM)
warpfdLM      = regList$warpfd
WfdLM         = regList$Wfd
```

The final logical argument value `TRUE` requires the warping functions h_i to themselves be strictly monotone functions.

The bottom panel of Figure 8.1 displays the same ten female growth acceleration curves after registering to the middle of the pubertal growth spurt. We see that the curves are now exactly aligned at the mean PGS (pubertal growth spurt) age, but that there is still some misalignment for the maximum and minimum acceleration ages. Our eye is now drawn to the curve for girl seven, whose acceleration minimum is substantially later than the others and who has still not reached zero acceleration by age 18. The long period of near zero acceleration for girl four prior to puberty also stands out as unusual. The mean curve is now much more satisfactory as a summary of the typical shape of growth acceleration curves, and in particular is nicely placed in the middle of the curves for the entire pubertal growth spurt period.

8.4 Continuous Registration with Function `register.fd`

We may need registration methods that use the entire curves rather than their values at specified points. A number of such methods have been developed, and the problem continues to be actively researched. Landmark registration is usually a good first step, but we need a more refined registration process if landmarks are not visible in all curves. For example, many but not all female growth acceleration curves have at least one peak prior to the pubertal growth spurt that might be considered a landmark. Even when landmarks are clear, identifying their timing may involve tedious interactive graphical procedures, and we might prefer a fully automatic method. Finally, as we saw in Figure 8.1, landmark registration using just a few landmarks can still leave aspects of the curves unregistered at other locations.

Here we illustrate the use of function `register.fd` to further improve the acceleration curves that have already been registered using function `landmarkreg`. The idea behind this method is that if an arbitrary sample registered curve $x[h(t)]$ and target curve $x_0(t)$ differ only in terms of amplitude variation, then their values

will tend to be proportional to one another across the range of t-values. That is, if we were to plot the values of the registered curve against the target curve, we would see something approaching a straight line tending to pass through the origin, although not necessarily at angle 45 degrees with respect to the axes of the plot. If this is true, then a principal components analysis of the following order two matrix $\mathbf{T}(h)$ of integrated products of these values should reveal essentially one component, and the smallest eigenvalue should be near 0:

$$\mathbf{C}(h) = \begin{bmatrix} \int \{x_0(t)\}^2\,dt & \int x_0(t)x[h(t)]\,dt \\ \int x_0(t)x[h(t)]\,dt & \int \{x[h(t)]\}^2\,dt \end{bmatrix}. \tag{8.2}$$

According to this rationale, then, estimating h so as to minimize the smallest eigenvalue of $\mathbf{C}(h)$ should do the trick. This is exactly what `register.fd` does for each curve in the sample.

If the curves are multivariate, such as coordinates of a moving point, then what is minimized is the sum of the smallest eigenvalues across the components of the curve vectors. We recall, too, that curve $x(t)$ may in fact be a derivative of the curve used to smooth the data.

In the following code, we use a more powerful basis than we used in Chapter 5, combined with a roughness penalty, for defining the functions $W(t)$ to estimate strictly monotone functions. Because the continuous registration process requires iterative numerical optimization techniques, we have to supply starting values for the coefficients defining the functions W. We do this by using zeros in defining the initial functional data object `Wfd0CR`.

```
wbasisCR = create.bspline.basis(c(1,18), 15, 5)
Wfd0CR   = fd(matrix(0,15,54),wbasisCR)
WfdParCR = fdPar(Wfd0CR, 1, 1)
regList  = register.fd(mean(accelfdLM),
                       accelfdLM, WfdfdPar)
accelfdCR = regList$regfd
warpfdCR  = regList$warpfd
WfdCR     = regList$Wfd
```

Figure 8.4 shows that the continuously registered height acceleration curves are now aligned over the entire PGS relative to the landmark–registered curves, although we have sacrificed a small amount of alignment of the zero–crossings of these curves. Figure 8.5 shows the impacts of the two types of registrations, and we see that, while both registrations provide average curves with maximum and minimum values much more typical of the individual curves, as well as the width of the PGS, the final continuously registered mean curve does a better job in the mid-spurt period centered on five years of age.

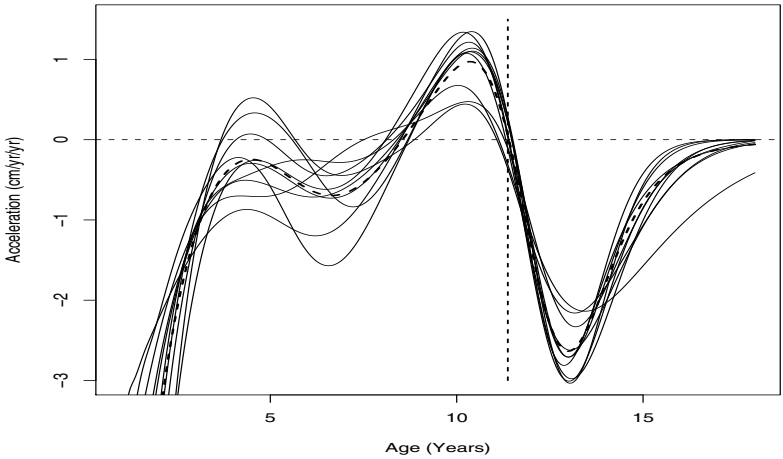

Fig. 8.4 The continuous registration of the landmark–registered height acceleration curves in Figure 8.1. The vertical dashed line indicates the target landmark age used in the landmark registration.

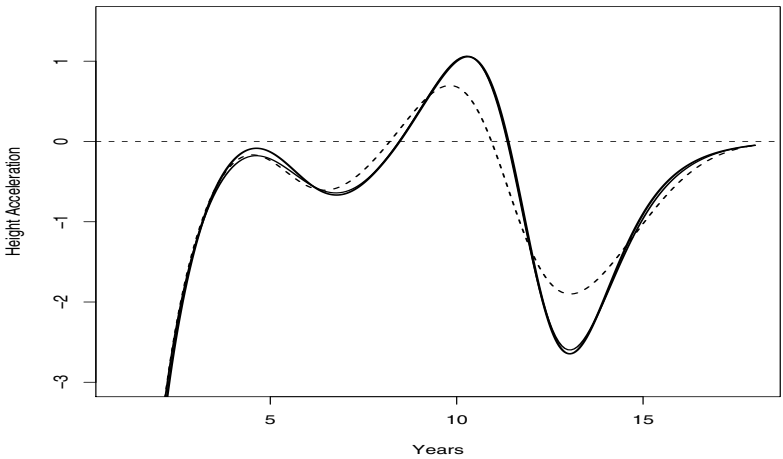

Fig. 8.5 The mean of the continuously registered acceleration curves is shown as a heavy solid line, while that of the landmark-registered curves is a light solid line. The light dashed line is the mean of the unregistered curves

8.5 A Decomposition into Amplitude and Phase Sums of Squares

Kneip and Ramsay (2008) developed a useful way of quantifying the amount of these two types of variation by comparing results for a sample of N functional observations before and after registration. The notation x_i stands for the unregistered version of the ith observation, y_i for its registered counterpart and h_i for associated warping function. The sample means of the unregistered and registered samples are $\bar{x}$ and $\bar{y}$, respectively.

The *total mean square error* is defined as

$$MSE_{total} = N^{-1} \sum_{i}^{N} \int [x_i(t) - \bar{x}(t)]^2 \, dt. \tag{8.3}$$

We define the constant C_R as

$$C_R = 1 + \frac{N^{-1} \sum_i^N \int [Dh_i(t) - N^{-1} \sum_i^N Dh_i(t)][y_i^2(t) - N^{-1} \sum_i^N y_i^2(t)] \, dt}{N^{-1} \sum_i^N \int y_i^2(t) \, dt}. \tag{8.4}$$

The structure of C_R indicates that $C_R - 1$ is related to the covariation between the deformation functions Dh_i and the squared registered functions y_i^2. When these two sets of functions are independent, the number of the ratio is 0 and $C_R = 1$.

The measures of amplitude and phase mean square error are, respectively,

$$\begin{aligned} MSE_{amp} &= C_R N^{-1} \sum_{i}^{N} \int [y_i(t) - \bar{y}(t)]^2 \, dt \\ MSE_{phase} &= C_R \int \bar{y}^2(t) \, dt - \int \bar{x}^2(t) \, dt. \end{aligned} \tag{8.5}$$

It can be shown that, defined in this way, $MSE_{total} = MSE_{amp} + MSE_{phase}$.

The interpretation of this decomposition is as follows. If we have registered our functions well, then the registered functions y_i will have higher and sharper peaks and valleys, since the main effect of mixing phase variation with amplitude variation is to smear variation over a wider range of t values, as we saw in Figure 1.2 and Figure 8.2. Consequently, the first term in MSE_{phase} will exceed the second and is a measure of how much phase variation has been removed from the y_i's by registration. On the other hand, MSE_{amp} is now a measure of pure amplitude variation to the extent that the registration has been successful. The decomposition does depend on the success of the registration step, however, since it is possible in principle for MSE_{phase} to be negative.

From this decomposition we can get a useful squared multiple correlation index of the proportion of the total variation due to phase:

$$R^2 = \frac{MSE_{phase}}{MSE_{total}}. \tag{8.6}$$

We applied used the decomposition to compare the unregistered acceleration curves with their landmark registered counterparts. Because the variation in the acceleration curves is far greater in the first few years than for the remaining years, which is the variation visible in Figure 8.1, we elected to use the decomposition over only the years from three to eighteen years. The function `AmpPhaseDecomp` returns a list with components `MS.amp`, `MS.pha`, `RSQR` and `C`. The commands

```
AmpPhasList = AmpPhaseDecomp(accffd, accregfdLM,
                             warpfd, c(3,18))
RSQR        = AmpPhasList$RSQR
```

after landmark registration of the growth acceleration curves yields the value $R^2 = 0.70$. That is, nearly 70% of the variation in acceleration over this period is due to phase.

On the other hand, if we use this decomposition to compare the landmark registered curves in Figure 8.1 with those for the continuously registered curves in Figure 8.4, we get the value –0.06. What does this mean? It means that "registered" is a rather fuzzy qualifier in the sense that we can define the registration process in different ways and get different answers. A careful comparison of the two figures might suggest that the landmark registration process has over–registered the pubertal growth spurt at the expense of earlier growth spurts visible in several of the curves. Or, alternatively, if our main concern is getting pubertal growth right, then the continuous registration process has deregistered the landmark–registered curves by about 6%.

8.6 Registering the Chinese Handwriting Data

The handwriting data discussed in Section 1.2.2 consisted of the writing of "statistics" in simplified Chinese 50 times. The average time of writing was six seconds, with the X-, Y- and Z-coordinates of the pen position being recorded 400 times per second. The handwriting involves 50 strokes, corresponding to about eight strokes per second, or 120 milliseconds per stroke. The processing of these data was done entirely in Matlab, and is too complex to describe in detail here.

The registration phase was carried out in two steps, as was the case for the growth data. In the first phase, three clear landmarks were visible in all curves in the vertical Z-coordinate corresponding to points where the pen was lifted from the paper. These were used in a preliminary landmark registration process for the Z-coordinate alone. The decomposition described above indicated that 66.6% of the variation in Z was due to phase. The warping functions were applied to the X- and Y-coordinates as well, and the decompositions indicated percentages of phase variation of 0% and 75%, respectively. This suggests that most of the phase variation in movement off the writing plane was associated with motion that was also vertical in the writing plane.

In a second registration phase, the scalar *tangential accelerations*,

$$TA_i(t) = \sqrt{D^2X_i(t) + D^2Y_i(t)},$$

of the tip of the pen along the writing path were registered using continuous registration. This corresponded to 48% of the variation in the landmark-registered tangential accelerations being due to phase. Figure 8.6 plots the tangential acceleration for all 50 replications before and after applying this two–stage registration procedure. After alignment, we see the remarkably small amount of amplitude variation in many of the acceleration peaks, and we also see how evenly spaced in time these peaks are. The pen hits acceleration of 30 meters/sec/sec, or three times the force of gravity. If sustained, this would launch a satellite into orbit in about seven minutes and put us in a plane's luggage rack if our seat belts were not fastened. It is also striking that near zero acceleration is found between these peaks.

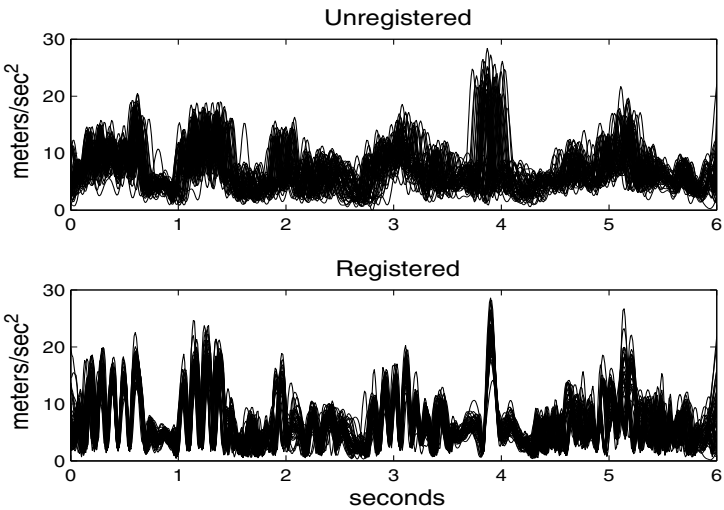

Fig. 8.6 The acceleration along the pen trajectory for all 50 replications of the script in Figure 1.9 before and after registration.

8.7 Details for Functions `landmarkreg` and `register.fd`

8.7.1 Function `landmarkreg`

The complete calling sequence for the R version is

```
landmarkreg(fdobj, ximarks, x0marks=xmeanmarks,
```

```
                WfdPar, monwrd=FALSE)
```

The arguments are as follows:

`fdobj` A functional data object containing the curves to be registered.

`ximarks` A matrix containing the timings or argument values associated with the landmarks for the observations in `fd` to be registered. The number of rows N equals the number of observations and the number of columns NL equals the number of landmarks. These landmark times must be in the interior of the interval over which the functions are defined.

`x0marks` A vector of times of landmarks for target curve. If not supplied, the mean of the landmark times in `ximarks` is used.

`WfdPar` A functional parameter object defining the warping functions that transform time in order to register the curves.

`monwrd` A logical value: if `TRUE`, the warping function is estimated using a monotone smoothing method; otherwise, a regular smoothing method is used, which is not guaranteed to give strictly monotonic warping functions.

`Landmarkreg` returns a list with two components:

`fdreg` A functional data object for the registered curves.

`warpfd` A functional data object for the warping functions.

It is essential that the location of every landmark be clearly defined in each of the curves as well as the template function. If this is not the case, consider using the continuous registration function `register.fd`. Although requiring that a monotone smoother be used to estimate the warping functions is safer, it adds considerably to the computation time since monotone smoothing is itself an iterative process. It is usually better to try an initial registration with this feature to see if there are any failures of monotonicity. Moreover, monotonicity failures can usually be cured by increasing the smoothing parameter defining `WfdPar`. Not much curvature is usually required in the warping functions, so a low-dimensional basis, whether B-splines or monomials, is suitable for defining the functional parameter argument `WfdPar`. A registration with a few prominent landmarks is often a good preliminary to using the more sophisticated but more lengthy process in `register.fd`.

8.7.2 Function ***register.fd***

The complete calling sequence for the R version is

```
register.fd(y0fd=NULL, yfd=NULL,
            WfdParobj=c(Lfdobj=2, lambda=1),
            conv=1e-04, iterlim=20, dbglev=1,
            periodic=FALSE, crit=2)
```

`y0fd` A functional data object defining the target for registration. If `yfd` is `NULL` and `y0fd` is a multivariate data object, then y0fd is assigned to `yfd` and `y0fd`

is replaced by its mean. Alternatively, if `yfd` is a multivariate functional data object and `y0fd` is missing, `y0fd` is replaced by the mean of `y0fd`. Otherwise, `y0fd` must be a univariate functional data object taken as the target to which `yfd` is registered.

`yfd` A multivariate functional data object defining the functions to be registered to target `y0fd`. If it is `NULL` and `y0fd` is a multivariate functional data object, `yfd` takes the value of `y0fd`.

`WfdParobj` A functional parameter object for a single function. This is used as the initial value in the estimation of a function $W(t)$ that defines the warping function $h(t)$ that registers a particular curve. The object also contains information on a roughness penalty and smoothing parameter to control the roughness of $h(t)$. Alternatively, this can be a vector or a list with components named `Lfdobj` and `lambda`, which are passed as arguments to `fdPar` to create the functional parameter form of `WfdParobj` required by the rest of the `register.fd` algorithm. The default `Lfdobj` of 2 penalizes curvature, thereby preferring no warping of time, with `lambda` indicating the strength of that preference. A common alternative is `Lfdobj` = 3, penalizing the rate of change of curvature.

`conv` A criterion for convergence of the iterations.

`iterlim` A limit on the number of iterations.

`dbglev` Either 0, 1, or 2. This controls the amount of information printed out on each iteration, with 0 implying no output, 1 intermediate output level, and 2 full output. (If this is run with output buffering, it may be necessary to turn off the output buffering to actually get the progress reports before the completion of computations.)

`periodic` A logical variable: if `TRUE`, the functions are considered to be periodic, in which case a constant can be added to all argument values after they are warped.

`crit` An integer that is either 1 or 2 that indicates the nature of the continuous registration criterion that is used. If 1, the criterion is least squares, and if 2, the criterion is the minimum eigenvalue of a cross-product matrix. In general, criterion 2 is to be preferred.

A named list of length 3 is returned containing the following components:

`regfd` A functional data object containing the registered functions.

`Wfd` A functional data object containing the functions $hW(t)$ that define the warping functions $h(t)$.

`shift` If the functions are periodic, this is a vector of time shifts.

8.8 Some Things to Try

1. At the end of Chapter 7 we suggested a principal components analysis of the log of the first derivative of the growth curves. This was, of course, before registra-

tion. Now repeat this analysis for the registered growth curves, and compare the results. What about the impact of the pubertal growth spurt now?

2. Try applying continuous registration to the unregistered growth curves. You will see that a few curves are badly misaligned, indicating that there are limits to how well continuous registration works. What should we do with these misaligned curves? Could we try, for example, starting the continous registrations off with initial estimates of function `Wfd` set up from the landmark registered results?
3. Using only those girls whose curves are well registered by continuous registration, now use canonical correlation analysis to explore the covariation between the Wfd object returned by function `register.fd` and the Wfd object from the monotone smooth. Look for interesting ways in which the amplitude variation in growth is related to its the phase variation.
4. **Medfly Data**: In Section 7.7, we suggested applying principal components analysis to the medfly data. Here, we suggest you extend that analysis as follows:

   a. Perform a functional linear regression to predict the total lifespan of the fly from their egg laying. Choose a smoothing parameter by cross-validation, and plot the coefficient function along with confidence intervals.
   b. Conduct a permutation test for the significance of the regression. Calculate the R^2 for your regression.
   c. Compare the results of the functional linear regression with the linear regression on the principal component scores from your analysis in Section 7.7.

8.9 More to Read

The classic paper on the estimation of time warping functions is Sakoe and Chiba (1978), who used dynamic programming to estimate the warping function in a context where there was no need for the warping function to be smooth.

Landmark registration has been studied in depth by Kneip and Gasser (1992) and Gasser and Kneip (1995), who refer to a landmark as a *structural feature*, its location as a *structural point*, to the distribution of landmark locations along the t axis as *structural intensity*, and to the process of averaging a set of curves after registration as *structural averaging*. Their papers contain various technical details on the asymptotic behavior of landmark estimates and warping functions estimated from them. Another source of much information on the study of landmarks and their use in registration is Bookstein (1991).

The literature on continuous registration is evolving rapidly, but is still somewhat technical. Gervini and Gasser (2004) and Liu and Müller (2004) are recent papers that review the literature and discuss some theoretical issues.