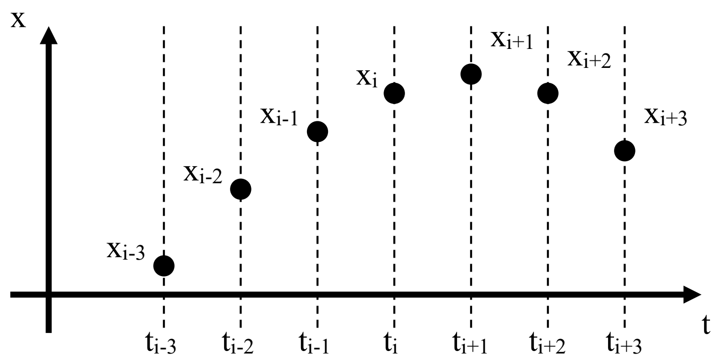
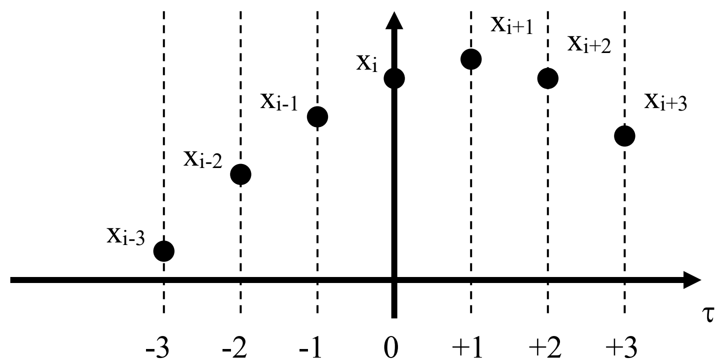
Numerical Derivatives Using Savitsky-Golay Floating Cubic Interpolants

measure position variable x_i at times t_i (note that the subscript i refers to time not body number)



for fixed time step h, $\tau = (t - t_i)/h$ $\quad \partial\tau = \partial t / h$ $\quad \partial\tau / \partial t = 1/h$



postulate $x = b_0 + b_1\,\tau + b_2\,\tau^2 + b_3\,\tau^3$

$\dot{x} = (\,b_1 + 2\,b_2\,\tau + 3\,b_3\,\tau^2\,)/h$ $\qquad \ddot{x} = (\,2\,b_2 + 6\,b_3\,\tau\,)/h^2$ $\qquad \dddot{x} = 6\,b_3/h^3$

using values for $x_i \approx [\,1 \;\; \tau_i \;\; \tau_i^2 \;\; \tau_i^3\,] \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{Bmatrix}$

$$\begin{Bmatrix} x_{i+3} \\ x_{i+2} \\ x_{i+1} \\ x_i \\ x_{i-1} \\ x_{i-2} \\ x_{i-3} \end{Bmatrix} \approx \begin{bmatrix} 1 & 3 & 9 & 27 \\ 1 & 2 & 4 & 8 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & -2 & 4 & -8 \\ 1 & -3 & 9 & -27 \end{bmatrix} \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

$$\{Y\} \approx [X]\{\beta\} \qquad \{Y\} = \begin{Bmatrix} x_{i+3} \\ x_{i+2} \\ x_{i+1} \\ x_i \\ x_{i-1} \\ x_{i-2} \\ x_{i-3} \end{Bmatrix} \qquad [X] = \begin{bmatrix} 1 & 3 & 9 & 27 \\ 1 & 2 & 4 & 8 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & -1 & 1 & -1 \\ 1 & -2 & 4 & -8 \\ 1 & -3 & 9 & -27 \end{bmatrix} \qquad \{\beta\} = \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

linear least-squares solution $\{\beta\} = ([X]^T [X])^{-1} [X]^T \{Y\}$

$$\begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{Bmatrix} = \left(([X]^T[X])^{-1}[X]^T\right)\begin{Bmatrix} x_{i+3} \\ x_{i+2} \\ x_{i+1} \\ x_i \\ x_{i-1} \\ x_{i-2} \\ x_{i-3} \end{Bmatrix} = \frac{1}{252}\begin{bmatrix} -24 & 36 & 72 & 84 & 72 & 36 & -24 \\ -22 & 67 & 58 & 0 & -58 & -67 & 22 \\ 15 & 0 & -9 & -12 & -9 & 0 & 15 \\ 7 & -7 & -7 & 0 & 7 & 7 & -7 \end{bmatrix}\begin{Bmatrix} x_{i+3} \\ x_{i+2} \\ x_{i+1} \\ x_i \\ x_{i-1} \\ x_{i-2} \\ x_{i-3} \end{Bmatrix}$$

interpolated values at $\tau = 0$ $\quad x_i^* = b_0 \quad \dot{x}_i^* = b_1 / h \quad \ddot{x}_i^* = 2\, b_2 / h^2 \quad \dddot{x}_i^* = 6\, b_3 / h^3$

$x_i^* = (- 2\, x_{i+3} + 3\, x_{i+2} + 6\, x_{i+1} + 7\, x_i + 6\, x_{i-1} + 3\, x_{i-2} - 2\, x_{i-3}) / 21$

$\dot{x}_i^* = (- 22\, x_{i+3} + 67\, x_{i+2} + 58\, x_{i+1} - 58\, x_{i-1} - 67\, x_{i-2} + 22\, x_{i-3}) / 252\, h$

$\ddot{x}_i^* = (5\, x_{i+3} - 3\, x_{i+1} - 4\, x_i - 3\, x_{i-1} + 5\, x_{i-3}) / 42\, h^2$

$\dddot{x}_i^* = (x_{i+3} - x_{i+2} - x_{i-1} + x_{i-1} + x_{i-2} - x_{i-3}) / 6\, h^3$

for first three values $\quad b_0 = x_4^* \quad b_1 = \dot{x}_4^*\, h \quad b_2 = \ddot{x}_4^*\, h^2 / 2 \quad b_3 = \dddot{x}_4^*\, h^3 / 6$

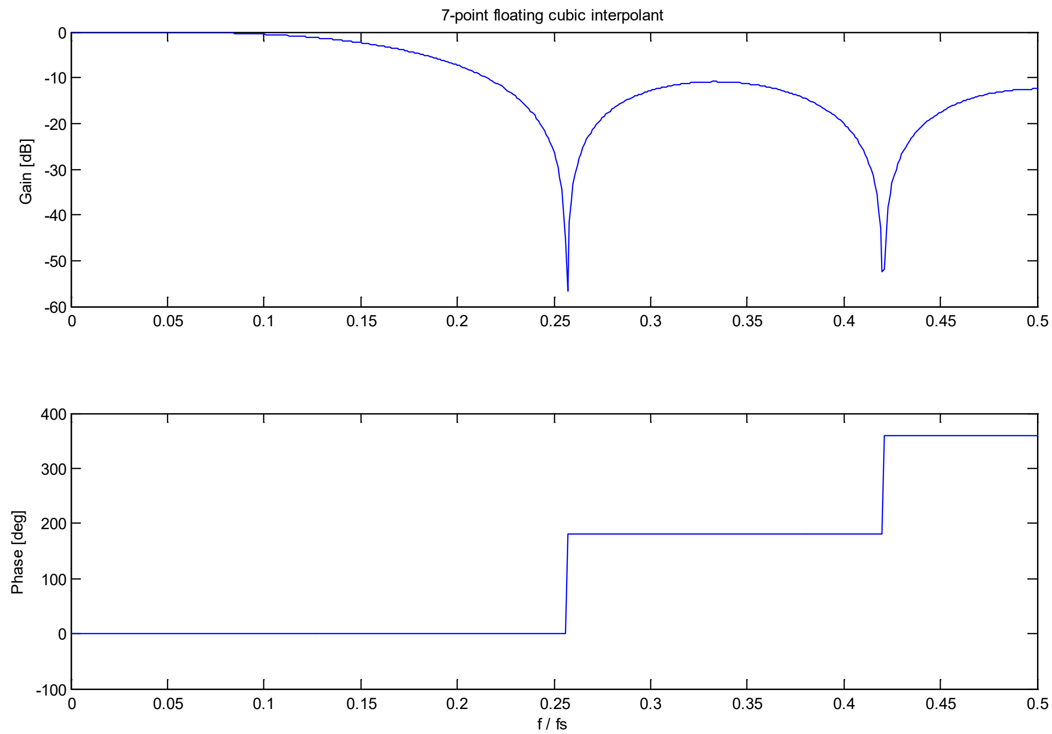
$x_1^* = b_0 - 3b_1 + 9b_2 - 27b_3 \quad \dot{x}_1^* = (b_1 - 6b_2 + 27b_3) / h \quad \ddot{x}_1^* = (2b_2 - 18b_3) / h^2 \quad \dddot{x}_2^* = \dddot{x}_4^*$
$x_2^* = b_0 - 2b_1 + 4b_2 - 8b_3 \quad \dot{x}_2^* = (b_1 - 4b_2 + 12b_3) / h \quad \ddot{x}_2^* = (2b_2 - 12b_3) / h^2 \quad \dddot{x}_2^* = \dddot{x}_4^*$
$x_3^* = b_0 - b_1 + b_2 - b_3 \quad \dot{x}_3^* = (b_1 - 2b_2 + 3b_3) / h \quad \ddot{x}_3^* = (2b_2 - 6b_3) / h^2 \quad \dddot{x}_3^* = \dddot{x}_4^*$

for last three values $\quad b_0 = x_{n-3}^* \quad b_1 = \dot{x}_{n-3}^*\, h \quad b_2 = \ddot{x}_{n-3}^*\, h^2 / 2 \quad b_3 = \dddot{x}_{n-3}^*\, h^3 / 6$

$x_{n-2}^* = b_0 + b_1 + b_2 + b_3 \quad \dot{x}_{n-2}^* = (b_1 + 2b_2 + 3b_3) / h \quad \ddot{x}_{n-2}^* = (2b_2 + 6b_3) / h^2 \quad \dddot{x}_{n-2}^* = \dddot{x}_{n-3}^*$
$x_{n-1}^* = b_0 + 2b_1 + 4b_2 + 8b_3 \quad \dot{x}_{n-1}^* = (b_1 + 4b_2 + 12b_3) / h \quad \ddot{x}_{n-1}^* = (2b_2 + 12b_3) / h^2 \quad \dddot{x}_{n-1}^* = \dddot{x}_{n-3}^*$
$x_n^* = b_0 + 3b_1 + 9b_2 + 27b_3 \quad \dot{x}_n^* = (b_1 + 6b_2 + 27b_3) / h \quad \ddot{x}_n^* = (2b_2 + 18b_3) / h^2 \quad \dddot{x}_n^* = \dddot{x}_{n-3}^*$

```
function [ p, v, a, j ] = filt_7pt_mat( x, h )
% Savitsky-Golay 7-point cubic interpolant and derivatives
%   -3dB low-pass cutoff at 16% of sampling frequency
%
% USAGE
% [ p, v, a, j ] = filt_7pt_mat( x, h )
%
% INPUTS
% x  - kxn matrix of raw samples
%      k = number of coordinates - scalar k=1, 2D k=2, 3D k=3, etc.
%      n = number of samples
% h - sampling interval
%
% OUTPUTS
% p - kxn position matrix
% v - kxn velocity matrix
% a - kxn acceleration matrix
% j - kxn jerk matrix

% HJSIII, 11.02.08 - tested under MATLAB v7.5
%
% H.J. Sommer III, Ph.D., Professor of Mechanical Engineering, 337 Leonhard Bldg
% The Pennsylvania State University, University Park, PA  16802
% (814)863-8997  FAX (814)865-9693  hjs1@psu.edu  www.mne.psu.edu/sommer/

% number of samples
[ k, n ] = size(x);
nm1 = n - 1;
nm2 = n - 2;
nm3 = n - 3;
nm4 = n - 4;
nm5 = n - 5;
nm6 = n - 6;

% initialize
p = zeros(k,n);
v = zeros(k,n);
a = zeros(k,n);
j = zeros(k,n);

% Savitsky-Golay 7 point cubic interpolant coefficients
% deriv  x(i+3)  x(i+2)  x(i+1)  x(i)    x(i-1)  x(i-2)  x(i-3)  divisor
% 0      -2      +3      +6      +7      +6      +3      -2      21
% 1      -22     +67     +58             -58     -67     +22     252*h
% 2      +5              -3      -4      -3              +5      42*h*h
% 3      +1      -1      -1              +1      +1      -1      6*h*h*h

p(:,4:nm3) = (  -2*x(:,7:n)    +3*x(:,6:nm1)  +6*x(:,5:nm2)  +7*x(:,4:nm3) ...
                +6*x(:,3:nm4)  +3*x(:,2:nm5)  -2*x(:,1:nm6) ) /21;

v(:,4:nm3) = ( -22*x(:,7:n)   +67*x(:,6:nm1) +58*x(:,5:nm2)              ...
               -58*x(:,3:nm4) -67*x(:,2:nm5) +22*x(:,1:nm6) ) /252 /h;

a(:,4:nm3) = (  +5*x(:,7:n)                  -3*x(:,5:nm2)  -4*x(:,4:nm3) ...
                -3*x(:,3:nm4)                +5*x(:,1:nm6) ) /42 /h /h;

j(:,4:nm3) = (     x(:,7:n)      -x(:,6:nm1)    -x(:,5:nm2)              ...
                  +x(:,3:nm4)    +x(:,2:nm5)    -x(:,1:nm6) ) /6 /h /h /h;

% first three
b0 = p(:,4);
b1 = v(:,4)*h;
b2 = a(:,4)*h*h/2;
b3 = j(:,4)*h*h*h/6;

p(:,1:3) = [ b0-3*b1+9*b2-27*b3  b0-2*b1+4*b2-8*b3  b0-b1+b2-b3  ];
v(:,1:3) = [ b1-6*b2+27*b3       b1-4*b2+12*b3      b1-2*b2+3*b3 ]/h;
a(:,1:3) = [ 2*b2-18*b3          2*b2-12*b3         2*b2-6*b3    ]/h/h;
j(:,1:3) = [ j(:,4)              j(:,4)             j(:,4)       ];

% last three
b0 = p(:,nm3);
b1 = v(:,nm3)*h;
```

```
b2 = a(:,nm3)*h*h/2;
b3 = j(:,nm3)*h*h*h/6;

p(:,nm2:n) = [ b0+b1+b2+b3   b0+2*b1+4*b2+8*b3  b0+3*b1+9*b2+27*b3 ];
v(:,nm2:n) = [ b1+2*b2+3*b3  b1+4*b2+12*b3      b1+6*b2+27*b3      ]/h;
a(:,nm2:n) = [ 2*b2+6*b3     2*b2+12*b3         2*b2+18*b3         ]/h/h;
j(:,nm2:n) = [ j(:,nm3)      j(:,nm3)           j(:,nm3)           ];

return

% bottom of filt_7pt_mat.m
```




```
% freq_7pt.m - frequency response for 7-point cubic interpolant
% HJSIII, 03.04.30

clear

% j*2*pi*f/fs
fdfs = ( 0 : 0.001 : 0.5 )';
t = j * 2 * pi * fdfs;

% transfer function
G = ( -2*exp(3*t) +3*exp(2*t) +6*exp(t) +7 +6*exp(-t) +3*exp(-2*t) -2*exp(-3*t) ) / 21;
amp = abs( G );
dB = 20 * log10( amp );
phi = unwrap( angle( G )) * 180 / pi;

figure( 1 )
  subplot( 2,1,1 )
  plot( fdfs,dB )
  ylabel( 'Gain [dB]' )
  title( '7-point floating cubic interpolant' )

  subplot( 2,1,2 )
  plot( fdfs,phi )
  xlabel( 'f / fs' )
  ylabel( 'Phase [deg]' )
```