sinc interpolation to reconstruct a signal from its samples

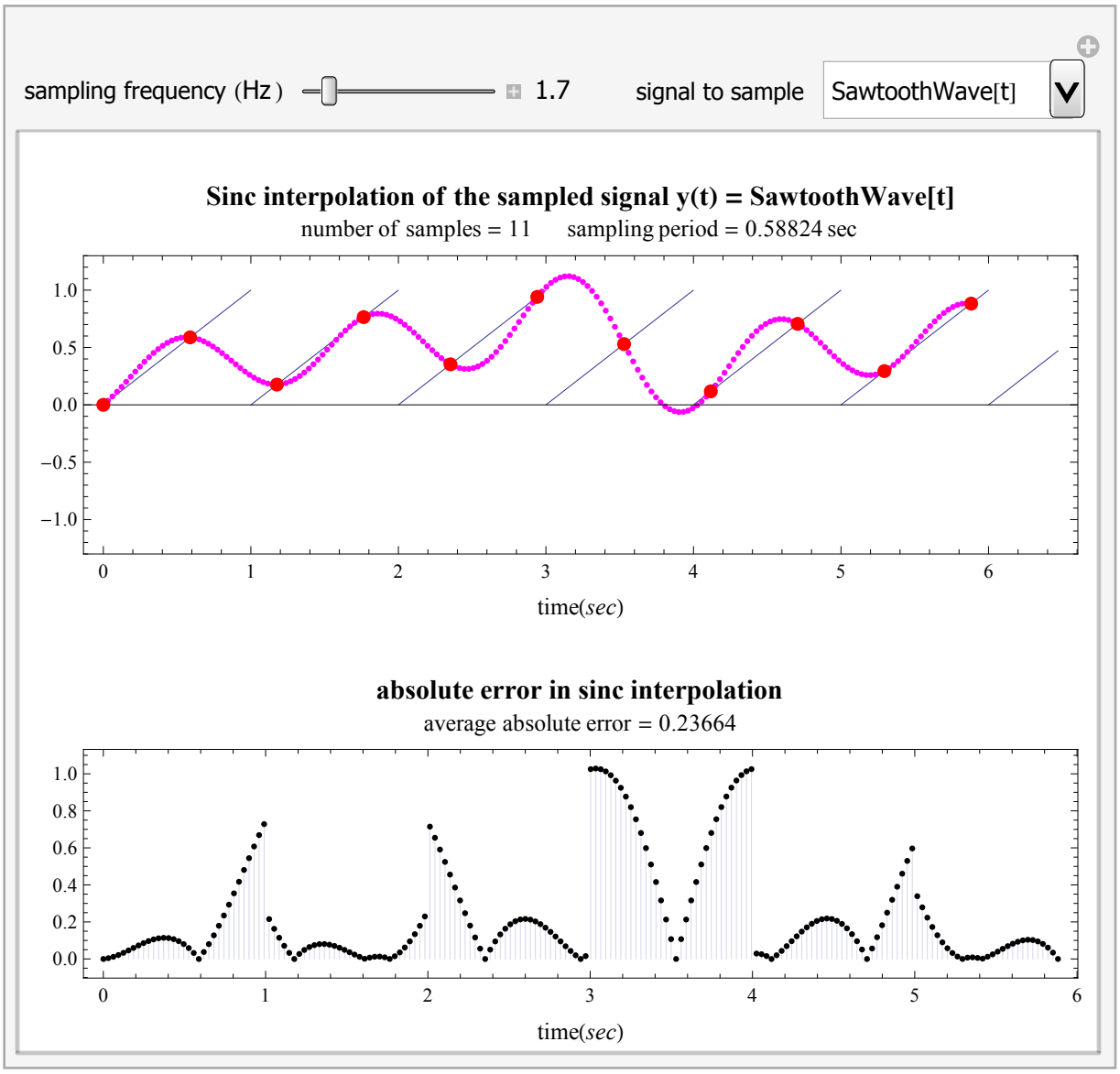
Initialization Code (optional)

Manipulate

```
Manipulate[process[ Fs, fun],
 Item[
  Grid[{
    {Control[{{Fs, 1, Row[{Style["sampling frequency", 12], Style[" (Hz)", Italic]}]},
       1, 10, .1, ImageSize → Small, Appearance → "Labeled"}],

     Control[{ {fun, 1, Style["signal to sample", 12]},
       {1 → Style["Cos[2 Pi t]", 10],
        2 → Style["Sin[2 Pi t]", 10],
        3 → Style["t", 10],
        4 → Style[t^2", 10],
        5 → Style[t^3", 10],
        6 → Style["SquareWave[t]", 10],
        6.1 -> Style["TriangleWave[t]", 10],
        6.2 -> Style["SawtoothWave[t]", 10],
        7 → Style["[Sin[2 Pi t]+Cos[3 Pi t]", 10],
        8 -> Style["Cos[2 Pi t]+t", 10]
       },
       ControlType → PopupMenu, ImageSize → Small}
     ]
    }
   }]
 ],
 FrameMargins → 0,
 ImageMargins → 0,
 ContinuousAction → False,
 SynchronousUpdating → False,
 AutorunSequencing → {{1, 60}},
 Initialization :→
  {

   (* This is the sinc interpolation formula *)
   sincInterpolate[t_, T_, samples_] := Module[{k},
     Sum[samples[[ k ]] * Sinc[Pi / T (t - (k - 1) T)], {k, 1, Length[samples]}]
    ];

   (* Main function called from Manipulate *)
   process[Fs_, fun_] := Module[
     {f, maxt, T, originalSignalPlot, p2, t, k, xa, samples, lines, absError, label, perror, sincDeltaTime,
      disks, sincNumberOfPoints = 20, ymin = 1.3, ymax = 1.3, numberOfSamples, functionName, averageAbsError},

     T = 1 / Fs;

     (* Check which function we are sampling and set appropriate plot limits*)
     Which[
```

```
 fun == 1, {f[t_] := Cos[2 Pi t]; maxt = 4; functionName = "y(t) = Cos[2 Pi t]"},
 fun == 2, {f[t_] := Sin[2 Pi t]; maxt = 4; functionName = "y(t) = Sin[2 Pi t]"},
 fun == 3, {f[t_] := t; maxt = 3; ymin = .5; ymax = 4; functionName = "y(t) = t"},

 fun == 4, {f[t_] := t^2; maxt = 4; ymin = .5; ymax = 20; functionName = "y(t) = t^2"},

 fun == 5, {f[t_] := t^3; maxt = 4; ymin = .5; ymax = 70; functionName = "y(t) = t^3"},

 fun == 6, {f[t_] := SquareWave[t]; maxt = 6; functionName = "y(t) = SquareWave[t]"},

 fun == 6.1, {f[t_] := TriangleWave[t]; maxt = 6; functionName = "y(t) = TriangleWave[t]"},

 fun == 6.2, {f[t_] := SawtoothWave[t]; maxt = 6; functionName = "y(t) = SawtoothWave[t]"},

 fun == 7, {f[t_] := Sin[2 Pi t] + Cos[3 Pi t]; maxt = 4;
  ymin = 2.5; ymax = 2.5; functionName = "y(t) = Sin[2 Pi t]+Cos[3 Pi t]]"},

 fun == 8, {f[t_] := Cos[2 Pi t] + t; maxt = 6; ymin = 1; ymax = 8; functionName = "y(t) = Cos[2 Pi t]+t"}

];

numberOfSamples = Floor[maxt / T] + 1;
samples = Table[f[k T], {k, 0, numberOfSamples - 1}];
maxt = (numberOfSamples - 1) * T;

(* find the sinc function sampling period *)
sincDeltaTime = T / (sincNumberOfPoints - 1);

(* This draws the samples lines, but not being used in this version *)
lines = Table[Line[{{t, 0}, {t, f[t]}}], {t, 0, maxt, T}];

(* This puts little dots the sample location *)
disks = Table[{PointSize[.014], Red, Opacity[1], Point[{t, f[t]}]}, {t, 0, maxt, T}];


(* format the plot label *)
label = Grid[
  {
   {Style["Sinc interpolation of the sampled signal " <> functionName, Bold, 14]
   },
   {
    Grid[{
       {
        Row[{Style["number of samples = ", 12], Style[Length[samples], 12]}], ,
        Row[
         {Style["sampling period = ", 12], Style[NumberForm[N[T], {7, 5}], 12], Style[" sec", 12]}]
        , SpanFromLeft
       }
     }, Alignment → Left
    ]
   }
  }, Alignment → Center
 ];


(* draw the original signal *)
originalSignalPlot =
 Plot[f[t], {t, 0, maxt + 0.1 * maxt},
  PlotRange → {Automatic, {-ymin, ymax}},
  Frame → True,
  ImagePadding → {{30, 1}, {45, 60}},
  ImageMargins → 0,
```

```
    ImageSize → 540,
    AspectRatio → .3,
    FrameTicksStyle -> Directive[10],
    AxesOrigin → {0, 0},
    FrameLabel → {{Null, Null}, { Row[{Style["time", 12], Style[ "(sec)", 12, Italic]}], label}},
    (*Epilog→{Dashed,Red,lines,disks}*)
    Epilog → {Dashed, Red, disks}
   ];

  (* Do the sinc interpolation sum *)
  xa = Table[{t,
      sincInterpolate[t, T, samples]}, {t, 0, maxt, sincDeltaTime}
   ];

  (* plot the result of the sinc interpolation *)
  p2 = ListPlot[xa, Joined → False, PlotRange → All, PlotStyle → {Dotted, Magenta}, ImageMargins → 5,
    ImageSize → 400];

  (* determine absolute error and plots it *)
  absError = Table[{xa[[i, 1]],
      Abs[ f[ xa[[i, 1]] ] - xa[[i, 2]] ]}, {i, 1, Length[xa]}];
  averageAbsError = Mean [absError[[All, 2]]] ;

  label = Grid[{
      {Style["absolute error in sinc interpolation", Bold, 14] , SpanFromLeft},
      {Row[{Style["average absolute error = ", 12],
         Style[NumberForm[N[averageAbsError], {8, 5}], 12]}], SpanFromLeft}},
    Alignment → Center
   ];

  perror =
   ListPlot[absError,
    Joined → False,
    Filling → Axis,
    PlotStyle → Black,
    PlotRange → All,
    Frame → True,
    ImagePadding → {{30, 1}, {35, 45}},
    ImageMargins → 0,
    ImageSize → 540,
    AspectRatio → .23,
    FrameTicksStyle -> Directive[10],
    AxesOrigin → {0, 0},
    FrameLabel → {{Null, Null}, { Row[{Style["time", 12], Style[ "(sec)", 12, Italic]}], label}}
   ];

  (*all done, puts all the plots in a grid *)
  Grid[
   {
    {Item[Show[{originalSignalPlot, p2}], ItemSize → Full], SpanFromLeft},
    {Item[perror, ItemSize → Full], SpanFromLeft}
   },
   Alignment → Center,
   Spacings → {1, 1},
   Frame → None
  ]
 ]
}
]
```



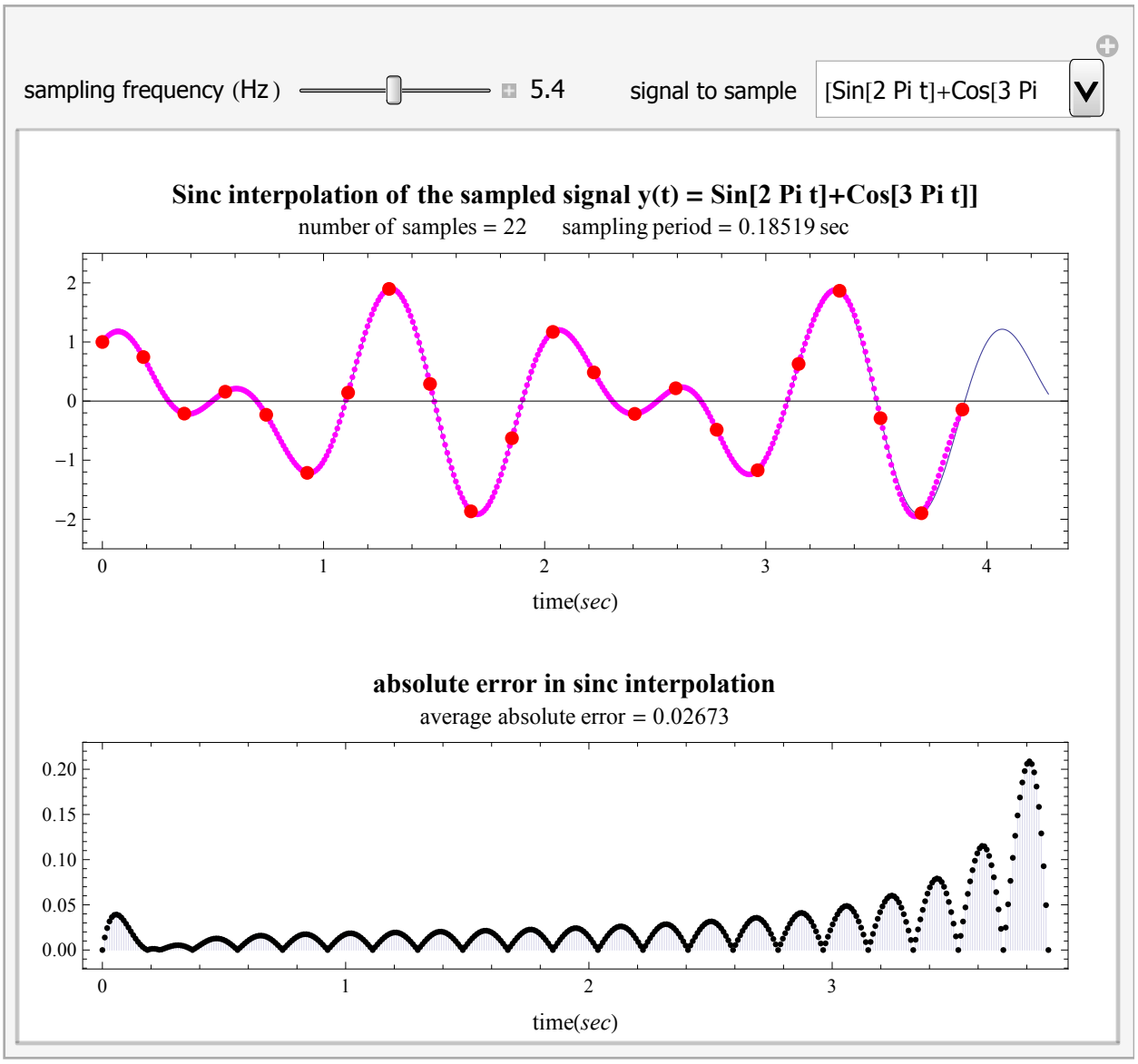
Caption

This demonstration illustrates the use of the Sinc interpolation formula to reconstruct a continues signal from its samples. The formula provides exact reconstructions for signals that are bandlimited and whose samples where obtained using the required Nyquist sampling frequency in order to eliminate aliasing in the reconstruction of the signal from its samples.

A number of continues signals are available to select from to apply the interpolation formula on. By increasing the sampling frequency a more accurate reconstruction of the continues function would result.

The original signal is shown in blue colored solid line, and the sample locations are shown using the red circles. The reconstructed signal is shown using the dotted magenta colored line and is superimposed on the original signal to make it easier to see the effect of increasing the sampling frequency on the reconstruction of the original signal from its samples.

Since a limited number of samples is used in this demonstration, there will be some error remaining in the reconstruction even at a higher sampling frequency than the Nyquist frequency. The absolute error (point by point error) between the original signal and the reconstructed signal is plotted to allow one to observe how the error decreases as the sampling rate increases.



Details (optional)

The Sinc interpolation formula is defined as $x(t) = \sum_{k=-\infty}^{\infty} x[n]\,\text{Sinc}\left[\frac{\text{Pi}}{T}(t - n\,T)\right]$ where T is the sampling period used to sample $x[n]$ from the original signal. $x(t)$ above is the reconstructed signal. The above formula represents a linear convolution between the sequence $x[n]$ and scaled and shifted samples of the Sinc function. In this demonstration, a limited number of samples $x[n]$ are generated, and the above sum is carried from $k = 0$ to $k = N - 1$ where N is the number of samples. Due to the shifting of the Sinc function by integer multiples of T, this results in $x(t)$ having the exact value of that sample which is located at such a multiple of T. This can be seen by observing that the absolute error is always zero at those instances of time which are integer multiples of T, in other words at the sample locations.
In this implementation, the sinc function is sampled at much higher rate than the sampling frequency used to sample the original function in order to produce a smoother plotting result.
Alan V. Oppenheim, Ronald W. Schafer, *Digital Signal processing*, Prentice-Hall, 1975.

Control Suggestions (optional)

☐ Slider Zoom

- ☐ Drag Locators
- ☐ Rotate and Zoom in 3D
- ☐ Automatic Animation
- ☐ Gamepad Controls
- ☐ Resize Images
- ☐ Bookmark Animation

Search Terms (optional)

sinc interpolation
Nyquist frequency
Shannon sampling theory
sampling frequency
aliasing

Related Links (optional)

Sinc function
Nyquist frequency

Authoring Information

Contributed by: Nasser M. Abbasi