

The **bodeplot** package

version 3.0.1

Rushikesh Kamalapurkar
rlkamalapurkar@gmail.com

November 21, 2025

Contents

1	Introduction	2
1.1	External Dependencies	2
1.2	Directory Structure	2
1.3	Limitations	2
2	TL;DR	3
3	Usage	10
3.1	Bode plots	10
3.1.1	Basic components up to first order	16
3.1.2	Basic components of the second order	18
3.2	Nyquist plots	19
3.3	Nichols charts	22
3.4	Pole-zero maps	23
4	Implementation	25
4.1	Initialization	25
4.2	PGF keys interface	27
4.3	Parametric function generators for poles, zeros, gains, and delays.	38
4.4	Second order systems.	39
4.5	Commands for Bode plots	40
4.5.1	User macros	40
4.5.2	Internal macros	48
4.6	Nyquist plots	55
4.6.1	User macros	55
4.6.2	Internal macros	59
4.7	Nichols charts	60
4.8	Pole-zero maps	63
4.8.1	User macros	63
4.8.2	Internal macros	66
A	TL;DR for package version 2.1.1 and earlier	72

1 Introduction

Generate Bode, Nyquist, and Nichols plots for transfer functions in the canonical (TF) form

$$G(s) = e^{-Ts} \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0} \quad (1)$$

and the zero-pole-gain (ZPK) form

$$G(s) = K e^{-Ts} \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)}. \quad (2)$$

In the equations above, b_m, \dots, b_0 and a_n, \dots, a_0 are real coefficients, $T \geq 0$ is the loop delay, z_1, \dots, z_m and p_1, \dots, p_n are complex zeros and poles of the transfer function, respectively, and $K \in \mathbb{R}$ is the loop gain.

For transfer functions in the ZPK format in (2) *with zero delay*, this package also supports linear and asymptotic approximation of Bode plots.

All phase plots use degrees as units. Use the `rad` package option or the optional argument `phase unit=rad` to generate plots in radians. The `phase unit` key accepts either `rad` or `deg` as inputs.

Frequency inputs and outputs are in radians per second. Use the `Hz` package option or the optional argument `frequency unit=Hz` to generate plots in Hertz. The `frequency unit` key accepts either `rad` or `Hz` as inputs.

1.1 External Dependencies

By default, the package uses `gnuplot` to do all the computations. If `gnuplot` is not available, the `pgf` package option can be used to do the calculations using the native `pgf` math engine. Compilation using the `pgf` math engine is typically slower, but the end result should be the identical (other than phase wrapping in the TF form, see limitations below).

1.2 Directory Structure

Since version 1.0.8, the `bodeplot` package places all `gnuplot` temporary files in the working directory. The package option `declutter` restores the original behavior where the temporary files are placed in a folder called `gnuplot`.

1.3 Limitations

- In `pgf` mode, Bode phase plots and Nichols charts in TF form wrap angles so that they are always between -180° and 180° or $-\pi$ and π radian. As such, these plots will show phase wrapping discontinuities. Since v1.1.1, in `gnuplot` mode, the package uses the `smooth unwrap` filter to correct wrapping discontinuities. As of now, I have not found a way to do this in `pgf` mode, any merge requests or ideas you may have are welcome! Since v1.1.4, you can redefine the `n@mod` macro using the commands `\makeatletter\renewcommand{\n@mod}{\n@mod@p}\makeatother` to wrap the phase between 0 and 360° or 0 and 2π radian. The commands `\makeatletter\renewcommand{\n@mod}{\n@mod@n}\makeatother` will wrap the phase between -360 and 0° or -2π and 0 radian.
- Use of the `declutter` option with other directory management tools such as a `tikzexternalize` prefix is not recommended.

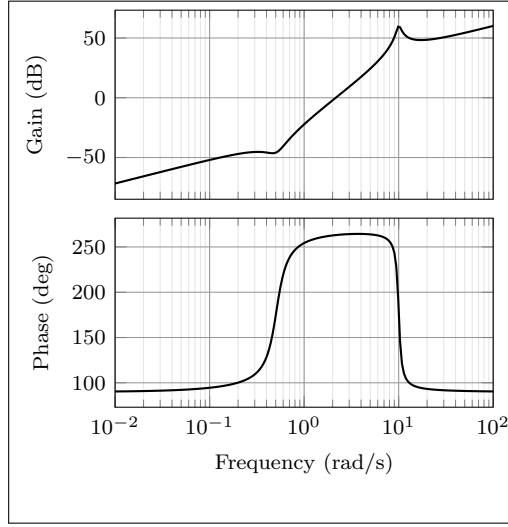
2 TL;DR

All Bode plots in this section are for the transfer function

$$G(s) = 10 \frac{s(s + 0.1 + 0.5i)(s + 0.1 - 0.5i)}{(s + 0.5 + 10i)(s + 0.5 - 10i)} = \frac{s(10s^2 + 2s + 2.6)}{(s^2 + s + 100.25)}. \quad (3)$$

Some of the examples include a transport delay. The examples use PGF-style key-value options and natural syntax for complex numbers in zeros and poles introduced in v3.0. For examples that show the legacy syntax (version 2.1.1 and earlier), please refer to Section A.

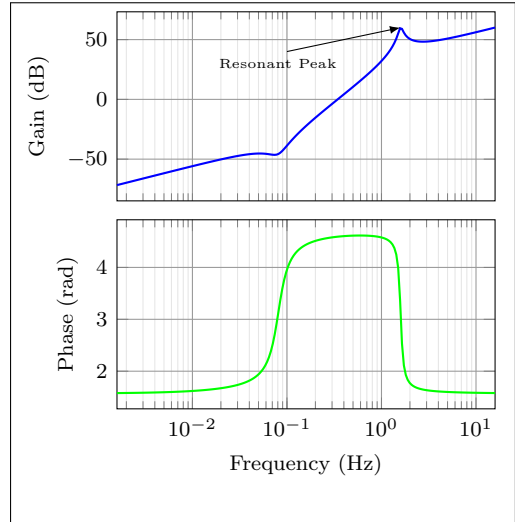
Bode plot in ZPK format



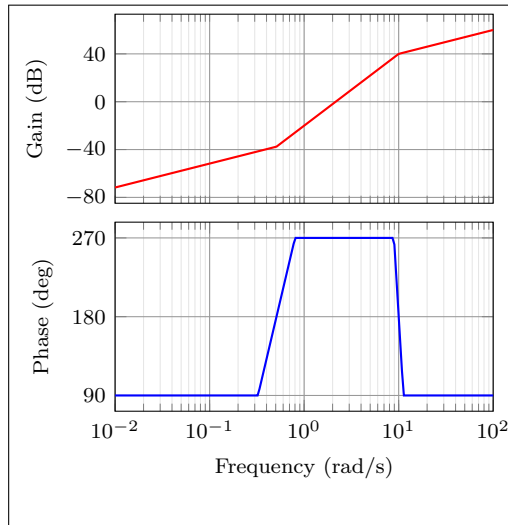
```
\BodeZPK[domain=0.01:100]{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10%
}
```

Same Bode plot over the same frequency range but supplied in Hz, in TF format with arrow decoration, transport delay, unit, and color customization (the phase plot may show wrapping if the **pgf** package option is used)

```
\BodeTF[%
domain=0.01/(2*pi):100/(2*pi),
samples=1000,
mag plot={blue,thick},
ph plot={green,thick},
tikz={%
>=latex,
phase unit=rad,
frequency unit=Hz%
},
mag commands={
\draw[>](axis cs:0.1,40) -- (axis cs:{10/(2*pi)},60);
\node at (axis cs: 0.08,30) {\tiny Resonant Peak};
}%
}
{%
numerator={10,2,2.6,0},
denominator={1,1,100.25}%
}
```



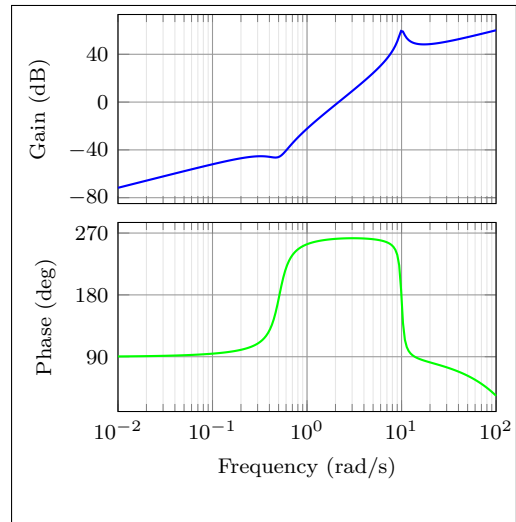
Linear approximation with customization



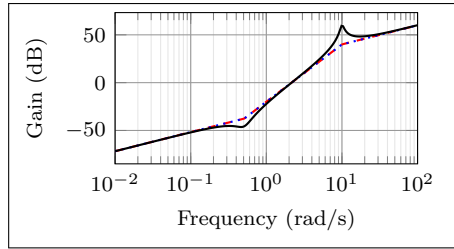
```
\BodeZPK[%
domain=0.01:100,
mag plot={red,thick},
ph plot={blue,thick},
mag axes={ytick distance=40},
ph axes={ytick distance=90},
approx=linear%
]{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10%
}
```

Plot with delay and customization

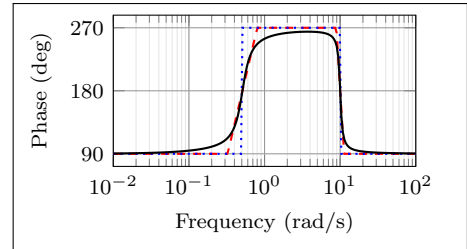
```
\BodeZPK[%
domain=0.01:100,
mag plot={blue,thick},
ph plot={green,thick},
mag axes={ytick distance=40},
ph axes={ytick distance=90}%
]{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10,
delay=0.01%
}
```



Individual gain and phase plots with more customization



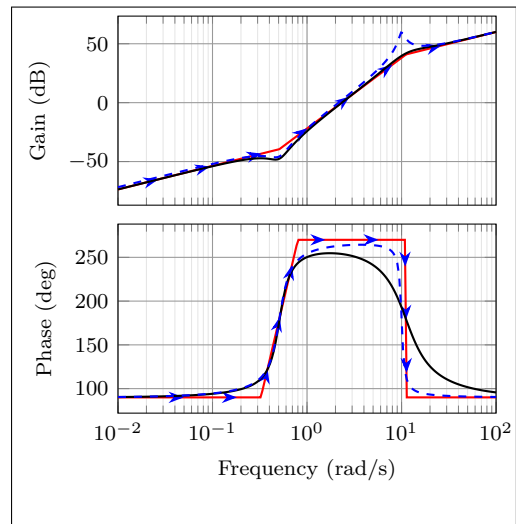
```
\begin{BodeMagPlot}{%
  axes={height=2cm,
    width=4cm},
  domain=0.01:100%
}
\addBodeZPKPlots{%
  true={black,thick},
  linear={red,dashed,thick},
  asymptotic={blue,dotted,thick}%
}
{magnitude}
{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
}\end{BodeMagPlot}
```



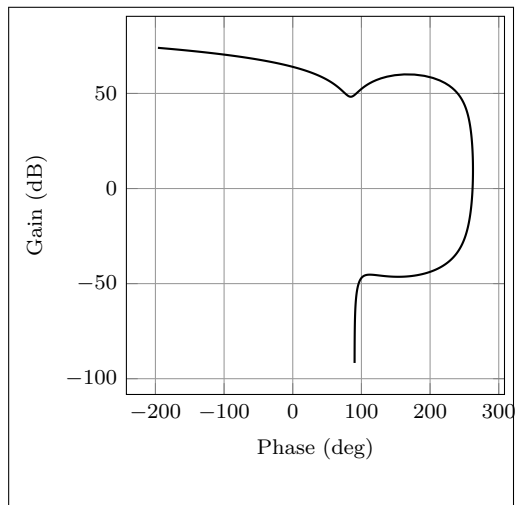
```
\begin{BodePhPlot}{%
  height=2cm,
  width=4cm,
  ytick distance=90,
  domain=0.01:100%
}
\addBodeZPKPlots{%
  true={black,thick},
  linear={red,dashed,thick},
  asymptotic={blue,dotted,thick}%
}
{phase}
{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
}\end{BodePhPlot}
```

Multiple transfer functions in a single Bode plot using the **BodePlot** environment and the **\addBodePlot** macro introduced in v2.1.

```
\begin{BodePlot}[domain=0.01:100]
\addBodePlot[red,postaction=decorate,
  decoration={%
    markings,
    mark=between positions 0.1 and 0.9 step 2em with {%
      \arrow{Stealth}[length=2mm,blue]}
    },linear]{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-5-10i,-5+10i},
  gain=10%
}
\addBodePlot[black,thick]{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-5-10i,-5+10i},
  gain=10%
}%
\addBodePlot[blue,dashed]{%
  numerator={10,2,2,6,0},
  denominator={1,1,100,25}%
}
}\end{BodePlot}
```



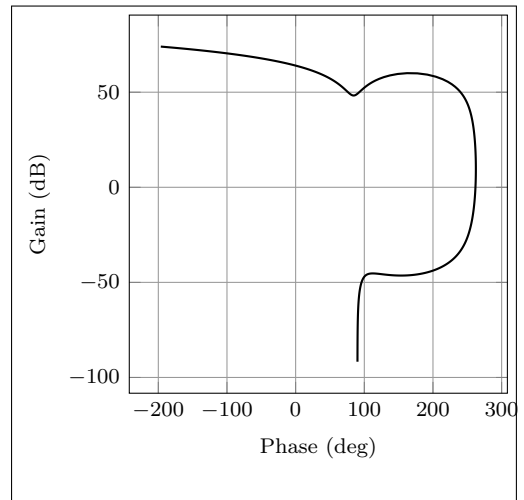
Nichols chart



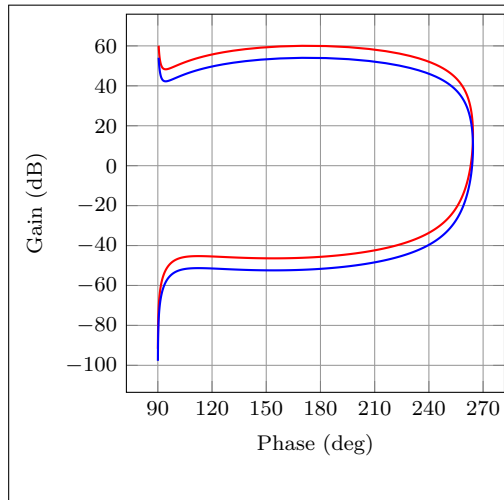
```
\NicholsZPK[domain=0.001:500,samples=1000]
{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10,
delay=0.01%
}
```

Same Nichols chart in TF format (may show wrapping in **pgf** mode)

```
\NicholsTF[domain=0.001:500,samples=1000]
{%
numerator={10,2,2.6,0},
denominator={1,1,100.25},
delay=0.01%
}
```



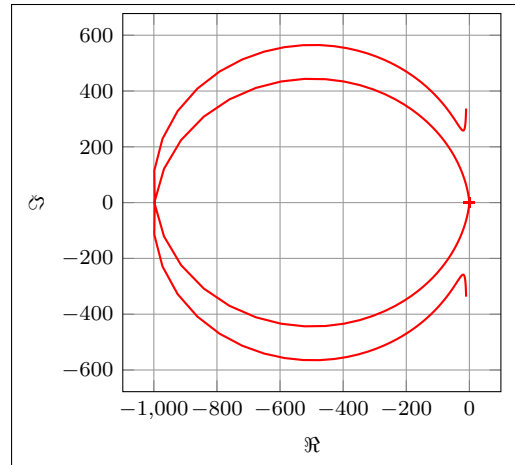
Multiple Nichols charts with customization



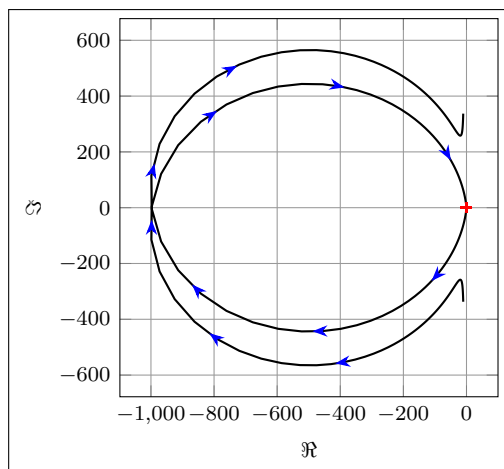
```
\begin{NicholsChart}{%
  ytick distance=20,
  xtick distance=30,
  domain=0.001:100%
}
\addNicholsZPKChart [red,samples=1000] {%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
\addNicholsZPKChart [blue,samples=1000] {%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=5%
}
\end{NicholsChart}
```

Nyquist plot

```
\NyquistZPK[domain=-30:30,plot={red,thick,samples=1000}]
{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
```



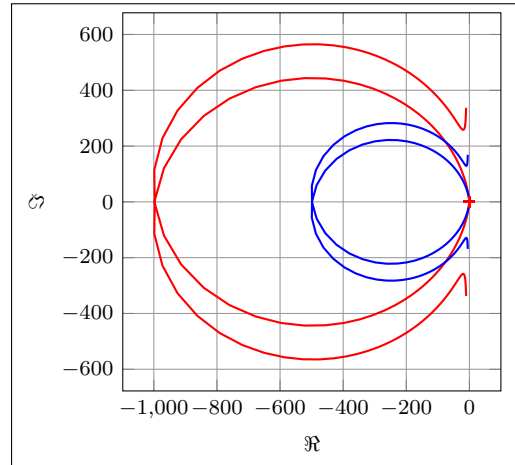
Nyquist plot in TF format with arrows



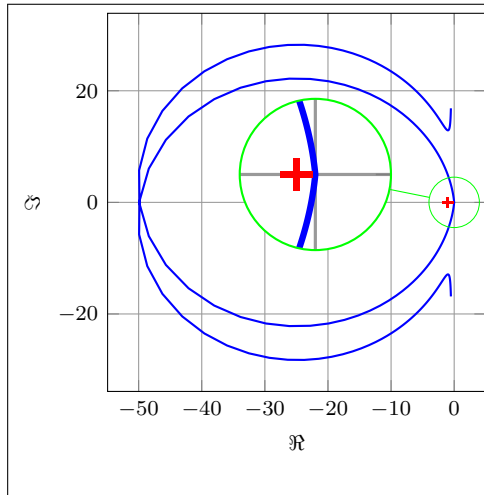
```
\NyquistTF[
  domain=-30:30,
  plot={%
    samples=1000,
    postaction=decorate,
    decoration={
      markings,
      mark=between positions 0.1 and 0.9 step 5em with {%
        \arrow{Stealth [length=2mm, blue]}
      }
    }
  }%
]{%
  numerator={10,2,2.6,0},
  denominator={1,1,100.25}%
}
```

Multiple Nyquist plots with customization

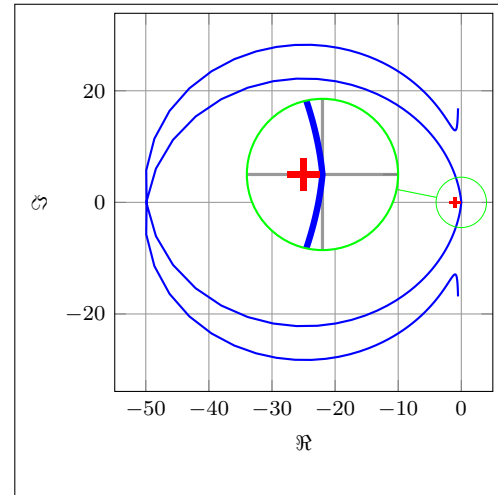
```
\begin{NyquistPlot}[domain=-30:30]
\addNyquistZPKPlot [red,samples=1000] {%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10%
}
\addNyquistZPKPlot [blue,samples=1000] {%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=5%
}
\end{NyquistPlot}
```



Nyquist plots with additional commands, using two different macros

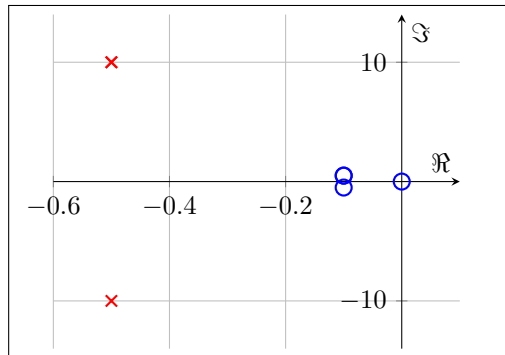


```
\begin{NyquistPlot}[%
tikz={
spy using outlines={%
circle,
magnification=3,
connect spies,
size=2cm
}
},
domain=-30:30%
]
\addNyquistZPKPlot [blue,samples=1000] {%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=0.5%
}
\coordinate (spyon) at (axis cs:0,0);
\coordinate (spyat) at (axis cs:-22,5);
\spy [green] on (spyon) in
node [fill=white] at (spyat);
\end{NyquistPlot}
```



```
\NyquistZPK[%
plot={blue,samples=1000},
tikz={
spy using outlines={%
circle,
magnification=3,
connect spies,
size=2cm
}
},
commands={
\coordinate (spyon) at (axis cs:0,0);
\coordinate (spyat) at (axis cs:-22,5);
\spy [green] on (spyon) in
node [fill=white] at (spyat);
},
domain=-30:30%
]
{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=0.5%
}
```

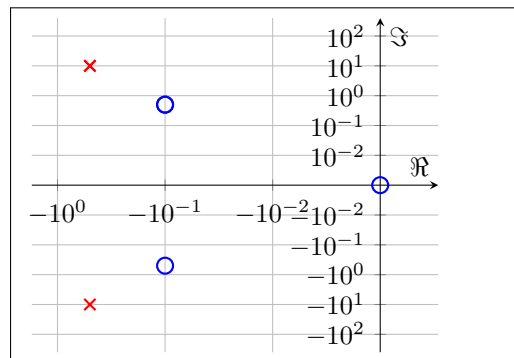

Pole-zero map



```
\PoleZeroMapZPK
{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
```

Pole-zero map (symmetric log scale)

```
\PoleZeroMapZPK[scale={log}]
{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
```



3 Usage

In all the macros described here, the frequency limits supplied by the user are assumed to be in **rad/s** unless either the **Hz** package option is used or the optional argument **frequency unit=Hz** is supplied by the user. All phase plots are generated in degrees unless either the **rad** package option is used or the optional argument **phase unit=rad** is supplied by the user.

3.1 Bode plots

\BodeZPK **\BodeZPK**[*<options>*]{*<zpk-spec>*}

Plots the Bode plot of a transfer function given in ZPK format using the **groupplot** environment.

zpk-spec: A pgfkeys-style specification using keys **zeros={...}**, **poles={...}**, **gain=...**, and **delay=...** within the **/bodeplot/zpk/** family. Zeros and poles are comma-separated lists with complex poles and zeros entered using the format **a+bi**.

options: The optional argument supports a comma-separated list of options in the form **key=value**. The following keys are supported:

- **domain=min:max:** Sets the frequency range, assumed to be **0.01:100** if not specified.
- **plot={opt}:** Options passed to **\addplot** for both magnitude and phase plots.
- **mag plot={opt}:** Options passed only to the magnitude plot.
- **ph plot={opt}:** Options passed only to the phase plot.
- **axes={opt}:** Options passed to **\nextgroupplot** for both plots.
- **mag axes={opt}:** Options passed only to magnitude axis.
- **ph axes={opt}:** Options passed only to phase axis.
- **group={opt}:** Options passed to the **groupplot** environment.
- **tikz={opt}:** Options passed to the **tikzpicture** environment.
- **commands={opt}:** TikZ commands added after both plots.
- **mag commands={opt}:** TikZ commands added after magnitude plot.
- **ph commands={opt}:** TikZ commands added after phase plot.
- **approx=<true/linear/asymptotic>:** Approximation type for the plot.
- **prefix={name}:** Prefix for gnuplot temporary files.
- **phase unit=<deg/rad>:** Phase axis units.
- **frequency unit=<rad/Hz>:** Frequency axis units.
- Any other key-value pair is passed as plot options to both plots.

For example, given a transfer function

$$G(s) = 10 \frac{s(s + 0.1 + 0.5i)(s + 0.1 - 0.5i)}{(s + 0.5 + 10i)(s + 0.5 - 10i)}, \quad (4)$$

its Bode plot over the frequency range $[0.01, 100]$ can be generated using

```
\BodeZPK [domain=0.01:100, blue, thick]
{zeros={0, -0.1-0.5i, -0.1+0.5i},
poles={-0.5-10i, -0.5+10i},
gain=10}
```

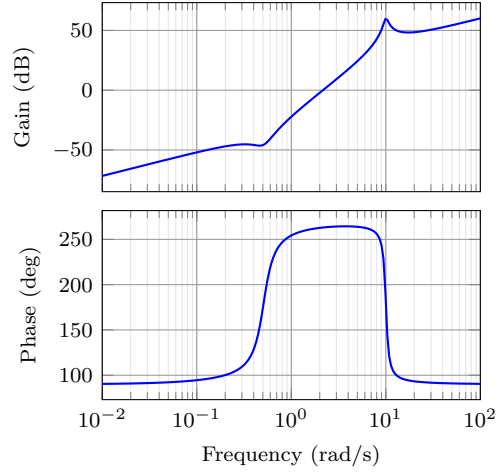


Figure 1: Output of the `\BodeZPK` macro.

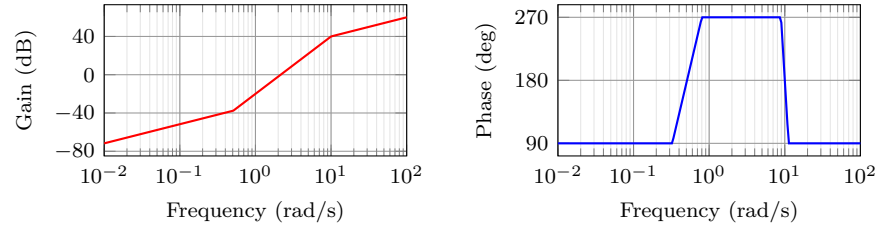


Figure 2: Customization of the default `\BodeZPK` macro.

which generates the plot in Figure 1. In this example, a delay is not specified, so it is assumed to be zero. A gain is not specified, so it is assumed to be 1. A single comma-separated list of options `[blue,thick]` is passed, so it is passed on to the `\addplot` commands in both the magnitude and the phase plots. The default plots are thick black lines and each of the axes, excluding ticks and labels, are 5cm wide and 2.5cm high.

The width and the height, along with other properties of the plots, the axes, and the group can be customized using native `pgf` keys. For example, a linear approximation of the Bode plot with customization of the plots, the axes, and the group can be generated using

```
\BodeZPK[%
  mag plot={red,thick},
  ph plot={blue,thick},
  mag axes={ytick distance=40,xmajorticks=true,xlabel={Frequency (rad/s)}},
  ph axes={ytick distance=90},
  group={group style={group size=2 by 1,horizontal sep=2cm,width=4cm,height=2cm}},
  approx=linear]
{zeros={0,-0.1-0.5i,-0.1+0.5i},poles={-0.5-10i,-0.5+10i},gain=10}
```

which generates the plot in Figure 2.

Legacy interface (v2.1.1 and earlier):

`\BodeZPK [options]{zpk-spec}{min-freq}{max-freq}`

If both frequency arguments are provided, the macro falls back to the legacy format. This format is supported for backward compatibility with versions 2.1.1 and earlier:

zpk-spec (v2.1.1 and earlier): `z/{zeros},p/{poles},k/{gain},d/{delay}` where zeros and poles are comma-separated lists of complex numbers of the form `{{real part 1,imaginary part 1},{real part 2,imaginary part 2},...}`. If the imaginary part is not provided, it is assumed to be zero. This format requires the frequency arguments `min-freq` and `max-freq` to be provided.

options (v2.1.1 and earlier): A comma-separated list of options that modify the plots. The following options are supported:

- **plot/typ/{opt}**: modify plot properties by adding options **{opt}** to the **\addplot** macro for the magnitude plot if **typ** is **mag** and the phase plot if **typ** is **ph**.
- **axes/typ/{opt}**: modify axis properties by adding options **{opt}** to the **\nextgroupplot** macro for the magnitude plot if **typ** is **mag** and the phase plot if **typ** is **ph**.
- **commands/typ/{opt}**: add any valid TikZ commands to the magnitude plot if **typ** is **mag** and the phase plot if **typ** is **ph**.
- **plot/{opt}**: adds options **{opt}** to **\addplot** macros for both the magnitude and the phase plots.
- **axes/{opt}**: adds options **{opt}** to **\nextgroupplot** macros for both the magnitude and the phase plots.
- **group/{opt}**: adds options **{opt}** to the **groupplot** environment.
- **tikz/{opt}**: adds options **{opt}** to the **tikzpicture** environment.
- **approx/linear**: plots linear approximation.
- **approx/asymptotic**: plots asymptotic approximation.
- **prefix/{opt}**: adds the string **opt** as a prefix to all temporary files generated by **gnuplot** for this plot.
- **{opt}** (bare options): add all of the supplied options to **\addplot** macros for both the magnitude and the phase plots.

Example of the legacy interface:

```
\BodeZPK [blue,thick]
  {z/{0},{-0.1,-0.5},{-0.1,0.5}},p/{{-0.5,-10},{-0.5,10}},k/10}
  {0.01}{100}
```

\BodeTF **\BodeTF**[*{options}*]{*{tf-spec}*}

Plots the Bode plot of a transfer function given in TF format.

tf-spec: The mandatory argument contains the pgf keys **numerator={coeffs}**, **denominator={coeffs}**, and optionally **delay=value**. The coefficients are entered as a comma-separated list, in order from the highest degree of s to the lowest, with zeros for missing degrees.

options: The frequency range is specified in the first optional argument using **domain=min:max**. The optional argument can also contain any of the keys supported by the **\BodeZPK** macro except for the **approx** key since linear/asymptotic approximation is not supported for TF format.

For example, given the same transfer function as (4) in TF form and with a small transport delay,

$$G(s) = e^{-0.01s} \frac{s(10s^2 + 2s + 2.6)}{(s^2 + s + 100.25)}, \quad (5)$$

its Bode plot over the frequency range [0.01, 100] can be generated using

```
\BodeTF[%
  domain=0.01:100,
  mag commands={\node at (axis cs: 2.1,0) [circle,fill,inner sep=0.05cm,
    label=below:{\omega_{gc}}]{};}}
  {numerator={10,2,2.6,0}, denominator={1,1,100.25}, delay=0.01}
```

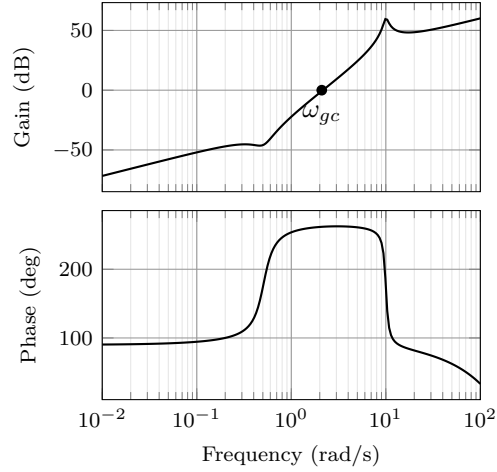


Figure 3: Output of the `\BodeTF` macro with an optional TikZ command used to mark the gain crossover frequency.

which generates the plot in Figure 3. Note the 0 added to the numerator coefficients to account for the fact that the numerator does not have a constant term in it. Note the semicolon after the TikZ command passed to the `\commands` option.

Legacy interface (v2.1.1 and earlier):

`\BodeTF` [*options*]{*tf-spec*}{*min-freq*}{*max-freq*}

If both frequency arguments are provided, the macro falls back to the legacy format. The three mandatory arguments include: (1) a list of tuples comprised of the coefficients in the numerator and the denominator of the transfer function and the transport delay (format: `num/{coeffs},den/{coeffs},d/{delay}`), (2) the lower end of the frequency range, and (3) the upper end of the frequency range. The coefficients are entered as a comma-separated list, in order from the highest degree of s to the lowest, with zeros for missing degrees. The optional arguments are the same as `\bodeZPK` except that linear/asymptotic approximation is not supported, so `approx/...` is ignored. Example of the legacy interface:

```
\BodeTF[%
  commands/mag/{\node at (axis cs: 2.1,0) [circle,fill,inner sep=0.05cm,
    label=below:{\omega_{gc}}]{};}}
{num/{10,2,2.6,0},den/{1,1,100.25},d/0.01}
{0.01}{100}
```

`BodeMagPlot` (*env.*) `\begin{BodeMagPlot}` [*options*]

`\addBode...`
`\end{BodeMagPlot}`

The `BodeMagPlot` environment works in conjunction with the parametric function generator macros `\addBodeZPKPlots`, `\addBodeTFPlot`, and `\addBodeComponentPlot`, intended to be used for magnitude plots. The first optional argument can contain keys from the `/bodeplot/env/` family, including `domain=min:max` to set the frequency range, `axes={...}` for axis options, `tikz={...}` for tikzpicture options, and `prefix={...}` for gnuplot file prefixes. The `prefix` option adds the provided string as a prefix to all temporary files generated by `gnuplot` for all plots in this environment. This option is useful if data generated by `gnuplot` for a specific plot needs to be uniquely identified. All other key-value pairs are passed as options to the `semilogaxis` environment. For examples, see the description of `\addBodeZPKPlots`, `\addBodeTFPlot`, and `\addBodeComponentPlot`.

Legacy interface (v2.1.1 and earlier):

`\begin{BodeMagPlot}` [*options*]{*min-freq*}{*max-freq*}

`\addBode...`
`\end{BodeMagPlot}`

If both frequency arguments are provided, the environment uses the legacy format. The first optional argument is comprised of a comma separated list of tuples, either `obj/{opt}` or just `{opt}`. Each tuple passes options to different `pgfplots` macros that generate the axes and the plots according to:

- Tuples of the form `obj/{opt}`:
 - `tikz/{opt}`: modify picture properties by adding options `{opt}` to the `tikzpicture` environment.
 - `axes/{opt}`: modify axis properties by adding options `{opt}` to the `semilogaxis` environment.
 - `prefix/{opt}`: adds the string `opt` as a prefix to all temporary files generated by `gnuplot` for all plots in this environment. This option is useful if data generated by `gnuplot` for a specific plot needs to be uniquely identified.
- Tuples of the form `{opt}` are passed directly to the `semilogaxis` environment.

The frequency limits are translated to the x-axis limits and the domain of the `semilogaxis` environment.

`BodePhPlot (env.) \begin{BodePhPlot}[\langle options \rangle]`
`\addBode...`
`\end{BodePhPlot}`

Same as `BodeMagPlot`, but intended to be used for phase plots.

`\addBodeZPKPlots \addBodeZPKPlots[\langle options \rangle]{\langle plot-type \rangle}{\langle zpk-spec \rangle}`

Generates the appropriate parametric functions and supplies them to multiple `\addplot` macros, one for each approximation type specified in the optional argument. If no optional argument is supplied, then a single `\addplot` command corresponding to a thick true Bode plot is generated. The `options` argument accepts keys `true={\langle plot-opts \rangle}`, `linear={\langle plot-opts \rangle}`, and `asymptotic={\langle plot-opts \rangle}`, where `\langle plot-opts \rangle` are TikZ/pgfplots styling options. Multiple approximations can be specified as `true={black,thick}`, `linear={red,dashed}`, etc. This macro can be used inside any `semilogaxis` environment as long as a domain for the x-axis is supplied in the environment or parent axis options. Use with the `BodeMagPlot` or `BodePhPlot` environments supplied with this package is recommended. The second mandatory argument, `plot-type` is either `magnitude` or `phase`. If it is not equal to `phase`, it is assumed to be `magnitude`. The third mandatory argument, `zpk-spec` is the same as that of the `\BodeZPK` macro described earlier.

For example, given the transfer function in (4), its linear, asymptotic, and true Bode plots can be superimposed using:

```
\begin{BodeMagPlot}[height=2cm,width=4cm,domain=0.01:100,xlabel={}]
  \addBodeZPKPlots[true={black,thick},
    linear={red,dashed,thick},
    asymptotic={blue,dotted,thick}]
    {magnitude}
    {zeros={0,-0.1-0.5i,-0.1+0.5i},poles={-0.5-10i,-0.5+10i},gain=10}
\end{BodeMagPlot}
\begin{BodePhPlot}[height=2cm,width=4cm,ytick distance=90,domain=0.01:100]
  \addBodeZPKPlots[true={black,thick},
    linear={red,dashed,thick},
    asymptotic={blue,dotted,thick}]
    {phase}
    {zeros={0,-0.1-0.5i,-0.1+0.5i},poles={-0.5-10i,-0.5+10i},gain=10}
\end{BodePhPlot}
```

which generates the plot in Figure 4.

For backward compatibility, the legacy format with explicit frequency arguments is also supported.

`\addBodeZPKPlots[\langle options \rangle]{\langle plot-type \rangle}{\langle zpk-spec \rangle}`

If an equals sign is not detected in the `zpk-spec`, then the macro falls back to the legacy format. In the legacy format, the macro generates the appropriate parametric

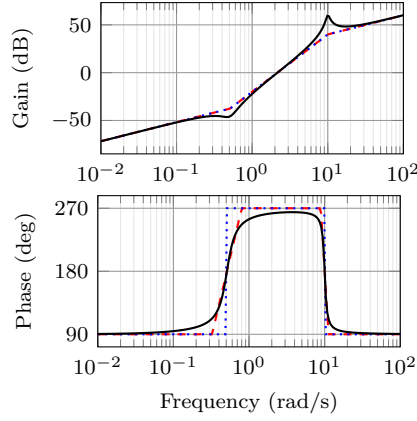


Figure 4: Superimposed approximate and true Bode plots using the `BodeMagPlot` and `BodePhPlot` environments and the `\addBodeZPKPlots` macro.

functions and supplies them to multiple `\addplot` macros, one for each `approx/{opt}` pair in the optional argument. If no optional argument is supplied, then a single `\addplot` command corresponding to a thick true Bode plot is generated. If an optional argument is supplied, it needs to be one of `true/{opt}`, `linear/{opt}`, or `asymptotic/{opt}`. For example, the plots in Figure 4 can also be generated using:

```
\begin{BodeMagPlot}[height=2cm,width=4cm] {0.01} {100}
  \addBodeZPKPlots[%
    true/{black,thick},
    linear/{red,dashed,thick},
    asymptotic/{blue,dotted,thick}]
    {magnitude}
    {z/{0,{-0.1,-0.5},{-0.1,0.5}},p/{{-0.5,-10},{-0.5,10}},k/10}
\end{BodeMagPlot}

\begin{BodePhPlot}[height=2cm, width=4cm, ytick distance=90] {0.01} {100}
  \addBodeZPKPlots[%
    true/{black,thick},
    linear/{red,dashed,thick},
    asymptotic/{blue,dotted,thick}]
    {phase}
    {z/{0,{-0.1,-0.5},{-0.1,0.5}},p/{{-0.5,-10},{-0.5,10}},k/10}
\end{BodePhPlot}
```

`\addBodeTFPlot` `\addBodeTFPlot[<options>]{<plot-type>}{<tf-spec>}`

Generates a single parametric function for either Bode magnitude or phase plot of a transfer function in TF form. The generated parametric function is passed to the `\addplot` macro. This macro can be used inside any `semilogaxis` environment as long as a domain for the x-axis is supplied through the `options` argument or in the optional argument of the container `semilogaxis` environment. Use with the `BodeMagPlot` and `BodePhPlot` environments supplied with this package is recommended. The `options` can include any options that are accepted by the `\addplot` macro. The mandatory argument `plot-type` is either `magnitude` or `phase`. If it is not equal to `phase`, it is assumed to be `magnitude`. The last mandatory argument supports both legacy format `num/{coeffs},den/{coeffs},d/{delay}` and new format (v3.0) using pgfkeys like `numerator={...},denominator={...},delay=...`. The frequency range is specified using `domain=min:max` in the `options` argument.

`\addBodeComponentPlot` `\addBodeComponentPlot[<plot-options>]{<plot-command>}`

Generates a single parametric function corresponding to the mandatory argument `plot-command` and passes it to the `\addplot` macro. The plot command can be any parametric function that uses `t` as the independent variable. The parametric function must be

gnuplot compatible (or pgfplots compatible if the package is loaded using the `pgf` option, **with angles passed to trigonometric functions in radian**). The intended use of this macro is to plot the parametric functions generated using the basic component macros described in Section 3.1.1 below.

`\addBodePlot` `\addBodePlot[$\langle plot-options \rangle$]{ $\langle system-data \rangle$ }`

Unified macro to add Bode plots for both ZPK and TF system representations. Unlike the `\addBodeZPKPlots` and `\addBodeTFPlots` macros, this macro can only be used inside a `BodePlot` environment. **It will not function if used in a generic semilogxaxis environment or in the BodeMagPlot and BodePhPlot environments.** For ZPK systems, the `system-data` has the same format `zeros={...},poles={...},gain=..., delay=...` as `\addBodeZPKPlots`. For TF systems, it has the same format `numerator={...},denominator={...},gain=...,delay=...` as `\addBodeTFPlot`. The optional arguments are the same as the `\addBodeTfPlot` macro, with two additions. Passing `linear` or `asymptotic` options will generate linear or asymptotic approximations if the system representation is ZPK. These two options are ignored for TF systems. The macro automatically detects the system representation based on the presence of `zeros` or `poles` keys (ZPK) versus `numerator` or `denominator` keys (TF) in the `system-data` argument.

Legacy format (v2.1.1 and earlier):

`\addBodePlot[$\langle plot-options \rangle$]{ $\langle system-type \rangle$ }{ $\langle system-data \rangle$ }`

If invoked using two mandatory arguments, the command falls back to the legacy format. In the legacy format, the command does not automatically detect the system representation. The second mandatory argument `system-type` should be either `zpk` or `tf` to specify the system representation.

`BodePlot (env.)` `\begin{BodePlot}[$\langle options \rangle$]`

`\addBodePlot...`

`\end{BodePlot}`

Starting from version 2.1, the deprecated `BodePlot` environment is revived exclusively to host multiple instances of the unified `\addBodePlot` macro. The `\addBodeZPKPlots` and `\addBodeTFPlot` macros **will not function inside this environment**. The purpose of this environment is to plot a Bode plot that contains both phase and magnitude plots for several different transfer functions.

The environment uses the `pgfkeys` interface from the `/bodeplot/combinedenv/` family, supporting keys like `domain=min:max`, `group={...}`, `approx=...`, `mag axes={...}`, `ph axes={...}`, `mag commands={...}`, `ph commands={...}`, `tikz={...}`, and `prefix={...}` similar to the `\BodeZPK` macro.

For example, the Bode plot of two different transfer functions, one in ZPK form and the other in TF form, can be generated using:

```
\begin{BodePlot}[domain=0.01:100,group={group style={horizontal sep=2cm}}]
  \addBodePlot[true={black,thick},linear={red,dashed,thick},asymptotic={blue,dotted,thick}]
    {zeros={0,-0.1-0.5i,-0.1+0.5i},poles={-0.5-10i,-0.5+10i},gain=10}
  \addBodePlot[blue,thick]
    {numerator={10,2,2.6,0}, denominator={1,1,100.25}, delay=0.01}
\end{BodePlot}
```

which generates the plot in Figure 5.

Legacy interface (v2.1.1 and earlier):

`\begin{BodePlot}[$\langle options \rangle$]{ $\langle min-freq \rangle$ }{ $\langle max-freq \rangle$ }`

`\addBodePlot...`

`\end{BodePlot}`

If both frequency arguments are provided, the environment falls back to the legacy interface, where the optional arguments are the same as the legacy interface for the `\BodeZPK` macro, supporting `plot/`, `axes/`, `group/`, `tikz/` options with `mag` and `ph` sub-types.

3.1.1 Basic components up to first order

`\TypeFeatureApprox` `\TypeFeatureApprox{ $\langle real-part \rangle$ }{ $\langle imaginary-part \rangle$ }`

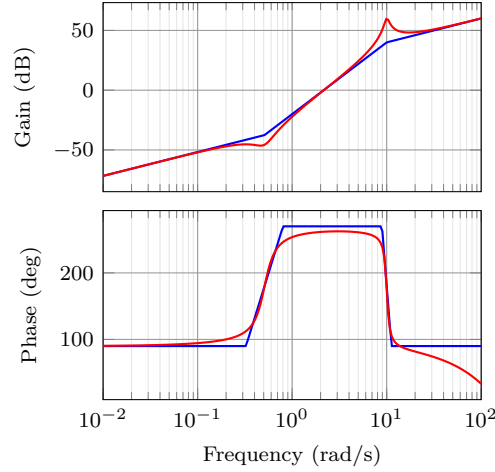


Figure 5: Bode plot combining ZPK and TF systems using the `BodePlot` environment and the unified `\addBodePlot` macro.

This entry describes 20 different macros of the form `\TypeFeatureApprox` that take the real part and the imaginary part of a complex number as arguments. The **Type** in the macro name should be replaced by either **Mag** or **Ph** to generate a parametric function corresponding to the magnitude or the phase plot, respectively. The **Feature** in the macro name should be replaced by one of **K**, **Pole**, **Zero**, or **Del**, to generate the Bode plot of a gain, a complex pole, a complex zero, or a transport delay, respectively. If the **Feature** is set to either **K** or **Del**, the `imaginary-part` mandatory argument is ignored. The **Approx** in the macro name should either be removed, or it should be replaced by **Lin** or **Asymp** to generate the true Bode plot, the linear approximation, or the asymptotic approximation, respectively. If the **Feature** is set to **Del**, then **Approx** has to be removed. For example,

- `\MagK{k}{0}` or `\MagK{k}{400}` generates a parametric function for the true Bode magnitude of $G(s) = k$
- `\PhPoleLin{a}{b}` generates a parametric function for the linear approximation of the Bode phase of $G(s) = \frac{1}{s-a-ib}$.
- `\PhDel{T}{200}` or `\PhDel{T}{0}` generates a parametric function for the Bode phase of $G(s) = e^{-Ts}$.

All 20 of the macros defined by combinations of **Type**, **Feature**, and **Approx**, and any `gnuplot` (or `pgfplot` if the `pgf` class option is loaded) compatible function of the 20 macros can be used as `plot-command` in the `addBodeComponentPlot` macro. This is sufficient to generate the Bode plot of any rational transfer function with delay. For example, the Bode phase plot in Figure 4 can also be generated using:

```
\begin{BodePhPlot}[ytick distance=90,domain=0.01:100]
  \addBodeComponentPlot[black,thick]{%
    \PhZero{0}{0} + \PhZero{-0.1}{-0.5} + \PhZero{-0.1}{0.5} +
    \PhPole{-0.5}{-10} + \PhPole{-0.5}{10} + \PhK{10}{0}}
  \addBodeComponentPlot[red,dashed,thick]{%
    \PhZeroLin{0}{0} + \PhZeroLin{-0.1}{-0.5} + \PhZeroLin{-0.1}{0.5} +
    \PhPoleLin{-0.5}{-10} + \PhPoleLin{-0.5}{10} + \PhKLin{10}{20}}
  \addBodeComponentPlot[blue,dotted,thick]{%
    \PhZeroAsymp{0}{0} + \PhZeroAsymp{-0.1}{-0.5} + \PhZeroAsymp{-0.1}{0.5} +
    \PhPoleAsymp{-0.5}{-10} + \PhPoleAsymp{-0.5}{10} + \PhKAsymp{10}{40}}
\end{BodePhPlot}
```

which gives us the plot in Figure 6.

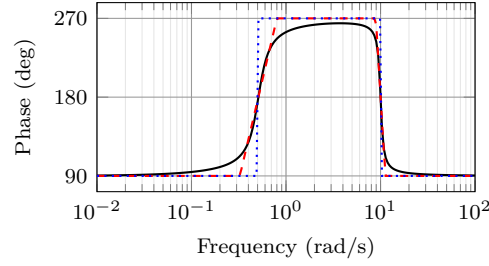


Figure 6: Superimposed approximate and true Bode Phase plot using the `BodePhPlot` environment, the `\addBodeComponentPlot` macro, and several macros of the `\TypeFeatureApprox` form.

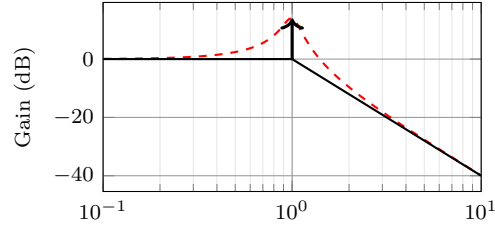


Figure 7: Resonant peak in asymptotic Bode plot using `\MagSOPolesPeak`.

3.1.2 Basic components of the second order

`\TypeS0FeatureApprox` `\TypeS0FeatureApprox{<a1>}{<a0>}`

This entry describes 12 different macros of the form `\TypeS0FeatureApprox` that take the coefficients a_1 and a_0 of a general second order system as inputs. The **Feature** in the macro name should be replaced by either **Poles** or **Zeros** to generate the Bode plot of $G(s) = \frac{1}{s^2 + a_1s + a_0}$ or $G(s) = s^2 + a_1s + a_0$, respectively. The **Type** in the macro name should be replaced by either **Mag** or **Ph** to generate a parametric function corresponding to the magnitude or the phase plot, respectively. The **Approx** in the macro name should either be removed, or it should be replaced by **Lin** or **Asymp** to generate the true Bode plot, the linear approximation, or the asymptotic approximation, respectively.

`\MagS0FeaturePeak` `\MagS0FeaturePeak[<draw-options>]{<a1>}{<a0>}`

This entry describes 2 different macros of the form `\MagS0FeaturePeak` that take the coefficients a_1 and a_0 of a general second order system as inputs, and draw a resonant peak using the `\draw` TikZ macro. The **Feature** in the macro name should be replaced by either **Poles** or **Zeros** to generate a peak for poles and a valley for zeros, respectively. For example, the command

```
\begin{BodeMagPlot}[xlabel={},domain=0.1:10]
  \addBodeComponentPlot[red,dashed,thick]{\MagSOPoles{0.2}{1}}
  \addBodeComponentPlot[black,thick]{\MagSOPolesLin{0.2}{1}}
  \MagSOPolesPeak[thick]{0.2}{1}
\end{BodeMagPlot}
```

generates the plot in Figure 7.

`\TypeCSFeatureApprox` `\TypeCSFeatureApprox{<zeta>}{<omega-n>}`

This entry describes 12 different macros of the form `\TypeCSFeatureApprox` that take the damping ratio, ζ , and the natural frequency, ω_n of a canonical second order system as inputs. The **Type** in the macro name should be replaced by either **Mag** or **Ph** to generate a parametric function corresponding to the magnitude or the phase plot, respectively. The **Feature** in the macro name should be replaced by either **Poles** or **Zeros** to generate the Bode plot of $G(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$ or $G(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$, respectively. The **Approx** in the macro name should either be removed, or it should be replaced by **Lin**

or **Asymp** to generate the true Bode plot, the linear approximation, or the asymptotic approximation, respectively.

`\MagCSFeaturePeak` `\MagCSFeaturePeak[$\langle draw-options \rangle$]{ $\langle zeta \rangle$ }{ $\langle omega-n \rangle$ }`

This entry describes 2 different macros of the form `\MagCSFeaturePeak` that take the damping ratio, ζ , and the natural frequency, ω_n of a canonical second order system as inputs, and draw a resonant peak using the `\draw` TikZ macro. The **Feature** in the macro name should be replaced by either **Poles** or **Zeros** to generate a peak for poles and a valley for zeros, respectively.

`\MagCCFeaturePeak` `\MagCCFeaturePeak[$\langle draw-options \rangle$]{ $\langle real-part \rangle$ }{ $\langle imaginary-part \rangle$ }`

This entry describes 2 different macros of the form `\MagCCFeaturePeak` that take the real and imaginary parts of a pair of complex conjugate poles or zeros as inputs, and draw a resonant peak using the `\draw` TikZ macro. The **Feature** in the macro name should be replaced by either **Poles** or **Zeros** to generate a peak for poles and a valley for zeros, respectively.

3.2 Nyquist plots

`\NyquistZPK` `\NyquistZPK[$\langle options \rangle$]{ $\langle zpk-spec \rangle$ }`

Generates the Nyquist diagram of a transfer function given in ZPK form and marks the critical point $(-1, 0)$ with a thick red $+$.

zpk-spec: Same as the `\BodeZPK` macro. Provide zeros, poles, gain, and optional delay with the pgfkeys-style format `zeros={...},poles={...},gain=...,delay=...`. Complex numbers are written as `a+bi`. The macro assumes unit gain and zero delay when those keys are omitted.

options:

- **domain=min:max:** Sets the frequency sweep for the contour. The default sweep is `-30:30` when the key is not provided.
- **plot={opt}:** Appends `opt` to the `\addplot` options that draw the Nyquist curve, e.g., `samples=2000`.
- **axes={opt}:** Adds `opt` to the surrounding `axis` environment.
- **commands={opt}:** Inserts `opt` immediately after the contour inside the `axis`, which is useful for annotations or additional plotting commands.
- **tikz={opt}:** Adds `opt` to the enclosing `tikzpicture`.
- **prefix={name}:** Overrides the prefix used for auxiliary gnuplot tables.

Any additional **key=value** entries are forwarded to `\addplot`, so native `pgfplots` options may be supplied without wrapping them in `plot={...}`. Linear or asymptotic approximations are not available for Nyquist plots. To indicate the direction of increasing frequency, load `decorations.markings` and `arrows.meta` and add `plot={postaction=decorate,decoration={...}}`. Very large sample counts in combination with decorations can trigger the warning **! Dimension too big**.

For example, the command

```
\NyquistZPK[
  domain=-30:30,
  plot={red,thick,samples=2000},
  axes={blue,thick}]
{zeros={0,-0.1-0.5i,-0.1+0.5i},
 poles={-0.5-10i,-0.5+10i},
 gain=10}
```

generates the Nyquist plot in Figure 8.

Legacy interface (v2.1.1 and earlier):

`\NyquistZPK[$\langle options \rangle$]{ $\langle zpk-spec \rangle$ }{ $\langle min-freq \rangle$ }{ $\langle max-freq \rangle$ }`

The optional argument accepts comma-separated tuples `plot/{opt}`, `axes/{opt}`, `commands/{opt}`, `tikz/{opt}`, and `prefix/{opt}`, while a bare `{opt}` is treated as

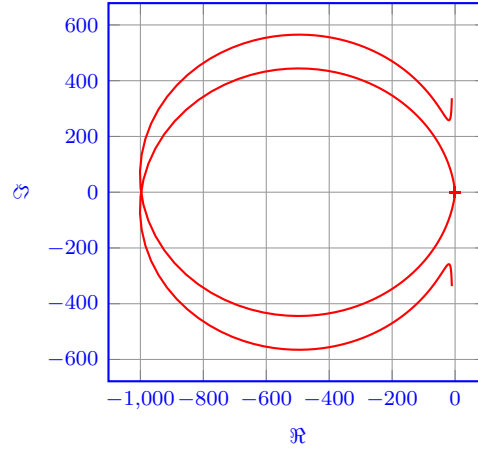


Figure 8: Output of the `\NyquistZPK` macro.

`plot/{opt}`. The `zpk-spec` uses the legacy format `z/{...},p/{...},k/...,d/...`, where complex numbers are expressed as `{a,b}`. Supplying the frequency arguments auto-selects the legacy behaviour and is intended only for maintaining older documents. Example of legacy usage:

```
\NyquistZPK[plot/{red,thick,samples=2000},axes/{blue,thick}]
  {z/{0,{-0.1,-0.5},{-0.1,0.5}},p/{{-0.5,-10},{-0.5,10}},k/10}
  {-30}{30}
```

`\NyquistTF` `\NyquistTF[<options>]{<tf-spec>}`

Generates the Nyquist diagram of a transfer function expressed in polynomial form. The `tf-spec` is the same as that of the `\BodeTF` macro described earlier and the `options` argument uses the same pgfkeys interface as `\NyquistZPK`. A legacy interface similar to the one in `\NyquistZPK` is also supported when both frequency arguments are provided. For example,

```
\NyquistTF[
  domain=-30:30,
  plot={green,thick,samples=500,postaction=decorate,
    decoration={markings,
      mark=between positions 0.1 and 0.9 step 5em
      with{\arrow{Stealth[length=2mm, blue]}}}}
  {numerator={10,2,2.6,0}, denominator={1,1,100.25}}
]
```

yields the Nyquist plot in Figure 9. As with the ZPK variant, combining dense sampling with `decorations.markings` may trigger the warning **! Dimension too big**.

Legacy syntax remains available:

```
\NyquistTF[plot/{green,thick,samples=500,postaction=decorate,
  decoration={markings,
    mark=between positions 0.1 and 0.9 step 5em
    with{\arrow{Stealth[length=2mm, blue]}}}}
  {num/{10,2,2.6,0},den/{1,1,100.25}}
  {-30}{30}]
```

`NyquistPlot (env.)` `\begin{NyquistPlot}[<options>]`

`\addNyquist...`

`\end{NyquistPlot}`

Wraps a `tikzpicture/axis` pair tailored for Nyquist diagrams and hosts one or more `\addNyquist...` commands.

options:

- `domain=min:max`: Sets the frequency sweep that is forwarded to the enclosed `\addplot` commands. The default sweep is `0.01:100`; choose a symmetric interval such as `-30:30` for traditional Nyquist diagrams.

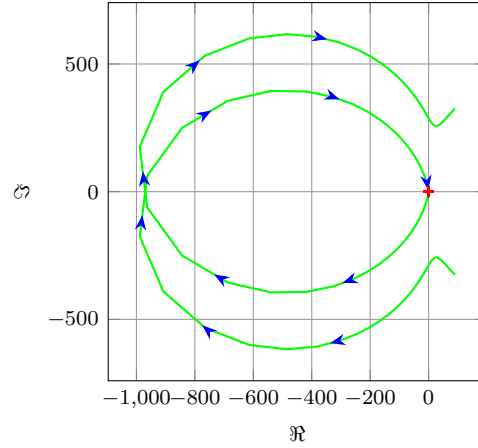


Figure 9: Output of the `\NyquistTF` macro with direction arrows. Increasing the number of samples can cause `decorations.markings` to throw errors.

- **axes={opt}**: Appends **opt** to the **axis** options (for example, **axis equal** or **grid=both**).
- **tikz={opt}**: Adds **opt** to the surrounding **tikzpicture**.
- **prefix={name}**: Overrides the prefix used for auxiliary gnuplot tables created inside the environment.
- **phase unit=deg/rad** and **frequency unit=rad/Hz**: Synchronises the environment with the package-wide unit switches.
- Additional **key=value** pairs are passed straight to the **axis** environment, so native **pgfplots** options can be provided without extra wrapping.

The environment marks the critical point $(-1, 0)$ with a red **+** by default. Pair it with `\addNyquistZPKPlot` or `\addNyquistTFPlot` to draw the actual contour.

Legacy interface (v2.1.1 and earlier):

```
\begin{NyquistPlot}[<options>]{<min-freq>}{<max-freq>}
  \addNyquist...
\end{NyquistPlot}
```

The legacy optional argument accepts comma-separated tuples **obj/{opt}** or bare **{opt}**. Valid **obj** tokens are **tikz**, **axes**, and **prefix**; unqualified entries are appended to the **axis** options. The mandatory frequency limits fix the axis and plot domain.

```
\addNyquistZPKPlot \addNyquistZPKPlot[<plot-options>]{<zpk-spec>}
```

Adds a Nyquist contour for a transfer function supplied in ZPK form.

zpk-spec: Same as the `\NyquistZPK` macro.

plot-options: Appended verbatim to `\addplot`, enabling control over sampling density, decorations, colors, and so on. Ensure that the surrounding **axis** (typically provided by `\begin{NyquistPlot}`) defines a suitable **domain** so that the generated parametric function matches the frequency sweep.

Legacy interface (v2.1.1 and earlier):

```
\addNyquistZPKPlot[<plot-options>]{<z/{...},p/{...},k/{...},d/{...}>}
```

The legacy format is preserved for backward compatibility; the optional argument continues to be forwarded directly to `\addplot`.

```
\addNyquistTFPlot \addNyquistTFPlot[<plot-options>]{<tf-spec>}
```

Identical to `\addNyquistZPKPlot` but expects **tf-spec** in the same polynomial format accepted by `\BodeTF`.

3.3 Nichols charts

`\NicholsZPK` `\NicholsZPK [<options>]{<zpk-spec>}`

Generates a Nichols chart (phase versus gain) for a transfer function in ZPK form.

The `zpk-spec` and `options` arguments are the same as those of the `\BodeZPK` macro described earlier except that the `domain=min:max` key now sets the frequency sweep for the Nichols contour (default `0.01:100`).

Supplying both frequency arguments activates the legacy behaviour for backward compatibility.

`\NicholsZPK [<options>]{<zpk-spec>}{<min-freq>}{<max-freq>}`

Example:

```
\NicholsZPK[
  domain=0.001:100,
  plot={red,thick,samples=2000},
  axes={ytick distance=20}]
{zeros={0,-0.1-0.5i,-0.1+0.5i},
 poles={-0.5-10i,-0.5+10i},
 gain=10}
```

Legacy usage remains available:

```
\NicholsZPK[plot/{red,thick,samples=2000}]
  {z/{0,-0.1,-0.5},{-0.1,0.5}},p/{-0.5,-10},{-0.5,10}},k/10}
{0.001}{100}
```

`\NicholsTF` `\NicholsTF [<options>]{<tf-spec>}`

Produces a Nichols chart for transfer functions expressed in polynomial form. Same as `\NicholsZPK` except that `tf-spec` is akin to `\BodeTF`. Supplying both frequency arguments reverts the macro to the legacy interface.

Example:

```
\NicholsTF[
  domain=0.001:100,
  plot={green,thick,samples=2000}]
{numerator={10,2,2.6,0},
 denominator={1,1,100.25},
 delay=0.01}
```

Legacy usage remains available:

```
\NicholsTF[plot/{green,thick,samples=2000}]
  {num/{10,2,2.6,0},den/{1,1,100.25},d/0.01}
{0.001}{100}
```

Both examples produce the chart in Figure 10.

`NicholsChart` (*env.*) `\begin{NicholsChart}[<options>]`

`\addNichols...`
`\end{NicholsChart}`

Provides a ready-made `tikzpicture/axis` scaffold for Nichols charts and collects `\addNichols...` commands. Current and legacy interfaces are similar to the `NyquistPlot` environment, except that the domain defaults to `0.01:100` if not supplied.

`\addNicholsZPKChart` `\addNicholsZPKChart[<plot-options>]{<zpk-spec>}`

Adds a Nichols contour for a transfer function expressed in ZPK form. Current and legacy interfaces are similar to those of `\addNyquistZPKPlot`, with the `domain` key now setting the frequency sweep for the Nichols contour, with a default of `0.01:100`.

`\addNicholsTFChart` `\addNicholsTFChart[<plot-options>]{<tf-spec>}`

Equivalent to `\addNicholsZPKChart` for transfer functions supplied in polynomial form similar to `\BodeTF`. Phase unwrapping in `gnuplot` mode mirrors the behaviour of `\BodeTF`.

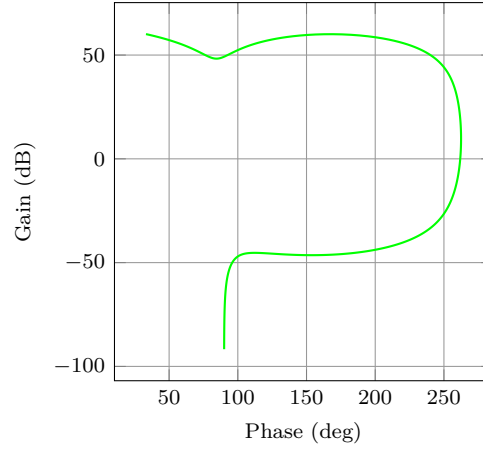


Figure 10: Output of the `\NicholsTF` macro.

3.4 Pole-zero maps

`\PoleZeroMapZPK` `\PoleZeroMapZPK` [*options*] [*zpk-spec*]

Plots the pole-zero map of a transfer function given in ZPK format, similar to MATLAB's `pzmap` function. The poles are marked with red 'x' symbols and the zeros are marked with blue 'o' symbols. The mandatory argument supports both the current `zeros={...},poles={...},gain=...` format with complex numbers expressed as `a+ib` and the legacy format `z/{zeros},p/{poles},k/{gain}` with complex numbers expressed as `{a,b}`. Note that the delay parameter `d` is ignored since delays do not affect pole-zero locations. The optional argument supports the same tuples as `\NyquistZPK`.

The `scale={log}` option enables symmetric logarithmic (symlog) scaling for both axes. This is particularly useful for systems with poles and zeros spanning multiple decades. The symlog scaling preserves the sign of coordinates while applying logarithmic scaling to their magnitude, allowing visualization of both positive and negative values on the same plot.

The following example generates a standard linear pole-zero map with the zeros at the origin and at $-0.1 \pm 0.5i$, and poles at $-0.5 \pm 10i$.

```
\PoleZeroMapZPK[axes/{grid=major}]
{zeros={0, -0.1-0.5i, -0.1+0.5i}, poles={-0.5-10i, -0.5+10i}, gain=10}
```

For logarithmic scaling:

```
\PoleZeroMapZPK[scale=log]
{zeros={0, -0.1-0.5i, -0.1+0.5i}, poles={-0.5-10i, -0.5+10i}, gain=10}
```

The legacy format remains available:

```
\PoleZeroMapZPK[axes/{grid=major}]
{z/{0,{-0.1,-0.5},{-0.1,0.5}},p/{-0.5,-10},{-0.5,10}},k/10}
\PoleZeroMapZPK[scale/log]
{zeros={0, -0.1-0.5i, -0.1+0.5i}, poles={-0.5-10i, -0.5+10i}, gain=10}
```

Both examples produce the pole-zero maps in Figure 11.

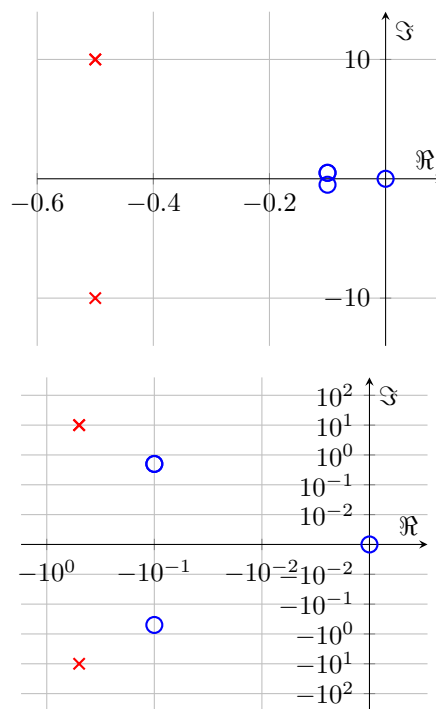


Figure 11: Pole-zero maps generated using the `\PoleZeroMapZPK` macro with linear and logarithmic scaling.

4 Implementation

4.1 Initialization

```

\n@mod We start by processing the class options.
\n@mod@p 1 \newif\if@pgfarg\@pgfargfalse
\n@mod@n 2 \DeclareOption{pgf}{
\n@pow 3 \@pgfargtrue
4 }
bp@gnuplot@id 5 \newif\if@declutterarg\@declutterargfalse
bp@gnu@prefix 6 \DeclareOption{declutter}{
7 \@declutterargtrue
8 }
9 \newif\if@radarg\@radargfalse
10 \DeclareOption{rad}{
11 \@radargtrue
12 }
13 \newif\if@hzarg\@hzargfalse
14 \DeclareOption{Hz}{
15 \@hzargtrue
16 }
17 \ProcessOptions\relax

```

New macros to unify **pgfplots** and **gnuplot**. New macros are defined for the **pow** and **mod** functions to address differences between the two math engines.

```

18 \newcommand{\n@mod}[2]{(#1)-((round((#1)/(#2)))*(#2))}
19 \newcommand{\n@mod@p}[2]{(#1)-((floor((#1)/(#2)))*(#2))}
20 \newcommand{\n@mod@n}[2]{(#1)-((floor((#1)/(#2))+1)*(#2))}
21 \if@pgfarg
22 \newcommand{\n@pow}[2]{(#1)^(#2)}
23 \else
24 \newcommand{\n@pow}[2]{(#1)**(#2)}

```

A counter so that a new data table is generated and for each new plot. If the plot macros have not changed, the tables, once generated, can be reused by **gnuplot**, which reduces compilation time. The **declutter** option is used to enable the **gnuplot** directory to declutter the working directory. The **gnuplot** prefix is set to be the name of the tex file unless the user supplies a prefix through the **prefix** option.

```

25 \newcounter{bp@gnuplot@id}
26 \setcounter{bp@gnuplot@id}{0}
27 \newcommand{\bp@prefix}{%
28 \ifx\bp@user@prefix\@empty
29 \if@declutterarg
30 gnuplot/\jobname
31 \else
32 \jobname
33 \fi
34 \else
35 \if@declutterarg
36 gnuplot/\bp@user@prefix
37 \else
38 \bp@user@prefix
39 \fi
40 \fi
41 }
42 \tikzset{
43 bp@gnu@prefix/.style={
44 id=\arabic{bp@gnuplot@id},
45 prefix=\bp@prefix
46 }
47 }

```

If the operating system is not Windows, and if the **declutter** option is passed, we create the **gnuplot** folder if it does not already exist.

```

48 \ifwindows\else
49   \if@declutterarg
50     \immediate\write18{mkdir -p gnuplot}
51   \fi
52 \fi
53 \fi

```

`\if@babel` Check if the `babel` package is loaded and generate a list of shorthands if it is. The code `\bp@short@list` is based on [this stackexchange answer](#).

```

54 \newif\if@babel\@babelfalse
55 \AtBeginDocument{%
56   \ifpackageloaded{babel}{%
57     \@babeltrue
58     \let\bp@short@list\@empty
59     \def\do#1{%
60       \begingroup
61         \lccode`\~=#1\relax
62         \lowercase{\ifbabelshorthand~{\g@addto@macro\bp@short@list{~}}{}}
63       \endgroup
64     }
65     \dospecials
66   }{}
67 }

```

`bp@style` Default axis properties for all plot macros are collected in this `pgf` style.

```

68 \pgfplotsset{
69   bp@style/.style = {
70     label style={font=\footnotesize},
71     tick label style={font=\footnotesize},
72     grid=both,
73     major grid style={color=gray!80},
74     minor grid style={color=gray!20},
75     x label style={at={(ticklabel cs:0.5)},anchor=near ticklabel},
76     y label style={at={(ticklabel cs:0.5)},anchor=near ticklabel},
77     scale only axis,
78     samples=200,
79     width=5cm,
80     log basis x=10
81   }
82 }

```

`bp@freq@filter` These macros handle the `Hz` and `rad` class options and two new `pgf` keys named `bp@freq@label` frequency unit and `bp@freq@scale` frequency unit for conversion of frequency and phase units, respectively.

```

\bp@ph@scale 83 \pgfplotsset{bp@freq@filter/.style = {}}
bp@ph@x@label 84 \def\bp@freq@scale{1}
bp@ph@y@label 85 \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (rad/s)}}}
86 \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (deg)}}}
87 \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (deg)}}}
88 \def\bp@ph@scale{180/pi}
89 \if@radarg
90   \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (rad)}}}
91   \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (rad)}}}
92   \def\bp@ph@scale{1}
93   \tikzset{
94     phase unit/.initial={rad},
95     phase unit/.default={rad},
96   }
97 \else
98   \tikzset{
99     phase unit/.initial={deg},
100    phase unit/.default={deg},
101  }

```

```

102 \fi
103 \if@hzarg
104   \def\bp@freq@scale{2*pi}
105   \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (Hz)}}}
106   \if@pgfarg
107     \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x-
log10(2*pi)}}}
108   \fi
109   \tikzset{
110     frequency unit/.initial={Hz},
111     frequency unit/.default={Hz},
112   }
113 \else
114   \tikzset{
115     frequency unit/.initial={rad},
116     frequency unit/.default={rad},
117   }
118 \fi
119 \tikzset{
120   phase unit/.is choice,
121   phase unit/deg/.code={
122     \pgfkeys{/bodeplot/phase unit=deg}
123   },
124   phase unit/rad/.code={
125     \pgfkeys{/bodeplot/phase unit=rad}
126   },
127   frequency unit/.is choice,
128   frequency unit/Hz/.code={
129     \pgfkeys{/bodeplot/frequency unit=Hz}
130   },
131   frequency unit/rad/.code={
132     \pgfkeys{/bodeplot/frequency unit=rad}
133   }
134 }

```

4.2 PGF keys interface

/bodeplot/env PGF keys for environment options.

```

135 \pgfkeys{
136   /bodeplot/env/.is family,
137   /bodeplot/env/.cd,
138   reset/.code={%
139     \pgfkeyssetvalue{/bodeplot/env/@tikz}{}
140     \pgfkeyssetvalue{/bodeplot/env/@prefix}{}
141     \pgfkeyssetvalue{/bodeplot/env/@domain}{}
142     \if@hzarg
143       \pgfkeys{/bodeplot/env/frequency unit=Hz}
144     \else
145       \pgfkeys{/bodeplot/env/frequency unit=rad}
146     \fi
147     \if@radarg
148       \pgfkeys{/bodeplot/env/phase unit=rad}
149     \else
150       \pgfkeys{/bodeplot/env/phase unit=deg}
151     \fi
152     \gdef\bp@domain@start{0.01}
153     \gdef\bp@domain@end{100}
154     \gdef\bp@axes{}
155   },
156   axes/.code={
157     \ifx\bp@axes\@empty
158       \xdef\bp@axes{\unexpanded\expandafter{#1}}
159     \else

```

```

160     \xdef\bp@axes{\unexpanded\expandafter{\bp@axes},
161     \unexpanded\expandafter{#1}}
162     \fi
163 },
164 axes/.value required,
165 tikz/.code={\pgfkeyssetvalue{/bodeplot/env/@tikz}{#1}},
166 tikz/.value required,
167 prefix/.code={\pgfkeyssetvalue{/bodeplot/env/@prefix}{#1}},
168 prefix/.value required,
169 domain/.code args={#1:#2}{%
170     \gdef\bp@domain@start{#1}%
171     \gdef\bp@domain@end{#2}%
172 },
173 domain/.value required,
174 phase unit/.initial={deg},
175 phase unit/.default={deg},
176 phase unit/.is choice,
177 phase unit/deg/.code={
178     \renewcommand{\bp@ph@scale}{180/pi}
179     \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (deg)}}}
180     \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (deg)}}}
181 },
182 phase unit/rad/.code={
183     \renewcommand{\bp@ph@scale}{1}
184     \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (rad)}}}
185     \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (rad)}}}
186 },
187 frequency unit/.initial={rad},
188 frequency unit/.default={rad},
189 frequency unit/.is choice,
190 frequency unit/Hz/.code={
191     \renewcommand{\bp@freq@scale}{2*pi}
192     \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (Hz)}}}
193     \ifpgfarg
194     \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x-
log10(2*pi)}}}
195     \fi
196 },
197 frequency unit/rad/.code={
198     \renewcommand{\bp@freq@scale}{1}
199     \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (rad/s)}}}
200     \ifpgfarg
201     \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x}}}
202     \fi
203 },
204 .unknown/.code={%
205     \edef\bp@full{\pgfkeyscurrentkey}%
206     \def\stripbodeprefix##1/bodeplot/env/##2\relax{##2}%
207     \edef\bp@short{\expandafter\stripbodeprefix\bp@full\relax}%
208     \edef\bp@checkfull{\bp@full}%
209     \edef\bp@checkshort{\bp@short}%
210     \ifx\bp@checkfull\bp@checkshort
211     \def\removeslash##1/##2\relax{##2}%
212     \edef\bp@short{\expandafter\removeslash\bp@full\relax}%
213     \fi
214     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue
215     \edef\bp@new{\bp@short}%
216     \else
217     \edef\bp@new{\bp@short=
218     {\unexpanded\expandafter{\pgfkeyscurrentvalue}}}%
219     \fi
220     \ifx\bp@new\@empty\else
221     \ifx\bp@axes\@empty

```

```

222         \xdef\bp@axes{\unexpanded\expandafter{\bp@new}}
223     \else
224         \xdef\bp@axes{\unexpanded\expandafter{\bp@axes},%
225         \unexpanded\expandafter{\bp@new}}
226     \fi
227 \fi
228 }
229 }

```

/bodeplot PGF keys for bode command options.

```

230 \pgfkeys{
231   /bodeplot/.is family,
232   /bodeplot/.cd,
233   reset/.code={%
234     \pgfkeyssetvalue{/bodeplot/@axes/mag}{}
235     \pgfkeyssetvalue{/bodeplot/@axes/ph}{}
236     \pgfkeyssetvalue{/bodeplot/@group}{}
237     \pgfkeyssetvalue{/bodeplot/@approx}{true}
238     \pgfkeyssetvalue{/bodeplot/@commands/mag}{}
239     \pgfkeyssetvalue{/bodeplot/@commands/ph}{}
240     \pgfkeyssetvalue{/bodeplot/@tikz}{}
241     \pgfkeyssetvalue{/bodeplot/@prefix}{}
242     \gdef\bp@domain@start{0.01}
243     \gdef\bp@domain@end{100}
244     \if@hzarg
245       \pgfkeys{/bodeplot/frequency unit=Hz}
246     \else
247       \pgfkeys{/bodeplot/frequency unit=rad}
248     \fi
249     \if@radarg
250       \pgfkeys{/bodeplot/phase unit=rad}
251     \else
252       \pgfkeys{/bodeplot/phase unit=deg}
253     \fi
254     \gdef\bp@mag@plot{}
255     \gdef\bp@ph@plot{}
256   },
257   plot/.code={%
258     \ifx\bp@mag@plot\@empty
259       \xdef\bp@mag@plot{\unexpanded\expandafter{#1}}
260     \else
261       \xdef\bp@mag@plot{\unexpanded\expandafter{\bp@mag@plot,#1}}
262     \fi
263     \ifx\bp@ph@plot\@empty
264       \xdef\bp@ph@plot{\unexpanded\expandafter{#1}}
265     \else
266       \xdef\bp@ph@plot{\unexpanded\expandafter{\bp@ph@plot,#1}}
267     \fi
268   },
269   plot/.value required,
270   mag plot/.code={%
271     \ifx\bp@mag@plot\@empty
272       \xdef\bp@mag@plot{\unexpanded\expandafter{#1}}
273     \else
274       \xdef\bp@mag@plot{\unexpanded\expandafter{\bp@mag@plot},
275       \unexpanded\expandafter{#1}}
276     \fi
277   },
278   mag plot/.value required,
279   ph plot/.code={%
280     \ifx\bp@ph@plot\@empty
281       \xdef\bp@ph@plot{\unexpanded\expandafter{#1}}
282     \else

```

```

283     \xdef\bp@ph@plot{\unexpanded\expandafter{\bp@ph@plot},
284     \unexpanded\expandafter{#1}}
285     \fi
286 },
287 ph plot/.value required,
288 axes/.code={%
289     \pgfkeysalso{/bodeplot/mag axes={#1}}
290     \pgfkeysalso{/bodeplot/ph axes={#1}}%
291 },
292 axes/.value required,
293 mag axes/.code={\pgfkeyssetvalue{/bodeplot/@axes/mag}{#1}},
294 mag axes/.value required,
295 ph axes/.code={\pgfkeyssetvalue{/bodeplot/@axes/ph}{#1}},
296 ph axes/.value required,
297 group/.code={\pgfkeyssetvalue{/bodeplot/@group}{#1}},
298 group/.value required,
299 approx/.code={\pgfkeyssetvalue{/bodeplot/@approx}{#1}},
300 approx/.value required,
301 commands/.code={%
302     \pgfkeysalso{/bodeplot/mag commands={#1}}
303     \pgfkeysalso{/bodeplot/ph commands={#1}}%
304 },
305 commands/.value required,
306 mag commands/.code={\pgfkeyssetvalue{/bodeplot/@commands/mag}{#1}},
307 mag commands/.value required,
308 ph commands/.code={\pgfkeyssetvalue{/bodeplot/@commands/ph}{#1}},
309 ph commands/.value required,
310 tikz/.code={\pgfkeyssetvalue{/bodeplot/@tikz}{#1}},
311 tikz/.value required,
312 prefix/.code={\pgfkeyssetvalue{/bodeplot/@prefix}{#1}},
313 prefix/.value required,
314 domain/.code args={#1:#2}{%
315     \gdef\bp@domain@start{#1}%
316     \gdef\bp@domain@end{#2}%
317 },
318 domain/.value required,
319 phase unit/.initial={deg},
320 phase unit/.default={deg},
321 phase unit/.is choice,
322 phase unit/deg/.code={
323     \renewcommand{\bp@ph@scale}{180/pi}
324     \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (deg)}}}
325     \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (deg)}}}
326 },
327 phase unit/rad/.code={
328     \renewcommand{\bp@ph@scale}{1}
329     \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (rad)}}}
330     \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (rad)}}}
331 },
332 frequency unit/.initial={rad},
333 frequency unit/.default={rad},
334 frequency unit/.is choice,
335 frequency unit/Hz/.code={
336     \renewcommand{\bp@freq@scale}{2*pi}
337     \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (Hz)}}}
338     \ifpgfarg
339     \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x-
log10(2*pi)}}}
340     \fi
341 },
342 frequency unit/rad/.code={
343     \renewcommand{\bp@freq@scale}{1}
344     \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (rad/s)}}}

```

```

345 \if@pgfarg
346 \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x}}}}
347 \fi
348 },
349 .unknown/.code={%
350 \edef\bp@full{\pgfkeyscurrentkey}%
351 \def\stripbodeprefix##1/bodeplot/##2\relax{##2}%
352 \edef\bp@short{\expandafter\stripbodeprefix\bp@full\relax}%
353 \edef\bp@checkfull{\bp@full}%
354 \edef\bp@checkshort{\bp@short}%
355 \ifx\bp@checkfull\bp@checkshort
356 \def\removeslash##1/##2\relax{##2}%
357 \edef\bp@short{\expandafter\removeslash\bp@full\relax}%
358 \fi
359 \ifx\pgfkeyscurrentvalue\pgfkeysnovalue
360 \edef\bp@new{\bp@short}%
361 \else
362 \edef\bp@new{\bp@short=
363 {\unexpanded\expandafter{\pgfkeyscurrentvalue}}}%
364 \fi
365 \ifx\bp@new\@empty\else
366 \ifx\bp@mag@plot\@empty
367 \xdef\bp@mag@plot{\unexpanded\expandafter{\bp@new}}
368 \else
369 \xdef\bp@mag@plot{\unexpanded\expandafter{\bp@mag@plot},%
370 \unexpanded\expandafter{\bp@new}}
371 \fi
372 \fi
373 \ifx\bp@new\@empty\else
374 \ifx\bp@ph@plot\@empty
375 \xdef\bp@ph@plot{\unexpanded\expandafter{\bp@new}}
376 \else
377 \xdef\bp@ph@plot{\unexpanded\expandafter{\bp@ph@plot},%
378 \unexpanded\expandafter{\bp@new}}
379 \fi
380 \fi
381 }
382 }

```

/bodeplot/combinedenv PGF keys for combined Bode environment options.

```

383 \pgfkeys{
384 /bodeplot/combinedenv/.is family,
385 /bodeplot/combinedenv/.cd,
386 reset/.code={%
387 \pgfkeyssetvalue{/bodeplot/combinedenv/@group}{}
388 \pgfkeyssetvalue{/bodeplot/combinedenv/@approx}{true}
389 \pgfkeyssetvalue{/bodeplot/combinedenv/@commands/mag}{}
390 \pgfkeyssetvalue{/bodeplot/combinedenv/@commands/ph}{}
391 \pgfkeyssetvalue{/bodeplot/combinedenv/@tikz}{}
392 \pgfkeyssetvalue{/bodeplot/combinedenv/@prefix}{}
393 \gdef\bp@domain@start{0.01}
394 \gdef\bp@domain@end{100}
395 \if@hzarg
396 \pgfkeys{/bodeplot/frequency unit=Hz}
397 \else
398 \pgfkeys{/bodeplot/frequency unit=rad}
399 \fi
400 \if@radarg
401 \pgfkeys{/bodeplot/phase unit=rad}
402 \else
403 \pgfkeys{/bodeplot/phase unit=deg}
404 \fi
405 \gdef\bp@mag@axes{}

```

```

406 \gdef\bp@ph@axes{}
407 },
408 axes/.code={%
409 \ifx\bp@mag@axes\@empty
410 \xdef\bp@mag@axes{\unexpanded\expandafter{#1}}
411 \else
412 \xdef\bp@mag@axes{\unexpanded\expandafter{\bp@mag@axes,#1}}
413 \fi
414 \ifx\bp@ph@axes\@empty
415 \xdef\bp@ph@axes{\unexpanded\expandafter{#1}}
416 \else
417 \xdef\bp@ph@axes{\unexpanded\expandafter{\bp@ph@axes,#1}}
418 \fi
419 },
420 axes/.value required,
421 mag axes/.code={%
422 \ifx\bp@mag@axes\@empty
423 \xdef\bp@mag@axes{\unexpanded\expandafter{#1}}
424 \else
425 \xdef\bp@mag@axes{\unexpanded\expandafter{\bp@mag@axes,#1}}
426 \fi
427 },
428 mag axes/.value required,
429 ph axes/.code={%
430 \ifx\bp@ph@axes\@empty
431 \xdef\bp@ph@axes{\unexpanded\expandafter{#1}}
432 \else
433 \xdef\bp@ph@axes{\unexpanded\expandafter{\bp@ph@axes,#1}}
434 \fi
435 },
436 ph axes/.value required,
437 group/.code={\pgfkeyssetvalue{/bodeplot/combinedenv/@group}{#1}},
438 group/.value required,
439 approx/.code={\pgfkeyssetvalue{/bodeplot/combinedenv/@approx}{#1}},
440 approx/.value required,
441 commands/.code={%
442 \pgfkeysalso{/bodeplot/combinedenv/mag commands={#1}}
443 \pgfkeysalso{/bodeplot/combinedenv/ph commands={#1}}%
444 },
445 commands/.value required,
446 mag commands/.code={\pgfkeyssetvalue{/bodeplot/combinedenv/@commands/mag}{#1}},
447 mag commands/.value required,
448 ph commands/.code={\pgfkeyssetvalue{/bodeplot/combinedenv/@commands/ph}{#1}},
449 ph commands/.value required,
450 tikz/.code={\pgfkeyssetvalue{/bodeplot/combinedenv/@tikz}{#1}},
451 tikz/.value required,
452 prefix/.code={\pgfkeyssetvalue{/bodeplot/combinedenv/@prefix}{#1}},
453 prefix/.value required,
454 domain/.code args={#1:#2}{%
455 \gdef\bp@domain@start{#1}%
456 \gdef\bp@domain@end{#2}%
457 },
458 domain/.value required,
459 phase unit/.initial={deg},
460 phase unit/.default={deg},
461 phase unit/.is choice,
462 phase unit/deg/.code={
463 \renewcommand{\bp@ph@scale}{180/pi}
464 \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (deg)}}}
465 \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (deg)}}}
466 },
467 phase unit/rad/.code={
468 \renewcommand{\bp@ph@scale}{1}

```



```

469 \pgfplotsset{bp@ph@y@label/.style = {ylabel={Phase (rad)}}}
470 \pgfplotsset{bp@ph@x@label/.style = {xlabel={Phase (rad)}}}
471 },
472 frequency unit/.initial={rad},
473 frequency unit/.default={rad},
474 frequency unit/.is choice,
475 frequency unit/Hz/.code={
476 \renewcommand{\bp@freq@scale}{2*pi}
477 \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (Hz)}}}
478 \ifpgfarg
479 \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x-
log10(2*pi)}}}
480 \fi
481 },
482 frequency unit/rad/.code={
483 \renewcommand{\bp@freq@scale}{1}
484 \pgfplotsset{bp@freq@label/.style = {xlabel = {Frequency (rad/s)}}}
485 \ifpgfarg
486 \pgfplotsset{bp@freq@filter/.style = {x filter/.expression={x}}}
487 \fi
488 },
489 .unknown/.code={%
490 \edef\bp@full{\pgfkeyscurrentkey}%
491 \def\stripbodeprefix##1/bodeplot/combinedenv/##2\relax{##2}%
492 \edef\bp@short{\expandafter\stripbodeprefix\bp@full\relax}%
493 \edef\bp@checkfull{\bp@full}%
494 \edef\bp@checkshort{\bp@short}%
495 \ifx\bp@checkfull\bp@checkshort
496 \def\removeslash##1/##2\relax{##2}%
497 \edef\bp@short{\expandafter\removeslash\bp@full\relax}%
498 \fi
499 \ifx\pgfkeyscurrentvalue\pgfkeysnovalue
500 \edef\bp@new{\bp@short}%
501 \else
502 \edef\bp@new{\bp@short=
503 {\unexpanded\expandafter{\pgfkeyscurrentvalue}}}%
504 \fi
505 \ifx\bp@new\@empty\else
506 \ifx\bp@mag@axes\@empty
507 \xdef\bp@mag@axes{\unexpanded\expandafter{\bp@new}}
508 \else
509 \xdef\bp@mag@axes{\unexpanded\expandafter{\bp@mag@axes},%
510 \unexpanded\expandafter{\bp@new}}
511 \fi
512 \fi
513 \ifx\bp@new\@empty\else
514 \ifx\bp@ph@axes\@empty
515 \xdef\bp@ph@axes{\unexpanded\expandafter{\bp@new}}
516 \else
517 \xdef\bp@ph@axes{\unexpanded\expandafter{\bp@ph@axes},%
518 \unexpanded\expandafter{\bp@new}}
519 \fi
520 \fi
521 }
522 }

```

/bodeplot/zpk PGF keys for supplying zero-pole-gain-delay (ZPK) representations.

```

523 \pgfkeys{
524 /bodeplot/zpk/.is family,
525 /bodeplot/zpk/.cd,
526 reset/.code={%
527 \pgfkeyssetvalue{/bodeplot/zpk/@zeros}{}
528 \pgfkeyssetvalue{/bodeplot/zpk/@poles}{}

```

```

529 \pgfkeyssetvalue{/bodeplot/zpk/@gain}{}
530 \pgfkeyssetvalue{/bodeplot/zpk/@delay}{}
531 },
532 zeros/.code={\pgfkeyssetvalue{/bodeplot/zpk/@zeros}{#1}},
533 zeros/.value required,
534 poles/.code={\pgfkeyssetvalue{/bodeplot/zpk/@poles}{#1}},
535 poles/.value required,
536 gain/.code={\pgfkeyssetvalue{/bodeplot/zpk/@gain}{#1}},
537 gain/.value required,
538 delay/.code={\pgfkeyssetvalue{/bodeplot/zpk/@delay}{#1}},
539 delay/.value required,
540 }

```

/bodeplot/tf PGF keys for supplying transfer function (TF) representations.

```

541 \pgfkeys{
542 /bodeplot/tf/.is family,
543 /bodeplot/tf/.cd,
544 reset/.code={%
545 \pgfkeyssetvalue{/bodeplot/tf/@numerator}{}
546 \pgfkeyssetvalue{/bodeplot/tf/@denominator}{}
547 \pgfkeyssetvalue{/bodeplot/tf/@delay}{}
548 },
549 numerator/.code={\pgfkeyssetvalue{/bodeplot/tf/@numerator}{#1}},
550 numerator/.value required,
551 denominator/.code={\pgfkeyssetvalue{/bodeplot/tf/@denominator}{#1}},
552 denominator/.value required,
553 delay/.code={\pgfkeyssetvalue{/bodeplot/tf/@delay}{#1}},
554 delay/.value required
555 }

```

/bodeplot/add PGF keys for adding asymptotic and linear plots.

```

556 \pgfkeys{
557 /bodeplot/add/.is family,
558 /bodeplot/add/.cd,
559 reset/.code={%
560 \gdef\bp@add@0{}%
561 },
562 true/.default={},
563 true/.code={%
564 \pgfutil@ifempty{#1}{
565 \ifx\bp@add@0\empty
566 \g@addto@macro\bp@add@0{true}%
567 \else
568 \g@addto@macro\bp@add@0{,true}%
569 \fi
570 }{%
571 \ifx\bp@add@0\empty
572 \g@addto@macro\bp@add@0{true/{#1}}%
573 \else
574 \g@addto@macro\bp@add@0{,true/{#1}}%
575 \fi
576 }
577 },
578 linear/.default={},
579 linear/.code={%
580 \pgfutil@ifempty{#1}{
581 \ifx\bp@add@0\empty
582 \g@addto@macro\bp@add@0{linear}%
583 \else
584 \g@addto@macro\bp@add@0{,linear}%
585 \fi
586 }{%
587 \ifx\bp@add@0\empty

```

```

588     \g@addto@macro\bp@add@0{linear/{#1}}%
589     \else
590     \g@addto@macro\bp@add@0{,linear/{#1}}%
591     \fi
592 }
593 },
594 asymptotic/.default={},
595 asymptotic/.code={%
596     \pgfutil@ifempty{#1}{
597         \ifx\bp@add@0\empty
598             \g@addto@macro\bp@add@0{asymptotic}%
599         \else
600             \g@addto@macro\bp@add@0{,asymptotic}%
601         \fi
602     }{%
603         \ifx\bp@add@0\empty
604             \g@addto@macro\bp@add@0{asymptotic/{#1}}%
605         \else
606             \g@addto@macro\bp@add@0{,asymptotic/{#1}}%
607         \fi
608     }%
609 },
610 .unknown/.code={%
611     \edef\bp@full{\pgfkeyscurrentkey}%
612     \def\stripbodeprefix##1/bodeplot/add/##2\relax{##2}%
613     \edef\bp@short{\expandafter\stripbodeprefix\bp@full\relax}%
614     \edef\bp@checkfull{\bp@full}%
615     \edef\bp@checkshort{\bp@short}%
616     \ifx\bp@checkfull\bp@checkshort
617         \def\removeslash##1/##2\relax{##2}%
618     \edef\bp@short{\expandafter\removeslash\bp@full\relax}%
619     \fi
620     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue
621         \edef\bp@new{\bp@short}%
622     \else
623         \edef\bp@new{\bp@short=
624             {\unexpanded\expandafter{\pgfkeyscurrentvalue}}}%
625     \fi
626     \ifx\bp@new\empty\else
627         \ifx\bp@add@0\empty
628             \xdef\bp@add@0{\unexpanded\expandafter{\bp@new}}
629         \else
630             \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0},%
631                 \unexpanded\expandafter{\bp@new}}
632         \fi
633     \fi
634 }
635 }

```

/bodeplot/nyquist PGF keys for Nyquist plot options.

```

636 \pgfkeys{
637     /bodeplot/nyquist/.is family,
638     /bodeplot/nyquist/.cd,
639     reset/.code={
640         \pgfkeyssetvalue{/bodeplot/nyquist/@axes}{}
641         \pgfkeyssetvalue{/bodeplot/nyquist/@plot}{}
642         \pgfkeyssetvalue{/bodeplot/nyquist/@commands}{}
643         \pgfkeyssetvalue{/bodeplot/nyquist/@tikz}{}
644         \pgfkeyssetvalue{/bodeplot/nyquist/@prefix}{}
645         \gdef\bp@domain@start{-30}
646         \gdef\bp@domain@end{30}
647         \gdef\bp@plot{}
648     },

```

```

649 axes/.code={\pgfkeyssetvalue{/bodeplot/nyquist/@axes}{#1}},
650 axes/.value required,
651 plot/.code={%
652   \xdef\bp@plot{\unexpanded\expandafter{#1}}
653 },
654 plot/.value required,
655 commands/.code={\pgfkeyssetvalue{/bodeplot/nyquist/@commands}{#1}},
656 commands/.value required,
657 tikz/.code={\pgfkeyssetvalue{/bodeplot/nyquist/@tikz}{#1}},
658 tikz/.value required,
659 prefix/.code={\pgfkeyssetvalue{/bodeplot/nyquist/@prefix}{#1}},
660 prefix/.value required,
661 domain/.code args={#1:#2}{\gdef\bp@domain@start{#1}\gdef\bp@domain@end{#2}},
662 domain/.value required,
663 .unknown/.code={%
664   \edef\bp@full{\pgfkeyscurrentkey}%
665   \def\stripbodeprefix##1/bodeplot/nyquist/##2\relax{##2}%
666   \edef\bp@short{\expandafter\stripbodeprefix\bp@full\relax}%
667   \edef\bp@checkfull{\bp@full}%
668   \edef\bp@checkshort{\bp@short}%
669   \ifx\bp@checkfull\bp@checkshort
670     \def\removeslash##1/##2\relax{##2}%
671     \edef\bp@short{\expandafter\removeslash\bp@full\relax}%
672   \fi
673   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue
674     \edef\bp@new{\bp@short}%
675   \else
676     \edef\bp@new{\bp@short=
677       {\unexpanded\expandafter{\pgfkeyscurrentvalue}}}%
678   \fi
679   \ifx\bp@new\@empty\else
680     \ifx\bp@plot\@empty
681       \xdef\bp@plot{\unexpanded\expandafter{\bp@new}}
682     \else
683       \xdef\bp@plot{\unexpanded\expandafter{\bp@plot},%
684         \unexpanded\expandafter{\bp@new}}
685     \fi
686   \fi
687 }
688 }

```

/bodeplot/nichols PGF keys for Nichols plot options.

```

689 \pgfkeys{
690   /bodeplot/nichols/.is family,
691   /bodeplot/nichols/.cd,
692   reset/.code={
693     \pgfkeyssetvalue{/bodeplot/nichols/@axes}{}
694     \pgfkeyssetvalue{/bodeplot/nichols/@plot}{}
695     \pgfkeyssetvalue{/bodeplot/nichols/@commands}{}
696     \pgfkeyssetvalue{/bodeplot/nichols/@tikz}{}
697     \pgfkeyssetvalue{/bodeplot/nichols/@prefix}{}
698     \gdef\bp@domain@start{0.01}
699     \gdef\bp@domain@end{100}
700     \gdef\bp@plot{}
701   },
702   axes/.code={\pgfkeyssetvalue{/bodeplot/nichols/@axes}{#1}},
703   axes/.value required,
704   plot/.code={%
705     \xdef\bp@plot{\unexpanded\expandafter{#1}}
706   },
707   plot/.value required,
708   commands/.code={\pgfkeyssetvalue{/bodeplot/nichols/@commands}{#1}},
709   commands/.value required,

```

```

710 tikz/.code={\pgfkeyssetvalue{/bodeplot/nichols/@tikz}{#1}},
711 tikz/.value required,
712 prefix/.code={\pgfkeyssetvalue{/bodeplot/nichols/@prefix}{#1}},
713 prefix/.value required,
714 domain/.code args={#1:#2}{\gdef\bp@domain@start{#1}\gdef\bp@domain@end{#2}},
715 domain/.value required,
716 .unknown/.code={%
717   \edef\bp@full{\pgfkeyscurrentkey}%
718   \def\stripbodeprefix##1/bodeplot/nichols/##2\relax{##2}%
719   \edef\bp@short{\expandafter\stripbodeprefix\bp@full\relax}%
720   \edef\bp@checkfull{\bp@full}%
721   \edef\bp@checkshort{\bp@short}%
722   \ifx\bp@checkfull\bp@checkshort
723     \def\removeslash##1/##2\relax{##2}%
724     \edef\bp@short{\expandafter\removeslash\bp@full\relax}%
725   \fi
726   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue
727     \edef\bp@new{\bp@short}%
728   \else
729     \edef\bp@new{\bp@short=
730       {\unexpanded\expandafter{\pgfkeyscurrentvalue}}}%
731   \fi
732   \ifx\bp@new\@empty\else
733     \ifx\bp@plot\@empty
734       \xdef\bp@plot{\unexpanded\expandafter{\bp@new}}
735     \else
736       \xdef\bp@plot{\unexpanded\expandafter{\bp@plot},%
737         \unexpanded\expandafter{\bp@new}}
738     \fi
739   \fi
740 }
741 }

```

/bodeplot/pzmap PGF keys for pole-zero map options.

```

742 \pgfkeys{
743   /bodeplot/pzmap/.is family,
744   /bodeplot/pzmap/.cd,
745   reset/.code={
746     \pgfkeyssetvalue{/bodeplot/pzmap/@axes}{}
747     \pgfkeyssetvalue{/bodeplot/pzmap/@plot}{}
748     \pgfkeyssetvalue{/bodeplot/pzmap/@commands}{}
749     \pgfkeyssetvalue{/bodeplot/pzmap/@tikz}{}
750     \pgfkeyssetvalue{/bodeplot/pzmap/@prefix}{}
751     \pgfkeyssetvalue{/bodeplot/pzmap/@scale}{linear}
752   },
753   axes/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@axes}{#1}},
754   axes/.value required,
755   plot/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@plot}{#1}},
756   plot/.value required,
757   commands/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@commands}{#1}},
758   commands/.value required,
759   tikz/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@tikz}{#1}},
760   tikz/.value required,
761   prefix/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@prefix}{#1}},
762   prefix/.value required,
763   scale/.initial=linear,
764   scale/.default=linear,
765   scale/.is choice,
766   scale/linear/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@scale}{linear}},
767   scale/log/.code={\pgfkeyssetvalue{/bodeplot/pzmap/@scale}{log}},
768 }

```

4.3 Parametric function generators for poles, zeros, gains, and delays.

All calculations are carried out assuming that frequency inputs are in rad/s. Magnitude outputs are in dB and phase outputs are in degrees or radians, depending on the value of `\bp@ph@scale`.

`\MagK` True, linear, and asymptotic magnitude and phase parametric functions for a pure gain
`\MagKAsymp` $G(s) = k + 0i$. The macros take two arguments corresponding to real and imaginary
`\MagKLin` part of the gain to facilitate code reuse between delays, gains, poles, and zeros, but only
`\PhK` real gains are supported. The second argument, if supplied, is ignored.

```

\PhKAsymp 769 \newcommand*\MagK[2]{(20*log10(abs(#1)))}
\PhKLin    770 \newcommand*\MagKAsymp{\MagK}
           771 \newcommand*\MagKLin{\MagK}
           772 \newcommand*\PhK[2]{((#1<0?-pi:0)*\bp@ph@scale)}
           773 \newcommand*\PhKAsymp{\PhK}
           774 \newcommand*\PhKLin{\PhK}

```

`\PhKAsymp` True magnitude and phase parametric functions for a pure delay $G(s) = e^{-Ts}$. The
`\PhKLin` macros take two arguments corresponding to real and imaginary part of the gain to
facilitate code reuse between delays, gains, poles, and zeros, but only real gains are
supported. The second argument, if supplied, is ignored.

```

           775 \newcommand*\MagDel[2]{0}
           776 \newcommand*\PhDel[2]{(-#1*t*\bp@ph@scale)}

```

`\MagPole` These macros are the building blocks for most of the plotting functions provided by this
`\MagPoleAsymp` package. We start with Parametric function for the true magnitude of a complex pole.
`\MagPoleLin` 777 \newcommand*\MagPole[2]

```

\PhPole    778 {( -20*log10(sqrt(\n@pow{#1}{2} + \n@pow{t - (#2)}{2})))}
\PhPoleAsymp Parametric function for linear approximation of the magnitude of a complex pole.
\PhPoleLin 779 \newcommand*\MagPoleLin[2]{(t < sqrt(\n@pow{#1}{2} + \n@pow{#2}{2}) ?
           780 -20*log10(sqrt(\n@pow{#1}{2} + \n@pow{#2}{2})) :
           781 -20*log10(t)
           782 )}
```

Parametric function for asymptotic approximation of the magnitude of a complex pole,
same as linear approximation.

```

           783 \newcommand*\MagPoleAsymp{\MagPoleLin}

```

Parametric function for the true phase of a complex pole.

```

           784 \newcommand*\PhPole[2]{((#1 > 0 ? (#2 > 0 ?
           785 (\n@mod@p{-atan2((t - (#2)),-(#1))}{2*pi}) :
           786 (-atan2((t - (#2)),-(#1)))) :
           787 (-atan2((t - (#2)),-(#1))))*\bp@ph@scale)}
```

Parametric function for linear approximation of the phase of a complex pole.

```

           788 \newcommand*\PhPoleLin[2]{
           789 ((abs(#1)+abs(#2) == 0 ? -pi/2 :
           790 (t < (sqrt(\n@pow{#1}{2} + \n@pow{#2}{2}) /
           791 (\n@pow{10}{sqrt(\n@pow{#1}{2}/(\n@pow{#1}{2} + \n@pow{#2}{2})))) ?
           792 (-atan2(-(#2),-(#1))) :
           793 (t >= (sqrt(\n@pow{#1}{2} + \n@pow{#2}{2}) *
           794 (\n@pow{10}{sqrt(\n@pow{#1}{2}/(\n@pow{#1}{2} + \n@pow{#2}{2})))) ?
           795 (#2>0?(#1>0?3*pi/2:-pi/2):-pi/2) :
           796 (-atan2(-(#2),-(#1)) + (log10(t/(sqrt(\n@pow{#1}{2} + \n@pow{#2}{2}) /
           797 (\n@pow{10}{sqrt(\n@pow{#1}{2}/(\n@pow{#1}{2} +
           798 \n@pow{#2}{2})))))))*((#2>0?(#1>0?3*pi/2:-pi/2):-pi/2) + atan2(-(#2),-
           799 (#1)))/
           800 (\log10(\n@pow{10}{sqrt((4*\n@pow{#1}{2})/
           (\n@pow{#1}{2} + \n@pow{#2}{2})))))))*\bp@ph@scale)}

```

Parametric function for asymptotic approximation of the phase of a complex pole.

```

           801 \newcommand*\PhPoleAsymp[2]{((t < (sqrt(\n@pow{#1}{2} + \n@pow{#2}{2})) ?

```

```

802 (-atan2(-(#2),-(#1))) :
803 (#2>0?(#1>0?3*pi/2:-pi/2):-pi/2))*\bp@ph@scale)}

```

\MagZero Plots of zeros are defined to be negative of plots of poles. The 0- is necessary due to a bug in gnuplot (fixed in version 5.4, patchlevel 3).

```

\MagZeroAsymp
\MagZeroLin 804 \newcommand*\MagZero{0-\MagPole}
\PhZero      805 \newcommand*\MagZeroLin{0-\MagPoleLin}
\PhZeroAsymp 806 \newcommand*\MagZeroAsymp{0-\MagPoleAsymp}
\PhZeroLin   807 \newcommand*\PhZero{0-\PhPole}
              808 \newcommand*\PhZeroLin{0-\PhPoleLin}
              809 \newcommand*\PhZeroAsymp{0-\PhPoleAsymp}

```

4.4 Second order systems.

Although second order systems can be dealt with using the macros defined so far, the following dedicated macros for second order systems involve less computation.

\MagCSPoles Consider the canonical second order transfer function $G(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$. We start with true, linear, and asymptotic magnitude plots for this transfer function.

```

\MagCSPolesAsymp
\MagCSPolesLin 810 \newcommand*\MagCSPoles[2]{(-20*log10(sqrt(\n@pow{\n@pow{#2}{2}
\PhCSPoles      811 - \n@pow{t}{2}}{2} + \n@pow{2*#1*#2*t}{2}}))}
\PhCSPolesAsymp 812 \newcommand*\MagCSPolesLin[2]{(t < #2 ? -40*log10(#2) : - 40*log10(t))}
\PhCSPolesLin   813 \newcommand*\MagCSPolesAsymp{\MagCSPolesLin}

```

\MagCSZeros True, linear, and asymptotic phase plots for the canonical second order transfer function.

```

\MagCSZerosAsymp 814 \newcommand*\PhCSPoles[2]{((-atan2((2*(#1)*(#2)*t),(\n@pow{#2}{2}
\MagCSZerosLin   815 - \n@pow{t}{2}))) * \bp@ph@scale)}
\PhCSZeros       816 \newcommand*\PhCSPolesLin[2]{((t < (#2 / (\n@pow{10}{abs(#1)})) ?
\PhCSZerosAsymp 817 0 :
\PhCSZerosLin   818 (t >= (#2 * (\n@pow{10}{abs(#1)})) ?
819 (#1>0 ? -pi : pi) :
820 (#1>0 ? (-pi*(log10(t*(\n@pow{10}{#1})/#2))/(2*#1)) :
821 (pi*(log10(t*(\n@pow{10}{abs(#1)})/#2))/(2*abs(#1)))) * \bp@ph@scale)}
822 \newcommand*\PhCSPolesAsymp[2]{((#1>0?(t<#2?0:-
pi):(t<#2?0:pi))*\bp@ph@scale)}

```

Plots of the inverse function $G(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$ are defined to be negative of plots of poles. The 0- is necessary due to a bug in gnuplot (fixed in version 5.4, patchlevel 3).

```

823 \newcommand*\MagCSZeros{0-\MagCSPoles}
824 \newcommand*\MagCSZerosLin{0-\MagCSPolesLin}
825 \newcommand*\MagCSZerosAsymp{0-\MagCSPolesAsymp}
826 \newcommand*\PhCSZeros{0-\PhCSPoles}
827 \newcommand*\PhCSZerosLin{0-\PhCSPolesLin}
828 \newcommand*\PhCSZerosAsymp{0-\PhCSPolesAsymp}

```

\MagCSPolesPeak These macros are used to add a resonant peak to linear and asymptotic plots of canonical second order poles and zeros. Since the plots are parametric, a separate \draw command is needed to add a vertical arrow.

```

829 \newcommand*\MagCSPolesPeak[3][]{
830 \draw[#1,->] (axis cs:{#3},{-40*log10(#3)}) --
831 (axis cs:{#3},{-40*log10(#3)-20*log10(2*abs(#2))})
832 }
833 \newcommand*\MagCSZerosPeak[3][]{
834 \draw[#1,->] (axis cs:{#3},{40*log10(#3)}) --
835 (axis cs:{#3},{40*log10(#3)+20*log10(2*abs(#2))})
836 }

```

\MagSOPoles Consider a general second order transfer function $G(s) = \frac{1}{s^2 + as + b}$. We start with true, linear, and asymptotic magnitude plots for this transfer function.

```

\MagSOPolesAsymp
\MagSOPolesLin 837 \newcommand*\MagSOPoles[2]{
\PhSOPoles      838 (-20*log10(sqrt(\n@pow{#2} - \n@pow{t}{2}}{2} + \n@pow{#1*t}{2}}))}
\PhSOPolesAsymp 839 \newcommand*\MagSOPolesLin[2]{
\PhSOPolesLin
\MagSOPoles
\MagSOPolesAsymp
\MagSOPolesLin
\PhSOPoles
\PhSOPolesAsymp

```

```

840 (t < sqrt(abs(#2)) ? -20*log10(abs(#2)) : - 40*log10(t))}
841 \newcommand*\MagSOPolesAsymp{\MagSOPolesLin}

```

True, linear, and asymptotic phase plots for the general second order transfer function.

```

842 \newcommand*\PhSOPoles[2]{((-atan2((#1)*t,((#2) - \n@pow{t}{2}))) * \bp@ph@scale)}
843 \newcommand*\PhSOPolesLin[2]{((#2>0 ?
844 \PhCSPolesLin{(#1/(2*sqrt(#2)))}{(sqrt(#2))} :
845 (#1>0 ? -pi : pi))}
846 \newcommand*\PhSOPolesAsymp[2]{((#2>0 ?
847 \PhCSPolesAsymp{(#1/(2*sqrt(#2)))}{(sqrt(#2))} :
848 (#1>0 ? -pi : pi))}

```

Plots of the inverse function $G(s) = s^2 + as + b$ are defined to be negative of plots of poles. The 0- is necessary due to a bug in **gnuplot** (fixed in version 5.4, patchlevel 3).

```

849 \newcommand*\MagSOPolesPeak{0-\MagSOPoles}
850 \newcommand*\MagSOPolesLinPeak{0-\MagSOPolesLin}
851 \newcommand*\MagSOPolesAsympPeak{0-\MagSOPolesAsymp}
852 \newcommand*\PhSOPolesPeak{0-\PhSOPoles}
853 \newcommand*\PhSOPolesLinPeak{0-\PhSOPolesLin}
854 \newcommand*\PhSOPolesAsympPeak{0-\PhSOPolesAsymp}

```

\MagSOPolesPeak These macros are used to add a resonant peak to linear and asymptotic plots of general
\MagSOPolesPeak second order poles and zeros. Since the plots are parametric, a separate **\draw** command
is needed to add a vertical arrow.

```

855 \newcommand*\MagSOPolesPeak[3]{\draw[>] (axis cs:{sqrt(abs(#3))},{-20*log10(abs(#3))} --
856 (axis cs:{sqrt(abs(#3))},{-20*log10(abs(#3)) -
857 20*log10(abs(#2/sqrt(abs(#3)))});
858 }
859 }
860 \newcommand*\MagSOPolesPeak[3]{\draw[>] (axis cs:{sqrt(abs(#3))},{20*log10(abs(#3))} --
861 (axis cs:{sqrt(abs(#3))},{20*log10(abs(#3)) +
862 20*log10(abs(#2/sqrt(abs(#3)))});
863 }
864 }

```

4.5 Commands for Bode plots

4.5.1 User macros

\BodeZPK This macro takes lists of complex poles and zeros of the form **{re,im}**, and values of gain and delay as inputs and constructs parametric functions for the Bode magnitude and phase plots. This is done by adding together the parametric functions generated by the macros for individual zeros, poles, gain, and delay, described above. The parametric functions are then plotted in a **tikzpicture** environment using the **\addplot** macro. Unless the package is loaded with the option **pgf**, the parametric functions are evaluated using **gnuplot**.

```

865 \NewDocumentCommand{\BodeZPK}{0}{m G{} G{}}{%

```

The macro now accepts an optional argument followed by a mandatory ZPK specification and two optional frequency arguments. If the frequency arguments are not provided, the **pgfkeys** interface introduced in v3.0 is used. Otherwise, the legacy interface is used for backward compatibility.

```

866 \pgfutil@ifempty{#3}{%
867 \pgfkeys{/bodeplot/.cd, reset}
868 \pgfkeys{/bodeplot/.cd, #1}
869 \pgfkeysgetvalue{/bodeplot/@axes/mag}{\bp@mag@axes}
870 \pgfkeysgetvalue{/bodeplot/@axes/ph}{\bp@ph@axes}
871 \pgfkeysgetvalue{/bodeplot/@group}{\bp@group}
872 \pgfkeysgetvalue{/bodeplot/@approx}{\bp@approx}
873 \pgfkeysgetvalue{/bodeplot/@commands/mag}{\bp@mag@commands}
874 \pgfkeysgetvalue{/bodeplot/@commands/ph}{\bp@ph@commands}
875 \pgfkeysgetvalue{/bodeplot/@tikz}{\bp@tikz}
876 \pgfkeysgetvalue{/bodeplot/@prefix}{\bp@user@prefix}

```



```

877 \bp@zpk@new@to@legacy{#2}
878 }{%
879 \if@radarg
880 \pgfkeys{/bodeplot/phase unit=rad}
881 \else
882 \pgfkeys{/bodeplot/phase unit=deg}
883 \fi
884 \if@hzarg
885 \pgfkeys{/bodeplot/frequency unit=Hz}
886 \else
887 \pgfkeys{/bodeplot/frequency unit=rad}
888 \fi
889 \bp@parse@opt{#1}
890 \edef\bp@legacy{#2}
891 \edef\bp@domain@start{#3}
892 \edef\bp@domain@end{#4}
893 }%
894
895 \edef\bp@tmp{\noexpand\begin{tikzpicture}
896 [\unexpanded\expandafter{\bp@tikz}]}
897 \bp@tmp
898 \gdef\bp@mag{}
899 \gdef\bp@ph{}
900 \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@approx}{\bp@legacy}
901 \edef\bp@tmp{\noexpand\begin{groupplot}[
902 bp@style,
903 xmin=\bp@domain@start,
904 xmax=\bp@domain@end,
905 domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
906 height=2.5cm,
907 xmode=log,
908 group style = {group size = 1 by 2,vertical sep=0.25cm},
909 \unexpanded\expandafter{\bp@group}
910 ]}
911 \bp@tmp
912 \edef\bp@tmp@mag{\noexpand\nextgroupplot
913 [ylabel={Gain (dB)}, xmajor ticks=false, \bp@mag@axes]
914 \noexpand\addplot [bp@freq@filter, variable=t, thick,
915 \unexpanded\expandafter{\bp@mag@plot}]}
916 \edef\bp@tmp@ph{\noexpand\nextgroupplot
917 [bp@ph@y@label, bp@freq@label, \bp@ph@axes]
918 \noexpand\addplot [bp@freq@filter, variable=t, thick, trig format plots=rad,
919 \unexpanded\expandafter{\bp@ph@plot}]}
920 \if@pgfarg
921 \bp@tmp@mag {\bp@mag};
922 \bp@mag@commands
923 \bp@tmp@ph {\bp@ph};
924 \bp@ph@commands
925 \else
926 \stepcounter{bp@gnuplot@id}
927 \edef\gnu@id{\arabic{bp@gnuplot@id}}
928 \bp@gnu@plot{\bp@mag}{\gnu@id}
929 \expandafter\bp@tmp@mag\bp@gnu@cmd
930 \bp@mag@commands
931 \stepcounter{bp@gnuplot@id}
932 \edef\gnu@id{\arabic{bp@gnuplot@id}}
933 \bp@gnu@plot{\bp@ph}{\gnu@id}
934 \expandafter\bp@tmp@ph\bp@gnu@cmd
935 \bp@ph@commands
936 \fi
937 \end{groupplot}
938 \end{tikzpicture}
939 }

```

The following code handles active characters to avoid conflicts with ‘babel.’

```

940 \AtBeginDocument{%
941   \if@babel
942   \let\Orig@BodeZPK\BodeZPK
943   \renewcommand{\BodeZPK}{%
944     \expandafter\shorthandoff\expandafter{\bp@short@list}
945     \BodeZPK@Shorthandoff
946   }
947   \newcommand{\BodeZPK@Shorthandoff}[4][{}]{%
948     \Orig@BodeZPK[#1]{#2}{#3}{#4}
949     \expandafter\shorthandon\expandafter{\bp@short@list}
950   }
951   \fi
952 }

```

\BodeTF Implementation of this macro is very similar to the **\BodeZPK** macro above. The only difference is the lack of linear and asymptotic plots and slightly different parsing of the mandatory arguments.

```

953 \NewDocumentCommand{\BodeTF}{ O{} m G{} G{} }{%
954   \pgfutil@ifempty{#3}{%
955     \pgfkeys{/bodeplot/.cd, reset}
956     \pgfkeys{/bodeplot/.cd, #1}
957     \pgfkeysgetvalue{/bodeplot/@axes/mag}{\bp@mag@axes}
958     \pgfkeysgetvalue{/bodeplot/@axes/ph}{\bp@ph@axes}
959     \pgfkeysgetvalue{/bodeplot/@group}{\bp@group}
960     \pgfkeysgetvalue{/bodeplot/@approx}{\bp@approx}
961     \pgfkeysgetvalue{/bodeplot/@commands/mag}{\bp@mag@commands}
962     \pgfkeysgetvalue{/bodeplot/@commands/ph}{\bp@ph@commands}
963     \pgfkeysgetvalue{/bodeplot/@tikz}{\bp@tikz}
964     \pgfkeysgetvalue{/bodeplot/@prefix}{\bp@user@prefix}
965     \bp@tf@new@to@legacy{#2}
966   }{%
967     \if@radarg
968       \pgfkeys{/bodeplot/phase unit=rad}
969     \else
970       \pgfkeys{/bodeplot/phase unit=deg}
971     \fi
972     \if@hzarg
973       \pgfkeys{/bodeplot/frequency unit=Hz}
974     \else
975       \pgfkeys{/bodeplot/frequency unit=rad}
976     \fi
977     \bp@parse@opt{#1}
978     \edef\bp@legacy{#2}
979     \edef\bp@domain@start{#3}
980     \edef\bp@domain@end{#4}
981   }%
982   \edef\bp@cmd{\noexpand\begin{tikzpicture}
983     [\unexpanded\expandafter{\bp@tikz}]}
984   \bp@cmd
985     \gdef\bp@mag{}
986     \gdef\bp@ph{}
987     \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}
988     \edef\bp@cmd{\noexpand\begin{groupplot}[
989       bp@style,
990       xmin=\bp@domain@start,
991       xmax=\bp@domain@end,
992       domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
993       height=2.5cm,
994       xmode=log,
995       group style = {group size = 1 by 2,vertical sep=0.25cm},
996       \unexpanded\expandafter{\bp@group}
997     ]}

```

```

998 \bp@cmd
999 \edef\bp@mag@cmd{\noexpand\nextgroupplot
1000 [ylabel={Gain (dB)}, xmajorticks=false, \bp@mag@axes]
1001 \noexpand\addplot [bp@freq@filter, variable=t, thick,
1002 \unexpanded\expandafter{\bp@mag@plot}]}
1003 \edef\bp@ph@cmd{\noexpand\nextgroupplot
1004 [bp@ph@y@label, bp@freq@label, \bp@ph@axes]
1005 \noexpand\addplot [bp@freq@filter, variable=t, thick, trig format plots=rad,
1006 \unexpanded\expandafter{\bp@ph@plot}]}
1007 \ifpgfarg
1008 \bp@mag@cmd {\bp@mag};
1009 \bp@mag@commands
1010 \bp@ph@cmd {\n@mod{\bp@ph}{2*pi*\bp@ph@scale}};
1011 \bp@ph@commands
1012 \else
1013 \stepcounter{bp@gnuplot@id}
1014 \edef\gnu@id{\arabic{bp@gnuplot@id}}
1015 \bp@gnu@plot{\bp@mag}{\gnu@id}
1016 \expandafter\bp@mag@cmd\bp@gnu@cmd
1017 \bp@mag@commands
1018 \stepcounter{bp@gnuplot@id}
1019 \edef\gnu@id{\arabic{bp@gnuplot@id}}
1020 \bp@gnu@unwrap@plot{\bp@ph}{\gnu@id}
1021 \expandafter\bp@ph@cmd\bp@gnu@cmd
1022 \bp@ph@commands
1023 \fi
1024 \end{groupplot}
1025 \end{tikzpicture}
1026 }

```

The following code handles active characters to avoid conflicts with ‘babel.’

```

1027 \AtBeginDocument{
1028 \if@babel
1029 \let\Orig@BodeTF\BodeTF
1030 \renewcommand{\BodeTF}{%
1031 \expandafter\shorthandoff\expandafter{\bp@short@list}
1032 \BodeTF@Shorthandoff
1033 }
1034 \newcommand{\BodeTF@Shorthandoff}[4][[]]{%
1035 \Orig@BodeTF[#1]{#2}{#3}{#4}
1036 \expandafter\shorthandon\expandafter{\bp@short@list}
1037 }
1038 \fi
1039 }

```

\addBodeZPKPlots This macro is designed to issue multiple `\addplot` macros for the same set of poles, zeros, gain, and delay. All of the work is done by the `\bp@ZPK@plot` macro.

```

1040 \NewDocumentCommand{\addBodeZPKPlots}{ 0{} m m }{%
1041 \bp@contains@equal{#3}{
1042 \pgfkeys{/bodeplot/add/.cd, reset}
1043 \pgfkeys{/bodeplot/add/.cd, #1}
1044 \bp@zpk@new@to@legacy{#3}
1045 \expandafter\bp@fix@add@opt\expandafter{\bp@add@0}
1046 }{
1047 \bp@fix@add@opt{#1}
1048 \edef\bp@legacy{#3}
1049 }
1050 \foreach \approx/\opt in \bp@add@0 {
1051 \ifx\approx\@empty\else
1052 \gdef\bp@plot{
1053 \gdef\bp@tmp{
1054 \ifnum\pdf@strcmp{#2}{phase}=0
1055 \bp@ZPK@plot{\bp@tmp}{\bp@plot}{\approx}{\bp@legacy}
1056 \else

```

```

1057         \bp@ZPK@plot{\bp@plot}{\bp@tmp}{\approx}{\bp@legacy}
1058     \fi
1059     \if@pgfarg
1060         \edef\bp@cmd{\noexpand\addplot [bp@freq@filter,
1061             domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
1062             variable=t, thick, trig format plots=rad, \unexpanded\expandafter{\opt}}
1063             \bp@cmd {\bp@plot};
1064     \else
1065         \stepcounter{bp@gnuplot@id}
1066         \edef\gnu@id{\arabic{bp@gnuplot@id}}
1067         \bp@gnu@plot[\bp@freq@scale]{\bp@plot}{\gnu@id}
1068         \edef\bp@cmd{\noexpand\addplot [variable=t, thick,
1069             \unexpanded\expandafter{\opt}}
1070             \expandafter\bp@cmd\bp@gnu@cmd
1071     \fi
1072 \fi
1073 }
1074 }

```

\addBodeTFPlot This macro is designed to issues a single **\addplot** macros for the set of coefficients and delay. All of the work is done by the **\bp@TF@plot** macro.

```

1075 \NewDocumentCommand{\addBodeTFPlot}{0}{m m}{%
1076     \bp@contains@equal{#3}{
1077         \bp@tf@new@to@legacy{#3}
1078     }{
1079         \edef\bp@legacy{#3}
1080     }
1081     \gdef\bp@plot{}
1082     \gdef\bp@tmp{}
1083     \ifnum\pdf@strcmp{#2}{phase}=0
1084         \bp@TF@plot{\bp@tmp}{\bp@plot}{\bp@legacy}
1085     \else
1086         \bp@TF@plot{\bp@plot}{\bp@tmp}{\bp@legacy}
1087     \fi
1088     \if@pgfarg
1089         \ifnum\pdf@strcmp{#2}{phase}=0
1090             \edef\bp@cmd{\noexpand\addplot [bp@freq@filter,
1091                 domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
1092                 variable=t, thick, trig format plots=rad,
1093                 \unexpanded\expandafter{#1}}
1094                 \bp@cmd {\n@mod{\bp@plot}{2*pi}};
1095         \else
1096             \edef\bp@cmd{\noexpand\addplot [bp@freq@filter,
1097                 domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
1098                 variable=t, thick, \unexpanded\expandafter{#1}}
1099                 \bp@cmd {\bp@plot};
1100         \fi
1101     \else
1102         \stepcounter{bp@gnuplot@id}
1103         \edef\gnu@id{\arabic{bp@gnuplot@id}}
1104         \edef\bp@cmd{\noexpand\addplot [variable=t, thick,
1105             \unexpanded\expandafter{#1}}
1106             \ifnum\pdf@strcmp{#2}{phase}=0
1107                 \bp@gnu@unwrap@plot[\bp@freq@scale]{\bp@plot}{\gnu@id}
1108                 \expandafter\bp@cmd\bp@gnu@cmd
1109             \else
1110                 \bp@gnu@plot[\bp@freq@scale]{\bp@plot}{\gnu@id}
1111                 \expandafter\bp@cmd\bp@gnu@cmd
1112             \fi
1113     \fi
1114 }

```

\addBodeComponentPlot This macro is designed to create a single **\addplot** macro capable of plotting linear

combinations of the basic components described in Section 3.1.1. The only work to do here is to handle the `pgf` package option.

```

1115 \newcommand{\addBodeComponentPlot}[2][]{
1116   \if@pgfarg
1117     \edef\bp@cmd{\noexpand\addplot [bp@freq@filter,
1118       domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
1119       variable=t, thick, trig format plots=rad, #1]}
1120     \bp@cmd {#2};
1121   \else
1122     \stepcounter{bp@gnuplot@id}
1123     \edef\gnu@id{\arabic{bp@gnuplot@id}}
1124     \edef\bp@cmd{\noexpand\addplot [variable=t, thick, #1]}
1125     \bp@gnu@plot[\bp@freq@scale]{#2}{\gnu@id}
1126     \expandafter\bp@cmd\bp@gnu@cmd
1127   \fi
1128 }

```

BodePhPlot (*env.*) An environment to host phase plot macros that pass parametric functions to `\addplot` macros. Uses the defaults specified in `bp@style` to create a shortcut that includes the `tikzpicture` and `semilogaxis` environments. The body of the environment is grabbed as a macro to maintain compatibility with externalization in `tikz`.

```

1129 \AtBeginDocument{%
1130   \if@babel
1131     \AddToHook{env/BodePhPlot/begin}{%
1132       \expandafter\shorthandoff\expandafter{\bp@short@list}
1133     }
1134     \AddToHook{env/BodePhPlot/end}{%
1135       \expandafter\shorthandon\expandafter{\bp@short@list}
1136     }
1137   \fi
1138 }
1139 \NewDocumentEnvironment{BodePhPlot}{0} G{} G{} +b {}{
1140   \pgfutil@ifempty{#2}{%
1141     \pgfkeys{/bodeplot/env/.cd, reset}
1142     \pgfkeys{/bodeplot/env/.cd, #1}
1143     \pgfkeysgetvalue{/bodeplot/env/@tikz}{\bp@tikz}
1144     \pgfkeysgetvalue{/bodeplot/env/@prefix}{\bp@user@prefix}
1145   }{%
1146     \bp@parse@env@opt{#1}
1147     \edef\bp@domain@start{#2}
1148     \edef\bp@domain@end{#3}
1149   }
1150   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1151     [\unexpanded\expandafter{\bp@tikz}]}
1152   \bp@cmd
1153     \edef\bp@cmd{\noexpand\begin{semilogaxis}[
1154       bp@ph@y@label,
1155       bp@freq@label,
1156       bp@style,
1157       xmin={\bp@domain@start},
1158       xmax={\bp@domain@end},
1159       domain=\bp@domain@start:\bp@domain@end,
1160       height=2.5cm,
1161       \unexpanded\expandafter{\bp@axes}
1162     ]}
1163   \bp@cmd
1164     #4
1165   \end{semilogaxis}
1166 \end{tikzpicture}
1167 }{}

```

BodeMagPlot (*env.*) An environment to host magnitude plot macros that pass parametric functions to `\addplot` macros. Uses the defaults specified in `bp@style` to create a shortcut that

includes the `tikzpicture` and `semilogaxis` environments.

```

1168 \AtBeginDocument{%
1169   \if@babel
1170     \AddToHook{env/BodeMagPlot/begin}{%
1171       \expandafter\shorthandoff\expandafter{\bp@short@list}
1172     }
1173     \AddToHook{env/BodeMagPlot/end}{%
1174       \expandafter\shorthandon\expandafter{\bp@short@list}
1175     }
1176   \fi
1177 }
1178 \NewDocumentEnvironment{BodeMagPlot}{0}{G}{G}{+b}{%
1179   \pgfutil@ifempty{#2}{%
1180     \pgfkeys{/bodeplot/env/.cd, reset}
1181     \pgfkeys{/bodeplot/env/.cd, #1}
1182     \pgfkeysgetvalue{/bodeplot/env/@tikz}{\bp@tikz}
1183     \pgfkeysgetvalue{/bodeplot/env/@prefix}{\bp@user@prefix}
1184   }{%
1185     \bp@parse@env@opt{#1}
1186     \edef\bp@domain@start{#2}
1187     \edef\bp@domain@end{#3}
1188   }
1189   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1190     [\unexpanded\expandafter{\bp@tikz}]}
1191   \bp@cmd
1192   \edef\bp@cmd{\noexpand\begin{semilogxaxis}[
1193     bp@style,
1194     bp@freq@label,
1195     xmin={\bp@domain@start},
1196     xmax={\bp@domain@end},
1197     domain=\bp@domain@start:\bp@domain@end,
1198     height=2.5cm,
1199     ylabel={Gain (dB)},
1200     \unexpanded\expandafter{\bp@axes}
1201   ]}
1202   \bp@cmd
1203   #4
1204   \end{semilogxaxis}
1205 \end{tikzpicture}
1206 }{}

```

\addBodePlot Unified macro to add Bode plots for both ZPK and TF system representations inside a BodePlot environment. Supports both pgfkeys interface introduced in v3.0 and the legacy interface.

```

1207 \NewDocumentCommand{\addBodePlot}{0}{m G}{ }{%
1208   \gdef\bp@mag{}
1209   \gdef\bp@ph{}
1210   \pgfutil@ifempty{#3}{%
1211     \pgfkeys{/bodeplot/add/.cd, reset}
1212     \pgfkeys{/bodeplot/add/.cd, #1}
1213     \expandafter\bp@parse@add@Bode@opt\expandafter{\bp@add@0}
1214     \bp@contains@num{#2}{%
1215       \bp@tf@new@to@legacy{#2}
1216       \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}
1217       \edef\bp@mode{tf}
1218     }{
1219       \bp@zpk@new@to@legacy{#2}
1220       \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@approx}{\bp@legacy}
1221       \edef\bp@mode{zpk}
1222     }
1223   }{
1224     \bp@parse@add@Bode@opt{#1}
1225     \ifnum\pdf@strcmp{#2}{zpk}=0

```

```

1226     \edef\bp@mode{zpk}
1227     \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@approx}{#3}
1228   \else
1229     \ifnum\pdf@strcmp{#2}{tf}=0
1230       \edef\bp@mode{tf}
1231       \bp@TF@plot{\bp@mag}{\bp@ph}{#3}
1232     \else
1233       \PackageError {bodeplot} {Unknown system representation `#2'.}
1234       {Supported representations are `zpk' and `tf'}.
1235     \fi
1236   \fi
1237 }
1238 \ifpgfarg
1239   \xdef\bp@mag@cmd{\unexpanded\expandafter{\bp@mag@cmd}\noexpand\addplot
1240   [bp@freq@filter, domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain},
1241   variable=t, thick, trig format plots=rad,
1242   \unexpanded\expandafter{\bp@plot}] {\bp@mag};}
1243   \xdef\bp@ph@cmd{\unexpanded\expandafter{\bp@ph@cmd}\noexpand\addplot
1244   [bp@freq@filter, domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain},
1245   variable=t, thick, trig format plots=rad,
1246   \unexpanded\expandafter{\bp@plot}] {\bp@ph};}
1247 \else
1248   \stepcounter{bp@gnuplot@id}
1249   \edef\gnu@id{\arabic{bp@gnuplot@id}}
1250   \bp@gnu@plot{\bp@mag}{\gnu@id}
1251   \xdef\bp@mag@cmd{\unexpanded\expandafter{\bp@mag@cmd}\noexpand\addplot
1252   [variable=t, thick, \unexpanded\expandafter{\bp@plot}]\bp@gnu@cmd}
1253   \stepcounter{bp@gnuplot@id}
1254   \edef\gnu@id{\arabic{bp@gnuplot@id}}
1255   \ifnum\pdf@strcmp{\bp@mode}{zpk}=0
1256     \bp@gnu@plot{\bp@ph}{\gnu@id}
1257   \else
1258     \ifnum\pdf@strcmp{\bp@mode}{tf}=0
1259       \bp@gnu@unwrap@plot{\bp@ph}{\gnu@id}
1260     \fi
1261   \fi
1262   \xdef\bp@ph@cmd{\unexpanded\expandafter{\bp@ph@cmd}\noexpand\addplot
1263   [variable=t, thick, \unexpanded\expandafter{\bp@plot}]\bp@gnu@cmd}
1264 \fi
1265 }

```

BodePlot (*env.*) An environment that works with the unified `\addBodePlot` macro. Creates a grouped plot with magnitude on top and phase on bottom, automatically collecting and inserting plot commands generated by `\addBodePlot` calls within the environment body.

```

1266 \NewDocumentEnvironment{BodePlot}{ 0{} G{} G{} +b }{
1267   \pgfutil@ifempty{#2}{%
1268     \pgfkeys{/bodeplot/combinedenv/.cd, reset}
1269     \pgfkeys{/bodeplot/combinedenv/.cd, #1}
1270     \pgfkeysgetvalue{/bodeplot/combinedenv/@group}{\bp@group}
1271     \pgfkeysgetvalue{/bodeplot/combinedenv/@approx}{\bp@approx}
1272     \pgfkeysgetvalue{/bodeplot/combinedenv/@commands/mag}{\bp@mag@commands}
1273     \pgfkeysgetvalue{/bodeplot/combinedenv/@commands/ph}{\bp@ph@commands}
1274     \pgfkeysgetvalue{/bodeplot/combinedenv/@tikz}{\bp@tikz}
1275     \pgfkeysgetvalue{/bodeplot/combinedenv/@prefix}{\bp@user@prefix}
1276   }{%
1277     \bp@parse@opt{#1}
1278     \edef\bp@domain@start{#2}
1279     \edef\bp@domain@end{#3}
1280   }
1281   \gdef\bp@mag@cmd{}
1282   \gdef\bp@ph@cmd{}
1283   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1284   [\unexpanded\expandafter{\bp@tikz}]}

```

```

1285 \bp@cmd
1286 \edef\bp@cmd{\noexpand\begin{groupplot}[
1287   bp@style,
1288   xmin=\bp@domain@start,
1289   xmax=\bp@domain@end,
1290   domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
1291   height=2.5cm,
1292   xmode=log,
1293   group style = {group size = 1 by 2,vertical sep=0.25cm},
1294   \unexpanded\expandafter{\bp@group}
1295 ]}
1296 \bp@cmd
1297 #4
1298 \edef\temp@mag@cmd{\noexpand\nextgroupplot
1299 [ylabel={Gain (dB)}, xmajorticks=false,
1300 \unexpanded\expandafter{\bp@mag@axes}]}
1301 \edef\temp@ph@cmd{\noexpand\nextgroupplot
1302 [bp@ph@y@label, bp@freq@label,
1303 \unexpanded\expandafter{\bp@ph@axes}]}
1304 \temp@mag@cmd
1305 \bp@mag@cmd
1306 \temp@ph@cmd
1307 \bp@ph@cmd
1308 \end{groupplot}
1309 \end{tikzpicture}
1310 {}

```

4.5.2 Internal macros

\bp@parse@complex Parses complex numbers in various formats (e.g., $1+2i$, $3-4i$, 5 , i). Uses `expl3` `regex` to extract real and imaginary parts. Sets `\bp@realpart`, `\bp@imagpart`, and `\bp@complex`.

```

1311 \ExplSyntaxOn
1312 \tl_new:N \l_real_tl
1313 \tl_new:N \l_imag_tl
1314 \tl_new:N \bp@complex
1315 \tl_new:N \bp@realpart
1316 \tl_new:N \bp@imagpart
1317 \NewDocumentCommand{\bp@parse@complex}{m}{
1318   \tl_set:Nn \l_tmpa_tl {#1}
1319   \tl_remove_all:Nn \l_tmpa_tl { ~ }
1320
1321   \regex_match:nVTF { i } \l_tmpa_tl
1322   {%
1323     \regex_extract_once:nVNTF { ^([+|-]?[0-9.]+)([+|-][0-
1324       9.]+)i$ } \l_tmpa_tl \l_tmpa_seq
1325     {%
1326       \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl
1327       \seq_pop_left:NN \l_tmpa_seq \l_real_tl
1328       \seq_pop_left:NN \l_tmpa_seq \l_imag_tl
1329     }
1330     \regex_extract_once:nVNTF { ^([+|-]?[0-
1331       9.]+)i$ } \l_tmpa_tl \l_tmpa_seq
1332     {
1333       \seq_pop_left:NN \l_tmpa_seq \l_tmpb_tl
1334       \seq_pop_left:NN \l_tmpa_seq \l_imag_tl
1335       \tl_set:Nn \l_real_tl {0}
1336       \tl_if_empty:NT \l_imag_tl { \tl_set:Nn \l_imag_tl {1} }
1337       \str_if_eq:VnT \l_imag_tl {+} { \tl_set:Nn \l_imag_tl {1} }
1338       \str_if_eq:VnT \l_imag_tl {-} { \tl_set:Nn \l_imag_tl {-1} }
1339     }
1340   }

```



```

1340         \tl_set:Nn \l_real_tl {0}
1341         \tl_set:Nn \l_imag_tl {0}
1342     }
1343 }
1344 }
1345 {%
1346     \tl_set_eq:NN \l_real_tl \l_tmpa_tl
1347     \tl_set:Nn \l_imag_tl {0}
1348 }
1349
1350 \tl_gset_eq:NN \bp@realpart \l_real_tl
1351 \tl_gset_eq:NN \bp@imagpart \l_imag_tl
1352 \tl_gset:Nx \bp@complex { { \l_real_tl } , { \l_imag_tl } } }
1353 }

```

`\bp_if_contains:nnTF` Helper function to check if a token list contains a substring.

```

1354 \tl_new:N \l__bode_check_tl
1355 \cs_new_protected:Npn \bp_if_contains:nnTF #1#2#3#4 {
1356     \tl_set:Nx \l__bode_check_tl {#1}
1357     \tl_if_in:VnTF \l__bode_check_tl {#2} {#3} {#4}
1358 }

```

`\bp@contains@equal` Checks if argument contains '=' to distinguish new from legacy interface.

```

1359 \NewDocumentCommand{\bp@contains@equal}{m m m}{%
1360     \bp_if_contains:nnTF {#1} {=} {#2} {#3}%
1361 }

```

`\bp@contains@num` Checks if argument contains 'numerator' keyword.

```

1362 \NewDocumentCommand{\bp@contains@num}{m m m}{%
1363     \bp_if_contains:nnTF {#1} {numerator} {#2} {#3}%
1364 }
1365 \ExplSyntaxOff

```

`\bp@fix@add@opt` Processes options for addplot commands, organizing by approximation type.

```

1366 \NewDocumentCommand{\bp@fix@add@opt}{ m }{%
1367     \gdef\bp@add@0{}
1368     \gdef\bp@add@tmp{}
1369     \foreach \approx/\opt in {#1} {
1370         \ifx\approx\@empty\else
1371             \ifnum\pdf@strcmp{\approx}{linear}=0
1372                 \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0}%
1373                     linear/{\unexpanded\expandafter{\opt}},}
1374             \else
1375                 \ifnum\pdf@strcmp{\approx}{asymptotic}=0
1376                     \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0}%
1377                         asymptotic/{\unexpanded\expandafter{\opt}},}
1378                 \else
1379                     \ifnum\pdf@strcmp{\approx}{true}=0
1380                         \xdef\bp@add@tmp{\unexpanded\expandafter{\bp@add@tmp}%
1381                             \unexpanded\expandafter{\opt}},}
1382                     \else
1383                         \xdef\bp@add@tmp{\unexpanded\expandafter{\bp@add@tmp}%
1384                             \unexpanded\expandafter{\approx}},}
1385                     \fi
1386                 \fi
1387             \fi
1388         \fi
1389     }
1390     \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0}%
1391         true/{\unexpanded\expandafter{\bp@add@tmp}}}
1392 }

```

`\bp@add` Helper macro to build up magnitude and phase expressions by adding contributions from zeros, poles, gain, and delay.

```

1393 \newcommand*\bp@add}[3]{
1394   \ifcat$\detokenize\expandafter{#1}$
1395     \xdef#1{\unexpanded\expandafter{#1 0+#2}}
1396   \else
1397     \xdef#1{\unexpanded\expandafter{#1+#2}}
1398   \fi
1399   \foreach \y [count=\n] in #3 {
1400     \xdef#1{\unexpanded\expandafter{#1}{\y}}
1401     \xdef\Last@LoopValue{\n}
1402   }
1403   \ifnum\Last@LoopValue=1
1404     \xdef#1{\unexpanded\expandafter{#1}{0}}
1405   \fi
1406 }

```

`\bp@ZPK@plot` Builds magnitude and phase plot expressions from zero-pole-gain representation. Handles linear, asymptotic, and true Bode plot approximations.

```

1407 \newcommand{\bp@ZPK@plot}[4]{
1408   \edef\bp@list{#4}
1409   \foreach \feature/\values in \bp@list {
1410     \ifx\values\empty\else
1411       \ifnum\pdf@strcmp{\feature}{z}=0
1412         \foreach \z in \values {
1413           \ifx\z\empty\else
1414             \ifnum\pdf@strcmp{#3}{linear}=0
1415               \bp@add{#2}{\PhZeroLin}{\z}
1416               \bp@add{#1}{\MagZeroLin}{\z}
1417             \else
1418               \ifnum\pdf@strcmp{#3}{asymptotic}=0
1419                 \bp@add{#2}{\PhZeroAsymp}{\z}
1420                 \bp@add{#1}{\MagZeroAsymp}{\z}
1421               \else
1422                 \bp@add{#2}{\PhZero}{\z}
1423                 \bp@add{#1}{\MagZero}{\z}
1424               \fi
1425             \fi
1426           \fi
1427         }
1428       \fi
1429       \ifnum\pdf@strcmp{\feature}{p}=0
1430         \foreach \p in \values {
1431           \ifx\p\empty\else
1432             \ifnum\pdf@strcmp{#3}{linear}=0
1433               \bp@add{#2}{\PhPoleLin}{\p}
1434               \bp@add{#1}{\MagPoleLin}{\p}
1435             \else
1436               \ifnum\pdf@strcmp{#3}{asymptotic}=0
1437                 \bp@add{#2}{\PhPoleAsymp}{\p}
1438                 \bp@add{#1}{\MagPoleAsymp}{\p}
1439               \else
1440                 \bp@add{#2}{\PhPole}{\p}
1441                 \bp@add{#1}{\MagPole}{\p}
1442               \fi
1443             \fi
1444           \fi
1445         }
1446       \fi
1447       \ifnum\pdf@strcmp{\feature}{k}=0
1448         \ifnum\pdf@strcmp{#3}{linear}=0
1449           \bp@add{#2}{\PhKLin}{\values}
1450           \bp@add{#1}{\MagKLin}{\values}

```

```

1451     \else
1452     \ifnum\pdf@strcmp{#3}{asymptotic}=0
1453     \bp@add{#2}{\PhKAsymp}{\values}
1454     \bp@add{#1}{\MagKAsymp}{\values}
1455     \else
1456     \bp@add{#2}{\PhK}{\values}
1457     \bp@add{#1}{\MagK}{\values}
1458     \fi
1459   \fi
1460 \fi
1461 \ifnum\pdf@strcmp{\feature}{d}=0
1462 \ifnum\pdf@strcmp{#3}{linear}=0
1463 \PackageError {bodeplot}
1464 {Linear approximation for pure delays is not supported.}
1465 {Plot the true Bode plot using `true' instead of `linear'.}
1466 \else
1467 \ifnum\pdf@strcmp{#3}{asymptotic}=0
1468 \PackageError {bodeplot}
1469 {Asymptotic approximation for pure delays is not supported.}
1470 {Plot the true Bode plot using `true' instead of `asymptotic'..}
1471 \else
1472 \ifdim\values pt < 0pt
1473 \PackageError {bodeplot} {Delay needs to be a positive number.}
1474 \fi
1475 \bp@add{#2}{\PhDel}{\values}
1476 \bp@add{#1}{\MagDel}{\values}
1477 \fi
1478 \fi
1479 \fi
1480 \fi
1481 }
1482 }

```

`\bp@gnu@plot` Generates gnuplot commands for computing Bode plot data.

```

1483 \newcommand{\bp@gnu@plot}[3][1]{
1484 \xdef\bp@gnu@cmd{ gnuplot [raw gnuplot, id=#3, prefix=\bp@prefix]
1485 { set table $meta;
1486 set dummy t;
1487 set logscale x 10;
1488 set xrange [#1*\pgfkeysvalueof{/pgfplots/domain}*\#1];
1489 set samples \pgfkeysvalueof{/pgfplots/samples};
1490 plot #2;
1491 set table "\bp@prefix#3.table";
1492 plot "$meta" using ($1/(\bp@freq@scale)):(#2);
1493 };}
1494 }

```

`\bp@zpk@new@to@legacy` Converts the pgfkeys interface format to the legacy ZPK format. Parses complex zeros and poles using `\bp@parse@complex`.

```

1495 \NewDocumentCommand{\bp@zpk@new@to@legacy}{ m }{%
1496 \pgfkeys{/bodeplot/zpk/.cd, reset}
1497 \pgfkeys{/bodeplot/zpk/.cd, #1}
1498 \pgfkeysgetvalue{/bodeplot/zpk/@zeros}{\bp@z}
1499 \gdef\bp@z@list{}
1500 \ifx\bp@z@empty\else
1501 \foreach \z in \bp@z {
1502 \bp@parse@complex{\z}
1503 \xdef\bp@z@list{\unexpanded\expandafter{\bp@z@list} \bp@complex,}
1504 }
1505 \fi
1506 \pgfkeysgetvalue{/bodeplot/zpk/@poles}{\bp@p}
1507 \gdef\bp@p@list{}
1508 \ifx\bp@p@empty\else

```

```

1509 \foreach \p in \bp@p {
1510 \bp@parse@complex{\p}
1511 \xdef\bp@p@list{\unexpanded\expandafter{\bp@p@list} \bp@complex,}
1512 }
1513 \fi
1514 \pgfkeysgetvalue{/bodeplot/zpk/@gain}{\bp@k}
1515 \pgfkeysgetvalue{/bodeplot/zpk/@delay}{\bp@d}
1516 \xdef\bp@legacy{z/{\bp@z@list},p/{\bp@p@list},k/{\bp@k},d/{\bp@d}
1517 }

```

`\bp@tf@new@to@legacy` Converts the pgfkeys interface format to the legacy TF format.

```

1518 \NewDocumentCommand{\bp@tf@new@to@legacy}{ m }{%
1519 \pgfkeys{/bodeplot/tf/.cd, reset}
1520 \pgfkeys{/bodeplot/tf/.cd, #1}
1521 \pgfkeysgetvalue{/bodeplot/tf/@numerator}{\bp@num}
1522 \pgfkeysgetvalue{/bodeplot/tf/@denominator}{\bp@den}
1523 \pgfkeysgetvalue{/bodeplot/tf/@delay}{\bp@d}
1524 \gdef\bp@num@list{}
1525 \ifx\bp@num\@empty\else
1526 \foreach \n in \bp@num {
1527 \xdef\bp@num@list{\unexpanded\expandafter{\bp@num@list} \n,}
1528 }
1529 \fi
1530 \gdef\bp@den@list{}
1531 \ifx\bp@den\@empty\else
1532 \foreach \d in \bp@den {
1533 \xdef\bp@den@list{\unexpanded\expandafter{\bp@den@list} \d,}
1534 }
1535 \fi
1536 \xdef\bp@legacy{num/{\bp@num@list},den/{\bp@den@list},d/{\bp@d}
1537 }

```

`\bp@TF@plot` Builds magnitude and phase plot expressions from transfer function representation.

```

1538 \newcommand{\bp@TF@plot}[3]{
1539 \gdef\bp@num@re{0}
1540 \gdef\bp@num@im{0}
1541 \gdef\bp@den@re{0}
1542 \gdef\bp@den@im{0}
1543 \gdef\bp@loop@delay{0}
1544 \edef\bp@list{#3}
1545 \foreach \feature/\values in \bp@list {
1546 \ifx\values\empty\else
1547 \ifnum\pdf@strcmp{\feature}{num}=0
1548 \foreach \numcoeff [count=\numpow] in \values {
1549 \ifx\numcoeff\empty\else
1550 \xdef\bp@num@deg{\numpow}
1551 \fi
1552 }
1553 \foreach \numcoeff [count=\numpow] in \values {
1554 \ifx\numcoeff\empty\else
1555 \pgfmathtruncatemacro{\currentdegree}{\bp@num@deg-\numpow}
1556 \ifnum\currentdegree = 0
1557 \xdef\bp@num@re{\bp@num@re+\numcoeff}
1558 \else
1559 \ifodd\currentdegree
1560 \xdef\bp@num@im{\bp@num@im+(\numcoeff*(\n@pow{-
1561 1}{(\currentdegree-1)/2})*%
1562 (\n@pow{t}{\currentdegree}))}
1563 \else
1564 \xdef\bp@num@re{\bp@num@re+(\numcoeff*(\n@pow{-
1565 1}{(\currentdegree)/2})*%
1566 (\n@pow{t}{\currentdegree}))}
1567 \fi
1568 }
1569 }
1570 }

```

```

1566         \fi
1567     \fi
1568 }
1569 \fi
1570 \ifnum\pdf@strcmp{\feature}{den}=0
1571     \foreach \dencoeff [count=\denpow] in \values {
1572         \ifx\dencoeff\empty\else
1573             \xdef\bp@den@deg{\denpow}
1574         \fi
1575     }
1576     \foreach \dencoeff [count=\denpow] in \values {
1577         \ifx\dencoeff\empty\else
1578             \pgfmathtruncatemacro{\currentdegree}{\bp@den@deg-\denpow}
1579             \ifnum\currentdegree = 0
1580                 \xdef\bp@den@re{\bp@den@re+\dencoeff}
1581             \else
1582                 \ifodd\currentdegree
1583                     \xdef\bp@den@im{\bp@den@im+(\dencoeff*(\n@pow{-
1584 1}}{(\currentdegree-1)/2}))*%
1585                     (\n@pow{t}{\currentdegree})))}
1586                 \else
1587                     \xdef\bp@den@re{\bp@den@re+(\dencoeff*(\n@pow{-
1588 1}}{(\currentdegree)/2}))*%
1589                     (\n@pow{t}{\currentdegree})))}
1590                 \fi
1591             \fi
1592         \fi
1593     }
1594     \ifnum\pdf@strcmp{\feature}{d}=0
1595         \xdef\bp@loop@delay{\values}
1596     \fi
1597 \fi
1598 \xdef#2{((atan2((\bp@num@im),(\bp@num@re))-atan2((\bp@den@im),%
1599 (\bp@den@re))- \bp@loop@delay*t)*(\bp@ph@scale))}
1600 \xdef#1{(20*log10(sqrt((\n@pow{\bp@num@re}{2})+(\n@pow{\bp@num@im}{2}))) -
1601 %
1602 20*log10(sqrt((\n@pow{\bp@den@re}{2})+(\n@pow{\bp@den@im}{2}))))}

```

`\bp@gnu@unwrap@plot` Generates gnuplot commands with phase unwrapping for TF plots.

```

1603 \newcommand{\bp@gnu@unwrap@plot}[3][1]{
1604     \xdef\bp@gnu@cmd{ gnuplot [raw gnuplot, id=#3, prefix=\bp@prefix]
1605     { set table $meta;
1606       set dummy t;
1607       set logscale x 10;
1608       set trange [#1*\pgfkeysvalueof{/pgfplots/domain}*#1];
1609       set samples \pgfkeysvalueof{/pgfplots/samples};
1610       plot '+' using (t) : ((#2)/(\bp@ph@scale)) smooth unwrap;
1611       set table "\bp@prefix#3.table";
1612       plot "$meta" using ($1/(\bp@freq@scale)):(($2*\bp@ph@scale));
1613     };}
1614 }

```

`\bp@parse@opt` Parses options supplied to the main Bode macros. A **for** loop over tuples of the form `\obj/\typ/\opt` with a long list of nested if-else statements does the job. If the input `\obj` is `plot`, `axes`, `group`, `approx`, or `tikz` the corresponding `\opt` are passed, unexpanded, to the `\addplot` macro, the `\nextgroupplot` macro, the `groupplot` environment, the `\build@ZPK@plot` macro, and the `tikzpicture` environment, respectively. If `\obj` is `commands`, the corresponding `\opt` are stored, unexpanded, in the macros `\bp@ph@commands` and `\bp@mag@commands`, to be executed in appropriate `axis` environments.

```

1615 \newcommand{\bp@parse@opt}[1]{
1616   \gdef\bp@mag@axes{}
1617   \gdef\bp@ph@axes{}
1618   \gdef\bp@ph@plot{}
1619   \gdef\bp@mag@plot{}
1620   \gdef\bp@group{}
1621   \gdef\bp@approx{}
1622   \gdef\bp@ph@commands{}
1623   \gdef\bp@mag@commands{}
1624   \gdef\bp@tikz{}
1625   \gdef\bp@user@prefix{}
1626   \foreach \obj/\typ/\opt in {#1} {
1627     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{plot}=0
1628       \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{mag}=0
1629         \xdef\bp@mag@plot{\unexpanded\expandafter{\opt}}
1630       \else
1631         \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{ph}=0
1632           \xdef\bp@ph@plot{\unexpanded\expandafter{\opt}}
1633         \else
1634           \xdef\bp@mag@plot{\unexpanded\expandafter{\opt}}
1635           \xdef\bp@ph@plot{\unexpanded\expandafter{\opt}}
1636         \fi
1637       \fi
1638     \else
1639       \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{axes}=0
1640         \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{mag}=0
1641           \xdef\bp@mag@axes{\unexpanded\expandafter{\opt}}
1642         \else
1643           \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{ph}=0
1644             \xdef\bp@ph@axes{\unexpanded\expandafter{\opt}}
1645           \else
1646             \xdef\bp@mag@axes{\unexpanded\expandafter{\opt}}
1647             \xdef\bp@ph@axes{\unexpanded\expandafter{\opt}}
1648           \fi
1649         \fi
1650       \else
1651         \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{group}=0
1652           \xdef\bp@group{\unexpanded\expandafter{\opt}}
1653         \else
1654           \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{approx}=0
1655             \xdef\bp@approx{\unexpanded\expandafter{\opt}}
1656           \else
1657             \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{commands}=0
1658               \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{ph}=0
1659                 \xdef\bp@ph@commands{\unexpanded\expandafter{\opt}}
1660               \else
1661                 \xdef\bp@mag@commands{\unexpanded\expandafter{\opt}}
1662               \fi
1663             \else
1664               \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{tikz}=0
1665                 \xdef\bp@tikz{\unexpanded\expandafter{\opt}}
1666               \else
1667                 \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{prefix}=0
1668                   \xdef\bp@user@prefix{\unexpanded\expandafter{\opt}}
1669                 \else
1670                   \xdef\bp@user@prefix{}
1671                   \xdef\bp@mag@plot{\unexpanded\expandafter{\bp@mag@plot},
1672                     \unexpanded\expandafter{\obj}}
1673                   \xdef\bp@ph@plot{\unexpanded\expandafter{\bp@ph@plot},
1674                     \unexpanded\expandafter{\obj}}
1675                 \fi
1676               \fi
1677             \fi

```

```

1678         \fi
1679     \fi
1680 \fi
1681 \fi
1682 }
1683 }

```

`\bp@parse@env@opt` Parses options supplied to the Bode, Nyquist, and Nichols environments. A `for` loop over tuples of the form `\obj/\opt`, processed using nested if-else statements does the job. The input `\obj` should either be `axes` or `tikz`, and the corresponding `\opt` are passed, unexpanded, to the `axis` environment and the `tikzpicture` environment, respectively.

```

1684 \newcommand{\bp@parse@env@opt}[1]{
1685     \gdef\bp@axes{}
1686     \gdef\bp@tikz{}
1687     \gdef\bp@user@prefix{}
1688     \foreach \obj/\opt in {#1} {
1689         \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{axes}=0
1690             \xdef\bp@axes{\unexpanded\expandafter{\opt}}
1691         \else
1692             \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{tikz}=0
1693                 \xdef\bp@tikz{\unexpanded\expandafter{\opt}}
1694             \else
1695                 \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{prefix}=0
1696                     \xdef\bp@user@prefix{\unexpanded\expandafter{\opt}}
1697                 \else
1698                     \xdef\bp@user@prefix{}
1699                     \xdef\bp@axes{\unexpanded\expandafter{\bp@axes},
1700                         \unexpanded\expandafter{\obj}}
1701                 \fi
1702             \fi
1703         \fi
1704     }
1705 }

```

`\bp@parse@add@Bode@opt` Parses options for the unified `\addBodePlot` macro. Handles `linear` and `asymptotic` approximation options plus general plot styling options.

```

1706 \newcommand{\bp@parse@add@Bode@opt}[1]{
1707     \gdef\bp@plot{}
1708     \gdef\bp@approx{}
1709     \foreach \opt in {#1} {
1710         \ifx\opt\@empty\else
1711             \ifnum\pdf@strcmp{\unexpanded\expandafter{\opt}}{linear}=0
1712                 \xdef\bp@approx{\unexpanded\expandafter{\opt}}
1713             \else
1714                 \ifnum\pdf@strcmp{\unexpanded\expandafter{\opt}}{asymptotic}=0
1715                     \xdef\bp@approx{\unexpanded\expandafter{\opt}}
1716                 \else
1717                     \xdef\bp@plot{\unexpanded\expandafter{\bp@plot}%
1718                         \unexpanded\expandafter{\opt},}
1719                 \fi
1720             \fi
1721         \fi
1722     }
1723 }

```

4.6 Nyquist plots

4.6.1 User macros

`\NyquistZPK` Converts magnitude and phase parametric functions built using `\bp@ZPK@plot` into real part and imaginary part parametric functions. A plot of these is the Nyquist plot. The parametric functions are then plotted in a `tikzpicture` environment using the

`\addplot` macro. Unless the package is loaded with the option `pgf`, the parametric functions are evaluated using `gnuplot`. A large number of samples is typically needed to get a smooth plot because frequencies near 0 result in plot points that are very close to each other. Linear frequency sampling is unnecessarily fine near zero and very coarse for large ω . Logarithmic sampling makes it worse, perhaps inverse logarithmic sampling will help, pull requests to fix that are welcome!

```

1724 \NewDocumentCommand{\NyquistZPK}{ 0{} m G{} G{} }{%
1725   \pgfutil@ifempty{#3}{%
1726     \pgfkeys{/bodeplot/nyquist/.cd, reset}
1727     \pgfkeys{/bodeplot/nyquist/.cd, #1}
1728     \pgfkeysgetvalue{/bodeplot/nyquist/@axes}{\bp@axes}
1729     \pgfkeysgetvalue{/bodeplot/nyquist/@commands}{\bp@commands}
1730     \pgfkeysgetvalue{/bodeplot/nyquist/@tikz}{\bp@tikz}
1731     \pgfkeysgetvalue{/bodeplot/nyquist/@prefix}{\bp@user@prefix}
1732     \bp@zpk@new@to@legacy{#2}
1733   }{%
1734     \bp@parse@N@opt{#1}
1735     \edef\bp@legacy{#2}
1736     \edef\bp@domain@start{#3}
1737     \edef\bp@domain@end{#4}
1738   }%
1739   \gdef\bp@mag{}\gdef\bp@ph{}%
1740   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1741     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
1742     \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1743     \edef\bp@cmd{\noexpand\begin{axis}[
1744       bp@style,
1745       domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
1746       height=5cm,
1747       xlabel={\Re$},
1748       ylabel={\Im$},
1749       samples=500,
1750       \unexpanded\expandafter{\bp@axes}
1751     ]}
1752     \bp@cmd
1753     \addplot [only marks,mark=+,thick,red] (-1 , 0);
1754     \edef\bp@cmd{\noexpand\addplot
1755       [variable=t, thick, trig format plots=rad,
1756       \unexpanded\expandafter{\bp@plot}]}
1757     \if@pgfarg
1758       \bp@cmd
1759       ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
1760       {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
1761       \bp@commands
1762     \else
1763       \stepcounter{bp@gnuplot@id}
1764       \bp@cmd gnuplot [parametric, bp@gnu@prefix] {
1765         \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
1766         \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
1767       };
1768       \bp@commands
1769     \fi
1770     \end{axis}
1771   \end{tikzpicture}
1772 }

```

The following code handles active characters to avoid conflicts with ‘babel.’

```

1773 \AtBeginDocument{%
1774   \if@babel
1775     \let\Orig@NyquistZPK\NyquistZPK
1776     \renewcommand{\NyquistZPK}{%
1777       \expandafter\shorthandoff\expandafter{\bp@short@list}
1778       \NyquistZPK@Shorthandoff

```



```

1779 }
1780 \newcommand{\NyquistZPK@Shorthandoff}[4][]{%
1781   \Orig@NyquistZPK[#1]{#2}{#3}{#4}
1782   \expandafter\shorthandon\expandafter{\bp@short@list}
1783 }
1784 \fi
1785 }

```

\NyquistTF Implementation of this macro is very similar to the \NyquistZPK macro above. The only difference is a slightly different parsing of the mandatory arguments via \bp@TF@plot.

```

1786 \NewDocumentCommand{\NyquistTF}{ O{} m G{} G{} }{%
1787   \pgfutil@ifempty{#3}{%
1788     \pgfkeys{/bodeplot/nyquist/.cd, reset}
1789     \pgfkeys{/bodeplot/nyquist/.cd, #1}
1790     \pgfkeysgetvalue{/bodeplot/nyquist/@axes}{\bp@axes}
1791     \pgfkeysgetvalue{/bodeplot/nyquist/@commands}{\bp@commands}
1792     \pgfkeysgetvalue{/bodeplot/nyquist/@tikz}{\bp@tikz}
1793     \pgfkeysgetvalue{/bodeplot/nyquist/@prefix}{\bp@user@prefix}
1794     \bp@tf@new@to@legacy{#2}
1795   }{%
1796     \bp@parse@N@opt{#1}
1797     \edef\bp@legacy{#2}
1798     \edef\bp@domain@start{#3}
1799     \edef\bp@domain@end{#4}
1800   }%
1801   \gdef\bp@mag{}\gdef\bp@ph{}%
1802   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1803     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
1804     \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1805     \edef\bp@cmd{\noexpand\begin{axis}[
1806       bp@style,
1807       domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
1808       height=5cm,
1809       xlabel={\Re$},
1810       ylabel={\Im$},
1811       samples=500,
1812       \unexpanded\expandafter{\bp@axes}
1813     ]}
1814     \bp@cmd
1815     \addplot [only marks, mark=+, thick, red] (-1 , 0);
1816     \edef\bp@cmd{\noexpand\addplot
1817       [variable=t, thick, trig format plots=rad,
1818       \unexpanded\expandafter{\bp@plot}]}
1819     \ifpgfarg
1820       \bp@cmd
1821       ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
1822         {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
1823       \bp@commands
1824     \else
1825       \stepcounter{bp@gnuplot@id}
1826       \bp@cmd gnuplot [parametric, bp@gnu@prefix] {
1827         \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
1828         \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
1829       };
1830       \bp@commands
1831     \fi
1832     \end{axis}
1833   \end{tikzpicture}
1834 }

```

The following code handles active characters to avoid conflicts with ‘babel.’

```

1835 \AtBeginDocument{%
1836   \if@babel
1837     \let\Orig@NyquistTF\NyquistTF

```

```

1838 \renewcommand{\NyquistTF}{%
1839   \expandafter\shorthandoff\expandafter{\bp@short@list}
1840   \NyquistTF@Shorthandoff
1841 }
1842 \newcommand{\NyquistTF@Shorthandoff}[4][]{%
1843   \Orig@NyquistTF[#1]{#2}{#3}{#4}
1844   \expandafter\shorthandon\expandafter{\bp@short@list}
1845 }
1846 \fi
1847 }

```

\addNyquistZPKPlot Adds Nyquist plot of a transfer function in ZPK form with dual interface support. Converts magnitude and phase to real and imaginary parts for parametric plotting.

```

1848 \NewDocumentCommand{\addNyquistZPKPlot}{0}{m}{%
1849   \bp@contains@equal{#2}{\bp@zpk@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
1850   \gdef\bp@mag{}\gdef\bp@ph{}%
1851   \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1852   \if@pgfarg
1853     \edef\bp@cmd{\noexpand\addplot
1854       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
1855       variable=t, thick, trig format plots=rad, #1]}%
1856     \bp@cmd ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
1857       {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
1858   \else
1859     \stepcounter{bp@gnuplot@id}%
1860     \edef\bp@cmd{\noexpand\addplot
1861       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale, thick, #1]
1862       \bp@cmd gnuplot [parametric, bp@gnu@prefix] {%
1863         \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
1864         \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
1865       };
1866   \fi
1867 }

```

\addNyquistTFPlot Adds Nyquist plot of a transfer function in TF form with dual interface support. Converts magnitude and phase to real and imaginary parts for parametric plotting.

```

1868 \NewDocumentCommand{\addNyquistTFPlot}{0}{m}{%
1869   \bp@contains@equal{#2}{\bp@tf@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
1870   \gdef\bp@mag{}\gdef\bp@ph{}%
1871   \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1872   \if@pgfarg
1873     \edef\bp@cmd{\noexpand\addplot
1874       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
1875       variable=t, thick, trig format plots=rad, #1]}%
1876     \bp@cmd ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
1877       {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
1878   \else
1879     \stepcounter{bp@gnuplot@id}%
1880     \edef\bp@cmd{\noexpand\addplot
1881       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale, thick, #1]
1882       \bp@cmd gnuplot [parametric, bp@gnu@prefix] {%
1883         \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
1884         \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
1885       };
1886   \fi
1887 }

```

NyquistPlot An environment to host `\addNyquist...` macros that pass parametric functions to `\addplot`. Uses the defaults specified in `bp@style` to create a shortcut that includes the `tikzpicture` and `axis` environments.

```

1888 \AtBeginDocument{%
1889   \if@babel
1890     \AddToHook{env/NyquistPlot/begin}{%

```

```

1891     \expandafter\shorthandoff\expandafter{\bp@short@list}
1892   }
1893   \AddToHook{env/NyquistPlot/end}{%
1894     \expandafter\shorthandon\expandafter{\bp@short@list}
1895   }
1896 \fi
1897 }
1898 \NewDocumentEnvironment{NyquistPlot}{0}{G}{G}{+b}{%
1899   \pgfutil@ifempty{#2}{%
1900     \pgfkeys{/bodeplot/env/.cd, reset}
1901     \pgfkeys{/bodeplot/env/.cd, #1}
1902     \pgfkeysgetvalue{/bodeplot/env/@tikz}{\bp@tikz}
1903     \pgfkeysgetvalue{/bodeplot/env/@prefix}{\bp@user@prefix}
1904   }{%
1905     \bp@parse@env@opt{#1}
1906     \edef\bp@domain@start{#2}%
1907     \edef\bp@domain@end{#3}%
1908   }
1909   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1910     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
1911     \edef\bp@cmd{\noexpand\begin{axis}[
1912       bp@style,
1913       height=5cm,
1914       domain=\bp@domain@start:\bp@domain@end,
1915       xlabel={\Re$},
1916       ylabel={\Im$},
1917       \unexpanded\expandafter{\bp@axes}%
1918     ]}\bp@cmd
1919     \addplot [only marks,mark=+,thick,red] (-1 , 0);
1920     #4
1921     \end{axis}
1922     \end{tikzpicture}
1923   }{}

```

4.6.2 Internal macros

`\bp@parse@N@opt` Parses options supplied to the main Nyquist and Nichols macros. A `for` loop over tuples of the form `\obj/\opt`, processed using nested if-else statements does the job. If the input `\obj` is `plot`, `axes`, `scale`, or `tikz` then the corresponding `\opt` are passed, unexpanded, to the `\addplot` macro, the `axis` environment, the scaling option, and the `tikzpicture` environment, respectively.

```

1924 \newcommand{\bp@parse@N@opt}[1]{
1925   \gdef\bp@axes{}
1926   \gdef\bp@plot{}
1927   \gdef\bp@commands{}
1928   \gdef\bp@tikz{}
1929   \gdef\bp@user@prefix{}
1930   \gdef\bp@scale{linear}
1931   \foreach \obj/\opt in {#1} {
1932     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{axes}=0
1933       \xdef\bp@axes{\unexpanded\expandafter{\opt}}
1934     \else
1935       \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{plot}=0
1936         \xdef\bp@plot{\unexpanded\expandafter{\opt}}
1937       \else
1938         \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{commands}=0
1939           \xdef\bp@commands{\unexpanded\expandafter{\opt}}
1940         \else
1941           \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{tikz}=0
1942             \xdef\bp@tikz{\unexpanded\expandafter{\opt}}
1943           \else
1944             \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{scale}=0
1945               \xdef\bp@scale{\unexpanded\expandafter{\opt}}

```

```

1946         \else
1947         \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{prefix}=0
1948         \xdef\bp@user@prefix{\unexpanded\expandafter{\opt}}
1949         \else
1950         \xdef\bp@user@prefix{}
1951         \xdef\bp@plot{\unexpanded\expandafter{\bp@plot},
1952         \unexpanded\expandafter{\obj}}
1953         \fi
1954     \fi
1955 \fi
1956 \fi
1957 \fi
1958 \fi
1959 }
1960 }

```

4.7 Nichols charts

\NicholsZPK

```

1961 \NewDocumentCommand{\NicholsZPK}{ O{ } m G{ } G{ } }{%
1962     \pgfutil@ifempty{#3}{%
1963         \pgfkeys{/bodeplot/nichols/.cd, reset}
1964         \pgfkeys{/bodeplot/nichols/.cd, #1}
1965         \pgfkeysgetvalue{/bodeplot/nichols/@axes}{\bp@axes}
1966         \pgfkeysgetvalue{/bodeplot/nichols/@commands}{\bp@commands}
1967         \pgfkeysgetvalue{/bodeplot/nichols/@tikz}{\bp@tikz}
1968         \pgfkeysgetvalue{/bodeplot/nichols/@prefix}{\bp@user@prefix}
1969         \bp@zpk@new@to@legacy{#2}
1970     }{%
1971         \bp@parse@N@opt{#1}
1972         \edef\bp@legacy{#2}
1973         \edef\bp@domain@start{#3}
1974         \edef\bp@domain@end{#4}
1975     }%
1976     \gdef\bp@mag{}\gdef\bp@ph{}%
1977     \edef\bp@cmd{\noexpand\begin{tikzpicture}
1978     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
1979     \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1980     \edef\bp@cmd{\noexpand\begin{axis}[
1981         bp@ph@x@label,
1982         bp@style,
1983         domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
1984         height=5cm,
1985         ylabel={Gain (dB)},
1986         samples=500,
1987         \unexpanded\expandafter{\bp@axes}
1988     ]}
1989     \bp@cmd
1990     \edef\bp@cmd{\noexpand\addplot
1991     [variable=t, thick, trig format plots=rad,
1992     \unexpanded\expandafter{\bp@plot}]}
1993     \if@pgfarg
1994         \bp@cmd ( {\bp@ph} , {\bp@mag} );
1995         \bp@commands
1996     \else
1997         \stepcounter{bp@gnuplot@id}
1998         \bp@cmd gnuplot [raw gnuplot, bp@gnu@prefix]
1999         { set table $meta;
2000         set logscale x 10;
2001         set dummy t;
2002         set samples \pgfkeysvalueof{/pgfplots/samples};
2003         set trange [\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale];
2004         plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));

```

```

2005         unset logscale x;
2006         set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2007         plot "$meta" using ($2*\bp@ph@scale):($1);
2008     };
2009     \bp@commands
2010 \fi
2011 \end{axis}
2012 \end{tikzpicture}
2013 }
2014 \AtBeginDocument{%
2015 \if@babel
2016 \let\Orig@NicholsZPK\NicholsZPK
2017 \renewcommand{\NicholsZPK}{%
2018 \expandafter\shorthandoff\expandafter{\bp@short@list}
2019 \NicholsZPK@Shorthandoff
2020 }
2021 \newcommand{\NicholsZPK@Shorthandoff}[4][{}]{%
2022 \Orig@NicholsZPK[#1]{#2}{#3}{#4}
2023 \expandafter\shorthandon\expandafter{\bp@short@list}
2024 }
2025 \fi
2026 }

```

\NicholsTF

```

2027 \NewDocumentCommand{\NicholsTF}{ O{} m G{} G{} }{%
2028 \pgfutil@ifempty{#3}{%
2029 \pgfkeys{/bodeplot/nichols/.cd, reset}
2030 \pgfkeys{/bodeplot/nichols/.cd, #1}
2031 \pgfkeysgetvalue{/bodeplot/nichols/@axes}{\bp@axes}
2032 \pgfkeysgetvalue{/bodeplot/nichols/@commands}{\bp@commands}
2033 \pgfkeysgetvalue{/bodeplot/nichols/@tikz}{\bp@tikz}
2034 \pgfkeysgetvalue{/bodeplot/nichols/@prefix}{\bp@user@prefix}
2035 \bp@tf@new@to@legacy{#2}
2036 }{%
2037 \bp@parse@N@opt{#1}
2038 \edef\bp@legacy{#2}
2039 \edef\bp@domain@start{#3}
2040 \edef\bp@domain@end{#4}
2041 }%
2042 \gdef\bp@mag{} \gdef\bp@ph{}%
2043 \edef\bp@cmd{\noexpand\begin{tikzpicture}
2044 [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
2045 \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2046 \edef\bp@cmd{\noexpand\begin{axis}[
2047 bp@ph@x@label,
2048 bp@style,
2049 domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
2050 height=5cm,
2051 ylabel={Gain (dB)},
2052 samples=500,
2053 \unexpanded\expandafter{\bp@axes}
2054 ]}
2055 \bp@cmd
2056 \edef\bp@cmd{\noexpand\addplot
2057 [variable=t, thick, trig format plots=rad,
2058 \unexpanded\expandafter{\bp@plot}]}%
2059 \if@pgfarg
2060 \bp@cmd ( {\n@mod{\bp@ph}{2*pi*\bp@ph@scale}} , {\bp@mag} );
2061 \bp@commands
2062 \else
2063 \stepcounter{bp@gnuplot@id}
2064 \bp@cmd gnuplot [raw gnuplot, bp@gnu@prefix]
2065 { set table $meta;

```

```

2066         set logscale x 10;
2067         set dummy t;
2068         set samples \pgfkeysvalueof{/pgfplots/samples};
2069         set trange [\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale];
2070         plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2071         unset logscale x;
2072         set table $meta2;
2073         plot "$meta1" using ($1):($2) smooth unwrap;
2074         set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2075         plot "$meta2" using ($2*\bp@ph@scale):($1);
2076     };
2077     \bp@commands
2078 \fi
2079 \end{axis}
2080 \end{tikzpicture}
2081 }
2082 \AtBeginDocument{
2083 \if@babel
2084 \let\Orig@NicholsTF\NicholsTF
2085 \renewcommand{\NicholsTF}{%
2086 \expandafter\shorthandoff\expandafter{\bp@short@list}
2087 \NicholsTF@Shorthandoff
2088 }
2089 \newcommand{\NicholsTF@Shorthandoff}[4][[%
2090 \Orig@NicholsTF[#1]{#2}{#3}{#4}
2091 \expandafter\shorthandon\expandafter{\bp@short@list}
2092 ]
2093 \AddToHook{env/NicholsChart/begin}{%
2094 \expandafter\shorthandoff\expandafter{\bp@short@list}
2095 }
2096 \AddToHook{env/NicholsChart/end}{%
2097 \expandafter\shorthandon\expandafter{\bp@short@list}
2098 }
2099 \fi
2100 }

```

NicholsChart

```

2101 \NewDocumentEnvironment{NicholsChart}{0}{G}{G}{+b}{
2102 \pgfutil@ifempty{#2}{%
2103 \pgfkeys{/bodeplot/env/.cd, reset}
2104 \pgfkeys{/bodeplot/env/.cd, #1}
2105 \pgfkeysgetvalue{/bodeplot/env/@tikz}{\bp@tikz}
2106 \pgfkeysgetvalue{/bodeplot/env/@prefix}{\bp@user@prefix}
2107 }{%
2108 \bp@parse@env@opt{#1}
2109 \edef\bp@domain@start{#2}
2110 \edef\bp@domain@end{#3}
2111 }
2112 \edef\bp@cmd{\noexpand\begin{tikzpicture}
2113 [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
2114 \edef\bp@cmd{\noexpand\begin{axis}[
2115 bp@ph@x@label,
2116 bp@style,
2117 domain=\bp@domain@start:\bp@domain@end,
2118 height=5cm,
2119 ylabel={Gain (dB)},
2120 \unexpanded\expandafter{\bp@axes}%
2121 ]}\bp@cmd
2122 #4
2123 \end{axis}
2124 \end{tikzpicture}
2125 }{}

```

`\addNicholsZPKChart` Generates Nichols chart for ZPK system with dual interface support. Plots phase vs

magnitude.

```

2126 \NewDocumentCommand{\addNicholsZPKChart}{ 0{ } m }{%
2127   \bp@contains@equal{#2}{\bp@zpk@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
2128   \gdef\bp@mag{}\gdef\bp@ph{}%
2129   \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2130   \if@pgfarg
2131     \edef\bp@cmd{\noexpand\addplot
2132       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
2133       variable=t, thick, trig format plots=rad, #1]}%
2134     \bp@cmd ( {\bp@ph} , {\bp@mag} );
2135   \else
2136     \stepcounter{bp@gnuplot@id}%
2137     \addplot [thick, #1] gnuplot [raw gnuplot, bp@gnu@prefix] {%
2138       set table $meta;
2139       set logscale x 10;
2140       set dummy t;
2141       set samples \pgfkeysvalueof{/pgfplots/samples};
2142       set trange [\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale];
2143       plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2144       unset logscale x;
2145       set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2146       plot "$meta" using ($2*\bp@ph@scale):($1);
2147     };
2148   \fi
2149 }

```

\addNicholsTFChart Generates Nichols chart for TF system with dual interface support. Plots phase vs magnitude with phase unwrapping.

```

2150 \NewDocumentCommand{\addNicholsTFChart}{ 0{ } m }{%
2151   \bp@contains@equal{#2}{\bp@tf@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
2152   \gdef\bp@mag{}\gdef\bp@ph{}%
2153   \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2154   \if@pgfarg
2155     \edef\bp@cmd{\noexpand\addplot
2156       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
2157       variable=t, thick, trig format plots=rad, #1]}%
2158     \bp@cmd ( {\n@mod{\bp@ph}{2*pi*\bp@ph@scale}} , {\bp@mag} );
2159   \else
2160     \stepcounter{bp@gnuplot@id}%
2161     \addplot [thick, #1] gnuplot [raw gnuplot, bp@gnu@prefix] {%
2162       set table $meta1;
2163       set logscale x 10;
2164       set dummy t;
2165       set samples \pgfkeysvalueof{/pgfplots/samples};
2166       set trange [\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale];
2167       plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2168       unset logscale x;
2169       set table $meta2;
2170       plot "$meta1" using ($1):($2) smooth unwrap;
2171       set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2172       plot "$meta2" using ($2*\bp@ph@scale):($1);
2173     };
2174   \fi
2175 }

```

4.8 Pole-zero maps

4.8.1 User macros

\PoleZeroMapZPK Creates a pole-zero map similar to MATLAB's **pzmap** function. Poles are plotted as 'x' markers and zeros as 'o' markers on the complex s-plane. The gain parameter is ignored since it does not affect pole-zero locations, and delay is also ignored since it does not add poles or zeros.

```

2176 \NewDocumentCommand{\PoleZeroMapZPK}{ 0{} m }{%
2177 \bp@contains@equal{#2}{%
2178   \pgfkeys{/bodeplot/pzmap/.cd, reset}
2179   \pgfkeys{/bodeplot/pzmap/.cd, #1}
2180   \pgfkeysgetvalue{/bodeplot/pzmap/@axes}{\bp@axes}
2181   \pgfkeysgetvalue{/bodeplot/pzmap/@plot}{\bp@plot}
2182   \pgfkeysgetvalue{/bodeplot/pzmap/@commands}{\bp@commands}
2183   \pgfkeysgetvalue{/bodeplot/pzmap/@tikz}{\bp@tikz}
2184   \pgfkeysgetvalue{/bodeplot/pzmap/@prefix}{\bp@user@prefix}
2185   \pgfkeysgetvalue{/bodeplot/pzmap/@scale}{\bp@scale}
2186   \bp@zpk@new@to@legacy{#2}
2187 }{%
2188   \bp@parse@N@opt{#1}
2189   \edef\bp@legacy{#2}
2190 }%
2191 \ifnum\pdf@strcmp{\bp@scale}{log}=0
2192   \bp@min@real@ZPK{\bp@legacy}
2193   \bp@min@im@ZPK{\bp@legacy}
2194   \pgfkeys{/pgf/fpu=true}
2195   \ifx\bp@has@positive@values\bp@min@false
2196     \xdef\bp@PZZPK@ticksXPos{0}
2197   \else
2198     \pgfmathparse{max(\bp@max@re@pos@pow@10 - \bp@min@re@pow@10 + 1, 1)}
2199     \pgfmathfloattoint{\pgfmathresult}
2200     \xdef\bp@PZZPK@ticksXPos{\pgfmathresult}
2201   \fi
2202   \ifx\bp@has@negative@values\bp@min@false
2203     \xdef\bp@PZZPK@ticksXNeg{0}
2204   \else
2205     \pgfmathparse{max(\bp@max@re@neg@pow@10 - \bp@min@re@pow@10 + 1, 1)}
2206     \pgfmathfloattoint{\pgfmathresult}
2207     \xdef\bp@PZZPK@ticksXNeg{\pgfmathresult}
2208   \fi
2209   \pgfkeys{/pgf/fpu=false}
2210   \def\PoleZeroMapZPK@formatXTick##1{%
2211     \pgfmathtruncatemacro{\PoleZeroMapZPK@tick}{##1}%
2212     \ifnum\PoleZeroMapZPK@tick=0
2213       $0$
2214     \else
2215       \pgfmathtruncatemacro{\PoleZeroMapZPK@exp}
2216       {\bp@min@re@pow@10 + abs(\PoleZeroMapZPK@tick) - 1}%
2217       \ifnum\PoleZeroMapZPK@tick>0
2218         $10^{\PoleZeroMapZPK@exp}$%
2219       \else
2220         $-10^{\PoleZeroMapZPK@exp}$%
2221       \fi
2222     \fi
2223   }
2224   \def\PoleZeroMapZPK@formatYTick##1{%
2225     \pgfmathtruncatemacro{\PoleZeroMapZPK@tick}{##1}%
2226     \ifnum\PoleZeroMapZPK@tick=0
2227       $0$
2228     \else
2229       \pgfmathtruncatemacro{\PoleZeroMapZPK@exp}
2230       {\bp@min@im@pow@10 + abs(\PoleZeroMapZPK@tick) - 1}%
2231       \ifnum\PoleZeroMapZPK@tick>0
2232         $10^{\PoleZeroMapZPK@exp}$%
2233       \else
2234         $-10^{\PoleZeroMapZPK@exp}$%
2235       \fi
2236     \fi
2237   }
2238   \def\PoleZeroMapZPK@xticklabel{\PoleZeroMapZPK@formatXTick{\tick}}

```



```

2239 \def\PoleZeroMapZPK@yticklabel{\PoleZeroMapZPK@formatYTick{\tick}}
2240 \fi
2241 \edef\bp@cmd{\noexpand\begin{tikzpicture}
2242 [\unexpanded\expandafter{\bp@tikz}]}
2243 \bp@cmd
2244 \ifnum\pdf@strcmp{\bp@scale}{log}=0
2245 \edef\bp@cmd{\noexpand\begin{axis}[
2246 xlabel={\Re$},
2247 ylabel={\Im$},
2248 axis lines=center,
2249 grid=major,
2250 height=6cm,
2251 enlarge x limits=0.2,
2252 enlarge y limits=0.2,
2253 xtick distance=1,
2254 ytick distance=1,
2255 xticklabel=\noexpand\PoleZeroMapZPK@xticklabel,
2256 yticklabel=\noexpand\PoleZeroMapZPK@yticklabel,
2257 x filter/.expression={abs(x) < \bp@min@re@threshold@result ?
2258 0 : (x >= 0 ? (log10(max(min(x, 1e100), 1e-
100)) - \bp@min@re@pow@10 + 1) :
2259 (-log10(max(min(-x, 1e100), 1e-100)) + \bp@min@re@pow@10 - 1))},
2260 y filter/.expression={abs(y) < \bp@min@im@threshold@result ?
2261 0 : (y >= 0 ? (log10(max(min(y, 1e100), 1e-
100)) - \bp@min@im@pow@10 + 1) :
2262 (-log10(max(min(-y, 1e100), 1e-100)) + \bp@min@im@pow@10 - 1))},
2263 \unexpanded\expandafter{\bp@axes}
2264 ]}
2265 \else
2266 \edef\bp@cmd{\noexpand\begin{axis}[
2267 xlabel={\Re$},
2268 ylabel={\Im$},
2269 axis lines=center,
2270 grid=major,
2271 height=6cm,
2272 enlarge x limits=0.2,
2273 enlarge y limits=0.2,
2274 \unexpanded\expandafter{\bp@axes}
2275 ]}
2276 \fi
2277 \bp@cmd
2278 \foreach \feature/\values in \bp@legacy {
2279 \ifnum\pdf@strcmp{\feature}{z}=0
2280 \foreach \z in \values {
2281 \foreach \y [count=\zcnt] in \z {
2282 \ifnum\zcnt=1
2283 \xdef\bp@zre{\y}
2284 \fi
2285 \ifnum\zcnt=2
2286 \xdef\bp@zim{\y}
2287 \fi
2288 \xdef\bp@Last@Loop@z{\zcnt}
2289 }
2290 \ifnum\bp@Last@Loop@z=1
2291 \xdef\bp@zim{0}
2292 \fi
2293 \edef\bp@cmd{\noexpand\addplot
2294 [only marks, mark=o, mark size=3pt, thick, blue,
2295 \unexpanded\expandafter{\bp@plot}]}
2296 \bp@cmd coordinates {(\bp@zre,\bp@zim)};
2297 }
2298 \fi
2299 \ifnum\pdf@strcmp{\feature}{p}=0

```

```

2300      \foreach \p in \values {
2301        \foreach \y [count=\pcnt] in \p {
2302          \ifnum\pcnt=1
2303            \xdef\bp@pre{\y}
2304          \fi
2305          \ifnum\pcnt=2
2306            \xdef\bp@pim{\y}
2307          \fi
2308          \xdef\bp@Last@Loop@p{\pcnt}
2309        }
2310      \ifnum\bp@Last@Loop@p=1
2311        \xdef\bp@pim{0}
2312      \fi
2313      \edef\bp@cmd{\noexpand\addplot
2314        [only marks, mark=x, mark size=3pt, thick, red,
2315        \unexpanded\expandafter{\bp@plot}]}
2316      \bp@cmd coordinates {(\bp@pre,\bp@pim)};
2317    }
2318  \fi
2319 }
2320 \bp@commands
2321 \end{axis}
2322 \end{tikzpicture}
2323 }

```

The following code handles active characters to avoid conflicts with ‘babel.’

```

2324 \AtBeginDocument{%
2325   \if@babel
2326     \let\Orig@PoleZeroMapZPK\PoleZeroMapZPK
2327     \renewcommand{\PoleZeroMapZPK}{%
2328       \expandafter\shorthandoff\expandafter{\bp@short@list}
2329       \PoleZeroMapZPK@Shorthandoff
2330     }
2331     \newcommand{\PoleZeroMapZPK@Shorthandoff}[2][1]{%
2332       \Orig@PoleZeroMapZPK[#1][#2]
2333       \expandafter\shandon\expandafter{\bp@short@list}
2334     }
2335   \fi
2336 }

```

4.8.2 Internal macros

`\bp@min@real@ZPK` Computes the minimum nonzero absolute value of real parts from all poles and zeros in ZPK format. This is used for automatically setting the threshold in logarithmic pole-zero maps. The result is stored in `\bp@min@re@threshold@result` and the corresponding power of 10 is stored in `\bp@min@re@pow@10`.

```

2337 \newcommand{\bp@min@real@ZPK}[1]{
2338   \gdef\bp@min@re@threshold@result{1000}
2339   \def\@bp@min@false{false}
2340   \gdef\bp@min@threshold@found{false}
2341   \global\let\bp@min@thresh@float\relax
2342   \pgfkeys{/pgf/fpu=true}
2343   \pgfmathparse{0}
2344   \global\let\bp@max@re@float=\pgfmathresult
2345   \pgfmathparse{0}
2346   \global\let\bp@max@re@pos@float=\pgfmathresult
2347   \pgfmathparse{0}
2348   \global\let\bp@max@re@neg@float=\pgfmathresult
2349   \pgfkeys{/pgf/fpu=false}
2350   \gdef\bp@max@re@value{0}
2351   \gdef\bp@has@positive@values{false}
2352   \gdef\bp@has@negative@values{false}
2353   \foreach \feature/\values in {#1} {

```

```

2354 \ifnum\pdf@strcmp{\feature}{z}=0
2355 \foreach \z in \values {
2356 \foreach \y [count=\zcnt] in \z {
2357 \ifnum\zcnt=1
2358 \pgfkeys{/pgf/fpu=true}
2359 \pgfmathparse{abs(\y)}
2360 \let\abs@valuefloat=\pgfmathresult
2361 \pgfmathfloattofixed{\abs@valuefloat}
2362 \edef\abs@value{\pgfmathresult}
2363 \pgfkeys{/pgf/fpu=false}
2364 \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2365 \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2366 \pgfkeys{/pgf/fpu=true}
2367 \pgfmathparse{\y >= 0 ? 1 : 0}
2368 \pgfmathfloattoint{\pgfmathresult}
2369 \pgfkeys{/pgf/fpu=false}
2370 \ifnum\pgfmathresult=1
2371 \gdef\bp@has@positive@values{true}
2372 \pgfkeys{/pgf/fpu=true}
2373 \pgfmathparse{\abs@valuefloat > \bp@max@re@pos@float ? 1 : 0}
2374 \pgfmathfloattoint{\pgfmathresult}
2375 \pgfkeys{/pgf/fpu=false}
2376 \ifnum\pgfmathresult=1
2377 \global\let\bp@max@re@pos@float=\abs@valuefloat
2378 \fi
2379 \else
2380 \gdef\bp@has@negative@values{true}
2381 \pgfkeys{/pgf/fpu=true}
2382 \pgfmathparse{\abs@valuefloat > \bp@max@re@neg@float ? 1 : 0}
2383 \pgfmathfloattoint{\pgfmathresult}
2384 \pgfkeys{/pgf/fpu=false}
2385 \ifnum\pgfmathresult=1
2386 \global\let\bp@max@re@neg@float=\abs@valuefloat
2387 \fi
2388 \fi
2389 \pgfkeys{/pgf/fpu=true}
2390 \pgfmathparse{\abs@valuefloat > \bp@max@re@float ? 1 : 0}
2391 \pgfmathfloattoint{\pgfmathresult}
2392 \pgfkeys{/pgf/fpu=false}
2393 \ifnum\pgfmathresult=1
2394 \global\let\bp@max@re@float=\abs@valuefloat
2395 \xdef\bp@max@re@value{\abs@value}
2396 \fi
2397 \ifx\bp@min@threshold@found\@bp@min@false
2398 \xdef\bp@min@re@threshold@result{\abs@value}
2399 \global\let\bp@min@thresh@float=\abs@valuefloat
2400 \gdef\bp@min@threshold@found{true}
2401 \else
2402 \pgfkeys{/pgf/fpu=true}
2403 \pgfmathparse{\abs@valuefloat < \bp@min@thresh@float ? 1 : 0}
2404 \pgfmathfloattoint{\pgfmathresult}
2405 \pgfkeys{/pgf/fpu=false}
2406 \ifnum\pgfmathresult=1
2407 \xdef\bp@min@re@threshold@result{\abs@value}
2408 \global\let\bp@min@thresh@float=\abs@valuefloat
2409 \fi
2410 \fi
2411 \fi
2412 \fi
2413 \fi
2414 }
2415 }
2416 \fi

```

```

2417 \ifnum\pdf@strcmp{\feature}{p}=0
2418 \foreach \p in \values {
2419 \foreach \y [count=\pcnt] in \p {
2420 \ifnum\pcnt=1
2421 \pgfkeys{/pgf/fpu=true}
2422 \pgfmathparse{abs(\y)}
2423 \let\abs@valuefloat=\pgfmathresult
2424 \pgfmathfloattofixed{\abs@valuefloat}
2425 \edef\abs@value{\pgfmathresult}
2426 \pgfkeys{/pgf/fpu=false}
2427 \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2428 \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2429 \pgfkeys{/pgf/fpu=true}
2430 \pgfmathparse{\y >= 0 ? 1 : 0}
2431 \pgfmathfloattoint{\pgfmathresult}
2432 \pgfkeys{/pgf/fpu=false}
2433 \ifnum\pgfmathresult=1
2434 \gdef\bp@has@positive@values{true}
2435 \pgfkeys{/pgf/fpu=true}
2436 \pgfmathparse{\abs@valuefloat > \bp@max@re@pos@float ? 1 : 0}
2437 \pgfmathfloattoint{\pgfmathresult}
2438 \pgfkeys{/pgf/fpu=false}
2439 \ifnum\pgfmathresult=1
2440 \global\let\bp@max@re@pos@float=\abs@valuefloat
2441 \fi
2442 \else
2443 \gdef\bp@has@negative@values{true}
2444 \pgfkeys{/pgf/fpu=true}
2445 \pgfmathparse{\abs@valuefloat > \bp@max@re@neg@float ? 1 : 0}
2446 \pgfmathfloattoint{\pgfmathresult}
2447 \pgfkeys{/pgf/fpu=false}
2448 \ifnum\pgfmathresult=1
2449 \global\let\bp@max@re@neg@float=\abs@valuefloat
2450 \fi
2451 \fi
2452 \ifx\bp@min@threshold@found\@bp@min@false
2453 \xdef\bp@min@re@threshold@result{\abs@value}
2454 \global\let\bp@min@thresh@float=\abs@valuefloat
2455 \gdef\bp@min@threshold@found{true}
2456 \else
2457 \pgfkeys{/pgf/fpu=true}
2458 \pgfmathparse{\abs@valuefloat < \bp@min@thresh@float ? 1 : 0}
2459 \pgfmathfloattoint{\pgfmathresult}
2460 \pgfkeys{/pgf/fpu=false}
2461 \ifnum\pgfmathresult=1
2462 \xdef\bp@min@re@threshold@result{\abs@value}
2463 \global\let\bp@min@thresh@float=\abs@valuefloat
2464 \fi
2465 \fi
2466 \fi
2467 \fi
2468 \fi
2469 }
2470 }
2471 \fi
2472 }
2473 \ifx\bp@min@threshold@found\@bp@min@false
2474 \gdef\bp@min@re@threshold@result{0.01}
2475 \fi
2476 \xdef\bp@min@threshold@result{\bp@min@re@threshold@result}
2477 \pgfkeys{/pgf/fpu=true}
2478 \pgfmathparse{log10(\bp@min@re@threshold@result)}
2479 \let\log@result=\pgfmathresult

```

```

2480 \pgfmathparse{\log@result + 1e-5}
2481 \let\log@adjusted=\pgfmathresult
2482 \pgfmathparse{floor(\log@adjusted)}
2483 \pgfmathfloattofixed{\pgfmathresult}
2484 \xdef\bp@min@re@pow@10{\pgfmathresult}
2485 \xdef\bp@min@pow@10{\bp@min@re@pow@10}
2486 \ifx\bp@has@positive@values\bp@min@false
2487 \xdef\bp@max@re@pos@pow@10{\bp@min@re@pow@10}
2488 \else
2489 \pgfmathparse{log10(max(\bp@max@re@pos@float,1e-100))}
2490 \let\log@bp@max@re@pos=\pgfmathresult
2491 \pgfmathparse{\log@bp@max@re@pos + 1e-5}
2492 \let\log@bp@max@re@pos@adjusted=\pgfmathresult
2493 \pgfmathparse{ceil(\log@bp@max@re@pos@adjusted)}
2494 \pgfmathfloattoint{\pgfmathresult}
2495 \xdef\bp@max@re@pos@pow@10{\pgfmathresult}
2496 \fi
2497 \ifx\bp@has@negative@values\bp@min@false
2498 \xdef\bp@max@re@neg@pow@10{\bp@min@re@pow@10}
2499 \else
2500 \pgfmathparse{log10(max(\bp@max@re@neg@float,1e-100))}
2501 \let\log@bp@max@re@neg=\pgfmathresult
2502 \pgfmathparse{\log@bp@max@re@neg + 1e-5}
2503 \let\log@bp@max@re@neg@adjusted=\pgfmathresult
2504 \pgfmathparse{ceil(\log@bp@max@re@neg@adjusted)}
2505 \pgfmathfloattoint{\pgfmathresult}
2506 \xdef\bp@max@re@neg@pow@10{\pgfmathresult}
2507 \fi
2508 \pgfmathparse{max(\bp@max@re@pos@float > 0 ?
2509 \bp@max@re@pos@float : 0, \bp@max@re@neg@float > 0 ? \bp@max@re@neg@float : 0)}
2510 \let\bp@max@re@valuefloat=\pgfmathresult
2511 \pgfmathparse{\bp@max@re@valuefloat > 0 ?
2512 \bp@max@re@valuefloat : \bp@min@re@threshold@result}
2513 \let\bp@max@re@valuefloat=\pgfmathresult
2514 \pgfmathparse{log10(max(\bp@max@re@valuefloat,1e-100))}
2515 \let\log@bp@max@re=\pgfmathresult
2516 \pgfmathparse{\log@bp@max@re + 1e-5}
2517 \let\log@bp@max@re@adjusted=\pgfmathresult
2518 \pgfmathparse{ceil(\log@bp@max@re@adjusted)}
2519 \pgfmathfloattoint{\pgfmathresult}
2520 \xdef\bp@max@re@pow@10{\pgfmathresult}
2521 \pgfkeys{/pgf/fpu=false}
2522 }

```

`\bp@min@im@ZPK` Computes the minimum nonzero absolute value of imaginary parts from all poles and zeros in ZPK format. This is used for automatically setting thresholds in logarithmic pole-zero maps for imaginary axis scaling. The result is stored in `\bp@min@im@threshold@result` and the corresponding power of 10 is stored in `\bp@min@im@pow@10`.

```

2523 \newcommand{\bp@min@im@ZPK}[1]{
2524 \gdef\bp@min@im@threshold@result{1000}
2525 \def\@bp@min@false{false}
2526 \gdef\bp@min@im@threshold@found{false}
2527 \global\let\bp@min@im@thresh@float\relax
2528 \pgfkeys{/pgf/fpu=true}
2529 \pgfmathparse{0}
2530 \global\let\bp@max@im@float=\pgfmathresult
2531 \pgfkeys{/pgf/fpu=false}
2532 \gdef\bp@max@im@value{0}
2533 \foreach \feature/\values in {#1} {
2534 \ifnum\pdf@strcmp{\feature}{z}=0
2535 \foreach \z in \values {
2536 \foreach \y [count=\zcnt] in \z {

```

```

2537 \ifnum\zcnt=2
2538 \pgfkeys{/pgf/fpu=true}
2539 \pgfmathparse{abs(\y)}
2540 \let\abs@valuefloat=\pgfmathresult
2541 \pgfmathfloattofixed{\abs@valuefloat}
2542 \edef\abs@value{\pgfmathresult}
2543 \pgfkeys{/pgf/fpu=false}
2544 \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2545 \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2546 \pgfkeys{/pgf/fpu=true}
2547 \pgfmathparse{\abs@valuefloat > \bp@max@im@float ? 1 : 0}
2548 \pgfmathfloattoint{\pgfmathresult}
2549 \pgfkeys{/pgf/fpu=false}
2550 \ifnum\pgfmathresult=1
2551 \global\let\bp@max@im@float=\abs@valuefloat
2552 \xdef\bp@max@im@value{\abs@value}
2553 \fi
2554 \ifx\bp@min@im@threshold@found\@bp@min@false
2555 \xdef\bp@min@im@threshold@result{\abs@value}
2556 \global\let\bp@min@im@thresh@float=\abs@valuefloat
2557 \gdef\bp@min@im@threshold@found{true}
2558 \else
2559 \pgfkeys{/pgf/fpu=true}
2560 \pgfmathparse{\abs@valuefloat < \bp@min@im@thresh@float ?
2561 1 : 0}
2562 \pgfmathfloattoint{\pgfmathresult}
2563 \pgfkeys{/pgf/fpu=false}
2564 \ifnum\pgfmathresult=1
2565 \xdef\bp@min@im@threshold@result{\abs@value}
2566 \global\let\bp@min@im@thresh@float=\abs@valuefloat
2567 \fi
2568 \fi
2569 \fi
2570 \fi
2571 \fi
2572 }
2573 }
2574 \fi
2575 \ifnum\pdf@strcmp{\feature}{p}=0
2576 \foreach \p in \values {
2577 \foreach \y [count=\pcnt] in \p {
2578 \ifnum\pcnt=2
2579 \pgfkeys{/pgf/fpu=true}
2580 \pgfmathparse{abs(\y)}
2581 \let\abs@valuefloat=\pgfmathresult
2582 \pgfmathfloattofixed{\abs@valuefloat}
2583 \edef\abs@value{\pgfmathresult}
2584 \pgfkeys{/pgf/fpu=false}
2585 \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2586 \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2587 \ifx\bp@min@im@threshold@found\@bp@min@false
2588 \xdef\bp@min@im@threshold@result{\abs@value}
2589 \global\let\bp@min@im@thresh@float=\abs@valuefloat
2590 \gdef\bp@min@im@threshold@found{true}
2591 \else
2592 \pgfkeys{/pgf/fpu=true}
2593 \pgfmathparse{\abs@valuefloat < \bp@min@im@thresh@float ?
2594 1 : 0}
2595 \pgfmathfloattoint{\pgfmathresult}
2596 \pgfkeys{/pgf/fpu=false}
2597 \ifnum\pgfmathresult=1
2598 \xdef\bp@min@im@threshold@result{\abs@value}
2599 \global\let\bp@min@im@thresh@float=\abs@valuefloat

```

```

2600             \fi
2601         \fi
2602     \fi
2603 \fi
2604 \fi
2605 }
2606 }
2607 \fi
2608 }
2609 \ifx\bp@min@im@threshold@found\@bp@min@false
2610     \gdef\bp@min@im@threshold@result{0.01}
2611 \fi
2612 \pgfkeys{/pgf/fpu=true}
2613 \pgfmathparse{log10(\bp@min@im@threshold@result)}
2614 \let\log@result=\pgfmathresult
2615 \pgfmathparse{\log@result + 1e-5}
2616 \let\log@adjusted=\pgfmathresult
2617 \pgfmathparse{floor(\log@adjusted)}
2618 \pgfmathfloattofixed{\pgfmathresult}
2619 \xdef\bp@min@im@pow@10{\pgfmathresult}
2620 \xdef\bp@min@Im@pow@10{\bp@min@im@pow@10}
2621 \pgfmathparse{\bp@max@im@float > 0 ?
2622 \bp@max@im@float : \bp@min@im@threshold@result}
2623 \let\bp@max@im@valuefloat=\pgfmathresult
2624 \pgfmathparse{log10(max(\bp@max@im@valuefloat,1e-10))}
2625 \let\log@bp@max@im=\pgfmathresult
2626 \pgfmathparse{\log@bp@max@im + 1e-5}
2627 \let\log@bp@max@im@adjusted=\pgfmathresult
2628 \pgfmathfloattoint{\pgfmathresult}
2629 \pgfmathfloattoint{\pgfmathresult}
2630 \xdef\bp@max@im@pow@10{\pgfmathresult}
2631 \pgfkeys{/pgf/fpu=false}
2632 }

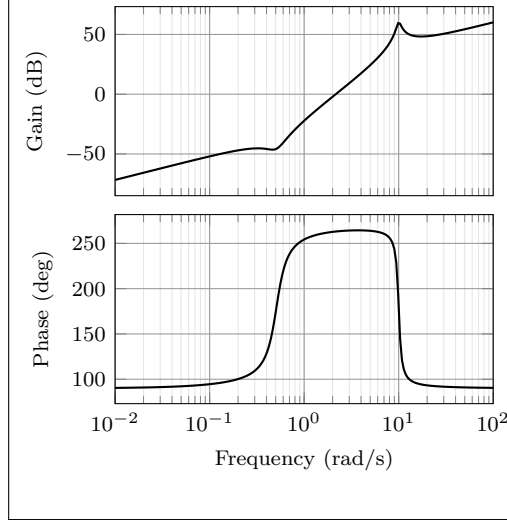
```

A TL;DR for package version 2.1.1 and earlier

All Bode plots in this section are for the transfer function (with and without a transport delay)

$$G(s) = 10 \frac{s(s + 0.1 + 0.5i)(s + 0.1 - 0.5i)}{(s + 0.5 + 10i)(s + 0.5 - 10i)} = \frac{s(10s^2 + 2s + 2.6)}{(s^2 + s + 100.25)}. \quad (6)$$

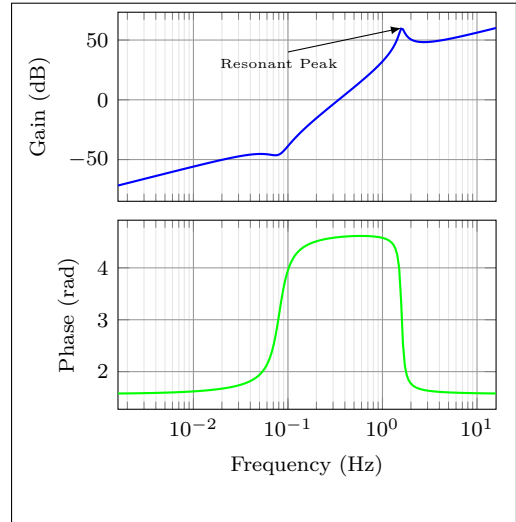
Bode plot in ZPK format



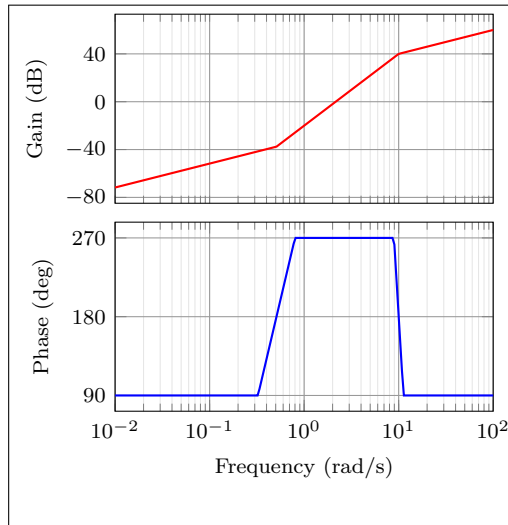
```
\BodeZPK{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
{0.01}
{100}
```

Same Bode plot over the same frequency range but supplied in Hz, in TF format with arrow decoration, transport delay, unit, and color customization (the phase plot may show wrapping if the **pgf** package option is used)

```
\BodeTF[%
  samples=1000,
  plot/mag/{blue,thick},
  plot/ph/{green,thick},
  tikz/{%
    >=latex,
    phase unit=rad,
    frequency unit=Hz%
  },
  commands/mag/{
    \draw[->](axis cs:0.1,40) -- (axis cs:{10/(2*pi)},60);
    \node at (axis cs: 0.08,30) {\tiny Resonant Peak};
  }%
]
{%
  num/{10,2,2.6,0},
  den/{1,1,100.25}%
}
{0.01/(2*pi)}
{100/(2*pi)}
```



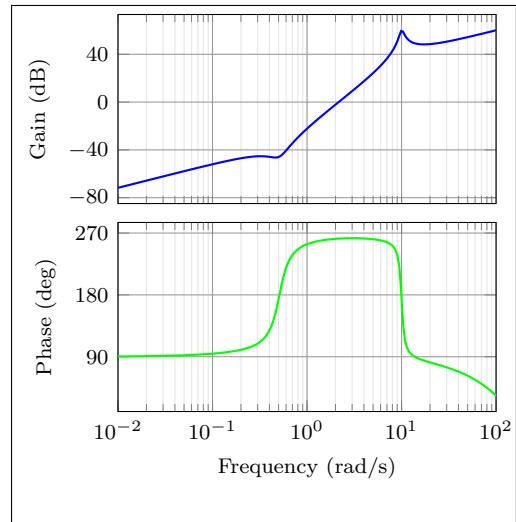
Linear approximation with customization



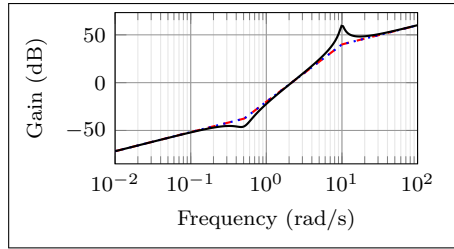
```
\BodeZPK[%
plot/mag/{red,thick},
plot/ph/{blue,thick},
axes/mag/{ytick distance=40},
axes/ph/{ytick distance=90},
approx/linear%
]{%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/10%
}
{0.01}
{100}
```

Plot with delay and customization

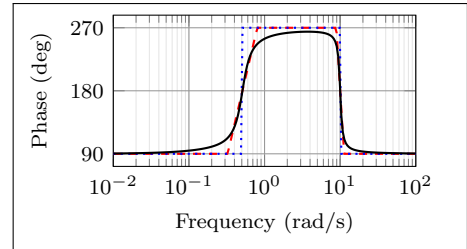
```
\BodeZPK[%
plot/mag/{blue,thick},
plot/ph/{green,thick},
axes/mag/{ytick distance=40},
axes/ph/{ytick distance=90%
}{%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/10,
d/0.01%
}
{0.01}
{100}
```



Individual gain and phase plots with more customization



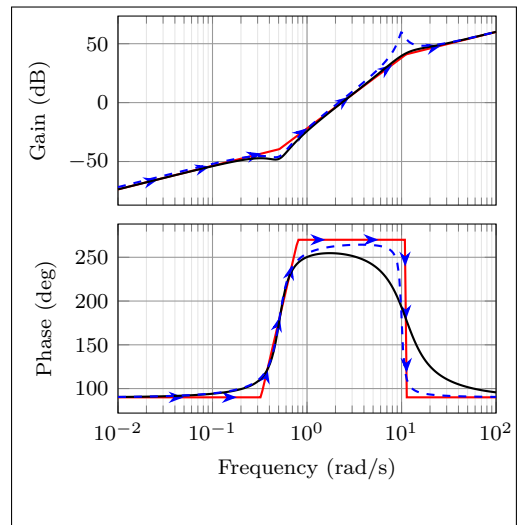
```
\begin{BodeMagPlot}{%
  axes/{height=2cm,
    width=4cm}%
}
{0.01}
{100}
\addBodeZPKPlots[%
  true/{black,thick},
  linear/{red,dashed,thick},
  asymptotic/{blue,dotted,thick}%
]
{magnitude}
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
}\end{BodeMagPlot}
```



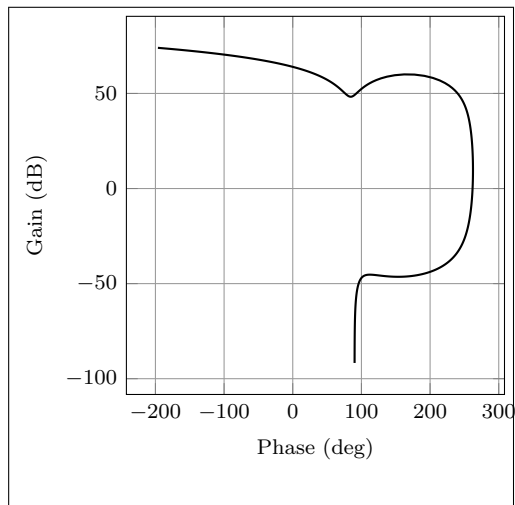
```
\begin{BodePhPlot}{%
  height=2cm,
  width=4cm,
  ytick distance=90
}
{0.01}
{100}
\addBodeZPKPlots[%
  true/{black,thick},
  linear/{red,dashed,thick},
  asymptotic/{blue,dotted,thick}%
]
{phase}
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
}\end{BodePhPlot}
```

Multiple transfer functions in a single Bode plot using the **BodePlot** environment and the **\addBodePlot** macro introduced in v2.1.

```
\begin{BodePlot}{0.01}{100}
\addBodePlot[red,postaction=decorate,
  decoration={%
    markings,
    mark=between positions 0.1 and 0.9 step 2em with {%
      \arrow{Stealth [length=2mm, blue]}
    }
  },linear]{zpk}{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-5,-10},{-5,10}},
  k/10%
}
\addBodePlot[black,thick]{zpk}{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-5,-10},{-5,10}},
  k/10%
}
\addBodePlot[blue,dashed]{tf}{%
  num/{10,2,2.6,0},
  den/{1,1,100.25}%
}
}\end{BodePlot}
```



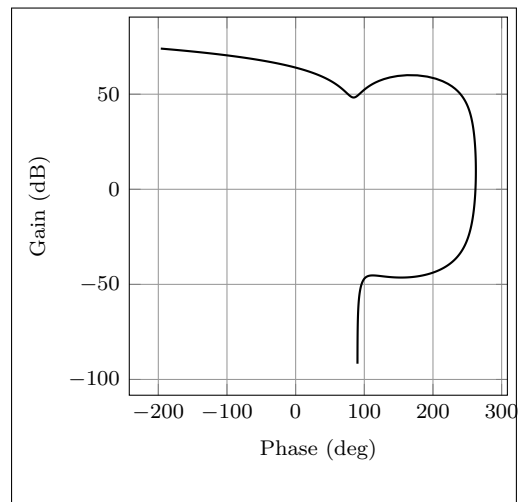
Nichols chart



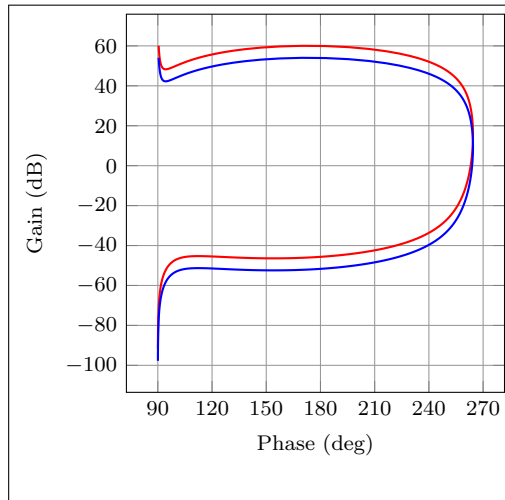
```
\NicholsZPK[samples=1000]
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10,
  d/0.01%
}
{0.001}
{500}
```

Same Nichols chart in TF format (may show wrapping in **pgf** mode)

```
\NicholsTF[samples=1000]
{%
  num/{10,2,2.6,0},
  den/{1,1,100.25},
  d/0.01%
}
{0.001}
{500}
```



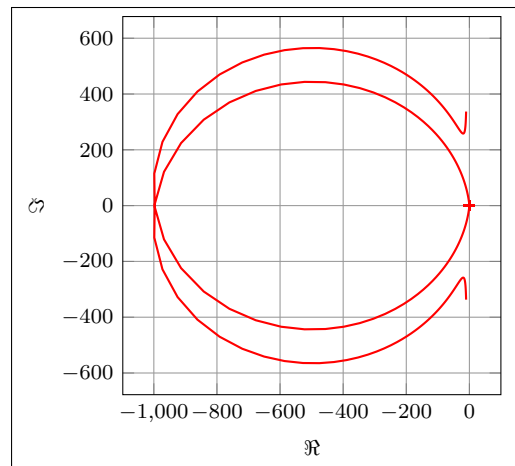
Multiple Nichols charts with customization



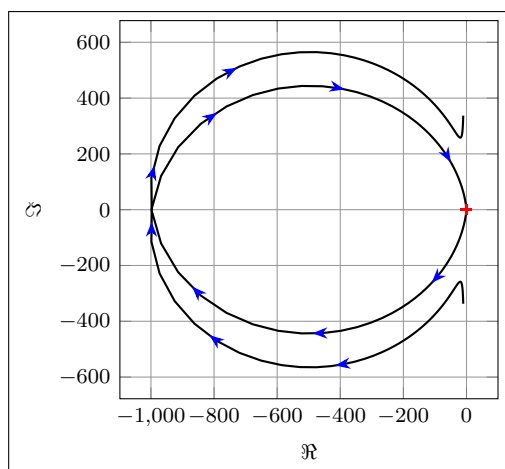
```
\begin{NicholsChart}{%
  ytick distance=20,
  xtick distance=30
}
{0.001}
{100}
\addNicholsZPKChart [red,samples=1000] {%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
\addNicholsZPKChart [blue,samples=1000] {%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/5%
}
\end{NicholsChart}
```

Nyquist plot

```
\NyquistZPK[plot/{red,thick,samples=1000}]
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
{-30}
{30}
```



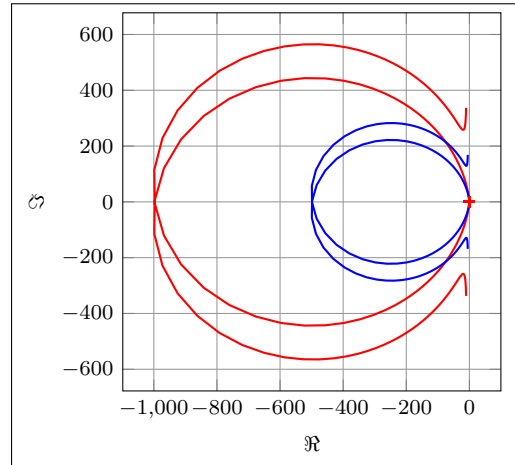
Nyquist plot in TF format with arrows



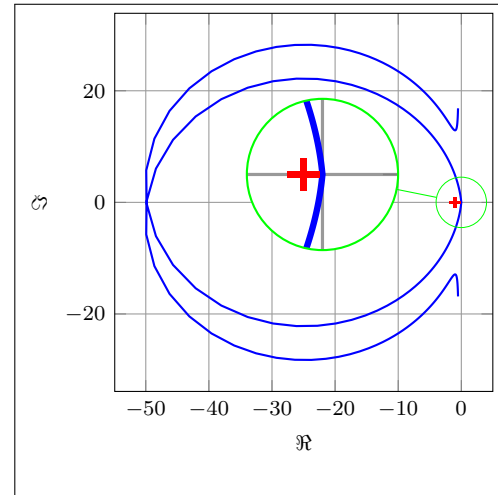
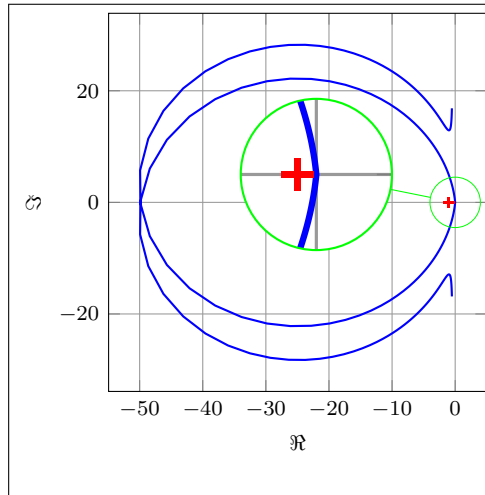
```
\NyquistTF[%
  plot/{%
    samples=1000,
    postaction=decorate,
    decoration={%
      markings,
      mark=between positions 0.1 and 0.9 step 5em with {%
        \arrow{Stealth [length=2mm, blue]}
      }
    }
  }%
]
{%
  num/{10,2,2.6,0},
  den/{1,1,100.25}%
}
{-30}
{30}
```

Multiple Nyquist plots with customization

```
\begin{NyquistPlot}{-30}{30}
\addNyquistZPKPlot [red,samples=1000] {%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/10%
}
\addNyquistZPKPlot [blue,samples=1000] {%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/5%
}
\end{NyquistPlot}
```



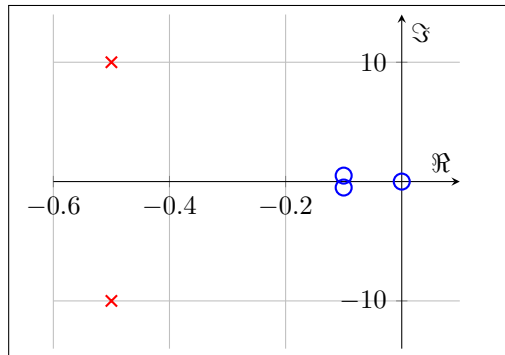
Nyquist plots with additional commands, using two different macros



```
\begin{NyquistPlot}{%
tikz/{
spy using outlines={%
circle,
magnification=3,
connect spies,
size=2cm
}
}%
}{-30}{30}
\addNyquistZPKPlot [blue,samples=1000] {%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/0.5%
}
\coordinate (spyon) at (axis cs:0,0);
\coordinate (spyat) at (axis cs:-22,5);
\spy [green] on (spyon) in
node [fill=white] at (spyat);
\end{NyquistPlot}
```

```
\NyquistZPK[%
plot/{blue,samples=1000},
tikz/{
spy using outlines={%
circle,
magnification=3,
connect spies,
size=2cm
}
},
commands/{
\coordinate (spyon) at (axis cs:0,0);
\coordinate (spyat) at (axis cs:-22,5);
\spy [green] on (spyon) in
node [fill=white] at (spyat);
}%
}
]{%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/0.5%
}
}{-30}
{30}
```

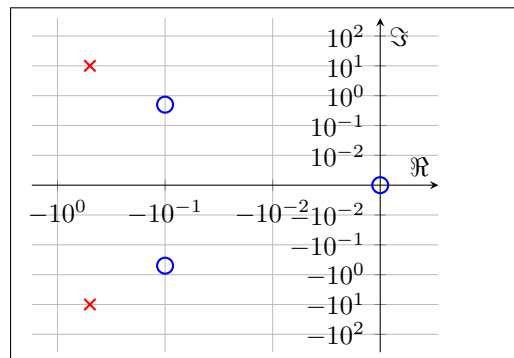
Pole-zero map



```
\PoleZeroMapZPK
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
```

Pole-zero map (symmetric log scale)

```
\PoleZeroMapZPK[scale/{log}]
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
```



Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in **roman** refer to the code lines where the entry is used.

79

152, 170, 242, 315,	\bp@Last@Loop@z 2288, 2290	2348, 2382, 2386,
393, 455, 645, 661,	\bp@legacy	2445, 2449, 2500, 2509
698, 714, 891, 903,	. . . 890, 900, 978,	\bp@max@re@neg@pow@ .
905, 979, 990, 992,	987, 1048, 1055,	. . . 2205, 2498, 2506
1147, 1157, 1186,	1057, 1079, 1084,	\bp@max@re@pos@float
1195, 1278, 1288,	1086, 1216, 1220, 2346,
1290, 1736, 1745,	1516, 1536, 1735,	2373, 2377, 2436,
1798, 1807, 1906,	1742, 1797, 1804,	2440, 2489, 2508, 2509
1973, 1983, 2003,	1849, 1851, 1869,	\bp@max@re@pos@pow@ .
2039, 2049, 2069, 2109	1871, 1972, 1979,	. . . 2198, 2487, 2495
\bp@domain@start: . . .	2038, 2045, 2127,	\bp@max@re@pow@ . . . 2520
. 1159,	2129, 2151, 2153,	\bp@max@re@value . . .
1197, 1914, 2117	2189, 2192, 2193, 2278 2350, 2395
\bp@fix@add@opt	\bp@list 1408,	\bp@max@re@valuefloat
. . . 1045, 1047, 1366	1409, 1544, 1545 2510,
\bp@freq@filter 83	\bp@loop@delay	2511, 2512, 2513, 2514
\bp@freq@label 83	. . . 1543, 1594, 1599	\bp@min@Im@pow@ . . . 2620
\bp@freq@scale	\bp@mag 898, 900,	\bp@min@im@pow@ 2230,
83, 191, 198, 336,	921, 928, 985, 987,	2261, 2262, 2619, 2620
343, 476, 483, 905,	1008, 1015, 1208,	\bp@min@im@thresh@float
992, 1061, 1067,	1216, 1220, 1227,	2527, 2556, 2560,
1091, 1097, 1107,	1231, 1242, 1250,	2566, 2589, 2593, 2599
1110, 1118, 1125,	1739, 1742, 1759,	\bp@min@im@threshold@found
1240, 1244, 1290,	1760, 1765, 1766, 2526, 2554,
1492, 1612, 1745,	1801, 1804, 1821,	2557, 2587, 2590, 2609
1807, 1854, 1861,	1822, 1827, 1828,	\bp@min@im@threshold@result
1874, 1881, 1983,	1850, 1851, 1856, 2260, 2524,
2003, 2049, 2069,	1857, 1863, 1864,	2555, 2565, 2588,
2132, 2142, 2156, 2166	1870, 1871, 1876,	2598, 2610, 2613, 2622
\bp@freq@scale:	1877, 1883, 1884,	\bp@min@im@ZPK 2193, 2523
. 905, 992,	1976, 1979, 1994,	\bp@min@pow@ 2485
1290, 1745, 1807,	2004, 2042, 2045,	\bp@min@re@pow@
1983, 2003, 2049, 2069	2060, 2070, 2128, 2198, 2205,
\bp@full 205, 207, 208,	2129, 2134, 2143,	2216, 2258, 2259,
212, 350, 352, 353,	2152, 2153, 2158, 2167	2484, 2485, 2487, 2498
357, 490, 492, 493,	\bp@mag@axes	\bp@min@re@threshold@result
497, 611, 613, 614,	. . . 405, 409, 410,	2257, 2338, 2398,
618, 664, 666, 667,	412, 422, 423, 425,	2407, 2453, 2462,
671, 717, 719, 720, 724	506, 507, 509, 869,	2474, 2476, 2478, 2512
\bp@gnu@cmd . 929, 934,	913, 957, 1000,	\bp@min@real@ZPK . . .
1016, 1021, 1070,	1300, 1616, 1641, 1646 2192, 2337
1108, 1111, 1126,	\bp@mag@cmd	\bp@min@thresh@float
1252, 1263, 1484, 1604	. 999, 1008, 1016,	2341, 2399, 2403,
\bp@gnu@plot 928, 933,	1239, 1251, 1281, 1305	2408, 2454, 2458, 2463
1015, 1067, 1110,	\bp@mag@commands . . .	\bp@min@threshold@found
1125, 1250, 1256, 1483 873, 922, 2340, 2397,
\bp@gnu@prefix 1	930, 961, 1009,	2400, 2452, 2455, 2473
\bp@gnu@unwrap@plot .	1017, 1272, 1623, 1661	\bp@min@threshold@result
. 1020,	\bp@mag@plot . . . 254, 2476
1107, 1259, 1603	258, 259, 261, 271,	\bp@mode . . 1217, 1221,
\bp@gnuplot@id 1	272, 274, 366, 367,	1226, 1230, 1255, 1258
\bp@group	369, 915, 1002,	\bp@new . 215, 217, 220,
871, 909, 959, 996,	1619, 1629, 1634, 1671	222, 225, 360, 362,
1270, 1294, 1620, 1652	\bp@max@im@float 2530,	365, 367, 370, 373,
\bp@has@negative@values	2547, 2551, 2621, 2622	375, 378, 500, 502,
. 2202,	\bp@max@im@pow@ . . . 2630	505, 507, 510, 513,
2352, 2380, 2443, 2497	\bp@max@im@value . . .	515, 518, 621, 623,
\bp@has@positive@values 2532, 2552	626, 628, 631, 674,
. 2195,	\bp@max@im@valuefloat	676, 679, 681, 684,
2351, 2371, 2434, 2486 2623, 2624	727, 729, 732, 734, 737
\bp@imagpart . 1316, 1351	\bp@max@re@float . . .	\bp@num . . 1521, 1525, 1526
\bp@k 1514, 1516	. . . 2344, 2390, 2394	\bp@num@deg . . 1550, 1555
\bp@Last@Loop@p 2308, 2310	\bp@max@re@neg@float	

<code>\bp@num@im</code> 1540,	1610, 1612, 1759,	1274, 1284, 1624,
1560, 1598, 1600	1760, 1765, 1766,	1665, 1686, 1693,
<code>\bp@num@list</code>	1821, 1822, 1827,	1730, 1741, 1792,
. . . 1524, 1527, 1536	1828, 1856, 1857,	1803, 1902, 1910,
<code>\bp@num@re</code> 1539,	1863, 1864, 1876,	1928, 1942, 1967,
1557, 1563, 1598, 1600	1877, 1883, 1884,	1978, 2033, 2044,
<code>\bp@p</code> . . . 1506, 1508, 1509	2004, 2007, 2060,	2105, 2113, 2183, 2242
<code>\bp@p@list</code> 1507, 1511, 1516	2070, 2075, 2143,	<code>\bp@tmp</code> . 895, 897, 901,
<code>\bp@parse@add@Bode@opt</code>	2146, 2158, 2167, 2172	911, 1053, 1055,
. . . 1213, 1224, <u>1706</u>	<code>\bp@phex@label</code> <u>83</u>	1057, 1082, 1084, 1086
<code>\bp@parse@complex</code> . . .	<code>\bp@ph@y@label</code> <u>83</u>	<code>\bp@tmp@mag</code> 912, 921, 929
. . . <u>1311</u> , 1502, 1510	<code>\bp@pim</code> . . 2306, 2311, 2316	<code>\bp@tmp@ph</code> . 916, 923, 934
<code>\bp@parse@env@opt</code> . . .	<code>\bp@plot</code> 647, 652,	<code>\bp@user@prefix</code> . 28,
. 1146,	680, 681, 683, 700,	36, 38, 876, 964,
1185, <u>1684</u> , 1905, 2108	705, 733, 734, 736,	1144, 1183, 1275,
<code>\bp@parse@N@opt</code>	1052, 1055, 1057,	1625, 1668, 1670,
. . . . 1734, 1796,	1063, 1067, 1081,	1687, 1696, 1698,
<u>1924</u> , 1971, 2037, 2188	1084, 1086, 1094,	1731, 1793, 1903,
<code>\bp@parse@opt</code>	1099, 1107, 1110,	1929, 1948, 1950,
889, 977, 1277, <u>1615</u>	1242, 1246, 1252,	1968, 2034, 2106, 2184
<code>\bp@ph</code>	1263, 1707, 1717,	<code>\bp@z</code> . . . 1498, 1500, 1501
899, 900, 923, 933,	1756, 1818, 1926,	<code>\bp@z@list</code> 1499, 1503, 1516
986, 987, 1010,	1936, 1951, 1992,	<code>\bp@zim</code> . . 2286, 2291, 2296
1020, 1209, 1216,	2058, 2181, 2295, 2315	<code>\bp@zpk@new@to@legacy</code>
1220, 1227, 1231,	<code>\bp@pre</code> 2303, 2316 877, 1044,
1246, 1256, 1259,	<code>\bp@prefix</code> 27, 45, 1484,	1219, <u>1495</u> , 1732,
1739, 1742, 1759,	1491, 1604, 1611,	1849, 1969, 2127, 2186
1760, 1765, 1766,	2006, 2074, 2145, 2171	<code>\bp@ZPK@plot</code>
1801, 1804, 1821,	<code>\bp@PZZPK@ticksXNeg</code> .	. 900, 1055, 1057,
1822, 1827, 1828, 2203, 2207	1220, 1227, <u>1407</u> ,
1850, 1851, 1856,	<code>\bp@PZZPK@ticksXPos</code> .	1742, 1851, 1979, 2129
1857, 1863, 1864, 2196, 2200	<code>\bp@zre</code> 2283, 2296
1870, 1871, 1876,	<code>\bp@realpart</code> . 1315, 1350	<code>\bp_if_contains:nnTF</code>
1877, 1883, 1884,	<code>\bp@scale</code> 1930,	. . . <u>1354</u> , 1360, 1363
1976, 1979, 1994,	1945, 2185, 2191, 2244	
2004, 2042, 2045,	<code>\bp@short</code> 207,	C
2060, 2070, 2128,	209, 212, 215, 217,	<code>\cs_new_protected:Npn</code>
2129, 2134, 2143,	352, 354, 357, 360, 1355
2152, 2153, 2158, 2167	362, 492, 494, 497,	<code>\currentdegree</code> . 1555,
<code>\bp@ph@axes</code>	500, 502, 613, 615,	1556, 1559, 1560,
. . . 406, 414, 415,	618, 621, 623, 666,	1561, 1563, 1564,
417, 430, 431, 433,	668, 671, 674, 676,	1578, 1579, 1582,
514, 515, 517, 870,	719, 721, 724, 727, 729	1583, 1584, 1586, 1587
917, 958, 1004,	<code>\bp@short@list</code>	D
1303, 1617, 1644, 1647	54, 944, 949, 1031,	<code>\dospecials</code> 65
<code>\bp@ph@cmd</code>	1036, 1132, 1135,	
1003, 1010, 1021,	1171, 1174, 1777,	E
1243, 1262, 1282, 1307	1782, 1839, 1844,	environments:
<code>\bp@ph@commands</code>	1891, 1894, 2018,	BodeMagPlot <u>1168</u>
. 874, 924,	2023, 2086, 2091,	BodePhPlot <u>1129</u>
935, 962, 1011,	2094, 2097, 2328, 2333	BodePlot <u>1266</u>
1022, 1273, 1622, 1659	<code>\bp@style</code> <u>68</u>	
<code>\bp@ph@plot</code> 255,	<code>\bp@tf@new@to@legacy</code>	G
263, 264, 266, 280, 965,	<code>\g@addto@macro</code>
281, 283, 374, 375,	1077, 1215, <u>1518</u> ,	62, 566, 568, 572,
377, 919, 1006,	1794, 1869, 2035, 2151	574, 582, 584, 588,
1618, 1632, 1635, 1673	<code>\bp@TF@plot</code>	590, 598, 600, 604, 606
<code>\bp@ph@scale</code> 987, 1084, 1086,	<code>\gnu@id</code>
. . . <u>83</u> , 178, 183,	1216, 1231, <u>1538</u> ,	927, 928, 932, 933,
323, 328, 463, 468,	1804, 1871, 2045, 2153	1014, 1015, 1019,
772, 776, 787, 800,	<code>\bp@tikz</code> 875, 896,	1020, 1066, 1067,
803, 815, 821, 822,	963, 983, 1143,	1103, 1107, 1110,
842, 1010, 1599,	1151, 1182, 1190,	

S	T	1314, 1315, 1316, 1354
\seq_pop_left:NN 1325,	\temp@mag@cmd 1298, 1304	\tl_remove_all:Nn .. 1319
1326, 1327, 1332, 1333	\temp@ph@cmd . 1301, 1306	\tl_set:Nn
\str_if_eq:VnT 1336, 1337	\tl_gset:Nx 1352	1334, 1335, 1336,
\stripbodeprefix ...	\tl_gset_eq:NN 1350, 1351	1337, 1340, 1341, 1347
... 206, 207, 351,	\tl_if_empty:NT ... 1335	\tl_set:No 1318
352, 491, 492, 612,	\tl_if_in:VnTF 1357	\tl_set:Nx 1356
613, 665, 666, 718, 719	\tl_new:N . 1312, 1313,	\tl_set_eq:NN 1346

Change History

v1.0	General: Initial release	1	\BodeTF: Fixed phase wrapping in gnuplot mode	42	
v1.0.1	\addBodeZPKPlots: Improved optional argument handling.	43	v1.1.1	General: Enable Hz and rad units	1
	\BodeZPK: Pass arbitrary TikZ commands as options.	40		\addBodeComponentPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	44
v1.0.2	bp@gnu@prefix: Fixed issue #1	25		\addBodeTFPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	44
v1.0.3	BodePlot: Added tikz option to environments	47		\addBodeZPKPlots: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	43
	\BodeTF: Added Tikz option	42		\addNicholsTFChart: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	63
	\bp@parse@env@opt: Added tikz option to environments	55		\addNyquistTFPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	58
	\bp@parse@N@opt: Added commands and tikz options	59		\addNyquistZPKPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	58
	\bp@parse@opt: Added Tikz option	53		BodeMagPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	45
	NicholsChart: Added tikz option to environments	62		BodePhPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	45
	\NicholsTF: Added commands and tikz options	61		BodePlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	47
	\NicholsZPK: Added commands and tikz options	60		\BodeTF: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	42
	\NyquistTF: Added commands and tikz options	57		\NyquistTF: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	57
	\NyquistZPK: Added commands and tikz options	56		\NyquistZPK: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	56
	bp@gnu@prefix: Added jobname to gnuplot prefix	25		bp@ph@y@label: New macros to enable ‘Hz’ and ‘rad’ units for frequency and phase, respectively	26
	NyquistPlot: Added tikz option to environments	58			
v1.0.4	General: Fixed unintended optional argument macro expansion	1			
v1.0.5	\bp@parse@opt: Fixed a bug	53			
v1.0.6	General: Fixed issue #3	1			
v1.0.7	General: Removed unnecessary semicolons	1			
	Updated documentation	1			
v1.0.8	General: Added a new class option ‘declutter’	1			
	bp@gnu@prefix: Fixed issue #6	25			
v1.1.0	General: Fixed phase wrapping in gnuplot mode	1			
	\addBodeTFPlot: Fixed phase wrapping in gnuplot mode	44			
	BodeMagPlot: Added separate environments for phase and magnitude plots	45			
	BodePhPlot: Added separate environments for phase and magnitude plots	45			
	BodePlot: Deprecated BodePlot environment	47			

<code>\PhS0ZerosLin</code> : Fix scaling bug introduced in v1.1.1	39	<code>\NicholsTF</code> : Added code to handle active characters	61
<code>NyquistPlot</code> : Defined using the ‘NewEniron’ command from the ‘environ’ package to fix conflicts with externalization	58	<code>\NicholsZPK</code> : Added code to handle active characters	60
v1.1.3		<code>\NyquistTF</code> : Added code to handle active characters	57
<code>\addBodeComponentPlot</code> : Changed implementation to respect user-supplied domain	44	<code>\NyquistZPK</code> : Added code to handle active characters	56
<code>\addBodeTFPlot</code> : Changed implementation to respect user-supplied domain	44	<code>NyquistPlot</code> : Defined using the ‘NewDocumentEnvironment’ command from the ‘xparse’ package and added a hook to handle active characters	58
<code>\addBodeZPKPlots</code> : Changed implementation to respect user-supplied domain	43	v1.1.6	
<code>\addNicholsTFChart</code> : Changed implementation to respect user-supplied domain	63	<code>\bp@short@list</code> : Detect ‘babel-french’ using ‘frenchbsetup’	26
<code>\addNicholsZPKChart</code> : Changed implementation to respect user-supplied domain	62	v1.1.7	
<code>\addNyquistTFPlot</code> : Changed implementation to respect user-supplied domain	58	General: Detect and turn off shorthands to improve ‘babel’ compatibility	1
<code>\addNyquistZPKPlot</code> : Changed implementation to respect user-supplied domain	58	<code>BodeMagPlot</code> : Use auto-generated list of active characters instead of manually entering them.	45
v1.1.4		<code>BodePhPlot</code> : Use auto-generated list of active characters instead of manually entering them.	45
<code>\addBodeTFPlot</code> : Changed phase wrapping in pgf mode	44	<code>BodePlot</code> : Use auto-generated list of active characters instead of manually entering them	47
<code>\addNicholsTFChart</code> : Changed phase wrapping in pgf mode	63	<code>\BodeTF</code> : Use auto-generated list of shorthands instead of manually specifying them	43
<code>\BodeTF</code> : Changed phase wrapping in pgf mode	42	<code>\BodeZPK</code> : Use auto-generated list of shorthands instead of manually specifying them	41
<code>bp@gnu@prefix</code> : Changed phase wrapping in pgf mode	25	<code>\bp@short@list</code> : Directly detect shorthands instead of detecting the language.	26
v1.1.5		<code>NicholsChart</code> : Use auto-generated list of active characters instead of manually entering them.	62
<code>BodeMagPlot</code> : Defined using the ‘NewDocumentEnvironment’ command from the ‘xparse’ package and added a hook to handle active characters	45	<code>\NicholsTF</code> : Use auto-generated list of shorthands instead of manually specifying them	61
<code>BodePhPlot</code> : Defined using the ‘NewDocumentEnvironment’ command from the ‘xparse’ package and added a hook to handle active characters	45	<code>\NicholsZPK</code> : Use auto-generated list of shorthands instead of manually specifying them	60
<code>BodePlot</code> : Defined using the ‘NewDocumentEnvironment’ command from the ‘xparse’ package and added a hook to handle active characters	47	<code>\NyquistTF</code> : Use auto-generated list of shorthands instead of manually specifying them	57
<code>\BodeTF</code> : Added code to handle active characters	43	<code>\NyquistZPK</code> : Use auto-generated list of shorthands instead of manually specifying them	56
<code>\BodeZPK</code> : Added code to handle active characters	41	<code>NyquistPlot</code> : Use auto-generated list of active characters instead of manually entering them.	58
<code>NicholsChart</code> : Defined using the ‘NewDocumentEnvironment’ command from the ‘xparse’ package and added a hook to handle active characters	62	v1.2	
		General: Removed global option to process pgf commands in radians	1
		<code>bp@gnu@prefix</code> : Removed global option to process pgf commands in radians	25

v2.0		/bodeplot: Added a pgfkeys-based simplified user interface	29
General: Added pole zero maps	1	/bodeplot/add: Added a pgfkeys-based simplified user interface	34
Added pole-zero map functionality	23	/bodeplot/combinedenv: Added a pgfkeys-based simplified user interface	31
\PoleZeroMapZPK: Added pole-zero map functionality	63	/bodeplot/env: Added a pgfkeys-based simplified user interface	27
v2.1		/bodeplot/nichols: Added a pgfkeys-based simplified user interface	36
General: Added an environment and macros to add multiple transfer functions to a single Bode plot	1	/bodeplot/nyquist: Added a pgfkeys-based simplified user interface	35
Added unified Bode plot macro supporting both ZPK and TF systems	16	/bodeplot/pzmap: Added a pgfkeys-based simplified user interface	37
Enhanced BodePlot environment with unified plot command collection	16	/bodeplot/tf: Added a pgfkeys-based simplified user interface	34
\addBodePlot: Added unified Bode plot macro supporting both ZPK and TF systems	46	/bodeplot/zpk: Added a pgfkeys-based simplified user interface	33
BodePlot: Enhanced BodePlot environment with unified plot command collection	47	\BodeTF: Added a pgfkeys-based simplified user interface	42
\bp@parse@add@Bode@opt: Added option parser for unified Bode plot macro	55	\BodeZPK: Added a pgfkeys-based simplified user interface	40
v2.1.1		Updated to use a pgfkeys-based interface	40
bp@gnu@prefix: Added a new helper command to allow the user to set per-plot gnuplot file names to address feature request #14	25	\bp@contains@equal: Added interface detector	49
v3.0		\bp@contains@num: Added keyword detector	49
General: Improved the interface to enable PGF-style options and more natural syntax for complex numbers in zeros and poles	1	\bp@fix@add@opt: Added option processor for add commands	49
\addBodeComponentPlot: Updated to use \bp@prefix for consistency	44	\bp@parse@complex: Added complex number parser for new interface	48
\addBodePlot: Added a pgfkeys-based simplified user interface	46	\bp@tf@new@to@legacy: Added converter from new to legacy TF interface	52
\addBodeTFPlot: Added a pgfkeys-based simplified user interface	44	\bp@zpk@new@to@legacy: Added converter from new to legacy ZPK interface	51
\addBodeZPKPlots: Added a pgfkeys-based simplified user interface	43	\bp_if_contains:nnTF: Added substring checker	49
\addNicholsTFChart: Added a pgfkeys-based simplified user interface	63	\NicholsTF: Added a pgfkeys-based simplified user interface	61
\addNicholsZPKChart: Added a pgfkeys-based simplified user interface	62	\NicholsZPK: Added a pgfkeys-based simplified user interface	60
\addNyquistTFPlot: Added a pgfkeys-based simplified user interface	58	\NyquistTF: Added a pgfkeys-based simplified user interface	57
\addNyquistZPKPlot: Added a pgfkeys-based simplified user interface	58	\NyquistZPK: Added a pgfkeys-based simplified user interface	56
BodeMagPlot: Added a pgfkeys-based simplified user interface	45	\PoleZeroMapZPK: Added a pgfkeys-based simplified user interface	63
BodePhPlot: Added a pgfkeys-based simplified user interface	45	v3.0.1	
BodePlot: Added a pgfkeys-based simplified user interface	47	BodePhPlot: Fixed a typo to resolve issue #17	45