

Week 10 Exercises

This week's exercises will cover issues related to temporal sampling of the hemodynamic response. You will work with idealized hemodynamic waveforms constructed by interpolating real data. You will evaluate how the measured hemodynamic response changes with different sampling parameters.

The MATLAB functions needed are `interp1`, `figure`, `plot`, `hold`.

Exercise 1: Evaluating Temporal Sampling

1.1 Key concepts

The basic concepts to understand in this exercise are 1) the effects of repetition time (TR) on sampling of the fMRI hemodynamic response, and 2) how the choice of TR affects your experimental analyses

TR: Repetition time. The time between successive acquisitions of a single image. For functional studies, TR usually ranges between 500 to 3000ms.

Sampling rate: The frequency of acquisition of data. For functional MR data, sampling rate is defined as $1/TR$. So, for a TR of 2s, the sampling rate would be 0.5Hz.

Sampling Phase: When your samples are taken, relative to some event. For any TR, you might sample a given slice at any one of a number of different time points relative to a stimulus onset.

Interpolation: The generation of intermediate points between adjacent samples. Consider the data set consisting of the following values sampled at 0,2,4, and 6s: 500, 508, 520, 500. You could use linear interpolation to generate the following values for every second from 0-6s: 500, 504, 508, 514, 520, 510, 500. Common ways of doing interpolation for fMRI data use linear, spline, or sinc functions.

1.2. Creating an idealized waveform

The sample hemodynamic waveform to be used is in the following file. Load it into MATLAB.

```
cd S:\Class.01\Examples\Hemodynamic_responses
load weights
```

The variable `weights` consists of a total of 19 time points going from -5s before the stimulus to +13s after the stimulus. Thus, the current TR for this waveform was 1000ms. We are now going to interpolate this waveform to simulate a 100ms TR (10x improvement in resolution). You should use the `interp1` command.

```
weights_new = interp1([-5:13],weights,[-5:.1:13],'spline');
weights_new = weights_new';
```

Note that the apostrophe transposes a matrix in Matlab. Since the `interp1` command gives output as a row, we can use the apostrophe to make the output into a column, for ease of use (optional).

1.3. Displaying the waveforms

To display the waveforms, you will use the `figure` and `plot` commands. `figure` creates a new window within which you can display data. `plot` creates a chart (line chart by default) within an open figure. So, you can open one figure and repeatedly plot different data into it, or you can open new figures each time. We will plot two figures, one for each data. [Note: you can use the `hold` command to over-plot one graph on top of another].

```
figure, plot([-5:13],weights)
```

```
figure, plot([-5:1:13],weights_new)
```

What is different about these two waveforms?

1.4. Evaluating the effects of interpolation

Now try different values for sampling the hemodynamic response. To change the simulated TR, just change the interval in the second array (as in below example, for 2s TR).

```
weights_new_2s = interp1([-5:13],weights,[-5:2:13],'spline',0);
```

Try a number of different TR values from very short to very long. **What happens to the hemodynamic response at long TRs?**

1.5. Choosing an optimal TR

Based on your results from exercise 1.4, what seems to be a good TR for sampling the hemodynamic response? Choose the largest value that, to your eyes, gives you a good representation of the hemodynamic response. Write a short description indicating why you chose this value.

Exercise 2: Effects of the Phase of Sampling

2.1. Key concepts

A second problem with poor sampling is that different phases of sampling can influence what the resulting data look like. Here, we will look at data sampled at a constant rate across different phases.

There are two primary ways in which the phase of sampling may be altered. The first way results from **interleaved slice acquisition**. In all multi-slice fMRI experiments done at BIAC, the slices are acquired at different time points within a TR. Thus, the time within a TR that a given part of the brain is sampled will depend upon its position within the imaging volume. The second way comes from **stimulus presentation**. Depending on when a stimulus is presented (e.g., at the onset of a TR or not), the hemodynamic response may or may not be time locked to image acquisition.

2.2. Adjusting the phase of sampling

A second problem with poor sampling is that different phases of sampling can influence what the resulting data look like. Here, we will look at data sampled at a constant rate across different phases. First, create the interpolated weights file like before.

```
weights_phase0 = interp1([-5:13],weights',[-5:2:13],'spline');  
weights_phase1 = interp1([-5:13],weights',[-4.5:2:13],'spline');  
weights_phase2 = interp1([-5:13],weights',[-4:2:13],'spline');  
weights_phase3 = interp1([-5:13],weights',[-3.5:2:13],'spline');
```

```
figure,plot([-5:2:13],weights_phase0);  
figure,plot([-4.5:2:13],weights_phase1);  
figure,plot([-4:2:13],weights_phase2);  
figure,plot([-3.5:2:13],weights_phase3);
```

How different are the resulting waveforms?

2.3 Results from long TRs

Repeat the analysis from exercise 2.2 with a longer TR of 3seconds. **How dissimilar are the results across phases? Can you think of a way to quantify the difference between different phases?**

2.4. Results from short TRs

Repeat the analysis from exercise 2.2 with whatever TR you decided upon above (in exercise 1.4). **Is the TR value you chose sufficiently small that the different sampling phases make little difference?**

Exercise 3: Sampling and Blocked Design Activity

In this exercise, we will examine the robustness of blocked designs to changes in sampling rate.

3.1. Display the sample data

You should load and display the sample data. This data set consisted of a task blocks from 0-24s and a non-task-block from 24-48s. The weights_blocked file has the data (**in 2s TRs**) from -4 time points of block onset through 23 time points after block onset.

```
cd S:\Class.01\Examples\Hemodynamic_responses  
load weights_blocked  
figure, plot([-8:2:46],weights_blocked)
```

What do the data look like? Is there a block effect between the task block and non-task block? Are there any transient changes associated with block onset or block offset?

3.2. Desampling the blocked data

Note that these data were acquired at a 2s TR. What happens to the block effect when you simulate sampling at a 3s TR? What about at a 4s TR? At what TR value does the block effect break down?

3.3. Comparing blocked vs. event-related effects

Now examine the transient activity associated with the task onset and with the task offset. There is a large peak following task onset and a small peak at task offset. At what TR does each of the peaks break down? Which peak is more robust to changes in TR? Why?