

# The **bodeplot** package

## version 3.0.4

Rushikesh Kamalapurkar  
rlkamalapurkar@gmail.com

May 14, 2026

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	External Dependencies . . . . .	2
1.2	Directory Structure . . . . .	2
1.3	Limitations . . . . .	2
<b>2</b>	<b>TL;DR</b>	<b>3</b>
<b>3</b>	<b>Usage</b>	<b>10</b>
3.1	Bode plots . . . . .	10
3.1.1	Basic components up to first order . . . . .	17
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
<b>4</b>	<b>Implementation</b>	<b>25</b>
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 . . . . .	58
4.6.1	User macros . . . . .	58
4.6.2	Internal macros . . . . .	61
4.7	Nichols charts . . . . .	62
4.8	Pole-zero maps . . . . .	66
4.8.1	User macros . . . . .	66
4.8.2	Internal macros . . . . .	69
<b>A</b>	<b>TL;DR for package version 2.1.1 and earlier</b>	<b>74</b>
	<b>Index</b>	<b>86</b>
	<b>Change History</b>	<b>86</b>

# 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  $-180^\circ$  and  $180^\circ$  or  $-\pi$  and  $\pi$  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  $0^\circ$  and  $360^\circ$  or  $0$  and  $2\pi$  radian. The commands `\makeatletter\renewcommand{\n@mod}{\n@mod@n}\makeatother` will wrap the phase between  $-360^\circ$  and  $0^\circ$  or  $-2\pi$  and  $0$  radian.
- Use of the `declutter` option with other directory management tools such as a `tikzexternalize` prefix is not recommended.

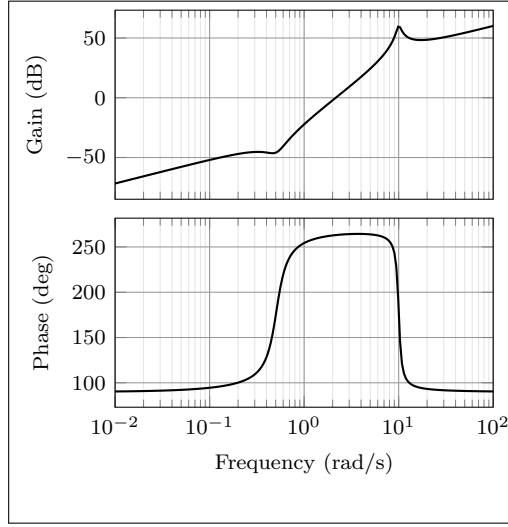
## 2 TL;DR

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.

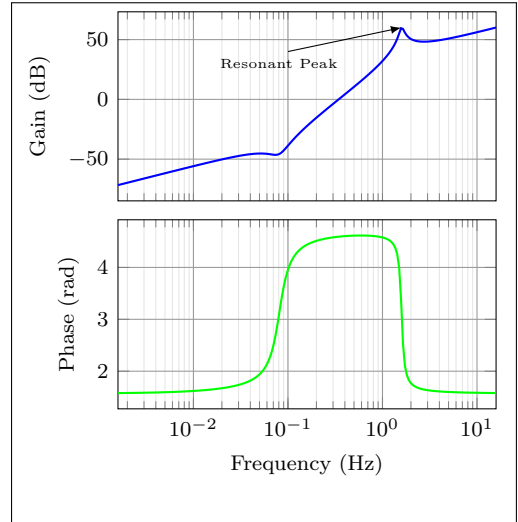
Bode plot in ZPK format



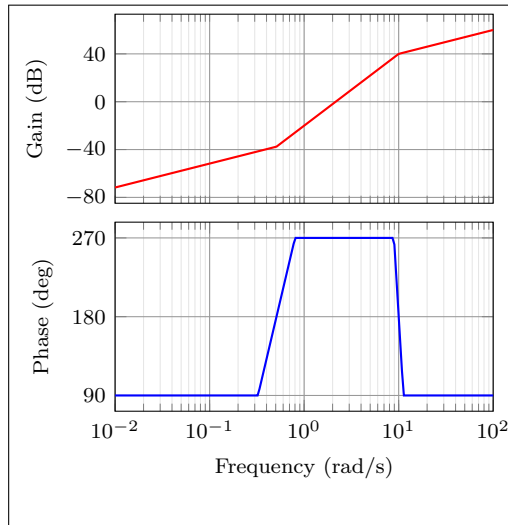
```
\BodeZPK[domain=0.01:100]{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10%
}
```

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}%
}
```



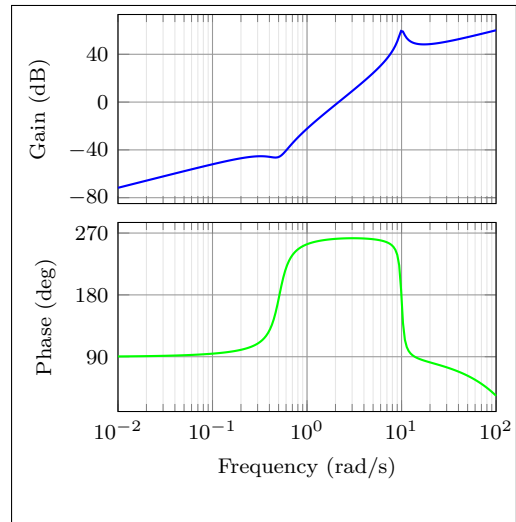
## Linear approximation with customization



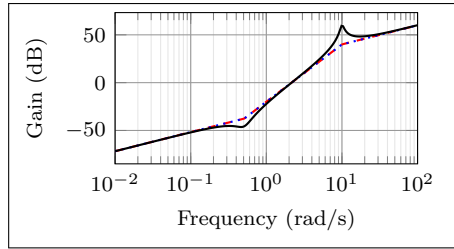
```
\BodeZPK[%
domain=0.01:100,
mag plot={red,thick},
ph plot={blue,thick},
mag axes={ytick distance=40},
ph axes={ytick distance=90},
approx=linear%
]{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10%
}
```

## Plot with delay and customization

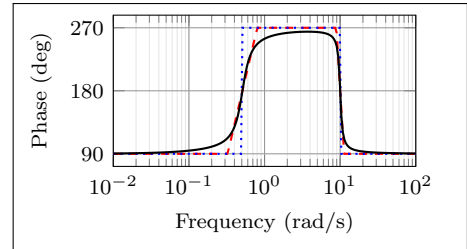
```
\BodeZPK[%
domain=0.01:100,
mag plot={blue,thick},
ph plot={green,thick},
mag axes={ytick distance=40},
ph axes={ytick distance=90}%
]{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10,
delay=0.01%
}
```



## Individual gain and phase plots with more customization



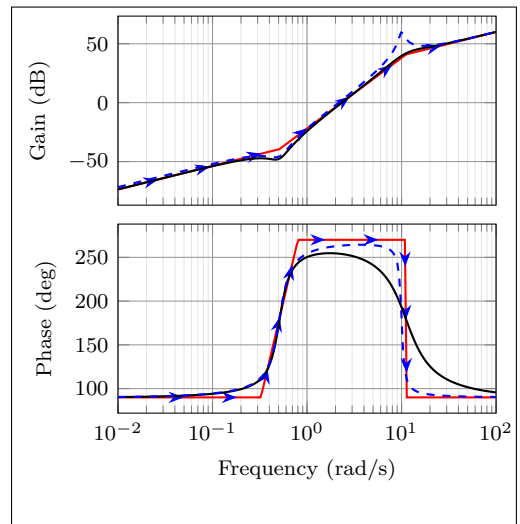
```
\begin{BodeMagPlot}{%
  axes={height=2cm,
    width=4cm},
  domain=0.01:100%
}
\addBodeZPKPlots{%
  true={black,thick},
  linear={red,dashed,thick},
  asymptotic={blue,dotted,thick}%
}
{magnitude}
{%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
}\end{BodeMagPlot}
```



```
\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}
```

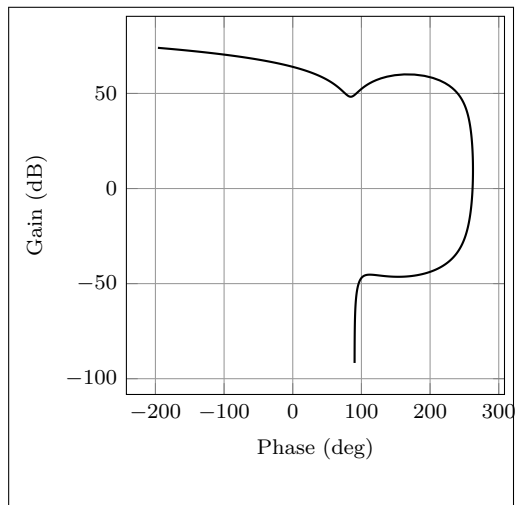
Multiple transfer functions in a single Bode plot using the **BodePlot** environment and the **\addBodePlot** macro introduced in v2.1.

```
\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

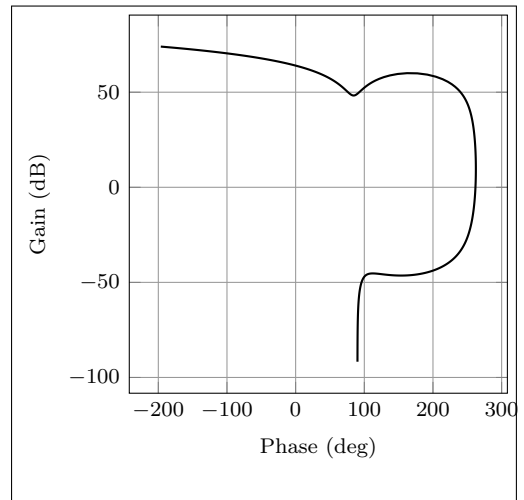


```
\NicholsZPK[domain=0.001:500,samples=1000]
{%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10,
delay=0.01%
}
```

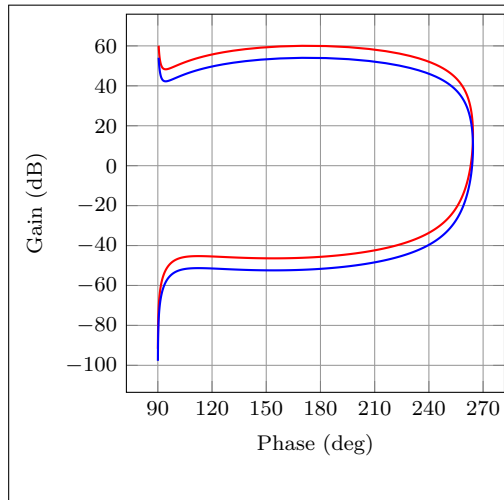
---

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%
}
```



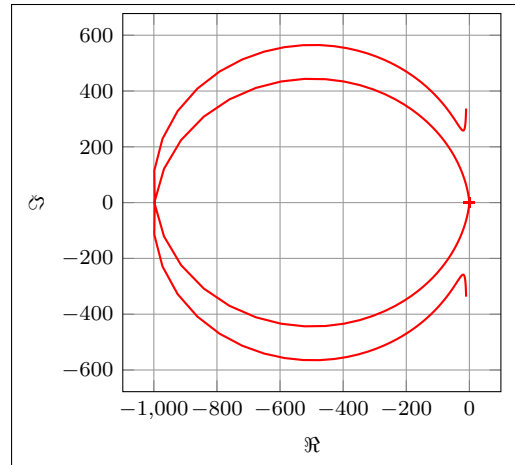
## Multiple Nichols charts with customization



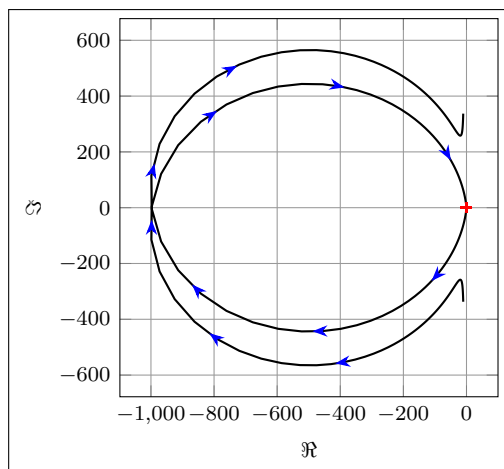
```
\begin{NicholsChart}{%
  ytick distance=20,
  xtick distance=30,
  domain=0.001:100%
}
\addNicholsZPKChart [red,samples=1000] {%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},
  gain=10%
}
\addNicholsZPKChart [blue,samples=1000] {%
  zeros={0,-0.1-0.5i,-0.1+0.5i},
  poles={-0.5-10i,-0.5+10i},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



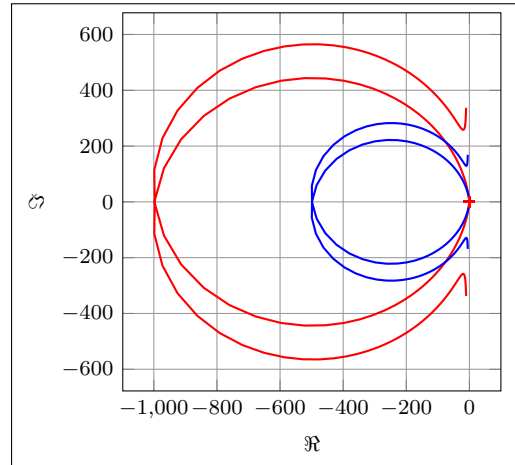
## Nyquist plot in TF format with arrows



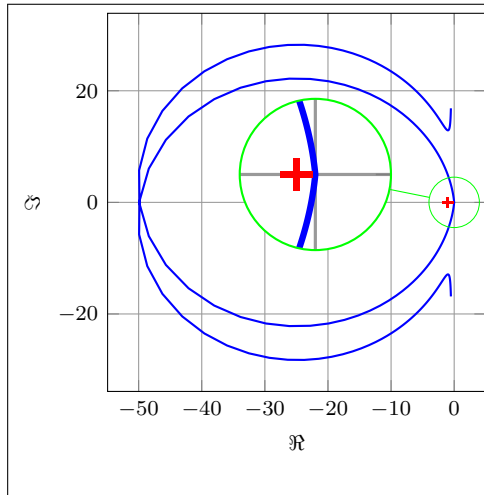
```
\NyquistTF[
  domain=-30:30,
  plot={%
    samples=1000,
    postaction=decorate,
    decoration={
      markings,
      mark=between positions 0.1 and 0.9 step 5em with {%
        \arrow{Stealth [length=2mm, blue]}
      }
    }
  }%
]{%
  numerator={10,2,2.6,0},
  denominator={1,1,100.25}%
}
```

## Multiple Nyquist plots with customization

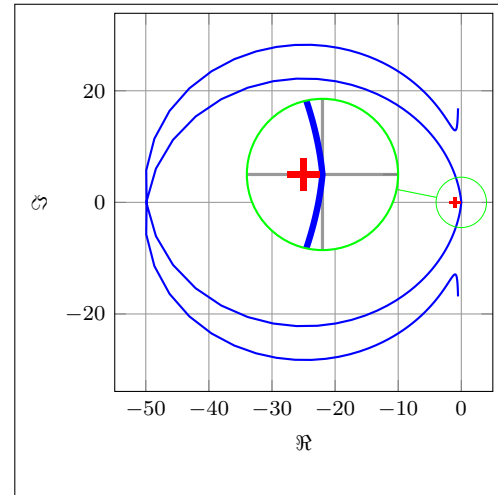
```
\begin{NyquistPlot}[domain=-30:30]
\addNyquistZPKPlot [red,samples=1000] {%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=10%
}
\addNyquistZPKPlot [blue,samples=1000] {%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=5%
}
\end{NyquistPlot}
```



## Nyquist plots with additional commands, using two different macros



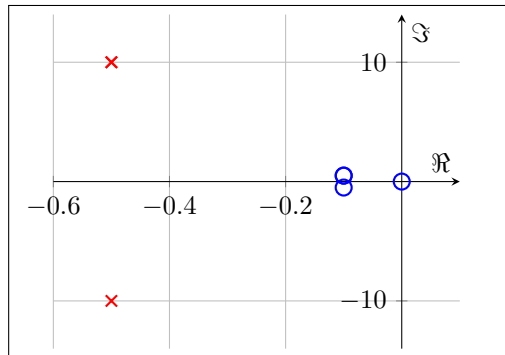
```
\begin{NyquistPlot}[%
tikz={
spy using outlines={%
circle,
magnification=3,
connect spies,
size=2cm
}
},
domain=-30:30%
]
\addNyquistZPKPlot [blue,samples=1000] {%
zeros={0,-0.1-0.5i,-0.1+0.5i},
poles={-0.5-10i,-0.5+10i},
gain=0.5%
}
\coordinate (spyon) at (axis cs:0,0);
\coordinate (spyat) at (axis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



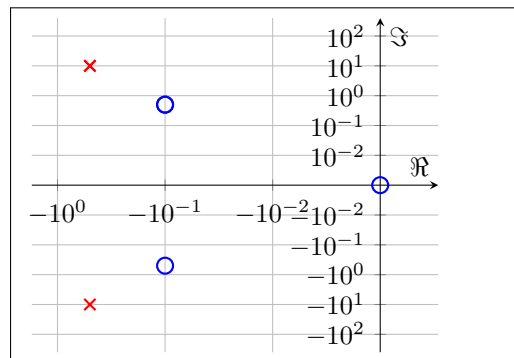
## Pole-zero map



```
\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**, where **a** and **b** can be arbitrary pgfmath/gnuplot expressions. The parser starts by detecting the trailing **i** and then finds the first top-level **+** or **-**. Everything between the **i** and the **+** or **-** is treated as the imaginary part, and everything before that is treated as the real part. If no **+** or **-** is found, the entire string except the trailing **i** is treated as the imaginary part and the real part is assumed to be zero. If the **gain** key is not specified, it is assumed to be 1. If the **delay** key is not specified, it is assumed to be zero.

**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.

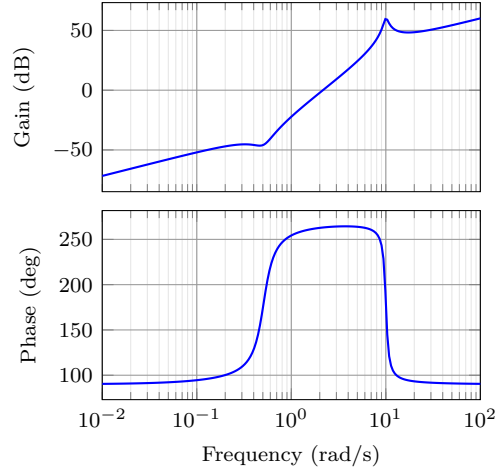


Figure 1: Output of the `\BodeZPK` macro.

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}
```

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:

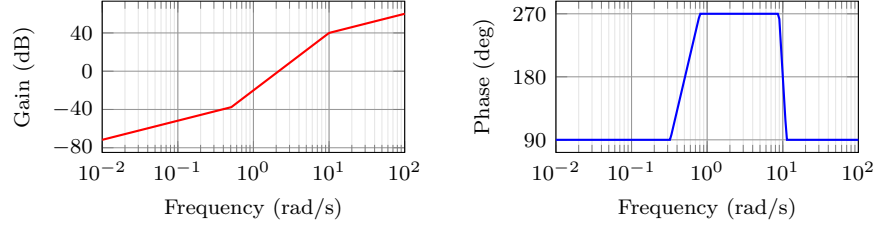


Figure 2: Customization of the default `\BodeZPK` macro.

- `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,

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

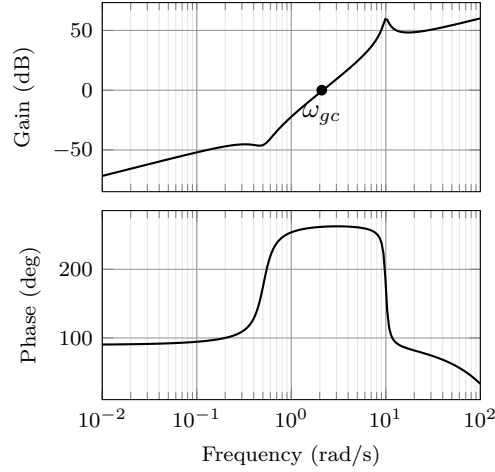


Figure 3: Output of the `\BodeTF` macro with an optional TikZ command used to mark the gain crossover frequency.

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}
```

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

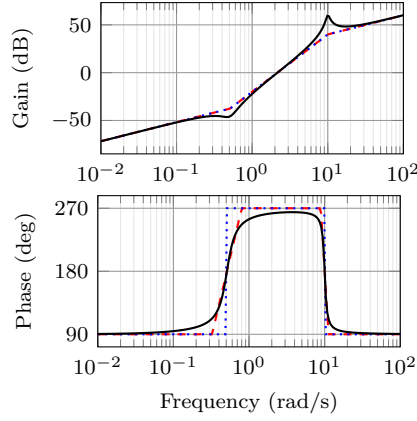


Figure 4: Superimposed approximate and true Bode plots using the **BodeMagPlot** and **BodePhPlot** environments and the `\addBodeZPKPlots` macro.

which generates the plot in Figure 4.

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

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

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 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[ $\langle plot-options \rangle$ ]{ $\langle plot-command \rangle$ }`

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



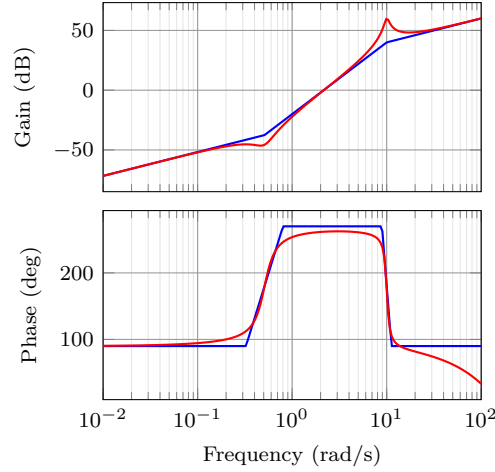


Figure 5: Bode plot combining ZPK and TF systems using the `BodePlot` environment and the unified `\addBodePlot` macro.

`\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{<real-part>}{<imaginary-part>}`

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]{%

```

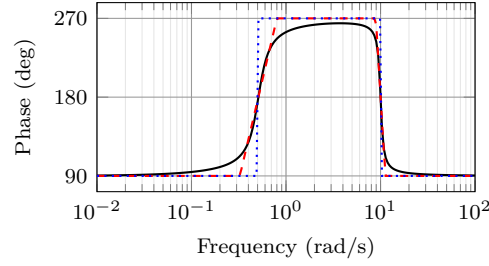


Figure 6: Superimposed approximate and true Bode Phase plot using the `BodePhPlot` environment, the `\addBodeComponentPlot` macro, and several macros of the `\TypeFeatureApprox` form.

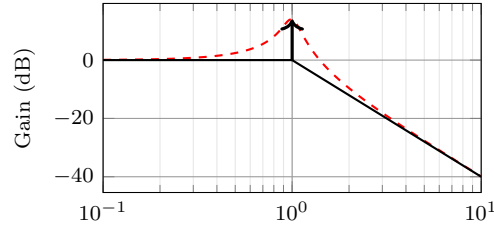


Figure 7: Resonant peak in asymptotic Bode plot using `\MagSOPolesPeak`.

```
\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.

### 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,  $\zeta$ , and the natural frequency,  $\omega_n$  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,  $\zeta$ , and the natural frequency,  $\omega_n$  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`, where `a` and `b` can be arbitrary pgfmath/gnuplot expressions. 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}
```

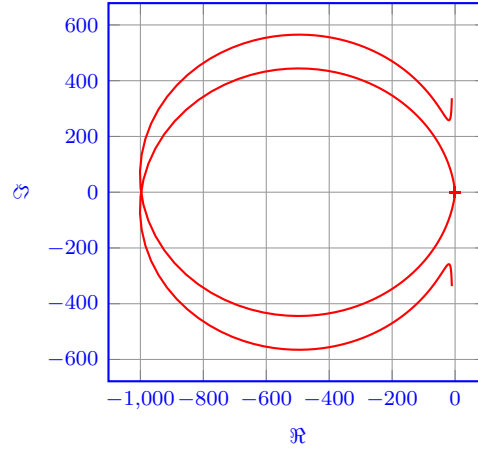


Figure 8: Output of the `\NyquistZPK` macro.

generates the Nyquist plot in Figure 8.

**Legacy interface (v2.1.1 and earlier):**

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

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 `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}
```

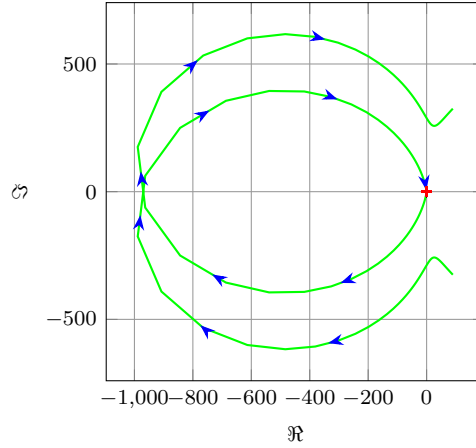


Figure 9: Output of the `\NyquistTF` macro with direction arrows. Increasing the number of samples can cause `decorations.markings` to throw errors.

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.
- **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}[\langle options \rangle]{\langle min-freq \rangle}{\langle max-freq \rangle}
  \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[\langle plot-options \rangle]{\langle zpk-spec \rangle}
```

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, 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[\langle plot-options \rangle]{\langle z/{...},p/{...},k/...,d/... \rangle}
```

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

`\addNyquistTFPlot` `\addNyquistTFPlot[ $\langle plot-options \rangle$ ]{ $\langle tf-spec \rangle$ }`  
 Identical to `\addNyquistZPKPlot` but expects **tf-spec** in the same polynomial format accepted by `\BodeTF`.

### 3.3 Nichols charts

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

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

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[ $\langle plot-options \rangle$ ]{ $\langle tf-spec \rangle$ }`

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

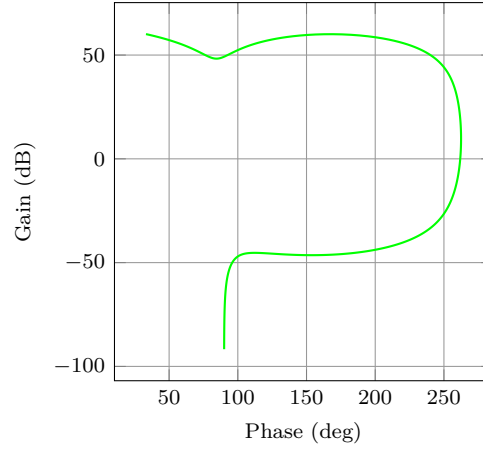


Figure 10: Output of the `\NicholsTF` macro.

### 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+bi` (where `a` and `b` can be arbitrary pgfmath/gnuplot expressions) 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.

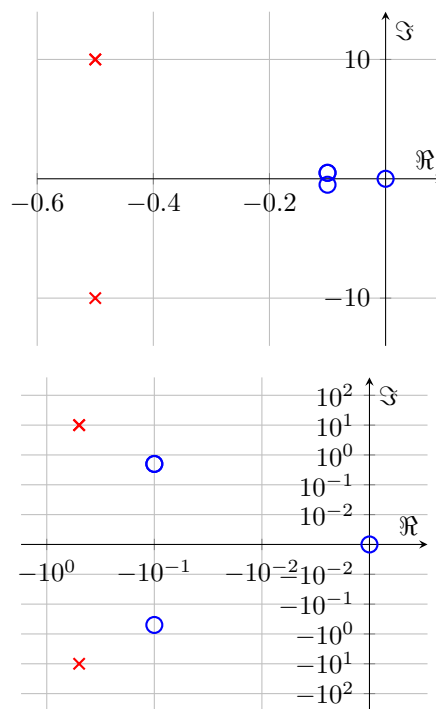


Figure 11: Pole-zero maps generated using the `\PoleZeroMapZPK` macro with linear and logarithmic scaling.



## 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.

```

\MagCSPolesLin 810 \newcommand*\MagCSPoles[2]{(-20*log10(sqrt(\n@pow{\n@pow{#2}{2}
\PhCSPoles      - \n@pow{t}{2}}{2} + \n@pow{2*#1*#2*t}{2}}))}
\PhCSPolesAsymp 811 \newcommand*\MagCSPolesLin[2]{(t < #2 ? -40*log10(#2) : - 40*log10(t))}
\PhCSPolesLin   812 \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  $G(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$  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.

```

\MagSOPolesLin 837 \newcommand*\MagSOPoles[2]{
\PhSOPoles      (-20*log10(sqrt(\n@pow{#2} - \n@pow{t}{2}}{2} + \n@pow{#1*t}{2}}))}
\PhSOPolesAsymp 838 \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*\MagSOPoles{0-\MagSOPoles}
850 \newcommand*\MagSOPolesLin{0-\MagSOPolesLin}
851 \newcommand*\MagSOPolesAsymp{0-\MagSOPolesAsymp}
852 \newcommand*\PhSOPoles{0-\PhSOPoles}
853 \newcommand*\PhSOPolesLin{0-\PhSOPolesLin}
854 \newcommand*\PhSOPolesAsymp{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][]{
856 \draw[#1,->] (axis cs:{sqrt(abs(#3))},{-20*log10(abs(#3))}) --
857 (axis cs:{sqrt(abs(#3))},{-20*log10(abs(#3)) -
858 20*log10(abs(#2/sqrt(abs(#3))))});
859 }
860 \newcommand*\MagSOPolesPeak[3][]{
861 \draw[#1,->] (axis cs:{sqrt(abs(#3))},{20*log10(abs(#3))}) --
862 (axis cs:{sqrt(abs(#3))},{20*log10(abs(#3)) +
863 20*log10(abs(#2/sqrt(abs(#3))))});
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   \NewDocumentCommand{\BodeZPK@Shorthandoff}{ 0{ } m G{ } G{ } }{%
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}{ 0{ } 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 \NewDocumentCommand{\BodeTF@Shorthandoff}{ 0{} m G{} G{} }{%
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{semilogxaxis}[
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{semilogxaxis}
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 the format  $a+bi$  where  $a$  and  $b$  can be arbitrary pgf-math/gnuplot expressions. The scans to the left starting from the trailing  $i$  and finds the first top-level  $+$  or  $-$ , i.e., the first such operator not inside braces or parentheses. Everything between the  $i$  and the  $+$  or  $-$  is treated as the imaginary part, and everything else is treated as the real part. 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
1318 % Variables for the complex number parser
1319 \tl_new:N \l__bp_input_tl
1320 \tl_new:N \l__bp_before_tl
1321 \tl_new:N \l__bp_after_tl
1322 \tl_new:N \l__bp_sep_tl
1323 \int_new:N \l__bp_pos_int
1324 \int_new:N \l__bp_seppos_int
1325 \int_new:N \l__bp_paren_int
1326 \bool_new:N \l__bp_found_bool
1327
1328 % Generate V variant for tl_if_single_token
1329 \prg_generate_conditional_variant:Nnn \tl_if_single_token:n { V } { TF, T, F }
1330
1331 \NewDocumentCommand{\bp@parse@complex}{m}{
1332   \tl_set:Nx \l__bp_input_tl {#1}
1333   \tl_remove_all:Nn \l__bp_input_tl { ~ }
1334
1335   % Check if it ends with 'i'
1336   \regex_match:nVTF { i$ } \l__bp_input_tl
1337   { \__bp_parse_complex_with_i: }
1338   { \__bp_parse_real_only: }

```



```

1339
1340 \tl_gset_eq:NN \bp@realpart \l_real_tl
1341 \tl_gset_eq:NN \bp@imagpart \l_imag_tl
1342 \tl_gset:Nx \bp@complex { { { \l_real_tl } , { \l_imag_tl } } }
1343 }
1344
1345 % Parse a pure real number (no trailing i)
1346 \cs_new_protected:Npn \__bp_parse_real_only:
1347 {
1348   \__bp_unwrap_braces:N \l__bp_input_tl
1349   \tl_set_eq:NN \l_real_tl \l__bp_input_tl
1350   \tl_set:Nn \l_imag_tl {0}
1351 }
1352
1353 % Parse a complex number (has trailing i)
1354 \cs_new_protected:Npn \__bp_parse_complex_with_i:
1355 {
1356   % Remove trailing 'i'
1357   \regex_replace_once:nnN { i$ } {} \l__bp_input_tl
1358
1359   % Find the last + or - at depth 0 (but not at position 0)
1360   \__bp_find_separator:
1361
1362   \bool_if:NTF \l__bp_found_bool
1363     { \__bp_split_at_separator: }
1364     { \__bp_parse_pure_imaginary: }
1365 }
1366
1367 % Parse pure imaginary (no separator found)
1368 \cs_new_protected:Npn \__bp_parse_pure_imaginary:
1369 {
1370   \tl_set:Nn \l_real_tl {0}
1371   \__bp_set_signed_value:NN \l_imag_tl \l__bp_input_tl
1372 }
1373
1374 % Set target to value with sign handling
1375 \cs_new_protected:Npn \__bp_set_signed_value:NN #1#2
1376 {
1377   \tl_if_empty:NTF #2
1378     { \tl_set:Nn #1 {1} }
1379     {
1380       \tl_set:Nx \l_tmpa_tl { \tl_head:N #2 }
1381       \str_if_eq:VnTF \l_tmpa_tl {-}
1382       {
1383         \tl_set:Nx \l_tmpb_tl { \tl_tail:N #2 }
1384         \tl_if_empty:NTF \l_tmpb_tl
1385           { \tl_set:Nn #1 {-1} }
1386           {
1387             \__bp_unwrap_braces:N \l_tmpb_tl
1388             \tl_set:Nx #1 { -( \l_tmpb_tl ) }
1389           }
1390       }
1391       {
1392         \str_if_eq:VnTF \l_tmpa_tl {+}
1393         {
1394           \tl_set:Nx \l_tmpb_tl { \tl_tail:N #2 }
1395           \tl_if_empty:NTF \l_tmpb_tl
1396             { \tl_set:Nn #1 {1} }
1397             {
1398               \__bp_unwrap_braces:N \l_tmpb_tl
1399               \tl_set_eq:NN #1 \l_tmpb_tl
1400             }
1401         }
1402       }
1403     }

```

```

1402         {
1403             \__bp_unwrap_braces:N #2
1404             \tl_set_eq:NN #1 #2
1405         }
1406     }
1407 }
1408 }
1409
1410 % Helper to unwrap a single braced group
1411 \cs_new_protected:Npn \__bp_unwrap_braces:N #1
1412 {
1413     \tl_if_single_token:VF #1
1414     {
1415         \tl_set:Nx \l_tmpc_tl { \tl_count:N #1 }
1416         \int_compare:nNnT { \l_tmpc_tl } = {1}
1417         { \tl_set:Nx #1 { \exp_after:wN \use:n #1 } }
1418     }
1419 }
1420
1421 % Process one token/group to find separator
1422 \cs_new_protected:Npn \__bp_process_item:n #1
1423 {
1424     \tl_if_single_token:nTF {#1}
1425     {
1426         % Track parenthesis depth
1427         \token_if_eq_charcode:NNT #1 ( { \int_incr:N \l__bp_paren_int }
1428         \token_if_eq_charcode:NNT #1 ) { \int_decr:N \l__bp_paren_int }
1429         % Only consider separators at position > 0 and paren depth 0
1430         \int_compare:nNnT { \l__bp_pos_int } > {0}
1431         {
1432             \int_compare:nNnT { \l__bp_paren_int } = {0}
1433             {
1434                 \bool_lazy_or:nnT
1435                 { \token_if_eq_charcode_p:NN #1 + }
1436                 { \token_if_eq_charcode_p:NN #1 - }
1437                 {
1438                     \bool_set_true:N \l__bp_found_bool
1439                     \int_set_eq:NN \l__bp_sep_pos_int \l__bp_pos_int
1440                     \tl_set:Nn \l__bp_sep_tl {#1}
1441                 }
1442             }
1443         }
1444     }
1445     { }
1446     \int_incr:N \l__bp_pos_int
1447 }
1448
1449 % Find the last + or - at depth 0
1450 \cs_new_protected:Npn \__bp_find_separator:
1451 {
1452     \bool_set_false:N \l__bp_found_bool
1453     \int_zero:N \l__bp_pos_int
1454     \int_zero:N \l__bp_paren_int
1455     \tl_map_function:NN \l__bp_input_tl \__bp_process_item:n
1456 }
1457
1458 % Collect items for splitting
1459 \cs_new_protected:Npn \__bp_collect_item:n #1
1460 {
1461     \int_compare:nNnTF { \l__bp_pos_int } < { \l__bp_sep_pos_int }
1462     { \tl_put_right:Nn \l__bp_before_tl {#1} }
1463     {
1464         \int_compare:nNnF { \l__bp_pos_int } = { \l__bp_sep_pos_int }

```

```

1465         { \tl_put_right:Nn \l__bp_after_tl {#1} }
1466     }
1467     \int_incr:N \l__bp_pos_int
1468 }
1469
1470 % Split the input at the separator position
1471 \cs_new_protected:Npn \__bp_split_at_separator:
1472 {
1473     \tl_clear:N \l__bp_before_tl
1474     \tl_clear:N \l__bp_after_tl
1475     \int_zero:N \l__bp_pos_int
1476
1477     \tl_map_function:NN \l__bp_input_tl \__bp_collect_item:n
1478
1479     \__bp_unwrap_braces:N \l__bp_before_tl
1480     \__bp_unwrap_braces:N \l__bp_after_tl
1481
1482     \tl_if_empty:NTF \l__bp_before_tl
1483     { \tl_set:Nn \l_real_tl {0} }
1484     { \tl_set_eq:NN \l_real_tl \l__bp_before_tl }
1485
1486     \tl_if_empty:NTF \l__bp_after_tl
1487     {
1488         \str_if_eq:VnTF \l__bp_sep_tl {-}
1489         { \tl_set:Nn \l_imag_tl {-1} }
1490         { \tl_set:Nn \l_imag_tl {1} }
1491     }
1492     {
1493         \str_if_eq:VnTF \l__bp_sep_tl {-}
1494         { \tl_set:Nx \l_imag_tl {-( \l__bp_after_tl ) } }
1495         { \tl_set_eq:NN \l_imag_tl \l__bp_after_tl }
1496     }
1497 }

```

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

```

1498 \tl_new:N \l__bode_check_tl
1499 \cs_new_protected:Npn \bp_if_contains:nnTF #1#2#3#4 {
1500     \tl_set:Nx \l__bode_check_tl {#1}
1501     \tl_if_in:VnTF \l__bode_check_tl {#2} {#3} {#4}
1502 }

```

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

```

1503 \NewDocumentCommand{\bp@contains@equal}{m m m}{%
1504     \bp_if_contains:nnTF {#1} {=} {#2} {#3}%
1505 }

```

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

```

1506 \NewDocumentCommand{\bp@contains@num}{m m m}{%
1507     \bp_if_contains:nnTF {#1} {numerator} {#2} {#3}%
1508 }
1509 \ExplSyntaxOff

```

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

```

1510 \NewDocumentCommand{\bp@fix@add@opt}{ m }{%
1511     \gdef\bp@add@0{}
1512     \gdef\bp@add@tmp{}
1513     \foreach \approx/\opt in {#1} {
1514         \ifx\approx\@empty\else
1515             \ifnum\pdf@strcmp{\approx}{linear}=0
1516                 \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0}%
1517                     linear/{\unexpanded\expandafter{\opt}}},}
1518             \else
1519                 \ifnum\pdf@strcmp{\approx}{asymptotic}=0

```

```

1520     \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0}%
1521     asymptotic/{\unexpanded\expandafter{\opt}},}
1522   \else
1523     \ifnum\pdf@strcmp{\approx}{true}=0
1524       \xdef\bp@add@tmp{\unexpanded\expandafter{\bp@add@tmp}%
1525       \unexpanded\expandafter{\opt}},}
1526     \else
1527       \xdef\bp@add@tmp{\unexpanded\expandafter{\bp@add@tmp}%
1528       \unexpanded\expandafter{\approx}},}
1529     \fi
1530   \fi
1531 \fi
1532 \fi
1533 }
1534 \ifx\bp@add@tmp\empty\else
1535   \xdef\bp@add@0{\unexpanded\expandafter{\bp@add@0}%
1536   true/{\unexpanded\expandafter{\bp@add@tmp}}}
1537 \fi
1538 }

```

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

```

1539 \newcommand*{\bp@add}[3]{
1540   \ifcat$\detokenize\expandafter{#1}$
1541     \xdef#1{\unexpanded\expandafter{#1 0+#2}}
1542   \else
1543     \xdef#1{\unexpanded\expandafter{#1+#2}}
1544   \fi
1545   \foreach \y [count=\n] in #3 {
1546     \xdef#1{\unexpanded\expandafter{#1}{\y}}
1547     \xdef\Last@LoopValue{\n}
1548   }
1549   \ifnum\Last@LoopValue=1
1550     \xdef#1{\unexpanded\expandafter{#1}{0}}
1551   \fi
1552 }

```

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

```

1553 \newcommand{\bp@ZPK@plot}[4]{
1554   \edef\bp@list{#4}
1555   \foreach \feature/\values in \bp@list {
1556     \ifx\values\empty\else
1557       \ifnum\pdf@strcmp{\feature}{z}=0
1558         \foreach \z in \values {
1559           \ifx\z\empty\else
1560             \ifnum\pdf@strcmp{#3}{linear}=0
1561               \bp@add{#2}{\PhZeroLin}{\z}
1562               \bp@add{#1}{\MagZeroLin}{\z}
1563             \else
1564               \ifnum\pdf@strcmp{#3}{asymptotic}=0
1565                 \bp@add{#2}{\PhZeroAsymp}{\z}
1566                 \bp@add{#1}{\MagZeroAsymp}{\z}
1567               \else
1568                 \bp@add{#2}{\PhZero}{\z}
1569                 \bp@add{#1}{\MagZero}{\z}
1570               \fi
1571             \fi
1572           \fi
1573         }
1574       \fi
1575       \ifnum\pdf@strcmp{\feature}{p}=0
1576         \foreach \p in \values {

```

```

1577         \ifx\p\empty\else
1578             \ifnum\pdf@strcmp{#3}{linear}=0
1579                 \bp@add{#2}{\PhPoleLin}{\p}
1580                 \bp@add{#1}{\MagPoleLin}{\p}
1581             \else
1582                 \ifnum\pdf@strcmp{#3}{asymptotic}=0
1583                     \bp@add{#2}{\PhPoleAsymp}{\p}
1584                     \bp@add{#1}{\MagPoleAsymp}{\p}
1585                 \else
1586                     \bp@add{#2}{\PhPole}{\p}
1587                     \bp@add{#1}{\MagPole}{\p}
1588                 \fi
1589             \fi
1590         \fi
1591     }
1592 \fi
1593 \ifnum\pdf@strcmp{\feature}{k}=0
1594     \ifnum\pdf@strcmp{#3}{linear}=0
1595         \bp@add{#2}{\PhKLin}{\values}
1596         \bp@add{#1}{\MagKLin}{\values}
1597     \else
1598         \ifnum\pdf@strcmp{#3}{asymptotic}=0
1599             \bp@add{#2}{\PhKAsymp}{\values}
1600             \bp@add{#1}{\MagKAsymp}{\values}
1601         \else
1602             \bp@add{#2}{\PhK}{\values}
1603             \bp@add{#1}{\MagK}{\values}
1604         \fi
1605     \fi
1606 \fi
1607 \ifnum\pdf@strcmp{\feature}{d}=0
1608     \ifnum\pdf@strcmp{#3}{linear}=0
1609         \PackageError {bodeplot}
1610         {Linear approximation for pure delays is not supported.}
1611         {Plot the true Bode plot using `true' instead of `linear'.}
1612     \else
1613         \ifnum\pdf@strcmp{#3}{asymptotic}=0
1614             \PackageError {bodeplot}
1615             {Asymptotic approximation for pure delays is not supported.}
1616             {Plot the true Bode plot using `true' instead of `asymptotic'}.}
1617         \else
1618             \ifdim\values pt < 0pt
1619                 \PackageError {bodeplot} {Delay needs to be a positive number.}
1620             \fi
1621             \bp@add{#2}{\PhDel}{\values}
1622             \bp@add{#1}{\MagDel}{\values}
1623         \fi
1624     \fi
1625 \fi
1626 \fi
1627 }
1628 }

```

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

```

1629 \newcommand{\bp@gnu@plot}[3][1]{
1630     \xdef\bp@gnu@cmd{ gnuplot [raw gnuplot, id=#3, prefix=\bp@prefix]
1631         { set table $meta;
1632           set dummy t;
1633           set logscale x 10;
1634           set xrange [#1*\pgfkeysvalueof{/pgfplots/domain}*#1];
1635           set samples \pgfkeysvalueof{/pgfplots/samples};
1636           plot #2;
1637           set table "\bp@prefix#3.table";

```

```

1638     plot "$meta" using ($1/(\bp@freq@scale)):($2);
1639 };}
1640 }

```

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

```

1641 \NewDocumentCommand{\bp@zpk@new@to@legacy}{ m }{%
1642   \pgfkeys{/bodeplot/zpk/.cd, reset}
1643   \pgfkeys{/bodeplot/zpk/.cd, #1}
1644   \pgfkeysgetvalue{/bodeplot/zpk/@zeros}{\bp@z}
1645   \gdef\bp@z@list{}
1646   \ifx\bp@z@empty\else
1647     \foreach \z in \bp@z {
1648       \bp@parse@complex{\z}
1649       \xdef\bp@z@list{\unexpanded\expandafter{\bp@z@list} \bp@complex,}
1650     }
1651   \fi
1652   \pgfkeysgetvalue{/bodeplot/zpk/@poles}{\bp@p}
1653   \gdef\bp@p@list{}
1654   \ifx\bp@p@empty\else
1655     \foreach \p in \bp@p {
1656       \bp@parse@complex{\p}
1657       \xdef\bp@p@list{\unexpanded\expandafter{\bp@p@list} \bp@complex,}
1658     }
1659   \fi
1660   \pgfkeysgetvalue{/bodeplot/zpk/@gain}{\bp@k}
1661   \pgfkeysgetvalue{/bodeplot/zpk/@delay}{\bp@d}
1662   \xdef\bp@legacy{z/{\bp@z@list},p/{\bp@p@list},k/\bp@k,d/\bp@d}
1663 }

```

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

```

1664 \NewDocumentCommand{\bp@tf@new@to@legacy}{ m }{%
1665   \pgfkeys{/bodeplot/tf/.cd, reset}
1666   \pgfkeys{/bodeplot/tf/.cd, #1}
1667   \pgfkeysgetvalue{/bodeplot/tf/@numerator}{\bp@num}
1668   \pgfkeysgetvalue{/bodeplot/tf/@denominator}{\bp@den}
1669   \pgfkeysgetvalue{/bodeplot/tf/@delay}{\bp@d}
1670   \gdef\bp@num@list{}
1671   \ifx\bp@num@empty\else
1672     \foreach \n in \bp@num {
1673       \xdef\bp@num@list{\unexpanded\expandafter{\bp@num@list} \n,}
1674     }
1675   \fi
1676   \gdef\bp@den@list{}
1677   \ifx\bp@den@empty\else
1678     \foreach \d in \bp@den {
1679       \xdef\bp@den@list{\unexpanded\expandafter{\bp@den@list} \d,}
1680     }
1681   \fi
1682   \xdef\bp@legacy{num/{\bp@num@list},den/{\bp@den@list},d/\bp@d}
1683 }

```

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

```

1684 \newcommand{\bp@TF@plot}[3]{
1685   \gdef\bp@num@re{0}
1686   \gdef\bp@num@im{0}
1687   \gdef\bp@den@re{0}
1688   \gdef\bp@den@im{0}
1689   \gdef\bp@loop@delay{0}
1690   \edef\bp@list{#3}
1691   \foreach \feature/\values in \bp@list {
1692     \ifx\values\empty\else
1693       \ifnum\pdf@strcmp{\feature}{num}=0

```

```

1694     \foreach \numcoeff [count=\numpow] in \values {
1695         \ifx\numcoeff\empty\else
1696             \xdef\bp@num@deg{\numpow}
1697         \fi
1698     }
1699     \foreach \numcoeff [count=\numpow] in \values {
1700         \ifx\numcoeff\empty\else
1701             \pgfmathtruncatemacro{\currentdegree}{\bp@num@deg-\numpow}
1702             \ifnum\currentdegree = 0
1703                 \xdef\bp@num@re{\bp@num@re+\numcoeff}
1704             \else
1705                 \ifodd\currentdegree
1706                     \xdef\bp@num@im{\bp@num@im+(\numcoeff*(\n@pow{-
1707 1}}{(\currentdegree-1)/2})*%
1708                     (\n@pow{t}{\currentdegree}))}
1709                 \else
1710                     \xdef\bp@num@re{\bp@num@re+(\numcoeff*(\n@pow{-
1711 1}}{(\currentdegree)/2})*%
1712                     (\n@pow{t}{\currentdegree}))}
1713                 \fi
1714             \fi
1715         \fi
1716         \ifnum\pdf@strcmp{\feature}{den}=0
1717             \foreach \dencoeff [count=\denpow] in \values {
1718                 \ifx\dencoeff\empty\else
1719                     \xdef\bp@den@deg{\denpow}
1720                 \fi
1721             }
1722             \foreach \dencoeff [count=\denpow] in \values {
1723                 \ifx\dencoeff\empty\else
1724                     \pgfmathtruncatemacro{\currentdegree}{\bp@den@deg-\denpow}
1725                     \ifnum\currentdegree = 0
1726                         \xdef\bp@den@re{\bp@den@re+\dencoeff}
1727                     \else
1728                         \ifodd\currentdegree
1729                             \xdef\bp@den@im{\bp@den@im+(\dencoeff*(\n@pow{-
1730 1}}{(\currentdegree-1)/2})*%
1731                             (\n@pow{t}{\currentdegree}))}
1732                         \else
1733                             \xdef\bp@den@re{\bp@den@re+(\dencoeff*(\n@pow{-
1734 1}}{(\currentdegree)/2})*%
1735                             (\n@pow{t}{\currentdegree}))}
1736                         \fi
1737                     \fi
1738                 \fi
1739                 \ifnum\pdf@strcmp{\feature}{d}=0
1740                     \xdef\bp@loop@delay{\values}
1741                 \fi
1742             \fi
1743         }
1744         \xdef#2{((atan2((\bp@num@im),(\bp@num@re))-atan2((\bp@den@im),%
1745         (\bp@den@re))-\bp@loop@delay*t)*(\bp@ph@scale))}
1746         \xdef#1{(20*log10(sqrt((\n@pow{\bp@num@re}{2})+(\n@pow{\bp@num@im}{2}))) -
1747         %
1748         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.

```

1749 \newcommand{\bp@gnu@unwrap@plot}[3][1]{

```

```

1750 \xdef\bp@gnu@cmd{ gnuplot [raw gnuplot, id=#3, prefix=\bp@prefix]
1751 { set table $meta;
1752   set dummy t;
1753   set logscale x 10;
1754   set trange [#1*\pgfkeysvalueof{/pgfplots/domain}*#1];
1755   set samples \pgfkeysvalueof{/pgfplots/samples};
1756   plot '+' using (t) : ((#2)/(\bp@ph@scale)) smooth unwrap;
1757   set table "\bp@prefix#3.table";
1758   plot "$meta" using ($1/(\bp@freq@scale)) : ($2*\bp@ph@scale);
1759   };}
1760 }

```

**\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.

```

1761 \newcommand{\bp@parse@opt}[1]{
1762   \gdef\bp@mag@axes{}
1763   \gdef\bp@ph@axes{}
1764   \gdef\bp@ph@plot{}
1765   \gdef\bp@mag@plot{}
1766   \gdef\bp@group{}
1767   \gdef\bp@approx{}
1768   \gdef\bp@ph@commands{}
1769   \gdef\bp@mag@commands{}
1770   \gdef\bp@tikz{}
1771   \gdef\bp@user@prefix{}
1772   \foreach \obj/\typ/\opt in {#1} {
1773     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{plot}=0
1774       \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{mag}=0
1775         \xdef\bp@mag@plot{\unexpanded\expandafter{\opt}}
1776       \else
1777         \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{ph}=0
1778           \xdef\bp@ph@plot{\unexpanded\expandafter{\opt}}
1779         \else
1780           \xdef\bp@mag@plot{\unexpanded\expandafter{\opt}}
1781           \xdef\bp@ph@plot{\unexpanded\expandafter{\opt}}
1782         \fi
1783       \fi
1784     \else
1785       \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{axes}=0
1786         \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{mag}=0
1787           \xdef\bp@mag@axes{\unexpanded\expandafter{\opt}}
1788         \else
1789           \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{ph}=0
1790             \xdef\bp@ph@axes{\unexpanded\expandafter{\opt}}
1791           \else
1792             \xdef\bp@mag@axes{\unexpanded\expandafter{\opt}}
1793             \xdef\bp@ph@axes{\unexpanded\expandafter{\opt}}
1794           \fi
1795         \fi
1796       \else
1797         \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{group}=0
1798           \xdef\bp@group{\unexpanded\expandafter{\opt}}
1799         \else
1800           \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{approx}=0
1801             \xdef\bp@approx{\unexpanded\expandafter{\opt}}
1802           \else
1803             \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{commands}=0

```



```

1804         \ifnum\pdf@strcmp{\unexpanded\expandafter{\typ}}{ph}=0
1805         \xdef\bp@ph@commands{\unexpanded\expandafter{\opt}}
1806     \else
1807         \xdef\bp@mag@commands{\unexpanded\expandafter{\opt}}
1808     \fi
1809 \else
1810     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{tikz}=0
1811     \xdef\bp@tikz{\unexpanded\expandafter{\opt}}
1812 \else
1813     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{prefix}=0
1814     \xdef\bp@user@prefix{\unexpanded\expandafter{\opt}}
1815 \else
1816     \xdef\bp@user@prefix{}
1817     \xdef\bp@mag@plot{\unexpanded\expandafter{\bp@mag@plot},
1818     \unexpanded\expandafter{\obj}}
1819     \xdef\bp@ph@plot{\unexpanded\expandafter{\bp@ph@plot},
1820     \unexpanded\expandafter{\obj}}
1821 \fi
1822 \fi
1823 \fi
1824 \fi
1825 \fi
1826 \fi
1827 \fi
1828 }
1829 }

```

`\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.

```

1830 \newcommand{\bp@parse@env@opt}[1]{
1831   \gdef\bp@axes{}
1832   \gdef\bp@tikz{}
1833   \gdef\bp@user@prefix{}
1834   \foreach \obj/\opt in {#1} {
1835     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{axes}=0
1836     \xdef\bp@axes{\unexpanded\expandafter{\opt}}
1837   \else
1838     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{tikz}=0
1839     \xdef\bp@tikz{\unexpanded\expandafter{\opt}}
1840   \else
1841     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{prefix}=0
1842     \xdef\bp@user@prefix{\unexpanded\expandafter{\opt}}
1843   \else
1844     \xdef\bp@user@prefix{}
1845     \xdef\bp@axes{\unexpanded\expandafter{\bp@axes},
1846     \unexpanded\expandafter{\obj}}
1847   \fi
1848 \fi
1849 \fi
1850 }
1851 }

```

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

```

1852 \newcommand{\bp@parse@add@Bode@opt}[1]{
1853   \gdef\bp@plot{}
1854   \gdef\bp@approx{}
1855   \foreach \opt in {#1} {
1856     \ifx\opt@empty\else
1857     \ifnum\pdf@strcmp{\unexpanded\expandafter{\opt}}{linear}=0

```

```

1858     \xdef\bp@approx{\unexpanded\expandafter{\opt}}
1859   \else
1860     \ifnum\pdf@strcmp{\unexpanded\expandafter{\opt}}{asymptotic}=0
1861     \xdef\bp@approx{\unexpanded\expandafter{\opt}}
1862   \else
1863     \xdef\bp@plot{\unexpanded\expandafter{\bp@plot}%
1864     \unexpanded\expandafter{\opt}},}
1865   \fi
1866 \fi
1867 \fi
1868 }
1869 }

```

## 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  $\omega$ . Logarithmic sampling makes it worse, perhaps inverse logarithmic sampling will help, pull requests to fix that are welcome!

```

1870 \NewDocumentCommand{\NyquistZPK}{ O{} m G{} G{} }{%
1871   \pgfutil@ifempty{#3}{%
1872     \pgfkeys{/bodeplot/nyquist/.cd, reset}
1873     \pgfkeys{/bodeplot/nyquist/.cd, #1}
1874     \pgfkeysgetvalue{/bodeplot/nyquist/@axes}{\bp@axes}
1875     \pgfkeysgetvalue{/bodeplot/nyquist/@commands}{\bp@commands}
1876     \pgfkeysgetvalue{/bodeplot/nyquist/@tikz}{\bp@tikz}
1877     \pgfkeysgetvalue{/bodeplot/nyquist/@prefix}{\bp@user@prefix}
1878     \bp@zpk@new@to@legacy{#2}
1879   }{%
1880     \bp@parse@N@opt{#1}
1881     \edef\bp@legacy{#2}
1882     \edef\bp@domain@start{#3}
1883     \edef\bp@domain@end{#4}
1884   }%
1885   \gdef\bp@mag{}\gdef\bp@ph{}%
1886   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1887   [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
1888   \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1889   \edef\bp@cmd{\noexpand\begin{axis}[
1890     bp@style,
1891     domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
1892     height=5cm,
1893     xlabel={$\Re$},
1894     ylabel={$\Im$},
1895     samples=500,
1896     \unexpanded\expandafter{\bp@axes}
1897   ]}
1898   \bp@cmd
1899   \addplot [only marks,mark=+,thick,red] (-1 , 0);
1900   \edef\bp@cmd{\noexpand\addplot
1901   [variable=t, thick, trig format plots=rad,
1902   \unexpanded\expandafter{\bp@plot}]}
1903   \if@pgfarg
1904     \bp@cmd
1905     ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
1906       {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );

```

```

1907     \bp@commands
1908   \else
1909     \stepcounter{bp@gnuplot@id}
1910     \bp@cmd gnuplot [parametric, bp@gnu@prefix] {
1911       \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
1912       \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
1913     };
1914     \bp@commands
1915   \fi
1916 \end{axis}
1917 \end{tikzpicture}
1918 }

```

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

```

1919 \AtBeginDocument{%
1920   \if@babel
1921     \let\Orig@NyquistZPK\NyquistZPK
1922     \renewcommand{\NyquistZPK}{%
1923       \expandafter\shorthandoff\expandafter{\bp@short@list}
1924       \NyquistZPK@Shorthandoff
1925     }
1926     \NewDocumentCommand{\NyquistZPK@Shorthandoff}{ 0{} m G{} G{} }{%
1927       \Orig@NyquistZPK[#1]{#2}{#3}{#4}
1928       \expandafter\shorthandon\expandafter{\bp@short@list}
1929     }
1930   \fi
1931 }

```

**\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**.

```

1932 \NewDocumentCommand{\NyquistTF}{ 0{} m G{} G{} }{%
1933   \pgfutil@ifempty{#3}{%
1934     \pgfkeys{/bodeplot/nyquist/.cd, reset}
1935     \pgfkeys{/bodeplot/nyquist/.cd, #1}
1936     \pgfkeysgetvalue{/bodeplot/nyquist/@axes}{\bp@axes}
1937     \pgfkeysgetvalue{/bodeplot/nyquist/@commands}{\bp@commands}
1938     \pgfkeysgetvalue{/bodeplot/nyquist/@tikz}{\bp@tikz}
1939     \pgfkeysgetvalue{/bodeplot/nyquist/@prefix}{\bp@user@prefix}
1940     \bp@tf@new@to@legacy{#2}
1941   }{%
1942     \bp@parse@N@opt{#1}
1943     \edef\bp@legacy{#2}
1944     \edef\bp@domain@start{#3}
1945     \edef\bp@domain@end{#4}
1946   }%
1947   \gdef\bp@mag{}\gdef\bp@ph{}%
1948   \edef\bp@cmd{\noexpand\begin{tikzpicture}
1949     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
1950     \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1951     \edef\bp@cmd{\noexpand\begin{axis}[
1952       bp@style,
1953       domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
1954       height=5cm,
1955       xlabel={\Re$},
1956       ylabel={\Im$},
1957       samples=500,
1958       \unexpanded\expandafter{\bp@axes}
1959     ]}
1960     \bp@cmd
1961     \addplot [only marks, mark=+, thick, red] (-1 , 0);
1962     \edef\bp@cmd{\noexpand\addplot
1963       [variable=t, thick, trig format plots=rad,
1964       \unexpanded\expandafter{\bp@plot}]}
1965     \if@pgfarg

```

```

1966      \bp@cmd
1967      ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
1968      {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
1969      \bp@commands
1970    \else
1971      \stepcounter{bp@gnuplot@id}
1972      \bp@cmd gnuplot [parametric, bp@gnu@prefix] {
1973        \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
1974        \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
1975      };
1976      \bp@commands
1977    \fi
1978  \end{axis}
1979 \end{tikzpicture}
1980 }

```

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

```

1981 \AtBeginDocument{%
1982   \if@babel
1983   \let\Orig@NyquistTF\NyquistTF
1984   \renewcommand{\NyquistTF}{%
1985     \expandafter\shorthandoff\expandafter{\bp@short@list}
1986     \NyquistTF@Shorthandoff
1987   }
1988   \NewDocumentCommand{\NyquistTF@Shorthandoff}{ 0{ } m G{ } G{ } }{%
1989     \Orig@NyquistTF[#1]{#2}{#3}{#4}
1990     \expandafter\shorthandon\expandafter{\bp@short@list}
1991   }
1992   \fi
1993 }

```

**\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.

```

1994 \NewDocumentCommand{\addNyquistZPKPlot}{ 0{ } m }{%
1995   \bp@contains@equal{#2}{\bp@zpk@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
1996   \gdef\bp@mag{\gdef\bp@ph}%
1997   \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
1998   \if@pgfarg
1999     \edef\bp@cmd{\noexpand\addplot
2000       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
2001       variable=t, thick, trig format plots=rad, #1]}%
2002     \bp@cmd ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
2003               {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
2004   \else
2005     \stepcounter{bp@gnuplot@id}%
2006     \edef\bp@cmd{\noexpand\addplot
2007       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale, thick, #1]
2008       \bp@cmd gnuplot [parametric, bp@gnu@prefix] {%
2009         \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
2010         \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
2011       };
2012     \fi
2013 }

```

**\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.

```

2014 \NewDocumentCommand{\addNyquistTFPlot}{ 0{ } m }{%
2015   \bp@contains@equal{#2}{\bp@tf@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
2016   \gdef\bp@mag{\gdef\bp@ph}%
2017   \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2018   \if@pgfarg
2019     \edef\bp@cmd{\noexpand\addplot
2020       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,

```

```

2021     variable=t, thick, trig format plots=rad, #1]}%
2022     \bp@cmd ( {\n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale))},
2023               {\n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))} );
2024 \else
2025     \stepcounter{bp@gnuplot@id}%
2026     \edef\bp@cmd{\noexpand\addplot
2027       [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale, thick, #1]
2028       \bp@cmd gnuplot [parametric, bp@gnu@prefix] {%
2029         \n@pow{10}{((\bp@mag)/20)}*cos((\bp@ph)/(\bp@ph@scale)),
2030         \n@pow{10}{((\bp@mag)/20)}*sin((\bp@ph)/(\bp@ph@scale))
2031       };
2032 \fi
2033 }

```

**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.

```

2034 \AtBeginDocument{%
2035   \if@babel
2036     \AddToHook{env/NyquistPlot/begin}{%
2037       \expandafter\shorthandoff\expandafter{\bp@short@list}
2038     }
2039     \AddToHook{env/NyquistPlot/end}{%
2040       \expandafter\shorthandon\expandafter{\bp@short@list}
2041     }
2042 \fi
2043 }
2044 \NewDocumentEnvironment{NyquistPlot}{0}{G}{G}{+b}{
2045   \pgfutil@ifempty{#2}{%
2046     \pgfkeys{/bodeplot/env/.cd, reset}
2047     \pgfkeys{/bodeplot/env/.cd, #1}
2048     \pgfkeysgetvalue{/bodeplot/env/@tikz}{\bp@tikz}
2049     \pgfkeysgetvalue{/bodeplot/env/@prefix}{\bp@user@prefix}
2050   }{%
2051     \bp@parse@env@opt{#1}
2052     \edef\bp@domain@start{#2}%
2053     \edef\bp@domain@end{#3}%
2054   }
2055   \edef\bp@cmd{\noexpand\begin{tikzpicture}
2056     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
2057   \edef\bp@cmd{\noexpand\begin{axis}[
2058     bp@style,
2059     height=5cm,
2060     domain=\bp@domain@start:\bp@domain@end,
2061     xlabel={\$Re\$},
2062     ylabel={\$Im\$},
2063     \unexpanded\expandafter{\bp@axes}%
2064   ]}\bp@cmd
2065     \addplot [only marks, mark=+, thick, red] (-1 , 0);
2066     #4
2067   \end{axis}
2068   \end{tikzpicture}
2069 }{}

```

#### 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.

```

2070 \newcommand{\bp@parse@N@opt}[1]{
2071   \gdef\bp@axes{

```

```

2072 \gdef\bp@plot{}
2073 \gdef\bp@commands{}
2074 \gdef\bp@tikz{}
2075 \gdef\bp@user@prefix{}
2076 \gdef\bp@scale{linear}
2077 \foreach \obj/\opt in {#1} {
2078   \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{axes}=0
2079     \xdef\bp@axes{\unexpanded\expandafter{\opt}}
2080   \else
2081     \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{plot}=0
2082       \xdef\bp@plot{\unexpanded\expandafter{\opt}}
2083     \else
2084       \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{commands}=0
2085         \xdef\bp@commands{\unexpanded\expandafter{\opt}}
2086       \else
2087         \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{tikz}=0
2088           \xdef\bp@tikz{\unexpanded\expandafter{\opt}}
2089         \else
2090           \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{scale}=0
2091             \xdef\bp@scale{\unexpanded\expandafter{\opt}}
2092           \else
2093             \ifnum\pdf@strcmp{\unexpanded\expandafter{\obj}}{prefix}=0
2094               \xdef\bp@user@prefix{\unexpanded\expandafter{\opt}}
2095             \else
2096               \xdef\bp@user@prefix{}
2097             \xdef\bp@plot{\unexpanded\expandafter{\bp@plot},
2098               \unexpanded\expandafter{\obj}}
2099             \fi
2100           \fi
2101         \fi
2102       \fi
2103     \fi
2104   \fi
2105 }
2106 }

```

## 4.7 Nichols charts

\NicholsZPK

```

2107 \NewDocumentCommand{\NicholsZPK}{ 0{} m G{} G{} }{%
2108   \pgfutil@ifempty{#3}{%
2109     \pgfkeys{/bodeplot/nichols/.cd, reset}
2110     \pgfkeys{/bodeplot/nichols/.cd, #1}
2111     \pgfkeysgetvalue{/bodeplot/nichols/@axes}{\bp@axes}
2112     \pgfkeysgetvalue{/bodeplot/nichols/@commands}{\bp@commands}
2113     \pgfkeysgetvalue{/bodeplot/nichols/@tikz}{\bp@tikz}
2114     \pgfkeysgetvalue{/bodeplot/nichols/@prefix}{\bp@user@prefix}
2115     \bp@zpk@new@to@legacy{#2}
2116   }{%
2117     \bp@parse@N@opt{#1}
2118     \edef\bp@legacy{#2}
2119     \edef\bp@domain@start{#3}
2120     \edef\bp@domain@end{#4}
2121   }%
2122   \gdef\bp@mag{}\gdef\bp@ph{}%
2123   \edef\bp@cmd{\noexpand\begin{tikzpicture}
2124     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
2125   \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2126     \edef\bp@cmd{\noexpand\begin{axis}[
2127       bp@ph@x@label,
2128       bp@style,
2129       domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
2130       height=5cm,

```

```

2131     ylabel={Gain (dB)},
2132     samples=500,
2133     \unexpanded\expandafter{\bp@axes}
2134 }
2135 \bp@cmd
2136 \edef\bp@cmd{\noexpand\addplot
2137 [variable=t, thick, trig format plots=rad,
2138 \unexpanded\expandafter{\bp@plot}]}
2139 \ifpgfarg
2140     \bp@cmd ( {\bp@ph} , {\bp@mag} );
2141     \bp@commands
2142 \else
2143     \stepcounter{bp@gnuplot@id}
2144     \bp@cmd gnuplot [raw gnuplot, bp@gnu@prefix]
2145     { set table $meta;
2146       set logscale x 10;
2147       set dummy t;
2148       set samples \pgfkeysvalueof{/pgfplots/samples};
2149       set trange [\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale];
2150       plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2151       unset logscale x;
2152       set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2153       plot "$meta" using ($2*\bp@ph@scale):($1);
2154     };
2155     \bp@commands
2156 \fi
2157 \end{axis}
2158 \end{tikzpicture}
2159 }
2160 \AtBeginDocument{%
2161     \if@babel
2162     \let\Orig@NicholsZPK\NicholsZPK
2163     \renewcommand{\NicholsZPK}{%
2164         \expandafter\shorthandoff\expandafter{\bp@short@list}
2165         \NicholsZPK@Shorthandoff
2166     }
2167     \NewDocumentCommand{\NicholsZPK@Shorthandoff}{ 0{} m G{} G{} }{%
2168         \Orig@NicholsZPK[#1]{#2}{#3}{#4}
2169         \expandafter\shorthandon\expandafter{\bp@short@list}
2170     }
2171 \fi
2172 }

```

\NicholsTF

```

2173 \NewDocumentCommand{\NicholsTF}{ 0{} m G{} G{} }{%
2174     \pgfutil@ifempty{#3}{%
2175         \pgfkeys{/bodeplot/nichols/.cd, reset}
2176         \pgfkeys{/bodeplot/nichols/.cd, #1}
2177         \pgfkeysgetvalue{/bodeplot/nichols/@axes}{\bp@axes}
2178         \pgfkeysgetvalue{/bodeplot/nichols/@commands}{\bp@commands}
2179         \pgfkeysgetvalue{/bodeplot/nichols/@tikz}{\bp@tikz}
2180         \pgfkeysgetvalue{/bodeplot/nichols/@prefix}{\bp@user@prefix}
2181         \bp@tf@new@to@legacy{#2}
2182     }{%
2183         \bp@parse@N@opt{#1}
2184         \edef\bp@legacy{#2}
2185         \edef\bp@domain@start{#3}
2186         \edef\bp@domain@end{#4}
2187     }%
2188     \gdef\bp@mag{}\gdef\bp@ph{}%
2189     \edef\bp@cmd{\noexpand\begin{tikzpicture}
2190     [\unexpanded\expandafter{\bp@tikz}]\bp@cmd
2191     \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%

```

```

2192 \edef\bp@cmd{\noexpand\begin{axis}[
2193     bp@ph@x@label,
2194     bp@style,
2195     domain=\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale,
2196     height=5cm,
2197     ylabel={Gain (dB)},
2198     samples=500,
2199     \unexpanded\expandafter{\bp@axes}
2200 ]}
2201 \bp@cmd
2202 \edef\bp@cmd{\noexpand\addplot
2203 [variable=t, thick, trig format plots=rad,
2204 \unexpanded\expandafter{\bp@plot}]]}%
2205 \ifpgfarg
2206     \bp@cmd ( {\n@mod{\bp@ph}{2*pi*\bp@ph@scale}} , {\bp@mag} );
2207     \bp@commands
2208 \else
2209     \stepcounter{bp@gnuplot@id}
2210     \bp@cmd gnuplot [raw gnuplot, bp@gnu@prefix]
2211     { set table $meta1;
2212       set logscale x 10;
2213       set dummy t;
2214       set samples \pgfkeysvalueof{/pgfplots/samples};
2215       set trange [\bp@domain@start*\bp@freq@scale:\bp@domain@end*\bp@freq@scale];
2216       plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2217       unset logscale x;
2218       set table $meta2;
2219       plot "$meta1" using ($1):($2) smooth unwrap;
2220       set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2221       plot "$meta2" using ($2*\bp@ph@scale):($1);
2222     };
2223     \bp@commands
2224 \fi
2225 \end{axis}
2226 \end{tikzpicture}
2227 }
2228 % The following code handles active characters to avoid conflicts with `babel.' \changes{
2229 % generated list of shorthands instead of manually specifying them}\changes{v3.0.4}{2026/05.
2230 % \begin{macrocode}
2231 \AtBeginDocument{
2232     \if@babel
2233     \let\Orig@NicholsTF\NicholsTF
2234     \renewcommand{\NicholsTF}{%
2235     \expandafter\shorthandoff\expandafter{\bp@short@list}
2236     \NicholsTF@Shorthandoff
2237     }
2238     \NewDocumentCommand{\NicholsTF@Shorthandoff}{ 0{} m G{} G{} }{%
2239     \Orig@NicholsTF[#1]{#2}{#3}{#4}
2240     \expandafter\shorthandon\expandafter{\bp@short@list}
2241     }
2242     \AddToHook{env/NicholsChart/begin}{%
2243     \expandafter\shorthandoff\expandafter{\bp@short@list}
2244     }
2245     \AddToHook{env/NicholsChart/end}{%
2246     \expandafter\shorthandon\expandafter{\bp@short@list}
2247     }
2248 \fi

```

#### NicholsChart

```

2249 \NewDocumentEnvironment{NicholsChart}{ 0{} G{} G{} +b }{
2250     \pgfutil@ifempty{#2}{%
2251     \pgfkeys{/bodeplot/env/.cd, reset}

```



```

2252 \pgfkeys{/bodeplot/env/.cd, #1}
2253 \pgfkeysgetvalue{/bodeplot/env/@tikz}{\bp@tikz}
2254 \pgfkeysgetvalue{/bodeplot/env/@prefix}{\bp@user@prefix}
2255 }{%
2256 \bp@parse@env@opt{#1}
2257 \edef\bp@domain@start{#2}
2258 \edef\bp@domain@end{#3}
2259 }
2260 \edef\bp@cmd{\noexpand\begin{tikzpicture}
2261 [\unexpanded\expandafter{\bp@tikz}]]\bp@cmd
2262 \edef\bp@cmd{\noexpand\begin{axis}[
2263 bp@ph@x@label,
2264 bp@style,
2265 domain=\bp@domain@start:\bp@domain@end,
2266 height=5cm,
2267 ylabel={Gain (dB)},
2268 \unexpanded\expandafter{\bp@axes}%
2269 ]]\bp@cmd
2270 #4
2271 \end{axis}
2272 \end{tikzpicture}
2273 }{}

```

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

```

2274 \NewDocumentCommand{\addNicholsZPKChart}{0}{m}{%
2275 \bp@contains@equal{#2}{\bp@zpk@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
2276 \gdef\bp@mag{}\gdef\bp@ph{}%
2277 \bp@ZPK@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2278 \ifpgfarg
2279 \edef\bp@cmd{\noexpand\addplot
2280 [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
2281 variable=t, thick, trig format plots=rad, #1]}%
2282 \bp@cmd ( {\bp@ph} , {\bp@mag} );
2283 \else
2284 \stepcounter{bp@gnuplot@id}%
2285 \addplot [thick, #1] gnuplot [raw gnuplot, bp@gnu@prefix] {%
2286 set table $meta;
2287 set logscale x 10;
2288 set dummy t;
2289 set samples \pgfkeysvalueof{/pgfplots/samples};
2290 set trange [\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale];
2291 plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2292 unset logscale x;
2293 set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2294 plot "$meta" using ($2*\bp@ph@scale):($1);
2295 };
2296 \fi
2297 }

```

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

```

2298 \NewDocumentCommand{\addNicholsTFChart}{0}{m}{%
2299 \bp@contains@equal{#2}{\bp@tf@new@to@legacy{#2}}{\edef\bp@legacy{#2}}%
2300 \gdef\bp@mag{}\gdef\bp@ph{}%
2301 \bp@TF@plot{\bp@mag}{\bp@ph}{\bp@legacy}%
2302 \ifpgfarg
2303 \edef\bp@cmd{\noexpand\addplot
2304 [domain=\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale,
2305 variable=t, thick, trig format plots=rad, #1]}%
2306 \bp@cmd ( {\nmod{\bp@ph}{2*pi*\bp@ph@scale}} , {\bp@mag} );
2307 \else
2308 \stepcounter{bp@gnuplot@id}%

```

```

2309 \addplot [thick, #1] gnuplot [raw gnuplot, bp@gnu@prefix] {%
2310     set table $meta1;
2311     set logscale x 10;
2312     set dummy t;
2313     set samples \pgfkeysvalueof{/pgfplots/samples};
2314     set trange [\bp@freq@scale*\pgfkeysvalueof{/pgfplots/domain}*\bp@freq@scale];
2315     plot '+' using (\bp@mag) : ((\bp@ph)/(\bp@ph@scale));
2316     unset logscale x;
2317     set table $meta2;
2318     plot "$meta1" using ($1):($2) smooth unwrap;
2319     set table "\bp@prefix\arabic{bp@gnuplot@id}.table";
2320     plot "$meta2" using ($2*\bp@ph@scale):($1);
2321 };
2322 \fi
2323 }

```

## 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.

```

2324 \NewDocumentCommand{\PoleZeroMapZPK}{ O{} m }{%
2325     \bp@contains@equal{#2}{%
2326         \pgfkeys{/bodeplot/pzmap/.cd, reset}
2327         \pgfkeys{/bodeplot/pzmap/.cd, #1}
2328         \pgfkeysgetvalue{/bodeplot/pzmap/@axes}{\bp@axes}
2329         \pgfkeysgetvalue{/bodeplot/pzmap/@plot}{\bp@plot}
2330         \pgfkeysgetvalue{/bodeplot/pzmap/@commands}{\bp@commands}
2331         \pgfkeysgetvalue{/bodeplot/pzmap/@tikz}{\bp@tikz}
2332         \pgfkeysgetvalue{/bodeplot/pzmap/@prefix}{\bp@user@prefix}
2333         \pgfkeysgetvalue{/bodeplot/pzmap/@scale}{\bp@scale}
2334         \bp@zpk@new@to@legacy{#2}
2335     }{%
2336         \bp@parse@N@opt{#1}
2337         \edef\bp@legacy{#2}
2338     }%
2339     \ifnum\pdf@strcmp{\bp@scale}{log}=0
2340         \bp@min@real@ZPK{\bp@legacy}
2341         \bp@min@im@ZPK{\bp@legacy}
2342         \pgfkeys{/pgf/fpu=true}
2343         \ifx\bp@has@positive@values\bp@min@false
2344             \xdef\bp@PZZPK@ticksXPos{0}
2345         \else
2346             \pgfmathparse{max(\bp@max@re@pos@pow@10 - \bp@min@re@pow@10 + 1, 1)}
2347             \pgfmathfloatoint{\pgfmathresult}
2348             \xdef\bp@PZZPK@ticksXPos{\pgfmathresult}
2349         \fi
2350         \ifx\bp@has@negative@values\bp@min@false
2351             \xdef\bp@PZZPK@ticksXNeg{0}
2352         \else
2353             \pgfmathparse{max(\bp@max@re@neg@pow@10 - \bp@min@re@pow@10 + 1, 1)}
2354             \pgfmathfloatoint{\pgfmathresult}
2355             \xdef\bp@PZZPK@ticksXNeg{\pgfmathresult}
2356         \fi
2357         \pgfkeys{/pgf/fpu=false}
2358         \def\PoleZeroMapZPK@formatXTick##1{%
2359             \pgfmathtruncatemacro{\PoleZeroMapZPK@tick}{##1}%
2360             \ifnum\PoleZeroMapZPK@tick=0
2361                 $0$
2362             \else

```

```

2363     \pgfmathtruncatemacro{\PoleZeroMapZPK@exp}
2364     {\bp@min@re@pow@10 + abs(\PoleZeroMapZPK@tick) - 1}%
2365     \ifnum\PoleZeroMapZPK@tick>0
2366         $10^{\PoleZeroMapZPK@exp}$%
2367     \else
2368         $-10^{\PoleZeroMapZPK@exp}$%
2369     \fi
2370 \fi
2371 }
2372 \def\PoleZeroMapZPK@formatYTick##1{%
2373     \pgfmathtruncatemacro{\PoleZeroMapZPK@tick}{##1}%
2374     \ifnum\PoleZeroMapZPK@tick=0
2375         $0$
2376     \else
2377         \pgfmathtruncatemacro{\PoleZeroMapZPK@exp}
2378         {\bp@min@im@pow@10 + abs(\PoleZeroMapZPK@tick) - 1}%
2379         \ifnum\PoleZeroMapZPK@tick>0
2380             $10^{\PoleZeroMapZPK@exp}$%
2381         \else
2382             $-10^{\PoleZeroMapZPK@exp}$%
2383         \fi
2384     \fi
2385 }
2386 \def\PoleZeroMapZPK@xticklabel{\PoleZeroMapZPK@formatXTick{\tick}}
2387 \def\PoleZeroMapZPK@yticklabel{\PoleZeroMapZPK@formatYTick{\tick}}
2388 \fi
2389 \edef\bp@cmd{\noexpand\begin{tikzpicture}}
2390 [\unexpanded\expandafter{\bp@tikz}}]
2391 \bp@cmd
2392 \ifnum\pdf@strcmp{\bp@scale}{log}=0
2393     \edef\bp@cmd{\noexpand\begin{axis}[
2394         xlabel={$\Re$},
2395         ylabel={$\Im$},
2396         axis lines=center,
2397         grid=major,
2398         height=6cm,
2399         enlarge x limits=0.2,
2400         enlarge y limits=0.2,
2401         xtick distance=1,
2402         ytick distance=1,
2403         xticklabel=\noexpand\PoleZeroMapZPK@xticklabel,
2404         yticklabel=\noexpand\PoleZeroMapZPK@yticklabel,
2405         x filter/.expression={abs(x) < \bp@min@re@threshold@result ?
2406             0 : (x >= 0 ? (log10(max(min(x, 1e100), 1e-
2407                 100)) - \bp@min@re@pow@10 + 1) :
2408                 (-log10(max(min(-x, 1e100), 1e-100)) + \bp@min@re@pow@10 - 1))},
2409         y filter/.expression={abs(y) < \bp@min@im@threshold@result ?
2410             0 : (y >= 0 ? (log10(max(min(y, 1e100), 1e-
2411                 100)) - \bp@min@im@pow@10 + 1) :
2412                 (-log10(max(min(-y, 1e100), 1e-100)) + \bp@min@im@pow@10 - 1))},
2413         \unexpanded\expandafter{\bp@axes}
2414     ]}
2415 \else
2416     \edef\bp@cmd{\noexpand\begin{axis}[
2417         xlabel={$\Re$},
2418         ylabel={$\Im$},
2419         axis lines=center,
2420         grid=major,
2421         height=6cm,
2422         enlarge x limits=0.2,
2423         enlarge y limits=0.2,
2424         \unexpanded\expandafter{\bp@axes}
2425     ]}

```

```

2424 \fi
2425 \bp@cmd
2426 \foreach \feature/\values in \bp@legacy {
2427 \ifnum\pdf@strcmp{\feature}{z}=0
2428 \foreach \z in \values {
2429 \foreach \y [count=\zcnt] in \z {
2430 \ifnum\zcnt=1
2431 \xdef\bp@zre{\y}
2432 \fi
2433 \ifnum\zcnt=2
2434 \xdef\bp@zim{\y}
2435 \fi
2436 \xdef\bp@Last@Loop@z{\zcnt}
2437 }
2438 \ifnum\bp@Last@Loop@z=1
2439 \xdef\bp@zim{0}
2440 \fi
2441 \edef\bp@cmd{\noexpand\addplot
2442 [only marks, mark=o, mark size=3pt, thick, blue,
2443 \unexpanded\expandafter{\bp@plot}]}
2444 \bp@cmd coordinates {(\bp@zre,\bp@zim)};
2445 }
2446 \fi
2447 \ifnum\pdf@strcmp{\feature}{p}=0
2448 \foreach \p in \values {
2449 \foreach \y [count=\pcnt] in \p {
2450 \ifnum\pcnt=1
2451 \xdef\bp@pre{\y}
2452 \fi
2453 \ifnum\pcnt=2
2454 \xdef\bp@pim{\y}
2455 \fi
2456 \xdef\bp@Last@Loop@p{\pcnt}
2457 }
2458 \ifnum\bp@Last@Loop@p=1
2459 \xdef\bp@pim{0}
2460 \fi
2461 \edef\bp@cmd{\noexpand\addplot
2462 [only marks, mark=x, mark size=3pt, thick, red,
2463 \unexpanded\expandafter{\bp@plot}]}
2464 \bp@cmd coordinates {(\bp@pre,\bp@pim)};
2465 }
2466 \fi
2467 }
2468 \bp@commands
2469 \end{axis}
2470 \end{tikzpicture}
2471 }

```

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

```

2472 \AtBeginDocument{%
2473 \if@babel
2474 \let\Orig@PoleZeroMapZPK\PoleZeroMapZPK
2475 \renewcommand{\PoleZeroMapZPK}{%
2476 \expandafter\shorthandoff\expandafter{\bp@short@list}
2477 \PoleZeroMapZPK@Shorthandoff
2478 }
2479 \newcommand{\PoleZeroMapZPK@Shorthandoff}[2][]{%
2480 \Orig@PoleZeroMapZPK[#1]{#2}
2481 \expandafter\shandon\expandafter{\bp@short@list}
2482 }
2483 \fi
2484 }

```

#### 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`.

```

2485 \newcommand{\bp@min@real@ZPK}[1]{
2486   \gdef\bp@min@re@threshold@result{1000}
2487   \def\@bp@min@false{false}
2488   \gdef\bp@min@threshold@found{false}
2489   \global\let\bp@min@thresh@float\relax
2490   \pgfkeys{/pgf/fpu=true}
2491   \pgfmathparse{0}
2492   \global\let\bp@max@re@float=\pgfmathresult
2493   \pgfmathparse{0}
2494   \global\let\bp@max@re@pos@float=\pgfmathresult
2495   \pgfmathparse{0}
2496   \global\let\bp@max@re@neg@float=\pgfmathresult
2497   \pgfkeys{/pgf/fpu=false}
2498   \gdef\bp@max@re@value{0}
2499   \gdef\bp@has@positive@values{false}
2500   \gdef\bp@has@negative@values{false}
2501   \foreach \feature/\values in {#1} {
2502     \ifnum\pdf@strcmp{\feature}{z}=0
2503       \foreach \z in \values {
2504         \foreach \y [count=\zcnt] in \z {
2505           \ifnum\zcnt=1
2506             \pgfkeys{/pgf/fpu=true}
2507             \pgfmathparse{abs(\y)}
2508             \let\abs@valuefloat=\pgfmathresult
2509             \pgfmathfloattofixed{\abs@valuefloat}
2510             \edef\abs@value{\pgfmathresult}
2511             \pgfkeys{/pgf/fpu=false}
2512             \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2513               \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2514                 \pgfkeys{/pgf/fpu=true}
2515                 \pgfmathparse{\y >= 0 ? 1 : 0}
2516                 \pgfmathfloattoint{\pgfmathresult}
2517                 \pgfkeys{/pgf/fpu=false}
2518                 \ifnum\pgfmathresult=1
2519                   \gdef\bp@has@positive@values{true}
2520                   \pgfkeys{/pgf/fpu=true}
2521                   \pgfmathparse{\abs@valuefloat > \bp@max@re@pos@float ? 1 : 0}
2522                   \pgfmathfloattoint{\pgfmathresult}
2523                   \pgfkeys{/pgf/fpu=false}
2524                   \ifnum\pgfmathresult=1
2525                     \global\let\bp@max@re@pos@float=\abs@valuefloat
2526                   \fi
2527                 \else
2528                   \gdef\bp@has@negative@values{true}
2529                   \pgfkeys{/pgf/fpu=true}
2530                   \pgfmathparse{\abs@valuefloat > \bp@max@re@neg@float ? 1 : 0}
2531                   \pgfmathfloattoint{\pgfmathresult}
2532                   \pgfkeys{/pgf/fpu=false}
2533                   \ifnum\pgfmathresult=1
2534                     \global\let\bp@max@re@neg@float=\abs@valuefloat
2535                   \fi
2536                 \fi
2537                 \pgfkeys{/pgf/fpu=true}
2538                 \pgfmathparse{\abs@valuefloat > \bp@max@re@float ? 1 : 0}
2539                 \pgfmathfloattoint{\pgfmathresult}
2540                 \pgfkeys{/pgf/fpu=false}
2541                 \ifnum\pgfmathresult=1

```

```

2542         \global\let\bp@max@re@float=\abs@valuefloat
2543         \xdef\bp@max@re@value{\abs@value}
2544     \fi
2545     \ifx\bp@min@threshold@found\@bp@min@false
2546         \xdef\bp@min@re@threshold@result{\abs@value}
2547         \global\let\bp@min@thresh@float=\abs@valuefloat
2548         \gdef\bp@min@threshold@found{true}
2549     \else
2550         \pgfkeys{/pgf/fpu=true}
2551         \pgfmathparse{\abs@valuefloat < \bp@min@thresh@float ? 1 : 0}
2552         \pgfmathfloattoint{\pgfmathresult}
2553         \pgfkeys{/pgf/fpu=false}
2554         \ifnum\pgfmathresult=1
2555             \xdef\bp@min@re@threshold@result{\abs@value}
2556             \global\let\bp@min@thresh@float=\abs@valuefloat
2557         \fi
2558     \fi
2559 \fi
2560 \fi
2561 \fi
2562 }
2563 }
2564 \fi
2565 \ifnum\pdf@strcmp{\feature}{p}=0
2566     \foreach \p in \values {
2567         \foreach \y [count=\pcnt] in \p {
2568             \ifnum\pcnt=1
2569                 \pgfkeys{/pgf/fpu=true}
2570                 \pgfmathparse{abs(\y)}
2571                 \let\abs@valuefloat=\pgfmathresult
2572                 \pgfmathfloattofixed{\abs@valuefloat}
2573                 \edef\abs@value{\pgfmathresult}
2574                 \pgfkeys{/pgf/fpu=false}
2575                 \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2576                     \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2577                         \pgfkeys{/pgf/fpu=true}
2578                         \pgfmathparse{\y >= 0 ? 1 : 0}
2579                         \pgfmathfloattoint{\pgfmathresult}
2580                         \pgfkeys{/pgf/fpu=false}
2581                         \ifnum\pgfmathresult=1
2582                             \gdef\bp@has@positive@values{true}
2583                             \pgfkeys{/pgf/fpu=true}
2584                             \pgfmathparse{\abs@valuefloat > \bp@max@re@pos@float ? 1 : 0}
2585                             \pgfmathfloattoint{\pgfmathresult}
2586                             \pgfkeys{/pgf/fpu=false}
2587                             \ifnum\pgfmathresult=1
2588                                 \global\let\bp@max@re@pos@float=\abs@valuefloat
2589                             \fi
2590                         \else
2591                             \gdef\bp@has@negative@values{true}
2592                             \pgfkeys{/pgf/fpu=true}
2593                             \pgfmathparse{\abs@valuefloat > \bp@max@re@neg@float ? 1 : 0}
2594                             \pgfmathfloattoint{\pgfmathresult}
2595                             \pgfkeys{/pgf/fpu=false}
2596                             \ifnum\pgfmathresult=1
2597                                 \global\let\bp@max@re@neg@float=\abs@valuefloat
2598                             \fi
2599                         \fi
2600                     \fi
2601                 \xdef\bp@min@re@threshold@result{\abs@value}
2602                 \global\let\bp@min@thresh@float=\abs@valuefloat
2603                 \gdef\bp@min@threshold@found{true}
2604             \else

```

```

2605         \pgfkeys{/pgf/fpu=true}
2606         \pgfmathparse{\abs@valuefloat < \bp@min@thresh@float ? 1 : 0}
2607         \pgfmathfloattoint{\pgfmathresult}
2608         \pgfkeys{/pgf/fpu=false}
2609         \ifnum\pgfmathresult=1
2610             \xdef\bp@min@re@threshold@result{\abs@value}
2611             \global\let\bp@min@thresh@float=\abs@valuefloat
2612         \fi
2613     \fi
2614 \fi
2615 \fi
2616 \fi
2617 }
2618 }
2619 \fi
2620 }
2621 \ifx\bp@min@threshold@found\@bp@min@false
2622     \gdef\bp@min@re@threshold@result{0.01}
2623 \fi
2624 \xdef\bp@min@threshold@result{\bp@min@re@threshold@result}
2625 \pgfkeys{/pgf/fpu=true}
2626 \pgfmathparse{log10(\bp@min@re@threshold@result)}
2627 \let\log@result=\pgfmathresult
2628 \pgfmathparse{\log@result + 1e-5}
2629 \let\log@adjusted=\pgfmathresult
2630 \pgfmathparse{floor(\log@adjusted)}
2631 \pgfmathfloattofixed{\pgfmathresult}
2632 \xdef\bp@min@re@pow@10{\pgfmathresult}
2633 \xdef\bp@min@pow@10{\bp@min@re@pow@10}
2634 \ifx\bp@has@positive@values\@bp@min@false
2635     \xdef\bp@max@re@pos@pow@10{\bp@min@re@pow@10}
2636 \else
2637     \pgfmathparse{log10(max(\bp@max@re@pos@float,1e-100))}
2638     \let\log@bp@max@re@pos=\pgfmathresult
2639     \pgfmathparse{\log@bp@max@re@pos + 1e-5}
2640     \let\log@bp@max@re@pos@adjusted=\pgfmathresult
2641     \pgfmathparse{ceil(\log@bp@max@re@pos@adjusted)}
2642     \pgfmathfloattoint{\pgfmathresult}
2643     \xdef\bp@max@re@pos@pow@10{\pgfmathresult}
2644 \fi
2645 \ifx\bp@has@negative@values\@bp@min@false
2646     \xdef\bp@max@re@neg@pow@10{\bp@min@re@pow@10}
2647 \else
2648     \pgfmathparse{log10(max(\bp@max@re@neg@float,1e-100))}
2649     \let\log@bp@max@re@neg=\pgfmathresult
2650     \pgfmathparse{\log@bp@max@re@neg + 1e-5}
2651     \let\log@bp@max@re@neg@adjusted=\pgfmathresult
2652     \pgfmathparse{ceil(\log@bp@max@re@neg@adjusted)}
2653     \pgfmathfloattoint{\pgfmathresult}
2654     \xdef\bp@max@re@neg@pow@10{\pgfmathresult}
2655 \fi
2656 \pgfmathparse{max(\bp@max@re@pos@float > 0 ?
2657 \bp@max@re@pos@float : 0, \bp@max@re@neg@float > 0 ? \bp@max@re@neg@float : 0)}
2658 \let\bp@max@re@valuefloat=\pgfmathresult
2659 \pgfmathparse{\bp@max@re@valuefloat > 0 ?
2660 \bp@max@re@valuefloat : \bp@min@re@threshold@result}
2661 \let\bp@max@re@valuefloat=\pgfmathresult
2662 \pgfmathparse{log10(max(\bp@max@re@valuefloat,1e-100))}
2663 \let\log@bp@max@re=\pgfmathresult
2664 \pgfmathparse{\log@bp@max@re + 1e-5}
2665 \let\log@bp@max@re@adjusted=\pgfmathresult
2666 \pgfmathparse{ceil(\log@bp@max@re@adjusted)}
2667 \pgfmathfloattoint{\pgfmathresult}

```

```

2668 \xdef\bp@max@re@pow@10{\pgfmathresult}
2669 \pgfkeys{/pgf/fpu=false}
2670 }

```

`\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`.

```

2671 \newcommand{\bp@min@im@ZPK}[1]{
2672   \gdef\bp@min@im@threshold@result{1000}
2673   \def\@bp@min@false{false}
2674   \gdef\bp@min@im@threshold@found{false}
2675   \global\let\bp@min@im@thresh@float\relax
2676   \pgfkeys{/pgf/fpu=true}
2677   \pgfmathparse{0}
2678   \global\let\bp@max@im@float=\pgfmathresult
2679   \pgfkeys{/pgf/fpu=false}
2680   \gdef\bp@max@im@value{0}
2681   \foreach \feature/\values in {#1} {
2682     \ifnum\pdf@strcmp{\feature}{z}=0
2683       \foreach \z in \values {
2684         \foreach \y [count=\zcnt] in \z {
2685           \ifnum\zcnt=2
2686             \pgfkeys{/pgf/fpu=true}
2687             \pgfmathparse{abs(\y)}
2688             \let\abs@valuefloat=\pgfmathresult
2689             \pgfmathfloattofixed{\abs@valuefloat}
2690             \edef\abs@value{\pgfmathresult}
2691             \pgfkeys{/pgf/fpu=false}
2692             \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2693               \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2694                 \pgfkeys{/pgf/fpu=true}
2695                 \pgfmathparse{\abs@valuefloat > \bp@max@im@float ? 1 : 0}
2696                 \pgfmathfloattoint{\pgfmathresult}
2697                 \pgfkeys{/pgf/fpu=false}
2698                 \ifnum\pgfmathresult=1
2699                   \global\let\bp@max@im@float=\abs@valuefloat
2700                   \xdef\bp@max@im@value{\abs@value}
2701                 \fi
2702                 \ifx\bp@min@im@threshold@found\@bp@min@false
2703                   \xdef\bp@min@im@threshold@result{\abs@value}
2704                   \global\let\bp@min@im@thresh@float=\abs@valuefloat
2705                   \gdef\bp@min@im@threshold@found{true}
2706                 \else
2707                   \pgfkeys{/pgf/fpu=true}
2708                   \pgfmathparse{\abs@valuefloat < \bp@min@im@thresh@float ?
2709                     1 : 0}
2710                   \pgfmathfloattoint{\pgfmathresult}
2711                   \pgfkeys{/pgf/fpu=false}
2712                   \ifnum\pgfmathresult=1
2713                     \xdef\bp@min@im@threshold@result{\abs@value}
2714                     \global\let\bp@min@im@thresh@float=\abs@valuefloat
2715                   \fi
2716                 \fi
2717               \fi
2718             \fi
2719           }
2720         }
2721       }
2722     \fi
2723     \ifnum\pdf@strcmp{\feature}{p}=0
2724       \foreach \p in \values {

```



```

2725 \foreach \y [count=\pcnt] in \p {
2726   \ifnum\pcnt=2
2727     \pgfkeys{/pgf/fpu=true}
2728     \pgfmathparse{abs(\y)}
2729     \let\abs@valuefloat=\pgfmathresult
2730     \pgfmathfloattofixed{\abs@valuefloat}
2731     \edef\abs@value{\pgfmathresult}
2732     \pgfkeys{/pgf/fpu=false}
2733     \ifnum\pdf@strcmp{\abs@value}{0}=0\else
2734       \ifnum\pdf@strcmp{\abs@value}{0.0}=0\else
2735         \ifx\bp@min@im@threshold@found\@bp@min@false
2736           \xdef\bp@min@im@threshold@result{\abs@value}
2737           \global\let\bp@min@im@thresh@float=\abs@valuefloat
2738           \gdef\bp@min@im@threshold@found{true}
2739         \else
2740           \pgfkeys{/pgf/fpu=true}
2741           \pgfmathparse{\abs@valuefloat < \bp@min@im@thresh@float ?
2742             1 : 0}
2743           \pgfmathfloattoint{\pgfmathresult}
2744           \pgfkeys{/pgf/fpu=false}
2745           \ifnum\pgfmathresult=1
2746             \xdef\bp@min@im@threshold@result{\abs@value}
2747             \global\let\bp@min@im@thresh@float=\abs@valuefloat
2748           \fi
2749         \fi
2750       \fi
2751     \fi
2752   \fi
2753 }
2754 }
2755 \fi
2756 }
2757 \ifx\bp@min@im@threshold@found\@bp@min@false
2758   \gdef\bp@min@im@threshold@result{0.01}
2759 \fi
2760 \pgfkeys{/pgf/fpu=true}
2761 \pgfmathparse{log10(\bp@min@im@threshold@result)}
2762 \let\log@result=\pgfmathresult
2763 \pgfmathparse{\log@result + 1e-5}
2764 \let\log@adjusted=\pgfmathresult
2765 \pgfmathparse{floor(\log@adjusted)}
2766 \pgfmathfloattofixed{\pgfmathresult}
2767 \xdef\bp@min@im@pow@10{\pgfmathresult}
2768 \xdef\bp@min@im@pow@10{\bp@min@im@pow@10}
2769 \pgfmathparse{\bp@max@im@float > 0 ?
2770 \bp@max@im@float : \bp@min@im@threshold@result}
2771 \let\bp@max@im@valuefloat=\pgfmathresult
2772 \pgfmathparse{log10(max(\bp@max@im@valuefloat,1e-100))}
2773 \let\log@bp@max@im=\pgfmathresult
2774 \pgfmathparse{\log@bp@max@im + 1e-5}
2775 \let\log@bp@max@im@adjusted=\pgfmathresult
2776 \pgfmathparse{ceil(\log@bp@max@im@adjusted)}
2777 \pgfmathfloattoint{\pgfmathresult}
2778 \xdef\bp@max@im@pow@10{\pgfmathresult}
2779 \pgfkeys{/pgf/fpu=false}
2780 }

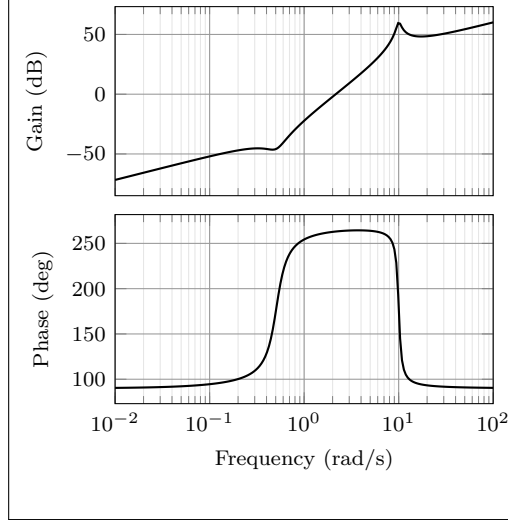
```

## 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)

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

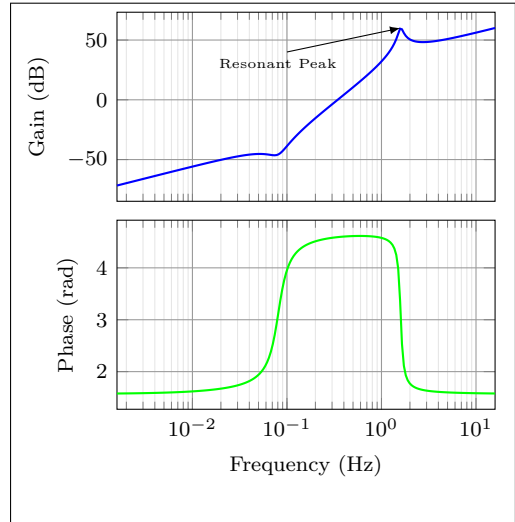
Bode plot in ZPK format



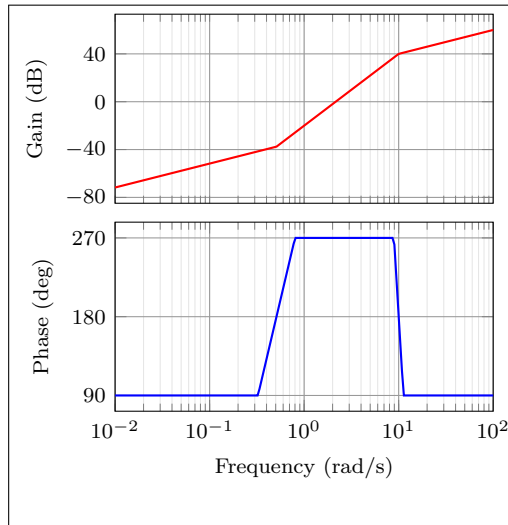
```
\BodeZPK{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
{0.01}
{100}
```

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)}
```



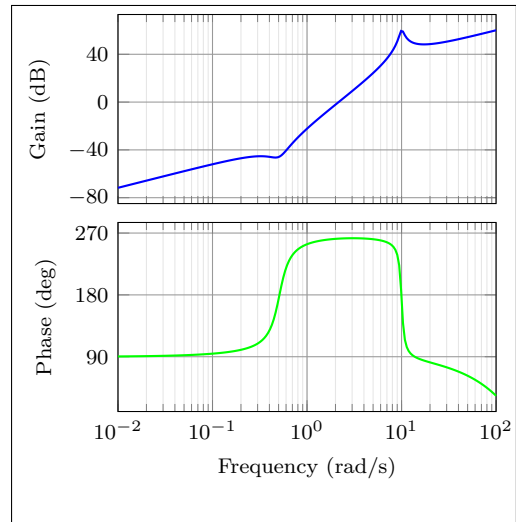
## Linear approximation with customization



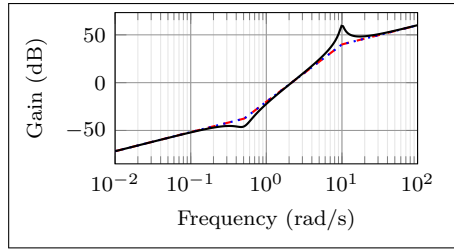
```
\BodeZPK[%
  plot/mag/{red,thick},
  plot/ph/{blue,thick},
  axes/mag/{ytick distance=40},
  axes/ph/{ytick distance=90},
  approx/linear%
]{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
{0.01}
{100}
```

## Plot with delay and customization

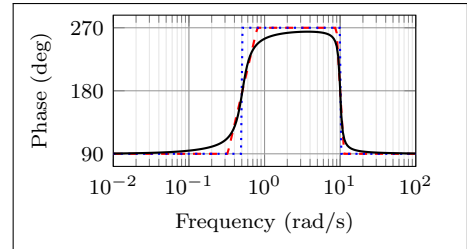
```
\BodeZPK[%
  plot/mag/{blue,thick},
  plot/ph/{green,thick},
  axes/mag/{ytick distance=40},
  axes/ph/{ytick distance=90%
]{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10,
  d/0.01%
}
{0.01}
{100}
```



## Individual gain and phase plots with more customization



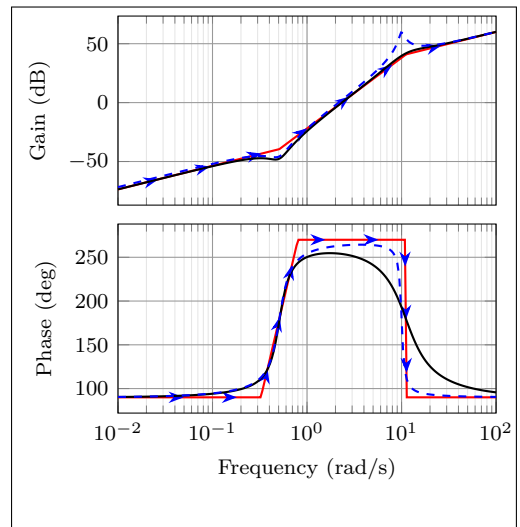
```
\begin{BodeMagPlot}{%
  axes/{height=2cm,
  width=4cm}%
}
{0.01}
{100}
\addBodeZPKPlots[%
  true/{black,thick},
  linear/{red,dashed,thick},
  asymptotic/{blue,dotted,thick}%
]
{magnitude}
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
}\end{BodeMagPlot}
```



```
\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}
```

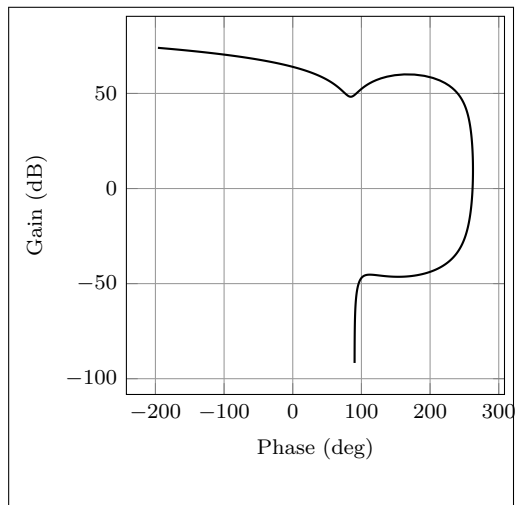
Multiple transfer functions in a single Bode plot using the **BodePlot** environment and the **\addBodePlot** macro introduced in v2.1.

```
\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

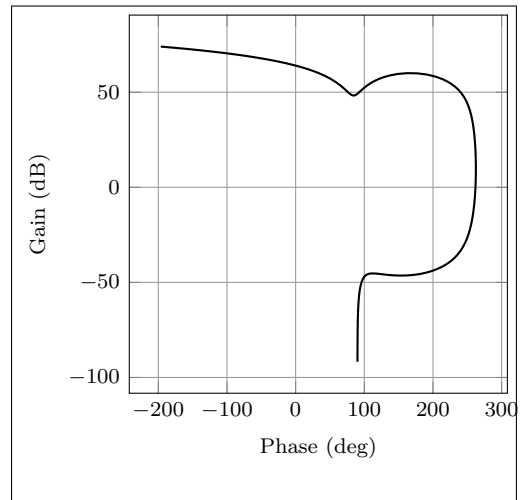


```
\NicholsZPK[samples=1000]
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10,
  d/0.01%
}
{0.001}
{500}
```

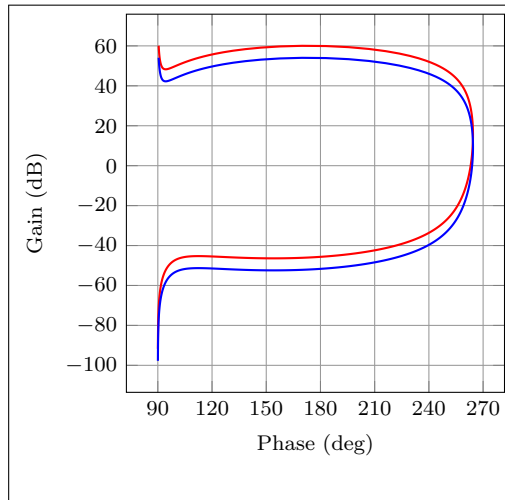
---

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}
```



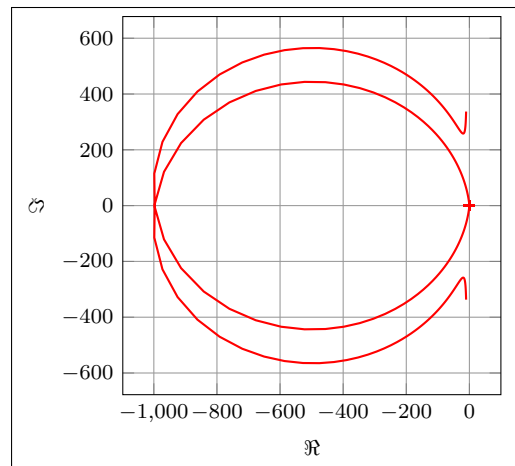
## Multiple Nichols charts with customization



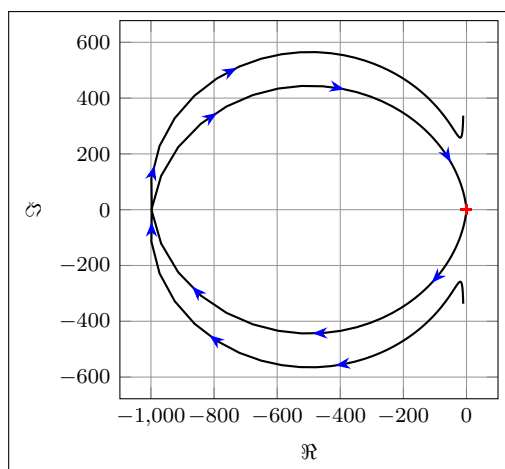
```
\begin{NicholsChart}{%
  ytick distance=20,
  xtick distance=30
}
{0.001}
{100}
\addNicholsZPKChart [red,samples=1000] {%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
\addNicholsZPKChart [blue,samples=1000] {%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
 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



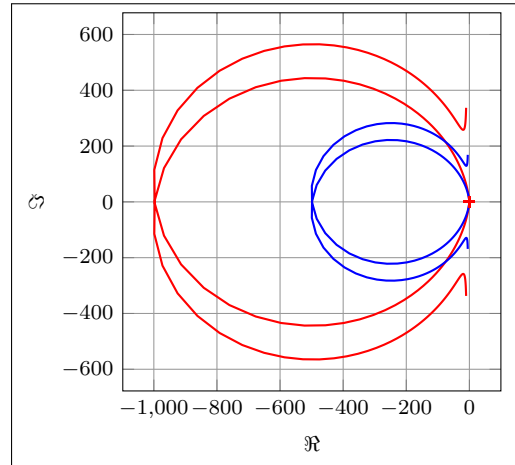
## Nyquist plot in TF format with arrows



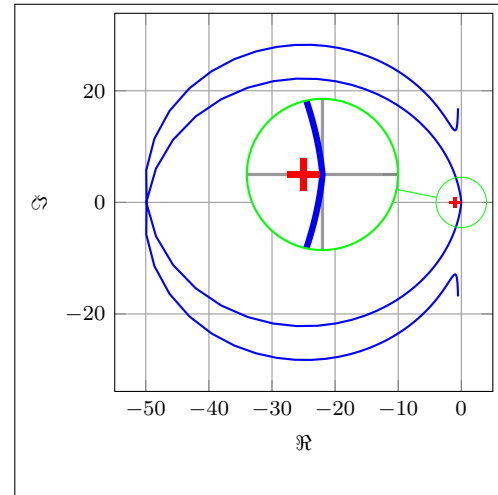
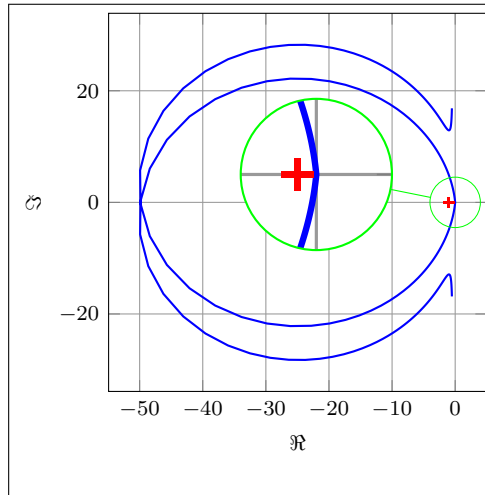
```
\NyquistTF[%
  plot/{%
    samples=1000,
    postaction=decorate,
    decoration={%
      markings,
      mark=between positions 0.1 and 0.9 step 5em with {%
        \arrow{Stealth [length=2mm, blue]}
      }
    }
  }%
]
{%
  num/{10,2,2.6,0},
  den/{1,1,100.25}%
}
{-30}
{30}
```

## 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

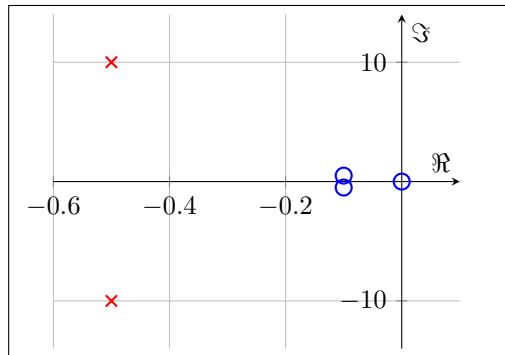


```
\begin{NyquistPlot}{%
tikz/{
spy using outlines={%
circle,
magnification=3,
connect spies,
size=2cm
}
}%
}
\addNyquistZPKPlot [blue,samples=1000] {%
z/{0,{-0.1,-0.5},{-0.1,0.5}},
p/{{-0.5,-10},{-0.5,10}},
k/0.5%
}
\coordinate (spyon) at (axis cs:0,0);
\coordinate (spyat) at (axis cs:-22,5);
\spy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

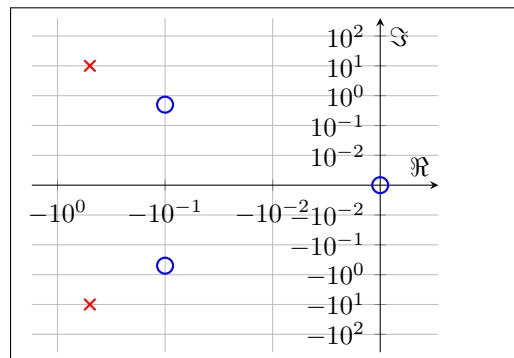


```
\PoleZeroMapZPK
{%
  z/{0,{-0.1,-0.5},{-0.1,0.5}},
  p/{{-0.5,-10},{-0.5,10}},
  k/10%
}
```

---

### 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%
}
```





# Index

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.

<b>Symbols</b>	2530, 2534, 2538,	222, 224, 1161,
<code>\bodeplot</code> ..... <u>230</u>	2542, 2547, 2551,	1200, 1831, 1836,
<code>\bodeplot/add</code> ..... <u>556</u>	2556, 2571, 2572,	1845, 1874, 1896,
<code>\bodeplot/combinedenv</code>	2584, 2588, 2593,	1936, 1958, 2063,
..... <u>383</u>	2597, 2602, 2606,	2071, 2079, 2111,
<code>\bodeplot/env</code> ..... <u>135</u>	2611, 2688, 2689,	2133, 2177, 2199,
<code>\bodeplot/nichols</code> .. <u>689</u>	2695, 2699, 2704,	2268, 2328, 2411, 2422
<code>\bodeplot/nyquist</code> .. <u>636</u>	2708, 2714, 2729,	<code>\bp@checkfull</code> .....
<code>\bodeplot/pzmap</code> ... <u>742</u>	2730, 2737, 2741, 2747	... 208, 210, 353,
<code>\bodeplot/tf</code> ..... <u>541</u>	<code>\addBodeComponentPlot</code>	355, 493, 495, 614,
<code>\bodeplot/zpk</code> ..... <u>523</u>	..... <u>1115</u>	616, 667, 669, 720, 722
<code>\@babelfalse</code> ..... <u>54</u>	<code>\addBodePlot</code> ..... <u>1207</u>	<code>\bp@checkshort</code> .....
<code>\@babeltrue</code> ..... <u>57</u>	<code>\addBodeTFPlot</code> .... <u>1075</u>	... 209, 210, 354,
<code>\@bp@min@false</code> .....	<code>\addBodeZPKPlots</code> .. <u>1040</u>	355, 494, 495, 615,
..... 2343, 2350,	<code>\addNicholsTFChart</code> . <u>2298</u>	616, 668, 669, 721, 722
2487, 2545, 2600,	<code>\addNicholsZPKChart</code> <u>2274</u>	<code>\bp@cmd</code> . 982, 984, 988,
2621, 2634, 2645,	<code>\addNyquistTFPlot</code> .. <u>2014</u>	998, 1060, 1063,
2673, 2702, 2735, 2757	<code>\addNyquistZPKPlot</code> . <u>1994</u>	1068, 1070, 1090,
<code>\@declutterargfalse</code> .. <u>5</u>		1094, 1096, 1099,
<code>\@declutterargtrue</code> ... <u>7</u>	<b>B</b>	1104, 1108, 1111,
<code>\@hzargfalse</code> ..... <u>13</u>	<code>BodeMagPlot</code> (env.) .. <u>1168</u>	1117, 1120, 1124,
<code>\@hzargtrue</code> ..... <u>15</u>	<code>BodePhPlot</code> (env.) ... <u>1129</u>	1126, 1150, 1152,
<code>\@pgfargfalse</code> ..... <u>1</u>	<code>BodePlot</code> (env.) ..... <u>1266</u>	1153, 1163, 1189,
<code>\@pgfargtrue</code> ..... <u>3</u>	<code>\BodeTF</code> ..... <u>953</u>	1191, 1192, 1202,
<code>\@radargfalse</code> ..... <u>9</u>	<code>\BodeTF@Shorthandoff</code>	1283, 1285, 1286,
<code>\@radargtrue</code> ..... <u>11</u>	..... 1032, 1034	1296, 1886, 1887,
<code>\__bp_collect_item:n</code>	<code>\BodeZPK</code> ..... <u>865</u>	1889, 1898, 1900,
..... 1459, 1477	<code>\BodeZPK@Shorthandoff</code>	1904, 1910, 1948,
<code>\__bp_find_separator:</code>	..... 945, 947	1949, 1951, 1960,
..... 1360, 1450	<code>\bool_if:NTF</code> ..... 1362	1962, 1966, 1972,
<code>\__bp_parse_complex_with_i:</code>	<code>\bool_lazy_or:nnT</code> .. 1434	1999, 2002, 2006,
..... 1337, 1354	<code>\bool_new:N</code> ..... 1326	2008, 2019, 2022,
<code>\__bp_parse_pure_imaginary:</code>	<code>\bool_set_false:N</code> .. 1452	2026, 2028, 2055,
..... 1364, 1368	<code>\bool_set_true:N</code> .. 1438	2056, 2057, 2064,
<code>\__bp_parse_real_only:</code>	<code>\bp@add</code> ... 1539, 1561,	2123, 2124, 2126,
..... 1338, 1346	1562, 1565, 1566,	2135, 2136, 2140,
<code>\__bp_process_item:n</code>	1568, 1569, 1579,	2144, 2189, 2190,
..... 1422, 1455	1580, 1583, 1584,	2192, 2201, 2202,
<code>\__bp_set_signed_value:NN</code>	1586, 1587, 1595,	2206, 2210, 2260,
..... 1371, 1375	1596, 1599, 1600,	2261, 2262, 2269,
<code>\__bp_split_at_separator:</code>	1602, 1603, 1621, 1622	2279, 2282, 2303,
..... 1363, 1471	<code>\bp@add@0</code> ... 560, 565,	2306, 2389, 2391,
<code>\__bp_unwrap_braces:N</code>	566, 568, 571, 572,	2393, 2414, 2425,
1348, 1387, 1398,	574, 581, 582, 584,	2441, 2444, 2461, 2464
1403, 1411, 1479, 1480	587, 588, 590, 597,	<code>\bp@commands</code> .....
	598, 600, 603, 604,	1875, 1907, 1914,
<b>A</b>	606, 627, 628, 630,	1937, 1969, 1976,
<code>\abs@value</code> .....	1045, 1050, 1213,	2073, 2085, 2112,
2510, 2512, 2513,	1511, 1516, 1520, 1535	2141, 2155, 2178,
2543, 2546, 2555,	<code>\bp@add@tmp</code> .... 1512,	2207, 2223, 2330, 2468
2573, 2575, 2576,	1524, 1527, 1534, 1536	<code>\bp@complex</code> .... 1314,
2601, 2610, 2690,	<code>\bp@approx</code> ..... 872,	1342, 1649, 1657
2692, 2693, 2700,	900, 960, 1220,	<code>\bp@contains@equal</code> ..
2703, 2713, 2731,	1227, 1271, 1767,	..... 1041,
2733, 2734, 2736, 2746	1801, 1854, 1858, 1861	1076, 1503, 1995,
<code>\abs@valuefloat</code> 2508,	<code>\bp@axes</code> ..... 154,	2015, 2275, 2299, 2325
2509, 2521, 2525,	157, 158, 160, 221,	<code>\bp@contains@num</code> ...

..... 1214, 1506	\bp@gnu@cmd . 929, 934,	913, 957, 1000,
\bp@d 1661, 1662, 1669, 1682	1016, 1021, 1070,	1300, 1762, 1787, 1792
\bp@den .. 1668, 1677, 1678	1108, 1111, 1126,	\bp@mag@cmd .....
\bp@den@deg .. 1719, 1724	1252, 1263, 1630, 1750	. 999, 1008, 1016,
\bp@den@im ..... 1688,	\bp@gnu@plot 928, 933,	1239, 1251, 1281, 1305
1729, 1744, 1747	1015, 1067, 1110,	\bp@mag@commands ...
\bp@den@list .....	1125, 1250, 1256, 1629	..... 873, 922,
... 1676, 1679, 1682	\bp@gnu@prefix ..... 1	930, 961, 1009,
\bp@den@re ..... 1687,	\bp@gnu@unwrap@plot .	1017, 1272, 1769, 1807
1726, 1732, 1745, 1747	..... 1020,	\bp@mag@plot ... 254,
\bp@domain@end .....	1107, 1259, 1749	258, 259, 261, 271,
..... 153, 171,	\bp@gnuplot@id ..... 1	272, 274, 366, 367,
243, 316, 394, 456,	\bp@group .....	369, 915, 1002,
646, 661, 699, 714,	871, 909, 959, 996,	1765, 1775, 1780, 1817
892, 904, 905, 980,	1270, 1294, 1766, 1798	\bp@max@im@float 2678,
991, 992, 1148,	\bp@has@negative@values	2695, 2699, 2769, 2770
1158, 1159, 1187,	..... 2350,	\bp@max@im@pow@ ... 2778
1196, 1197, 1279,	2500, 2528, 2591, 2645	\bp@max@im@value ...
1289, 1290, 1883,	\bp@has@positive@values	..... 2680, 2700
1891, 1945, 1953,	..... 2343,	\bp@max@im@valuefloat
2053, 2060, 2120,	2499, 2519, 2582, 2634	..... 2771, 2772
2129, 2149, 2186,	\bp@imagpart . 1316, 1341	\bp@max@re@float ...
2195, 2215, 2258, 2265	\bp@k ..... 1660, 1662	... 2492, 2538, 2542
\bp@domain@start ...	\bp@Last@Loop@p 2456, 2458	\bp@max@re@neg@float
152, 170, 242, 315,	\bp@Last@Loop@z 2436, 2438	2496, 2530, 2534,
393, 455, 645, 661,	\bp@legacy .....	2593, 2597, 2648, 2657
698, 714, 891, 903,	... 890, 900, 978,	\bp@max@re@neg@pow@ .
905, 979, 990, 992,	987, 1048, 1055,	... 2353, 2646, 2654
1147, 1157, 1186,	1057, 1079, 1084,	\bp@max@re@pos@float
1195, 1278, 1288,	1086, 1216, 1220,	..... 2494,
1290, 1882, 1891,	1662, 1682, 1881,	2521, 2525, 2584,
1944, 1953, 2052,	1888, 1943, 1950,	2588, 2637, 2656, 2657
2119, 2129, 2149,	1995, 1997, 2015,	\bp@max@re@pos@pow@ .
2185, 2195, 2215, 2257	2017, 2118, 2125,	... 2346, 2635, 2643
\bp@domain@start: ...	2184, 2191, 2275,	\bp@max@re@pow@ ... 2668
..... 1159,	2277, 2299, 2301,	\bp@max@re@value ...
1197, 2060, 2265	2337, 2340, 2341, 2426	..... 2498, 2543
\bp@fix@add@opt ....	\bp@list ..... 1554,	\bp@max@re@valuefloat
... 1045, 1047, 1510	1555, 1690, 1691	..... 2658,
\bp@freq@filter .... 83	\bp@loop@delay .....	2659, 2660, 2661, 2662
\bp@freq@label ..... 83	... 1689, 1740, 1745	\bp@min@im@pow@ ... 2768
\bp@freq@scale .....	\bp@mag ..... 898, 900,	\bp@min@im@pow@ 2378,
83, 191, 198, 336,	921, 928, 985, 987,	2409, 2410, 2767, 2768
343, 476, 483, 905,	1008, 1015, 1208,	\bp@min@im@thresh@float
992, 1061, 1067,	1216, 1220, 1227,	2675, 2704, 2708,
1091, 1097, 1107,	1231, 1242, 1250,	2714, 2737, 2741, 2747
1110, 1118, 1125,	1885, 1888, 1905,	\bp@min@im@threshold@found
1240, 1244, 1290,	1906, 1911, 1912,	.... 2674, 2702,
1638, 1758, 1891,	1947, 1950, 1967,	2705, 2735, 2738, 2757
1953, 2000, 2007,	1968, 1973, 1974,	\bp@min@im@threshold@result
2020, 2027, 2129,	1996, 1997, 2002,	.... 2408, 2672,
2149, 2195, 2215,	2003, 2009, 2010,	2703, 2713, 2736,
2280, 2290, 2304, 2314	2016, 2017, 2022,	2746, 2758, 2761, 2770
\bp@freq@scale: ....	2023, 2029, 2030,	\bp@min@im@ZPK 2341, 2671
..... 905, 992,	2122, 2125, 2140,	\bp@min@pow@ ..... 2633
1290, 1891, 1953,	2150, 2188, 2191,	\bp@min@re@pow@ ....
2129, 2149, 2195, 2215	2206, 2216, 2276,	.... 2346, 2353,
\bp@full 205, 207, 208,	2277, 2282, 2291,	2364, 2406, 2407,
212, 350, 352, 353,	2300, 2301, 2306, 2315	2632, 2633, 2635, 2646
357, 490, 492, 493,	\bp@mag@axes .....	\bp@min@re@threshold@result
497, 611, 613, 614,	... 405, 409, 410,	2405, 2486, 2546,
618, 664, 666, 667,	412, 422, 423, 425,	2555, 2601, 2610,
671, 717, 719, 720, 724	506, 507, 509, 869,	2622, 2624, 2626, 2660

<code>\bp@min@real@ZPK</code> ...	<code>\bp@ph@axes</code> .....	500, 502, 613, 615,
..... 2340, <u>2485</u>	... 406, 414, 415,	618, 621, 623, 666,
<code>\bp@min@thresh@float</code>	417, 430, 431, 433,	668, 671, 674, 676,
2489, 2547, 2551,	514, 515, 517, 870,	719, 721, 724, 727, 729
2556, 2602, 2606, 2611	917, 958, 1004,	
<code>\bp@min@threshold@found</code>	1303, 1763, 1790, 1793	<code>\bp@short@list</code> .....
.... 2488, 2545,	<code>\bp@ph@cmd</code> .....	54, 944, 949, 1031,
2548, 2600, 2603, 2621	1003, 1010, 1021,	1036, 1132, 1135,
<code>\bp@min@threshold@result</code>	1243, 1262, 1282, 1307	1171, 1174, 1923,
..... 2624	<code>\bp@ph@commands</code> .....	1928, 1985, 1990,
<code>\bp@mode</code> .. 1217, 1221,	..... 874, 924,	2037, 2040, 2164,
1226, 1230, 1255, 1258	935, 962, 1011,	2169, 2234, 2239,
<code>\bp@new</code> . 215, 217, 220,	1022, 1273, 1768, 1805	2242, 2245, 2476, 2481
222, 225, 360, 362,	<code>\bp@ph@plot</code> .... 255,	<code>\bp@style</code> ..... <u>68</u>
365, 367, 370, 373,	263, 264, 266, 280,	<code>\bp@tf@new@to@legacy</code>
375, 378, 500, 502,	281, 283, 374, 375,	..... 965,
505, 507, 510, 513,	377, 919, 1006,	1077, 1215, <u>1664</u> ,
515, 518, 621, 623,	1764, 1778, 1781, 1819	1940, 2015, 2181, 2299
626, 628, 631, 674,	<code>\bp@ph@scale</code> .....	<code>\bp@TF@plot</code> .....
676, 679, 681, 684,	... 83, 178, 183,	. 987, 1084, 1086,
727, 729, 732, 734, 737	323, 328, 463, 468,	1216, 1231, <u>1684</u> ,
<code>\bp@num</code> .. 1667, 1671, 1672	772, 776, 787, 800,	1950, 2017, 2191, 2301
<code>\bp@num@deg</code> .. 1696, 1701	803, 815, 821, 822,	<code>\bp@tikz</code> .... 875, 896,
<code>\bp@num@im</code> ..... 1686,	842, 1010, 1745,	963, 983, 1143,
1706, 1744, 1746	1756, 1758, 1905,	1151, 1182, 1190,
<code>\bp@num@list</code> .....	1906, 1911, 1912,	1274, 1284, 1770,
... 1670, 1673, 1682	1967, 1968, 1973,	1811, 1832, 1839,
<code>\bp@num@re</code> ..... 1685,	1974, 2002, 2003,	1876, 1887, 1938,
1703, 1709, 1744, 1746	2009, 2010, 2022,	1949, 2048, 2056,
<code>\bp@p</code> ... 1652, 1654, 1655	2023, 2029, 2030,	2074, 2088, 2113,
<code>\bp@p@list</code> 1653, 1657, 1662	2150, 2153, 2206,	2124, 2179, 2190,
<code>\bp@parse@add@Bode@opt</code>	2216, 2221, 2291,	2253, 2261, 2331, 2390
... 1213, 1224, <u>1852</u>	2294, 2306, 2315, 2320	<code>\bp@tmp</code> . 895, 897, 901,
<code>\bp@parse@complex</code> ...	<code>\bp@ph@x@label</code> ..... <u>83</u>	911, 1053, 1055,
... <u>1311</u> , 1648, 1656	<code>\bp@ph@y@label</code> ..... <u>83</u>	1057, 1082, 1084, 1086
<code>\bp@parse@env@opt</code> ...	<code>\bp@pim</code> .. 2454, 2459, 2464	<code>\bp@tmp@mag</code> 912, 921, 929
..... 1146,	<code>\bp@plot</code> .... 647, 652,	<code>\bp@tmp@ph</code> . 916, 923, 934
1185, <u>1830</u> , 2051, 2256	680, 681, 683, 700,	<code>\bp@user@prefix</code> . 28,
<code>\bp@parse@N@opt</code> ....	705, 733, 734, 736,	36, 38, 876, 964,
.... 1880, 1942,	1052, 1055, 1057,	1144, 1183, 1275,
<u>2070</u> , 2117, 2183, 2336	1063, 1067, 1081,	1771, 1814, 1816,
<code>\bp@parse@opt</code> .....	1084, 1086, 1094,	1833, 1842, 1844,
889, 977, 1277, <u>1761</u>	1099, 1107, 1110,	1877, 1939, 2049,
<code>\bp@ph</code> .....	1242, 1246, 1252,	2075, 2094, 2096,
899, 900, 923, 933,	1263, 1853, 1863,	2114, 2180, 2254, 2332
986, 987, 1010,	1902, 1964, 2072,	<code>\bp@z</code> ... 1644, 1646, 1647
1020, 1209, 1216,	2082, 2097, 2138,	<code>\bp@z@list</code> 1645, 1649, 1662
1220, 1227, 1231,	2204, 2329, 2443, 2463	<code>\bp@zim</code> .. 2434, 2439, 2444
1246, 1256, 1259,	<code>\bp@pre</code> ..... 2451, 2464	<code>\bp@zpk@new@to@legacy</code>
1885, 1888, 1905,	<code>\bp@prefix</code> 27, 45, 1630,	..... 877, 1044,
1906, 1911, 1912,	1637, 1750, 1757,	1219, <u>1641</u> , 1878,
1947, 1950, 1967,	2152, 2220, 2293, 2319	1995, 2115, 2275, 2334
1968, 1973, 1974,	<code>\bp@PZZPK@ticksXNeg</code> .	<code>\bp@ZPK@plot</code> .....
1996, 1997, 2002,	..... 2351, 2355	. 900, 1055, 1057,
2003, 2009, 2010,	<code>\bp@PZZPK@ticksXPos</code> .	1220, 1227, <u>1553</u> ,
2016, 2017, 2022,	..... 2344, 2348	1888, 1997, 2125, 2277
2023, 2029, 2030,	<code>\bp@realpart</code> . 1315, 1340	<code>\bp@zre</code> ..... 2431, 2444
2122, 2125, 2140,	<code>\bp@scale</code> ..... 2076,	<code>\bp_if_contains:nnTF</code>
2150, 2188, 2191,	2091, 2333, 2339, 2392	... <u>1498</u> , 1504, 1507
2206, 2216, 2276,	<code>\bp@short</code> ..... 207,	
2277, 2282, 2291,	209, 212, 215, 217,	
2300, 2301, 2306, 2315	352, 354, 357, 360,	
	362, 492, 494, 497,	

## C

`\changes` ..... 2228

`\cs_new_protected:Npn`

1346, 1354, 1368,

1375, 1411, 1422, 1450, 1459, 1471, 1499	<code>\l_bp_after_tl</code> .... 1321, 1465, 1474, 1480, 1486, 1494, 1495	<code>\MagCSZerosLin</code> .... <b>810</b> <code>\MagCSZerosPeak</code> .... <b>829</b> <code>\MagDel</code> ..... <b>775, 1622</b> <code>\MagK</code> ..... <b>769, 1603</b> <code>\MagKAsymp</code> .... <b>769, 1600</b> <code>\MagKLin</code> ..... <b>769, 1596</b> <code>\MagPole</code> ... <b>777, 804, 1587</b> <code>\MagPoleAsymp</code> ..... ..... <b>777, 806, 1584</b> <code>\MagPoleLin</code> <b>777, 805, 1580</b> <code>\MagSOPoles</code> ..... <b>837</b> <code>\MagSOPolesAsymp</code> ... <b>837</b> <code>\MagSOPolesLin</code> .... <b>837</b> <code>\MagSOPolesPeak</code> .... <b>855</b> <code>\MagS0Zeros</code> ..... <b>837</b> <code>\MagS0ZerosAsymp</code> ... <b>837</b> <code>\MagS0ZerosLin</code> ..... <b>837</b> <code>\MagS0ZerosPeak</code> .... <b>855</b> <code>\MagZero</code> ..... <b>804, 1569</b> <code>\MagZeroAsymp</code> . <b>804, 1566</b> <code>\MagZeroLin</code> ... <b>804, 1562</b>
<b>D</b>	<code>\l_bp_found_bool</code> ... ..... <b>1326,</b> <b>1362, 1438, 1452</b>	<b>N</b>
<code>\dospecials</code> ..... <b>65</b>	<code>\l_bp_input_tl</code> .... 1319, 1332, 1333, 1336, 1348, 1349, 1357, 1371, 1455, 1477	<code>\n@mod</code> ..... <b>1,</b> 1010, 1094, 2206, 2306 <code>\n@mod@n</code> ..... <b>1</b> <code>\n@mod@p</code> ..... <b>1, 785</b> <code>\n@pow</code> ..... <b>1, 778,</b> 779, 780, 790, 791, 793, 794, 796, 797, 798, 799, 800, 801, 810, 811, 814, 815, 816, 818, 820, 821, 838, 842, 1706, 1707, 1709, 1710, 1729, 1730, 1732, 1733, 1746, 1747, 1905, 1906, 1911, 1912, 1967, 1968, 1973, 1974, 2002, 2003, 2009, 2010, 2022, 2023, 2029, 2030 <code>\NicholsChart</code> ..... <b>2249</b> <code>\NicholsTF</code> ..... <b>2173</b> <code>\NicholsTF@Shorthandoff</code> ..... <b>2235, 2237</b> <code>\NicholsZPK</code> ..... <b>2107</b> <code>\NicholsZPK@Shorthandoff</code> ..... <b>2165, 2167</b> <code>\NyquistPlot</code> ..... <b>2034</b> <code>\NyquistTF</code> ..... <b>1932</b> <code>\NyquistTF@Shorthandoff</code> ..... <b>1986, 1988</b> <code>\NyquistZPK</code> ..... <b>1870</b> <code>\NyquistZPK@Shorthandoff</code> ..... <b>1924, 1926</b>
<b>E</b>	<code>\l_bp_sep_tl</code> .. <b>1322,</b> <b>1440, 1488, 1493</b>	
environments:	<code>\l_bp_sep_pos_int</code> ... ..... <b>1324,</b> <b>1439, 1461, 1464</b>	
<code>BodeMagPlot</code> .... <b>1168</b>	<code>\l_imag_tl</code> 1313, 1341, 1342, 1350, 1371, 1489, 1490, 1494, 1495	
<code>BodePhPlot</code> .... <b>1129</b>	<code>\l_real_tl</code> ..... 1312, 1340, 1342, 1349, 1370, 1483, 1484	
<code>BodePlot</code> ..... <b>1266</b>	<code>\l_tmpa_tl</code> <b>1380, 1381, 1392</b>	
<code>\exp_after:wN</code> ..... <b>1417</b>	<code>\l_tmpb_tl</code> ..... <b>1383,</b> 1384, 1387, 1388, 1394, 1395, 1398, 1399	
<b>G</b>	<code>\l_tmpc_tl</code> ... <b>1415, 1416</b>	
<code>\g@addto@macro</code> ..... 62, 566, 568, 572, 574, 582, 584, 588, 590, 598, 600, 604, 606	<code>\Last@LoopValue</code> <b>1547, 1549</b>	
<code>\gnu@id</code> ..... 927, 928, 932, 933, 1014, 1015, 1019, 1020, 1066, 1067, 1103, 1107, 1110, 1123, 1125, 1249, 1250, 1254, 1256, 1259	<code>\log@adjusted</code> .. <b>2629,</b> <b>2630, 2764, 2765</b>	
<b>I</b>	<code>\log@bp@max@im</code> <b>2773, 2774</b>	
<code>\if@babel</code> ..... <b>54,</b> 941, 1028, 1130, 1169, 1920, 1982, 2035, 2161, 2231, 2473	<code>\log@bp@max@im@adjusted</code> ..... <b>2775, 2776</b>	
<code>\if@declutterarg</code> ... ..... <b>5, 29, 35, 49</b>	<code>\log@bp@max@re</code> <b>2663, 2664</b>	
<code>\if@hzarg</code> .... <b>13, 103,</b> <b>142, 244, 395, 884, 972</b>	<code>\log@bp@max@re@adjusted</code> ..... <b>2665, 2666</b>	
<code>\if@pgfarg</code> .... <b>1, 21,</b> 106, 193, 200, 338, 345, 478, 485, 920, 1007, 1059, 1088, 1116, 1238, 1903, 1965, 1998, 2018, 2139, 2205, 2278, 2302	<code>\log@bp@max@re@neg</code> .. ..... <b>2649, 2650</b>	
<code>\if@radarg</code> .... <b>9, 89,</b> <b>147, 249, 400, 879, 967</b>	<code>\log@bp@max@re@neg@adjusted</code> ..... <b>2651, 2652</b>	
<code>\int_compare:nNnF</code> .. <b>1464</b>	<code>\log@bp@max@re@pos</code> .. ..... <b>2638, 2639</b>	
<code>\int_compare:nNnT</code> ... ... <b>1416, 1430, 1432</b>	<code>\log@bp@max@re@pos@adjusted</code> ..... <b>2640, 2641</b>	
<code>\int_compare:nNnTF</code> . <b>1461</b>	<code>\log@result</code> .... <b>2627,</b> <b>2628, 2762, 2763</b>	
<code>\int_decr:N</code> ..... <b>1428</b>	<code>\lowercase</code> ..... <b>62</b>	
<code>\int_incr:N</code> ..... ... <b>1427, 1446, 1467</b>	<b>M</b>	<b>O</b>
<code>\int_new:N</code> <b>1323, 1324, 1325</b>	<code>\MagCSPOles</code> ..... <b>810</b>	<code>\Orig@BodeTF</code> . <b>1029, 1035</b>
<code>\int_set_eq:NN</code> .... <b>1439</b>	<code>\MagCSPOlesAsymp</code> ... <b>810</b>	<code>\Orig@BodeZPK</code> .. <b>942, 948</b>
<code>\int_zero:N</code> ..... ... <b>1453, 1454, 1475</b>	<code>\MagCSPOlesLin</code> ..... <b>810</b>	<code>\Orig@NicholsTF</code> <b>2232, 2238</b>
<b>L</b>	<code>\MagCSPOlesPeak</code> .... <b>829</b>	<code>\Orig@NicholsZPK</code> ... ..... <b>2162, 2168</b>
<code>\l_bode_check_tl</code> ... ... <b>1498, 1500, 1501</b>	<code>\MagCSZeros</code> ..... <b>810</b>	<code>\Orig@NyquistTF</code> <b>1983, 1989</b>

\Orig@NyquistZPK ...	\PoleZeroMapZPK@tick	\tl_count:N ..... 1415
..... 1921, 1927	..... 2359,	\tl_gset:Nx ..... 1342
\Orig@PoleZeroMapZPK	2360, 2364, 2365,	\tl_gset_eq:NN 1340, 1341
..... 2474, 2480	2373, 2374, 2378, 2379	\tl_head:N ..... 1380
<b>P</b>	\PoleZeroMapZPK@xticklabel	\tl_if_empty:NTF 1377,
\PhCSPoles ..... 810	..... 2386, 2403	1384, 1395, 1482, 1486
\PhCSPolesAsymp 810, 847	\PoleZeroMapZPK@yticklabel	\tl_if_in:VnTF .... 1501
\PhCSPolesLin .. 810, 844	..... 2387, 2404	\tl_if_single_token:n
\PhCSZeros ..... 810	\prg_generate_conditional_variant:Nnn ..... 1329	\prg_generate_conditional_variant:Nnn ..... 1329
\PhCSZerosAsymp .... 810	..... 1329	\tl_if_single_token:nTF
\PhCSZerosLin ..... 810	<b>R</b>	..... 1424
\PhDel ..... 776, 1621	\regex_match:nVTF .. 1336	\tl_if_single_token:VF
\PhK ..... 769, 1602	\regex_replace_once:nnN	..... 1413
\PhKAsymp .. 769, 775, 1599	..... 1357	\tl_map_function:NN .
\PhKLin .... 769, 775, 1595	\removeslash ..... 1455, 1477	\tl_new:N ..... 1312, 1313, 1314,
\PhPole .... 777, 807, 1586	... 211, 212, 356,	1315, 1316, 1319,
\PhPoleAsymp 777, 809, 1583	357, 496, 497, 617,	1320, 1321, 1322, 1498
\PhPoleLin . 777, 808, 1579	618, 670, 671, 723, 724	\tl_put_right:Nn ...
\PhSOPoles ..... 837	\renewcommand ..... 1462, 1465	..... 1462, 1465
\PhSOPolesAsymp .... 837	178, 183, 191, 198,	\tl_remove_all:Nn .. 1333
\PhSOPolesLin ..... 837	323, 328, 336, 343,	\tl_set:Nn 1350, 1370,
\PhS0Zeros ..... 837	463, 468, 476, 483,	1378, 1385, 1396,
\PhS0ZerosAsymp .... 837	943, 1030, 1922,	1440, 1483, 1489, 1490
\PhS0ZerosLin ..... 837	1984, 2163, 2233, 2475	\tl_set:Nx 1332, 1380,
\PhZero ..... 804, 1568	<b>S</b>	1383, 1388, 1394,
\PhZeroAsymp .. 804, 1565	\str_if_eq:VnTF 1381,	1415, 1417, 1494, 1500
\PhZeroLin .... 804, 1561	1392, 1488, 1493	\tl_set_eq:NN .. 1349,
\PoleZeroMapZPK ... 2324	\stripbodeprefix ... 1399, 1404, 1484, 1495	\tl_tail:N ... 1383, 1394
\PoleZeroMapZPK@exp .	... 206, 207, 351,	\token_if_eq_charcode:NNT
..... 2363, 2366,	352, 491, 492, 612,	..... 1427, 1428
2368, 2377, 2380, 2382	613, 665, 666, 718, 719	\token_if_eq_charcode_p:NN
\PoleZeroMapZPK@formatXTick	..... 2358, 2386	..... 1435, 1436
\PoleZeroMapZPK@formatYTick	<b>T</b>	
..... 2372, 2387	\temp@mag@cmd 1298, 1304	
\PoleZeroMapZPK@Shorthandofftemp@ph@cmd . 1301, 1306		<b>U</b>
..... 2477, 2479	\tl_clear:N .. 1473, 1474	\use:n ..... 1417

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	65
	\bp@parse@env@opt: Added tikz option to environments	57		\addNyquistTFPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	60
	\bp@parse@N@opt: Added commands and tikz options	61		\addNyquistZPKPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	60
	\bp@parse@opt: Added Tikz option	56		BodeMagPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	45
	NicholsChart: Added tikz option to environments	64		BodePhPlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	45
	\NicholsTF: Added commands and tikz options	63		BodePlot: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	47
	\NicholsZPK: Added commands and tikz options	62		\BodeTF: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	42
	\NyquistTF: Added commands and tikz options	59		\NyquistTF: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	59
	\NyquistZPK: Added commands and tikz options	58		\NyquistZPK: Enabled ‘Hz’ and ‘rad’ units for frequency and phase, respectively	58
	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	61			
v1.0.4	General: Fixed unintended optional argument macro expansion	1			
v1.0.5	\bp@parse@opt: Fixed a bug	56			
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	63
<code>NyquistPlot</code> : Defined using the ‘NewEniron’ command from the ‘environ’ package to fix conflicts with externalization	61	<code>\NicholsZPK</code> : Added code to handle active characters	62
v1.1.3		<code>\NyquistTF</code> : Added code to handle active characters	60
<code>\addBodeComponentPlot</code> : Changed implementation to respect user-supplied domain	44	<code>\NyquistZPK</code> : Added code to handle active characters	59
<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	61
<code>\addBodeZPKPlots</code> : Changed implementation to respect user-supplied domain	43	v1.1.6	
<code>\addNicholsTFChart</code> : Changed implementation to respect user-supplied domain	65	<code>\bp@short@list</code> : Detect ‘babel-french’ using ‘frenchbsetup’	26
<code>\addNicholsZPKChart</code> : Changed implementation to respect user-supplied domain	65	v1.1.7	
<code>\addNyquistTFPlot</code> : Changed implementation to respect user-supplied domain	60	General: Detect and turn off shorthands to improve ‘babel’ compatibility	1
<code>\addNyquistZPKPlot</code> : Changed implementation to respect user-supplied domain	60	<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	65	<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.	64
<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	63
<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	62
<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	60
<code>\BodeTF</code> : Added code to handle active characters	43	<code>\NyquistZPK</code> : Use auto-generated list of shorthands instead of manually specifying them	59
<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.	61
<code>NicholsChart</code> : Defined using the ‘NewDocumentEnvironment’ command from the ‘xparse’ package and added a hook to handle active characters	64	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 . . . . .	66	/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 . . . . .	17	/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 . . . . .	57	\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 . . . . .	51
v3.0		\bp@contains@num: Added keyword detector . . . . .	51
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 . . . . .	51
\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 . . . . .	54
\addBodeTFPlot: Added a pgfkeys-based simplified user interface . . . . .	44	\bp@zpk@new@to@legacy: Added converter from new to legacy ZPK interface . . . . .	54
\addBodeZPKPlots: Added a pgfkeys-based simplified user interface . . . . .	43	\bp_if_contains:nnTF: Added substring checker . . . . .	51
\addNicholsTFChart: Added a pgfkeys-based simplified user interface . . . . .	65	\NicholsTF: Added a pgfkeys-based simplified user interface . . . . .	63
\addNicholsZPKChart: Added a pgfkeys-based simplified user interface . . . . .	65	\NicholsZPK: Added a pgfkeys-based simplified user interface . . . . .	62
\addNyquistTFPlot: Added a pgfkeys-based simplified user interface . . . . .	60	\NyquistTF: Added a pgfkeys-based simplified user interface . . . . .	59
\addNyquistZPKPlot: Added a pgfkeys-based simplified user interface . . . . .	60	\NyquistZPK: Added a pgfkeys-based simplified user interface . . . . .	58
BodeMagPlot: Added a pgfkeys-based simplified user interface . . . . .	45	\PoleZeroMapZPK: Added a pgfkeys-based simplified user interface . . . . .	66
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



v3.0.2		command argument specification for grabbing arguments . . . . .	43
\bp@fix@add@opt:	Added a conditional to avoid always plotting the true plot, fixes issue #18 . . . . .	\BodeZPK: Fixed babel wrapper command argument specification for grabbing arguments . . . . .	41
v3.0.3		\NyquistTF: Fixed babel wrapper command argument specification for grabbing arguments . . . . .	60
\bp@parse@complex:	Rewritten to support arbitrary expressions with nested braces and parentheses . .	\NyquistZPK: Fixed babel wrapper command argument specification for grabbing arguments . . . . .	59
v3.0.4			
\BodeTF:	Fixed babel wrapper		