1

Introduction

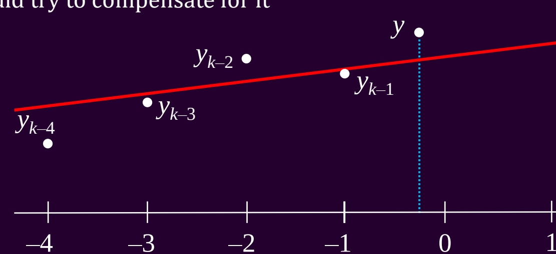
- In this topic, we will
  - Describe the concept of jitter
  - Consider what happens if samples are not read periodically
  - We will try to, as reasonably as possible, adjust the read value to compensate for the error in the time of the reading
    - We will not eliminate the noise
    - The formulas we use mitigate the noise

2

2

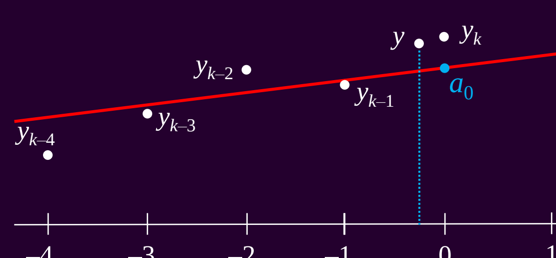
Introducing jitter

- *Jitter* are the small anomalies that will occur in a system
  - For example, while there is noise in the signal read, the sample may also not be taken exactly every h s
  - Thus, the most recent reading may have been taken at $t_k + \delta h$ instead of exactly t_k
    - We assume δ is small—no more than 0.5
  - If we know a reading was taken early or late, we should try to compensate for it



Dealing with jitter

- Question: how do we correct for the value of y_k if that value was read at the wrong time?
  - Issue: all the least-squares formulas we've looked at require that the samples are taken periodically
  - Idea: the points we have calculated can estimate the value of the function at t_k
    - Chose that value y_k such that a_0 remains unchanged



Strategy

- For this, suppose that we are attempting to read y_k,
  but the reading taken was the value y and it was taken at $t_k + \delta h$
- Find the constant coefficient of the least-squares best-fitting polynomial that passes through:

$$(-n, y_{k-n}),(1-n, y_{k-n+1}),\ldots,(-1, y_{k-1}),(\delta, y)$$

- Find the constant coefficient of the least-squares best-fitting polynomial that passes through:

$$(-k, y_{k-n}),(1-k, y_{k-n+1}),\ldots,(-1, y_{k-1}),(0, y_k)$$

- Equate these and solve for y_k

Linear example when $n = 4$

- This requires symbolic computation,
  so a language such as Maple is excellent for this

$$\frac{(y_{k-4}+y_{k-3}+y_{k-2}+y_{k-1})\delta^2+(4y_{k-4}+3y_{k-3}+2y_{k-2}+y_{k-1}+10y)\delta+10(-y_{k-4}+y_{k-2}+2y_{k-1}+3y)}{4\delta^2+20\delta+50}$$

$$=\frac{-y_{k-4}+y_{k-2}+2y_{k-1}+3y_k}{5}$$

- Solving for y_k yields

$$y_k \leftarrow \frac{(9y_{k-4}+5y_{k-3}+y_{k-2}-3y_{k-1})\delta^2+5(8y_{k-4}+3y_{k-3}-2y_{k-2}-7y_{k-1}+10y)\delta+150y}{12\delta^2+60\delta+150}$$

- A first order approximation (if δ is sufficiently small) is:

$$y_n \leftarrow y+\frac{\delta}{30}(8y_{k-4}+3y_{k-3}-2y_{k-2}-7y_{k-1}-2y)$$

Quadratic example when $n = 4$

- You could also do this for least-squares best-fitting quadratic polynomials and get a 2nd-order series solution around δ:

$$y_k \leftarrow y + \left(-\frac{484}{1085} y_{k-4} + \frac{1107}{2170} y_{k-3} + \frac{701}{1085} y_{k-2} - \frac{83}{2170} y_{k-1} - \frac{729}{1085} y \right) \delta$$
$$+ \left(\frac{19519}{75950} y_{k-4} - \frac{33283}{75950} y_{k-3} - \frac{1189}{3038} y_{k-2} + \frac{30193}{75950} y_{k-1} + \frac{6648}{37975} y \right) \delta^2$$

Maple code

- Here is the Maple code:

```
> restart;
> n := 4:
> ts := seq( k, k = -n..-1 );          # -4, -3, -2, -1
> ys := seq( y||(-k), k = -n..-1 ):    # y4, y3, y2, y1
> # curve_expr := a1*t + a0:
> curve_expr := a2*t^2 + a1*t + a0:
> pt := CurveFitting:-LeastSquares( [ts, 0], [ys, y0], t,
                                    'curve' = curve_expr ):
> at := eval( pt, t = 0 ):
> assume( delta::real );
> p := CurveFitting:-LeastSquares( [ts, delta], [ys, y], t,
                                   'curve' = curve_expr ):
> a := simplify( eval(p, t = 0 ), size):
> y0 := simplify( solve( at = a, y0 ), 'size' ):
> series( y0, delta = 0, 3 );
```

Summary

- Following this topic, you now
  - Understand the idea of jitter
  - Understand that the system may not have a perfectly periodic sampling
  - Know you can attempt to compensate for that errors in sampling
  - Understand that the formulas we have require periodic sampling
  - Know that it is possible to set a rule and then find an appropriate value of y_k that satisfies that rule

References

[1] https://en.wikipedia.org/wiki/Least_squares
[2] Maplesoft: https://www.Maplesoft.com/

Acknowledgments

None so far.

Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.
Mathematical equations are prepared in MathType by Design Science, Inc.
Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

Disclaimer

These slides are provided for the ECE 204 *Numerical methods* course taught at the University of Waterloo. The material in it reflects the author's best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.

13