

# yLpWin V3.02 \*†‡§¶

YEES ‖

## 概要

急遽、フィルタ検討のため Win7 向けに修正しました。使う用事が発生した時点でバージョンアップしています。簡単な伝達関数でもブレイクポイント周波数を計算するのが面倒で、ボード線図作成に使用しています。

旧 Macintosh SymantecThinkC → SymantecC++ → Windows98 VC6 → VS20xx とコンパイル環境も変わっています。自作ソフトのなかで最も長期使用しているものです。

## 目次

|     |            |   |     |                |   |
|-----|------------|---|-----|----------------|---|
| 1   | 機能全般.....  | 1 | 2.2 | 操作手順.....      | 3 |
| 1.1 | 機能.....    | 1 | 2.3 | プログラムについて..... | 5 |
| 2   | 使い方.....   | 2 | 2.4 | グラフデータ保存.....  | 6 |
| 2.1 | ツールバー..... | 2 | 2.5 | グラフ画像保存.....   | 6 |
|     |            |   | 2.6 | プログラミング仕様..... | 7 |

## 1 機能全般

### 1.1 機能

- 伝達関数  $f(s)$  からボード線図を作成する
- 伝達関数  $f(s)$  の数値ラプラス逆変換を行う (FILT アルゴリズムで古いものです)
- データ ファイル保存機能 (周波数 vs ゲイン&位相、CSV 形式)
- 生成したグラフ 画像ファイル保存機能 (BMP, JPG, PNG 形式)

\* 2014/08/07 V3.02 グラフ画像保存サイズ変更追加

† 2014/08/05 V3.01 BODE 設定 グラフ表示選択追加、メモリ管理変更

‡ 2014/08/04 V3.00 Win7 向け新規

§ 2014/08/04 ソフトウェア yLpWin は使用配布自由です。

¶ 2014/08/04 ソフトウェア yLpWin および関連文書 利用の結果生じた損害について一切責任を負いません。

‖ yeess@nifty.com ( <http://homepage2.nifty.com/yees/> )

## 2 使い方

### 2.1 ツールバー



図1 標準ツールバー

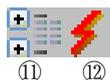


図2 BODE 設定 ツールバー

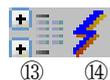


図3 FILT 設定 ツールバー



図4 プログラムソース ツールバー

- ① 新規ファイル 作業中のプログラムソース、グラフデータを破棄します
- ② ⑮ プログラムソースファイル（テキストファイル）を開きます
- ③ ⑯ プログラムソースファイル（テキストファイル）を保存します
- ④ カット（テキスト編集時）
- ⑤ コピー（テキスト編集時）
- ⑥ ペースト（テキスト編集時）
- ⑦ ⑫ ⑱ プログラムソースをコンパイルし、ボード線図を作成します
- ⑧ ⑭ ⑲ プログラムソースをコンパイルし、数値ラプラス逆変換を実行します
- ⑨ データを保存します（ボード線図：周波数 vs ゲイン&位相、FILT:時間 vs レベル、CSV 形式）
- ⑩ バージョン情報を表示します
  
- ⑪ BODE 設定の全項目を展開します
  
- ⑬ FILT 設定の全項目を展開します
  
- ⑰ プログラムソースウィンドウ内のフォントを変更します（ソース編集用のフォント）

## 2.2 操作手順

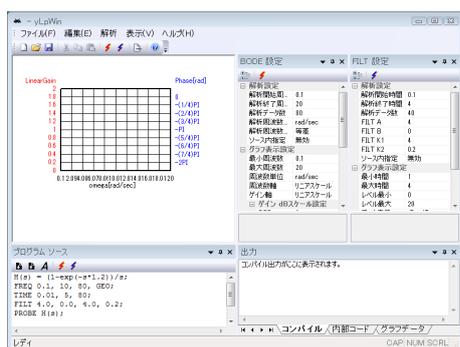


図 5 起動

起動時、ソースプログラム例が予め入力されます。プログラムソース例は 0 次ホールド (1.2sec サンプル&ホールド) です。

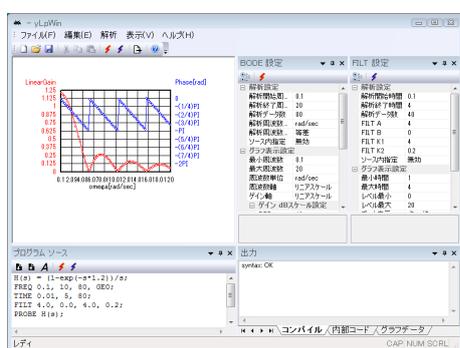


図 6 画像入力

そのまま、ツールバーボタン⑦または⑩または⑱で、プログラムソースをコンパイルし、ボード線図を作成します。ボード線図作成条件は、BODE 設定ウインドウ内の指定によるものです。

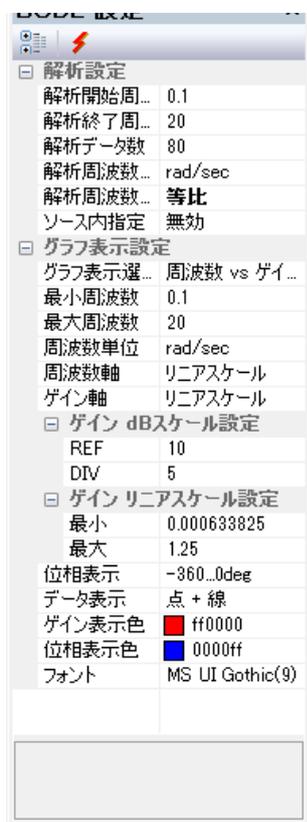


図 7 画像入力

解析設定 各項目は、データ計算時の条件です。

開始周波数 データ計算の開始周波数(間隔が等比の場合 0 指定不可です)

終了周波数 データ計算の開始周波数(間隔が等比の場合 0 指定不可です)

データ数 データ数を指定します。

周波数数間隔

等比 周波数軸が対数の場合、等間隔でデータを作成します。

等差 周波数軸がリニアの場合、等間隔でデータを作成します。

ソース内指定... 開始周波数/終了周波数/データ数は、プログラムソース内で指定できます (FREQ 行、または、OMEGA 行) そのソース内指定を有効とするか無効とするかを指定します。

グラフ表示設定 各項目は、グラフ表示についての設定です。各項目の変更後フォーカスを失うと(他項目や他ウィンドウをクリック)反映されます。

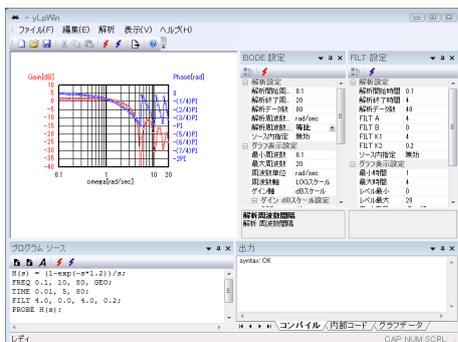


図 8 画像入力

「周波数数間隔」を「等比」で実行した例

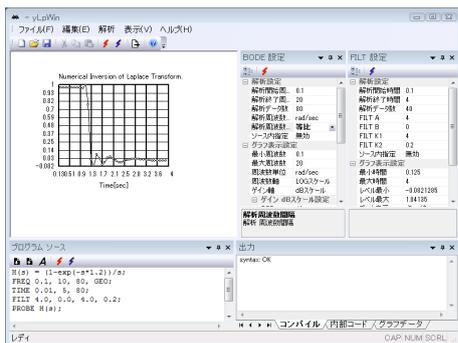


図 9 画像入力

起動直後、そのまま、  
 ツールバーボタン⑧または⑭または⑯で、数値ラプラス逆変換を実行した例。プログラムソース例の0次ホールド(1.2sec サンプル&ホールド)を、そのまま逆変換したもので、デルタ関数の積分値(=1)が1.2sec 間ホールドされます。

**解析設定** 各項目は、データ計算時の条件です。

- 開始時間 データ計算の開始時間 (0 以下不可)
- 終了時間 データ計算の開始時間 (0 以下不可)
- データ数 データ数を指定します。

**FILT A** 近似誤差を決める定数。概ね有効桁数。

**FILT B**  $f(s)$  が不安定系の場合の収束座標。

(参考) ステップ応答を求める場合、伝達関数に  $(1/s)$  を掛けるので、 $B = 0$  を指定。

**FILT K1**

**FILT K2**  $K1$  と  $K2$  を使って、打ち切り誤差を決定する。詳細については下記文献を参照下さい。

電子通信学会誌 (現、電子情報通信学会)  
 数値ラプラス変換 [ I ], [ II ], [ III ]  
 1982/8 ~ 1982/10 (UDC 517.442 - 37)

著者：細野敏夫

**ソース内指定...** 開始時間/終了時間/データ数を、プログラムソース内の **TIME** 行で指定できます。また、**A/B/K1/K2** をソース内の **FILT** 行で指定できます。それらのソース内指定を有効とするか無効とするかを指定します。

**グラフ表示設定** 各項目は、グラフ表示についての設定です。各項目の変更後フォーカスを失うと (他項目や他ウィンドウをクリック) 反映されます。



図 10 画像入力

### 2.3 プログラムについて



図 11 画像入力

サンプルソースファイル「BiQuad.txt」を開き、ボード線図を作成した例。

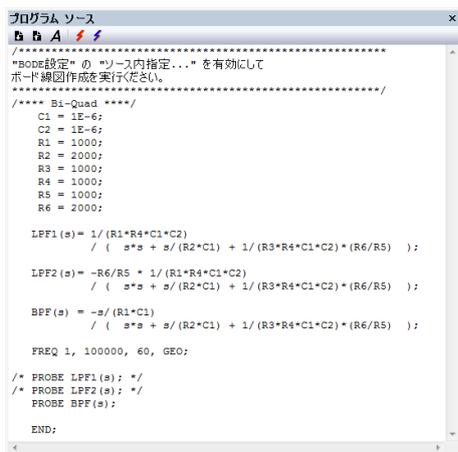


図 12 画像入力

図は、サンプルソースファイル「BiQuad.txt」の内容です。

- C 言語の数式記述に準じます。if, while, for, goto 等の繰り返しや分岐制御に対応していません。
- /\* ~ \*/ はコメントです。//行はコメント行として使用できません。//は並列演算を記述する 2 項演算子としています。A/B は、A\*B/(A+B) と同等です。
- 英数字は、実数変数です。C1, R1 等
- 英数字 (s) は、s の関数を扱う関数変数です。LFF1(s) 等
- FREQ は、ボード線図の開始周波数 (Hz)、終了周波数 (Hz)、データ数を指定する行です。
- OMEGA を使用すれば、開始周波数 (rad/s)、終了周波数 (rad/s)、データ数を指定できます。
- PROBE は、どの関数変数についてボード線図を作成するのかを指定する行です。

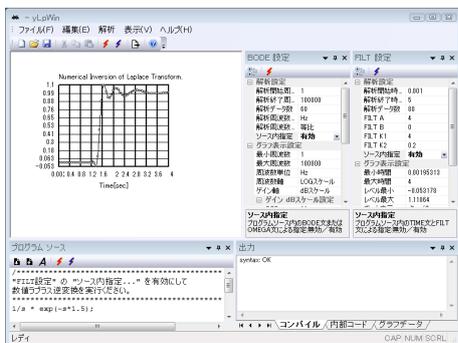


図 13 画像入力

サンプルソースファイル「step関数の1.5sec遅れ.txt」を開き、数値ラプラス逆変換を実施した例。

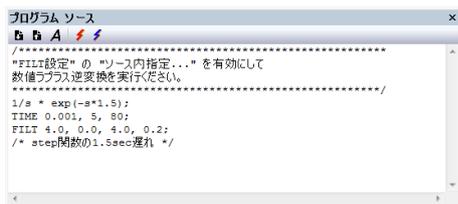


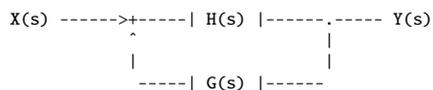
図 14 画像入力

図は、サンプルソースファイル「step関数の1.5sec遅れ.txt」の内容です。

- s の関数変数を使用せず、s の関数内容をひとつだけ記述すれば、それを解析対象とします。これはボード線図作成についても同様です。
- TIME は、ラプラス逆変換の開始時間 (sec)、終了時間 (sec)、データ数を指定する行です。
- FILT は、定数 A、B、K1、K2 を指定する行です。

===== 注意点 =====

yLpWin は、行列演算を行っていません。従って接点方程式を解くことはできません。入力式を順序処理するのみです。フィードバックがある場合



以下の例のように、 $Y(s)$  にフィードバックの公式を記述すれば、解析できます。

```
/* example Feed-back *****/

H(s) = 1/ ( 100+s );
G(s) = (1+s)*(50+s)/(100+s);
Y(s) = H(s)/(1-H(s)*G(s));

FREQ 1, 1000, 50, GEO;
/* in Hz */

/* PROBE H(s); */
/* PROBE G(s); */
PROBE Y(s);
/*****/
```

## 2.4 グラフデータ保存

| No. | omega[rad/s] | gain[linear] | phase[rad] |
|-----|--------------|--------------|------------|
| 1   | 0.1          | 5E-005       | -1.57082   |
| 2   | 0.351899     | 0.000175949  | -1.57088   |
| 3   | 0.603797     | 0.000301899  | -1.57095   |
| 4   | 0.855696     | 0.000427848  | -1.57101   |
| 5   | 1.10759      | 0.000553798  | -1.57107   |
| 6   | 1.35949      | 0.000679747  | -1.57114   |
| 7   | 1.61139      | 0.000805697  | -1.5712    |
| 8   | 1.86329      | 0.000931647  | -1.57126   |
| 9   | 2.11519      | 0.0010576    | -1.57133   |
| 10  | 2.36709      | 0.00118355   | -1.57139   |
| 11  | 2.61899      | 0.0013095    | -1.57145   |
| 12  | 2.87089      | 0.00143545   | -1.57151   |
| 13  | 3.12278      | 0.0015614    | -1.57158   |
| 14  | 3.37468      | 0.00168735   | -1.57164   |
| 15  | 3.62658      | 0.0018133    | -1.5717    |
| 16  | 3.87848      | 0.00193925   | -1.57177   |
| 17  | 4.13038      | 0.00206521   | -1.57183   |
| 18  | 4.38228      | 0.00219116   | -1.57189   |
| 19  | 4.63418      | 0.00231711   | -1.57195   |
| 20  | 4.88608      | 0.00244307   | -1.57202   |
| 21  | 5.13797      | 0.00256902   | -1.57208   |
| 22  | 5.38987      | 0.00269497   | -1.57214   |

図 15 画像入力

ツールバーボタン⑨または、メニュー「解析」内の「グラフデータファイル (CSV) 保存 ...」でグラフデータを保存できます。「出力」ウインドウの「グラフデータ」タブに表示中のデータを保存します。

## 2.5 グラフ画像保存

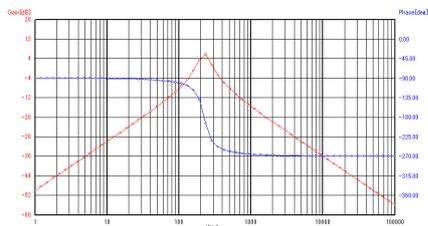


図 16 画像入力

メニュー「ファイル (F)」内の「表示グラフを保存 ...」で表示グラフを画像ファイルに保存できます。

グラフを画像のデフォルトサイズは、水平方向 400 ピクセル、垂直方向 300 ピクセルです。メニュー「ファイル (F)」内の「画像サイズ変更 ...」で変更できます。

## 2.6 プログラミング仕様

### プログラミング仕様

(1) 予約語 (大文字、小文字またその混合区別なく)

```
S
FREQ FREQUENCY
OMEGA
TIME
FILT
PROBE
ARITH ARITHMETIC GEO GEOMETRIC
EXP LOG LN SIN COS TAN SINH COSH TANH SQRT
END
```

(2) 演算子

```
+ 加算
- 減算
* 乗算
/ 除算
** 累乗
// 並列 (x//y は、x*y/(x+y) と同等。演算の優先順位は乗除算と同等。)
```

(3) 関数

```
EXP(x)
LOG(x)      対数 (10底)
LOG10(x)    対数 (10底)
LN(x)       自然対数
SIN(x)
COS(x)
TAN(x)
SINH(x)
COSH(x)
TANH(x)
SQRT(x)
```

(3) パーサ規則

```
ソース : 文リスト
;
```

```
文リスト
:
| 文リスト 文
;
```

文

```
: /* コメント */
| ラプラス式
| ラプラス式 END
| 英数字 = 実数式 ;
| ラプラス関数 = ラプラス式 ;
| FREQ[UENCY] 数値 , 数値 , 数値 , 周波数列タイプ ; (開始, 終了, データ数, ARITH または GEO の並び)
| OMEGA 数値 , 数値 , 数値 , 周波数列タイプ ; (開始, 終了, データ数, ARITH または GEO の並び)
| TIME 数値 , 数値 , 数値 ; (開始, 終了, データ数 の並び)

| FILT 数値 , 数値 , 数値 , 数値 ; (A, B, K1, K2 の並び)
| PROBE 英数字 ;
| PROBE ラプラス関数 ;
| ;
| END
;
```

周波数列タイプ

```
: ARITH[METRIC] 等差数列の周波数で解析 (リニアスケール上で等間隔)
| GEO[METRIC] 等比数列の周波数で解析 (ログスケール上で等間隔)
;
```

実数式

```
: 実数式 + 実数式
| 実数式 - 実数式
| 実数式 * 実数式
| 実数式 / 実数式
| - 実数式 %prec -
| 実数式 ** 実数式
| 実数式 // 実数式
| EXP ( 実数式 )
| LOG ( 実数式 )
| LN ( 実数式 )
| SIN ( 実数式 )
| COS ( 実数式 )
| TAN ( 実数式 )
| SINH ( 実数式 )
| COSH ( 実数式 )
| TANH ( 実数式 )
| SQRT ( 実数式 )
| ( 実数式 )
| 英数字
| 数値
;
```

ラプラス式  
: ラプラス式 + ラプラス式  
| ラプラス式 - ラプラス式  
| ラプラス式 \* ラプラス式  
| ラプラス式 / ラプラス式  
| - ラプラス式  
| ラプラス式 \*\* ラプラス式  
| ラプラス式 // ラプラス式  
| EXP ( ラプラス式 )  
| LOG ( ラプラス式 )  
| LN ( ラプラス式 )  
| SIN ( ラプラス式 )  
| COS ( ラプラス式 )  
| TAN ( ラプラス式 )  
| SINH ( ラプラス式 )  
| COSH ( ラプラス式 )  
| TANH ( ラプラス式 )  
| SQRT ( ラプラス式 )  
| ( ラプラス式 )  
| 英数字  
| 数値  
| s  
| ラプラス式関数  
;

ラプラス式関数  
: 英数字 (s)  
;