Using Impulse Invariance to Convert an Analog to a Discrete System

Initialization Code (optional)

Manipulate

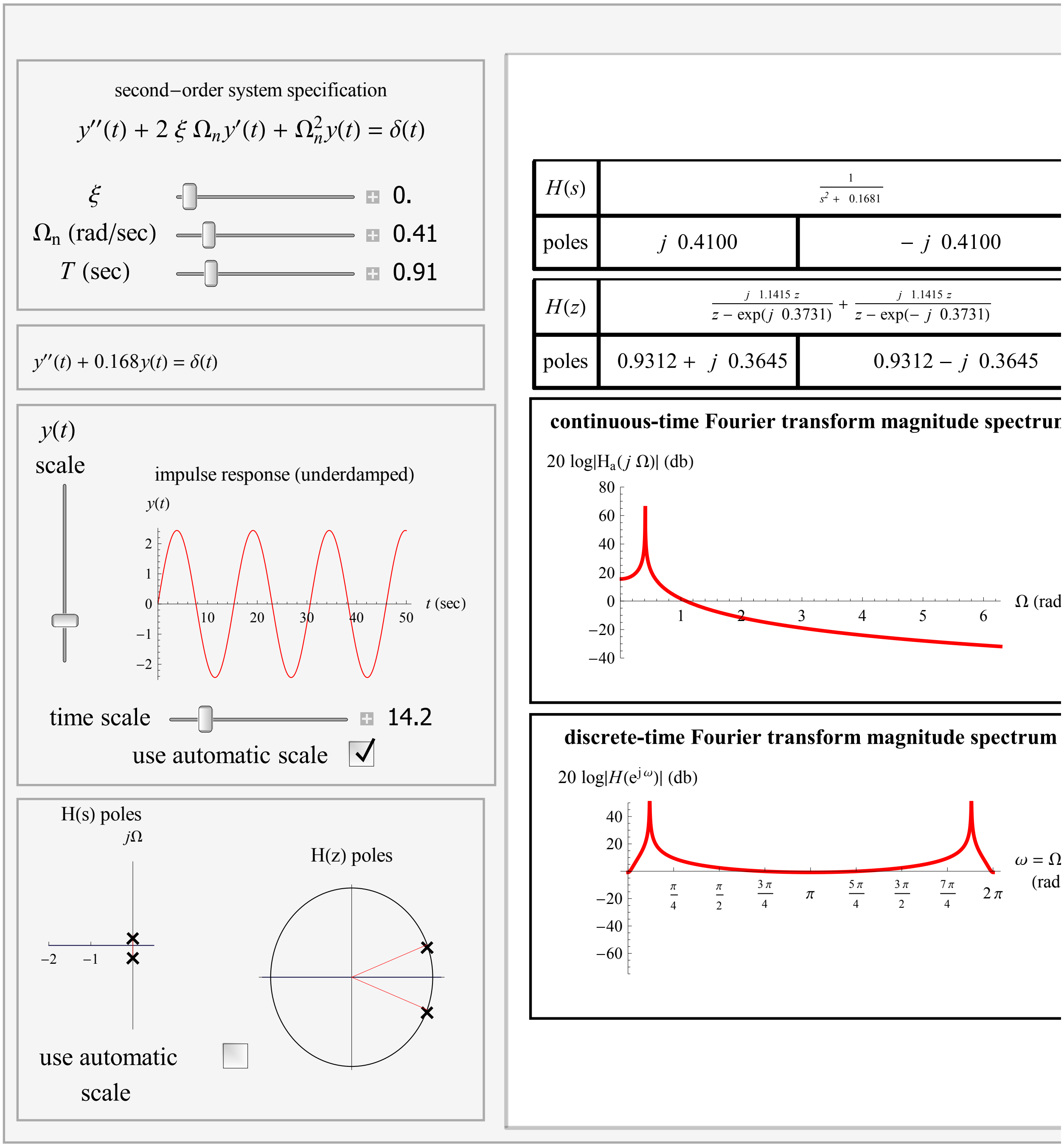
```
Manipulate[
 process[ξ, Ωn, T],
 Panel[Labeled[Grid[{
     {Text["ξ"], Control[{{ξ, .07, ""}, 0, 1.2, .1, Appearance → "Labeled" , ImageSize → Small}]},
     {Text["Ωn (rad/sec)"], Control[{{Ωn, 1, ""}, 0.01, Pi, .1, Appearance → "Labeled" , ImageSize → Small}]},
     {Text@Row[{Style["T", Italic], " (sec)"}],
      Control[{{T, 1, ""}, 0.01, 2 Pi, .1, Appearance → "Labeled" , ImageSize → Small}]}
    }, Spacings → {0, 0}], Column[{Text@Style["second-order system specification", 11],
     Text@Row[{Style["y", Italic]'', "(", Style["t", Italic], ") + 2 ξ ", Ω_Style["n",Italic],
        Style["y", Italic]', "(", Style["t", Italic], ") + ", Ω^2_Style["n",Italic], Style["y", Italic],
        "(", Style["t", Italic], ") = δ(", Style["t", Italic], ")"}]}, Center],
   {{Top, Center}}, Spacings → {0, 1}
  ], FrameMargins → 7, ImageSize → 260],
 Panel[Grid[{{Dynamic[If[ξ > 0, Text@Style[Row[{
           Style["y", Italic]'', "(", Style["t", Italic], ") + ",
           ToString[NumberForm[2 ξ Ωn, {4, 3}, NumberPadding → {"", "0"}]], Style["y", Italic]', "(",
           Style["t", Italic], ") + ", ToString[NumberForm[Ωn^2, {4, 3}, NumberPadding → {"", "0"}]],
           Style["y", Italic], "(", Style["t", Italic], ") = δ(", Style["t", Italic], ")"}], 11],
       Text@Style[Row[{Style["y", Italic]'', "(", Style["t", Italic], ") + ",
           ToString[NumberForm[Ωn^2, {4, 3}, NumberPadding → {"", "0"}]], Style["y", Italic],
           "(", Style["t", Italic], ") = δ(", Style["t", Italic], ")"}], 11]]]}},
   Spacings → {0, 0}, Alignment → Left, ItemSize → 30, Frame → None], FrameMargins →
   7,
  ImageSize →
   260
 ], Panel[Grid[{{Labeled[Control[{{maxy, 1, ""}, 0.1, 10, .1,
        ImageSize → Small,
        ControlType → VerticalSlider, Enabled → Dynamic@TrueQ[fixScale == 0]}],
      Text@Column[{Row[{Style["y", Italic], "(", Style["t", Italic], ")"}], " scale"}, Alignment → Center],
      {{Top, Center}}, Spacings → {0, 0}],
     Dynamic[Plot[If[ξ == 1, t Exp[-t Ωn], -(E^(t (-ξ Ωn - Ωn Sqrt[-1 + ξ^2])) - E^(t (-ξ Ωn + Ωn Sqrt[-1 + ξ^2])))/(2 Ωn Sqrt[-1 + ξ^2])], {t, 0, If[TrueQ[fixScale == 1],
         50, maxt]}, ImageMargins → 0, ImageSize → 220, ImagePadding → {{40, 40}, {10, 20}}, AspectRatio → .6,
       PlotRange → {{0, If[TrueQ[fixScale == 1], Automatic, maxt]},
         If[TrueQ[fixScale == 1], All, {-maxy, maxy}]},
       AxesLabel → {Text@Row[{Style["t", Italic], " (sec)"}],
         Text@Row[{Style["y", Italic], "(", Style["t", Italic], ")"}]},
       PlotLabel → Text@Row[{Style["impulse response", 11],
           If[ξ < 1, Style[" (underdamped)", 11],
            If[ξ > 1, Style[" (overdamped)", 11], Style[" (critcally damped)", 11]]]
          },
          Alignment → Center],
       AxesOrigin → {0, 0}, TicksStyle -> Directive[8], PlotStyle → Red]]},
```

```
   {Control[{{maxt, 50, Text["time scale"]}, 0.1, 100, .1, Appearance → "Labeled" ,
      ImageSize → Small, Enabled → Dynamic@TrueQ[fixScale == 0]}], SpanFromLeft},
   {Control[{{fixScale, 1, Text["use automatic scale"]}, {0, 1} , ControlType → Checkbox, ImageSize → Small}],
    SpanFromLeft}
  }, Alignment → Center, Spacings → {0, 0}, Frame → None],
 FrameMargins → 5
],
Panel[Grid[{
   {Column[{Dynamic[Plot[0, {t, .5, If[ξ < .5, -2, If[Ωn < 1, -2, If[Ωn < 2, -4.5, -6]]] },
        ImageSize → {130}, ImageMargins → 0, ImagePadding → {{15, 5}, {5, 20}},
        AspectRatio → 1.6,
        PlotRange → If[TrueQ[fixScaleHsPoles == 1], If[ξ < 1, {{-1.2 Ωn ξ, .2 Ωn ξ},
            {-1.2 (Ωn Sqrt[1 - ξ^2]), 1.2 Ωn Sqrt[1 - ξ^2]}}, If[ξ > 1, {{-1.2 (Ωn ξ + Ωn Sqrt[ξ^2 - 1]), .5}, {-.2, .2}},
            {{-Ωn, .5}, {-.2, .2}}]], If[ξ < .5, {{-2, .5}, {-3.5, 3.5}}, If[Ωn < 1,
            {{-2, .5}, {-3.5, 3.5}}, If[Ωn < 2, {{-4.5, .5}, {-3.5, 3.5}}, {{-6, .5}, {-3.5, 3.5}}]]]],
        AxesOrigin → {0, 0},
        Ticks → If[TrueQ[fixScaleHsPoles == 1], If[ξ < 1, {{-1.2 Ωn ξ, -.5 Ωn ξ}, Automatic},
           If[ξ > 1, {{-1.2 (Ωn ξ + Ωn Sqrt[ξ^2 - 1]), -.5 (Ωn ξ + Ωn Sqrt[ξ^2 - 1])}, Automatic},
            {{-Ωn, .5}, Automatic}]], If[ξ < .5, {{-2, -1}, , {-3, 3}},
           If[Ωn < 1, {{-2, -1}, , {-3, 3}}, If[Ωn < 2, {{-4, -2}, , {-3, 3}}, {{-6, -4, -2}, , {-3, 3}}]]]],
        TicksStyle -> Directive[8], PlotLabel → Text@Row[{Style[H[s], 11], Style[" poles", 11]}],
        AxesLabel → {None, Text@Row[{Style["j", Italic], "Ω"}]}, Epilog → {{Thin, Red, Line[
            {{0, 0}, If[ξ < 1, {-Ωn ξ, Ωn Sqrt[1 - ξ^2]}, If[ξ > 1, {-Ωn ξ + Ωn Sqrt[ξ^2 - 1], 0}, {-Ωn, 0}]]}]}, {Thin,
           Red, Line[{{0, 0}, If[ξ < 1, {-Ωn ξ, -Ωn Sqrt[1 - ξ^2]}, If[ξ > 1, {-Ωn ξ - Ωn Sqrt[ξ^2 - 1], 0}, {-Ωn, 0}]]}]},
          Text[Style["×", Bold, 16], If[ξ < 1, {-Ωn ξ, Ωn Sqrt[1 - ξ^2]},
            If[ξ > 1, {-Ωn ξ + Ωn Sqrt[ξ^2 - 1], 0}, {-Ωn, 0}]], {0, 0}], Text[Style["×", Bold, 16],
           If[ξ < 1, {-Ωn ξ, -Ωn Sqrt[1 - ξ^2]}, If[ξ > 1, {-Ωn ξ - Ωn Sqrt[ξ^2 - 1], 0}, {-Ωn, 0}]], {0, 0}]}]]
      , Control[{{fixScaleHsPoles, 0, Text["use automatic scale"]}, {0, 1} ,
        ControlType → Checkbox, ImageSize → Small}]}, Spacings → {0, 0}],
    Dynamic[Plot[0, {t, -1.1, 1.1}, ImageSize → {130},
      ImageMargins → 0,
      ImagePadding → {{1, 1}, {5, 10}}, AspectRatio → 1, PlotRange → All, AxesOrigin → {0, 0},
      PlotLabel → Style[Text@Row[{H[z], Style[" poles"]}], 11],
      Ticks → None, Epilog → {Circle[{0, 0}, 1],
        Text[Style["×", Bold, 16], If[ξ < 1, {Exp[-Ωn ξ T] Cos[Ωn T Sqrt[1 - ξ^2]], Exp[-Ωn ξ T] Sin[T Ωn Sqrt[1 - ξ^2]]},
          If[ξ > 1, {Exp[-Ωn ξ T + Ωn T Sqrt[-1 + ξ^2]], 0}, {Exp[-Ωn T], 0}]], {0, 0}],
        Text[Style["×", Bold, 16], If[ξ < 1, {Exp[-Ωn ξ T] Cos[-T Ωn Sqrt[1 - ξ^2]], Exp[-T Ωn ξ] Sin[-T Ωn Sqrt[1 - ξ^2]]},
          If[ξ > 1, {Exp[-Ωn ξ T - Ωn T Sqrt[-1 + ξ^2]], 0}, {Exp[-Ωn T], 0}]], {0, 0}],
        {Thin, Red, Line[{{0, 0}, If[ξ < 1, {Exp[-Ωn ξ T] Cos[T Ωn Sqrt[1 - ξ^2]], Exp[-T Ωn ξ] Sin[T Ωn Sqrt[1 - ξ^2]]},
            If[ξ > 1, {Exp[-T Ωn ξ + T Ωn Sqrt[-1 + ξ^2]], 0}, {Exp[-T Ωn], 0}]]}]},
        {Thin, Red, Line[{{0, 0}, If[ξ < 1, {Exp[-T Ωn ξ] Cos[-T Ωn Sqrt[1 - ξ^2]], Exp[-T Ωn ξ] Sin[-T Ωn Sqrt[1 - ξ^2]]},
```

```
            If[ξ > 1, {Exp[-T Ωn ξ - T Ωn Sqrt[-1 + ξ^2]], 0}, {Exp[-T Ωn], 0}]]}]}
        }]]}},
  Alignment → Left,
  Spacings → {0, 0}, Frame → None, ItemSize → {12, 5}], FrameMargins → 1, ImageSize → {{260, 180}}
],

{{t, 0}, ControlType → None},
{{maxy, 1}, ControlType → None},
{{maxt, 50}, ControlType → None},
{{fixScale, 1}, ControlType → None},
ContinuousAction → False,
SynchronousUpdating → True,
AutorunSequencing → {1, 2, 3},
ControlPlacement → Left,
Initialization :→
 {

  str[expr_] := Module[{}, StringReplace[
     ToString[expr, FormatType → TraditionalForm], c : LetterCharacter ~~ "$" ~~ DigitCharacter .. :→ c]
   ];

  numIt[v_, s1_, s2_] := Module[{},
    ToString[AccountingForm[Chop[v], {s1, s2}, NumberPadding → {" ", "0"}, NumberSigns → {"-", ""}]]
   ];

  formatHsPoles[Ωn_, ξ_, T_] := Module[{p1, p2},
    p1 = If[ξ == 0, Row[{Style["j", Italic], numIt[T Ωn, 6, 4]}], If[ξ < 1, Row[{numIt[-Ωn ξ T, 6, 4], " + ",
         Style["j", Italic], numIt[Ωn T Sqrt[1 - ξ^2], 6, 4]}], Row[{numIt[-Ωn ξ T + Ωn T Sqrt[ξ^2 - 1], 6, 4]}]]];

    p2 = If[ξ == 0, Row[{Style["- j", Italic], numIt[Ωn T, 6, 4]}], If[ξ < 1, Row[{numIt[-Ωn ξ T, 6, 4], " - ",
         Style["j", Italic], numIt[Ωn T Sqrt[1 - ξ^2], 6, 4]}], Row[{numIt[-Ωn ξ T - Ωn T Sqrt[ξ^2 - 1], 6, 4]}]]];
    {p1, p2}
   ];

  formatHsPoles[Ωn_, ξ_] := Module[{p1, p2},
    p1 = If[ξ == 0, Row[{Style["j", Italic], numIt[Ωn, 6, 4]}], If[ξ < 1, Row[{numIt[-Ωn ξ, 6, 4], " + ",
         Style["j", Italic], numIt[Ωn Sqrt[1 - ξ^2], 6, 4]}], Row[{numIt[-Ωn ξ + Ωn Sqrt[ξ^2 - 1], 6, 4]}]]];

    p2 = If[ξ == 0, Row[{Style["- j", Italic], numIt[Ωn, 6, 4]}], If[ξ < 1, Row[{numIt[-Ωn ξ, 6, 4], " - ",
         Style["j", Italic], numIt[Ωn Sqrt[1 - ξ^2], 6, 4]}], Row[{numIt[-Ωn ξ - Ωn Sqrt[ξ^2 - 1], 6, 4]}]]];
    {p1, p2}
   ];

  formatHzPoles[Ωn_, ξ_, T_] := Module[{p1, p2, t1, tt1, t2, tt2, t3, tt3},
    (*for  zeta < 1*)
    tt1 = N[Exp[-Ωn ξ T] Cos[Ωn T Sqrt[1 - ξ^2]]];
    tt2 = N[Exp[-Ωn ξ T] Sin[T Ωn Sqrt[1 - ξ^2]]];

    (*for  zeta > 1*)
```

```
  t1 = N[Exp[T (-Ωn ξ + Ωn Sqrt[ξ^2 - 1])]];
  t2 = N[Exp[T (-Ωn ξ - Ωn Sqrt[ξ^2 - 1])]];

  (*for  zeta = 1*)
  t3 = N[Exp[-Ωn ξ T + Ωn T Sqrt[ξ^2 - 1]]];
  tt3 = N[Exp[-Ωn ξ T - Ωn T Sqrt[ξ^2 - 1]]];

  p1 = If[ξ < 1, Row[{numIt[tt1, 6, 4], If[Sign[tt2] == 1, " + ", " - "], Style[" j", Italic],
       numIt[Abs[tt2], 6, 4]}], If[ξ > 1, Row[{numIt[t1, 6, 4]}], numIt[N[Exp[- Ωn T]], 6, 4]]];

  p2 = If[ξ < 1, Row[{numIt[tt1, 6, 4], If[Sign[tt2] == 1, " - ", " + "], Style["j", Italic],
       numIt[Abs[tt2], 6, 4]}], If[ξ > 1, Row[{numIt[t2, 6, 4]}], numIt[N[Exp[-Ωn T]], 6, 4]]];

  {p1, p2}
 ];

process[ξ_, Ωn_, Ts_] := Module[{s, Ω, plotOptions, hz, z, dtft, ω, fourierTransformPlot,
   dtftPlot, tfLaplace, tfFourier, ps1, ps2, absMag, fourierTransformPlotLabel,
   ps1Str, ps2Str, f1, f2, dtftPlotLabel, pz1Str, pz2Str, f3, f4, p1Coeff, p2Coeff},

  plotOptions = {ImageSize → {310}, ImageMargins → 0, ImagePadding → {{45, 55}, {20, 30}},
    AspectRatio → .45, PlotStyle → {Thick, Red}};

  ps1 = Chop[-Ωn ξ - Ωn Sqrt[-1 + ξ^2]]; ps2 = Chop[ -Ωn ξ + Ωn Sqrt[-1 + ξ^2] ];
  {ps1Str, ps2Str} = formatHsPoles[Ωn, Chop[ξ]];
  {pz1Str, pz2Str} = formatHzPoles[Ωn, Chop[ξ], Ts];
  tfLaplace = 1/(s^2 + 2 ξ Ωn s + Ωn^2);

  tfFourier = tfLaplace /. s → I Ω;
  absMag = ComplexExpand[Abs[tfFourier]];

  fourierTransformPlotLabel = Grid[{
     {Graphics[Text@Style["H(s)", Italic, 12], ImageSize → {35, 30}],
      Graphics[Text["1"/If[ξ > 0, Style[Row[{Style["s", Italic]^2, " +",
             ToString[NumberForm[2 ξ Ωn, {6, 4}, NumberPadding → {" ", "0"}]], Style[" s", Italic], " +",
             ToString[NumberForm[Ωn^2, {6, 4}, NumberPadding → {" ", "0"}]]}], ImageMargins → 0], 12],
          Style[Row[{Style["s^2", Italic], " + ", ToString[NumberForm[Ωn^2, {6, 4},
               NumberPadding → {" ", "0"}]]}]]]], ImageSize → {280, 30}],
      SpanFromLeft}, {Graphics[Text[Style["poles", 12]], ImageSize → {35, 30}],
      Graphics[Text@Style[ps1Str, 12], ImageSize → {110, 30}],
      Graphics[Text@Style[ps2Str, 12], ImageSize → {110, 30}]}
    }, Frame → All, Alignment → Center, Spacings → {0, 0}];

  fourierTransformPlot = Framed@Plot[20 Log[10, absMag], {Ω, 0, 2 Pi},
     AxesOrigin → {0, 0},
     Evaluate[plotOptions],
     PlotRange → {{0, 2 Pi}, {-40, 80}},
     AxesLabel → {Style[Text@TraditionalForm@Row[{"Ω ", "(rad/sec)"}], 10],
       Style[Text@TraditionalForm@Row[{"20 log|", H_a, "(", Style["j", Italic], " Ω)|", " (db)"}], 10]},
     PlotLabel → Style["continuous-time Fourier transform magnitude spectrum", Bold, 12]];

  If[ξ == 1, hz = (Ts z Exp[ps1 Ts])/(Exp[ps1 Ts] - z)^2, hz = (Ts z)/(z - Exp[ps1 Ts]) + (Ts z)/(z - Exp[ps2 Ts])];
```

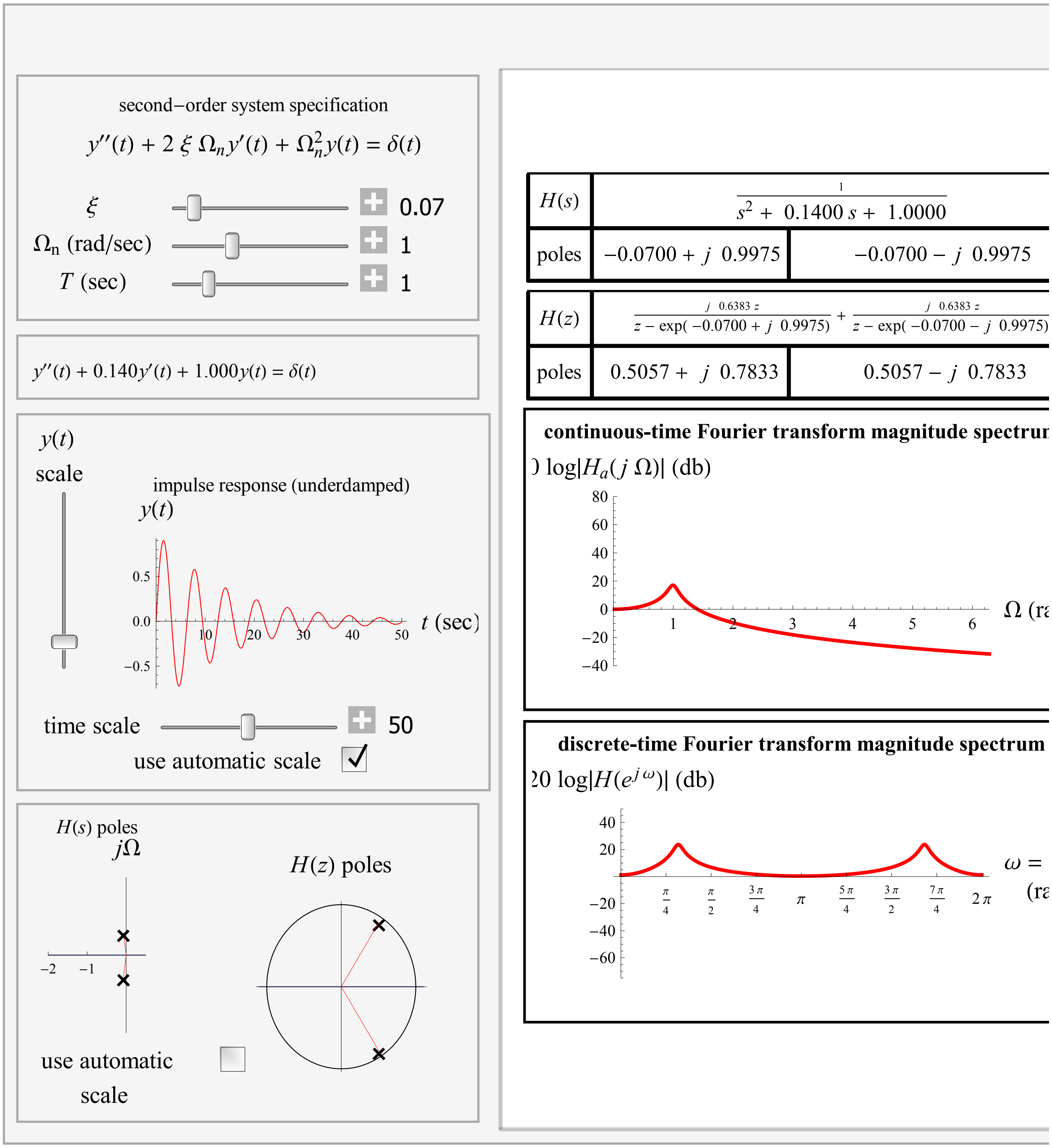
```
If[ξ ≠ 1, {f1 = (-Ts Exp[ξ Ωn]) / (2 Sin[Ωn Sqrt[1 - ξ^2]]); f2 = (Ts Exp[ξ Ωn]) / (Exp[Ωn Sqrt[ξ^2 - 1]] - Exp[-Ωn Sqrt[ξ^2 - 1]]);
   f3 = -f1; f4 = (Ts Exp[ξ Ωn]) / (Exp[-Ωn Sqrt[ξ^2 - 1]] - Exp[Ωn Sqrt[ξ^2 - 1]])}];

p1Coeff = numIt[If[ξ < 1, Abs[f1], If[ξ > 1, Abs[f2], 0]], 6, 4];
p2Coeff = numIt[If[ξ < 1, Abs[f3], If[ξ > 1, Abs[f4], 0]], 6, 4];

{ps1Str, ps2Str} = formatHsPoles[Ωn, Chop[ξ], Ts];
dtftPlotLabel = Grid[{
    {Graphics[Text@Style["H(z)", Italic, 12], ImageSize → {35, 30}],
     Graphics[
       If[ξ == 1, Style[Text@Row[{Row[{numIt[Ts, 6, 4], Style[" z ", Italic], "exp", "(", ps1Str, ")"}] /
             Row[{"(", Style["z", Italic], " - ", "exp", "(", ps1Str, ") )^2"}]}], 9], Text@Row[{Row[
             {If[ξ < 1, If[f1 < 0, Style["j ", Italic], Style["- j ", Italic]], If[f2 < 0, "- "]], p1Coeff,
              Style[" z", Italic]}] / Style[Row[{Style["z", Italic], " - ", "exp", "(", ps1Str, ")"}], 9],
           If[ξ < 1, If[f3 < 0, " - ", " + "], If[f4 < 0, " - ", " + "]], Row[{If[ξ < 1,
               Style[" j ", Italic]], p2Coeff, Style[" z", Italic]}] / Style[Row[{Style["z", Italic],
               " - ", "exp", "(", ps2Str, ")"}], 9]}]], ImageSize → {280, 30}], SpanFromLeft},
    {Graphics[Text@Style["poles", 12], ImageSize → {35, 30}], Graphics[Text@Style[pz1Str, 12],
       ImageSize → {110, 30}], Graphics[Text@Style[pz2Str, 12], ImageSize → {110, 30}]}
   }, Frame → All, Alignment → Center, Spacings → {0, 0}];

dtft = hz /. z → Exp[I ω];
absMag = ComplexExpand[Abs[dtft]];
dtftPlot = Framed@Plot[20 Log[10, absMag], {ω, 0.001, 2 Pi},
    Evaluate[plotOptions],
    AxesOrigin → {0, 0},
    PlotRange → {All, {-75, 50}},
    AxesLabel → {Style[Grid[{{Text@Row[{"ω = Ω ", Style[" T", Italic]}]},
          {Text@TraditionalForm["(rad)"]}}, Spacings → {0, 0}], 10],
       Style[Text@Row[{"20 log|", Style["H", Italic], "(e^(j ω))|", " (db)"}], 10]},
    Ticks → {{0, Pi / 4, Pi / 2, 3 / 4 Pi, Pi, 5 / 4 Pi, 6 / 4 Pi, 7 / 4 Pi, 2 Pi}, Automatic},
    PlotLabel → Style["discrete-time Fourier transform magnitude spectrum", Bold, 12]];

Grid[
  {{fourierTransformPlotLabel}, {dtftPlotLabel}, {fourierTransformPlot}, {dtftPlot}},
  Alignment → Center, Spacings → {0, .5}, Frame → None
 ]]}
]
```



Caption

This Demonstration illustrates the impulse invariance method used to convert an analog to a discrete system representation. The analog system consisting of the Laplace transfer function $H(s)$ is converted to the discrete system $H(z)$, the Z transfer function. This analog system is the response of a standard second-order system (with damping and stiffness) to a given impulse with zero initial conditions. The functions $H(s)$ and $H(z)$ are displayed with their pole locations.

Thumbnail



Snapshots

Details

(optional)

In this Demonstration, $j = \sqrt{-1}$.

Using the impulse invariance method, $H(z)$ is directly generated from $H(s)$ using a mapping that depends on the sampling period and the locations of the poles of $H(s)$. Because the input is an impulse, the system transfer function $H(s)$ is the same as the Laplace transform of the response $y(t)$.

The method starts by expressing the Laplace transfer function $H(s)$ in partial-fraction form $H(s) = \sum_{i=1}^{N} \frac{A_i}{s-p_i}$, where the N poles are located at the points p_i. Then the discrete system can be written as $H(z) = \sum_{i=1}^{N} \frac{T A_i}{1-z^{-1} \exp[T p_i]}$, where T is the sampling period for the analog system. This formula applies when the poles p_i are all distinct. In the case of a pole of order two, which pertains to the damping ratio $\xi = 1$, $H(z) = \sum_{i=1}^{N} \frac{T z \exp[T p_i]}{(\exp[T p_i]-z)^2}$.

Plots of the magnitude of the frequency response are generated for both $H(s)$ and $H(z)$ to compare the effect of changing the system parameters, including the sampling period T. Aliasing effects (which occur in the impulse invariance method) can be observed by making T larger and comparing the frequency response shapes of the analog and discrete systems. In these plots, the frequency axis (x axis) is a linear scale (not the more usual logarithmic scale) to better illustrate the method.

Locations of the poles of $H(s)$ and $H(z)$ show that stable poles in the left s-half-plane are mapped to stable poles inside the unit circle in z-space.

This Demonstration can also be used to analyze the impulse response of a second-order system as the system's natural frequency and damping ratio are varied.

You can change the system damping ratio ξ, the natural frequency Ω_n, and the sampling period T to observe how the poles of $H(s)$ and $H(z)$ change. The analytic forms of $H(s)$ and $H(z)$ are displayed at the top-center of the graphic with the numerical values of the poles. The system response $y(t)$ is plotted, with the option to scale the x axis and the y axis manually.

Reference: A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Upper Saddle River, NJ: Prentice Hall, 1975 pp. 201–203.

Control Suggestions (optional)

- [x] Resize Images
- [ ] Rotate and Zoom in 3D
- [ ] Drag Locators
- [ ] Create and Delete Locators
- [x] Slider Zoom
- [ ] Gamepad Controls
- [x] Automatic Animation
- [ ] Bookmark Animation

Search Terms (optional)

impulse invariance

Laplace transform

Z transform

second order system

impulse response

pole

stable

unit circle

frequency response

Related Links (optional)

Laplace transform

Z transform

Authoring Information

Contributed by: Nasser M. Abbasi

