Problem description

"This data set consists of measurements from 25 experiments. Each experiment is removing material from a surface in steps. Each experiment has 131 steps. The data is from a simulation of the removal of material from 1500A to 200A in 10A steps. At each step, 30 measurements are taken. These measurements are all plagued with noise. Some of the noise acts as process noise, other noise sources distort the underlying spectra a bit at each of the 30 measurement sites. This means that even if we removed the process noise, the results for the 30 measurements in each step would be a bit different. The measurement sites are in fixed places that are consistent between steps but not between wafers. Each wafer may enter the removal process rotated slightly differently from the previous wafer so the measurement sites of wafer 1 won't be affected in the same way as the sites on wafer 2. But during removal, a wafer does not rotate so the sites between steps will stay the same. Each row consists of 601 intensity values. These values are the intensities detected at each wavelength from 2000A (in column 1) to 8000A (in column 601) in 10A steps. The 602nd value is the thickness of the surface (in angstroms) for that row.

There is a training data set and a validation data set. The training data has all 25 wafers one after the next in a single file. The 131 removal steps for each wafer are in order one after the other. Each step has all 30 spectra measurements together. Thus 31 rows into the data section, is the first measurement after the first removal step has completed for the first experiment.
In this case, the validation data set also has the answers for the depths of the surface at each row. Your challenge is to build an algorithm that at every step that uses only the current and previous measurement data (for this and all previous steps) to predict the current surface depth for the validation data set. You don't have to use previous step data, but it is an option. This of course precludes using the final column, as that is only for being able to judge the effectiveness of your algorithm. You also can't code in an assumption that the removal rate is always 10A. But you can assume that the removal rate is linear. The experiment does remove roughly the same amount each pass.
This exercise isn't all about making the best model (though it doesn't hurt), but having a problem where you can document your thinking. Please keep notes on how you tackle this problem. What did you try? Why? Do the best you can in a reasonable amount of time, and make notes about what you would try if you had more time and why.
Please upload to the repository any scripts or code you used for your work, along with your notes on your thinking, a plot of your results as predicted vs. actual depth, and anything else you think would be useful. If your script requires special libraries please document this as well. Your results should be reproducible. You are free to use any tool you prefer."