

グラフで設計仕様を与えることができる制御系設計支援ツール Computer Aided Control System Design Tool based on Design Specification in Graph

古賀 雅伸†
Masanobu Koga

森 智宏†
Tomohiro Mori

1. はじめに

実験データやシミュレーションデータを把握するには個々のデータの値を調べるだけでなく、データ全体を描画したグラフを分析する必要がある。数値から特徴の詳細が、グラフから特徴の概略が分かる。また、制御系の設計プロセスにおいて、設計仕様をグラフで与える方が数値で与えるより簡単な場合がある。

例えば、ボード線図を用いて設計を行う H_∞ 制御 [1] や古典制御 [2] などでは、制御系の式をグラフ表示して、グラフからパラメータを読み取り、その値を用いて新たな式を導出する、といった作業を繰り返す。このような場合、数値で仕様を与える代わりに、仕様を表すグラフを描画し、グラフに対応する式を仕様として与えられると設計が容易になる。

しかし、これまでのグラフを用いる設計方法としては、数式を変更しながらグラフを確認するという方法がほとんどであり、グラフを変えて数式に反映させるという方法はあまりとられていない。

そこで本研究では、与えられた数式のグラフを表示し、そのグラフをマウスで変形すると、数式の構造は保たれたまま、その係数(パラメータ)がグラフに対応する値に変化するグラフツールを開発する。そして、このツールを用いて、グラフで設計仕様を与えることができる制御系設計支援ツール Jpit(Java Plot Interactive Tool)を開発する。

開発言語は GUI をネイティブサポートし、再利用性の優れたオブジェクト指向言語である Java 言語 [3] を用いる。

2. 数式とグラフの関連

グラフで設計仕様を与えるためには数式とグラフとを関連付ける仕組みが必要である。すなわち、数式の変化とグラフの変化を一対一に対応させるとともに、グラフをインタラクティブに移動できる必要がある。

しかし、一般に任意のグラフから対応する数式を唯一に決定することは困難である。そこで、Jpit ではあらかじめ数式の構造を決定し、構造を保つ制約のもと係数(パラメータ)を変化させられるよう、ユーザに変更可能な点を与える。これにより、インタラクティブにパラメータを調節することが可能となる。

3. Jpit

Jpit は時間応答と周波数応答を表示可能であり、伝達関数 [4] の構造を保ったまま周波数応答をインタラクティブに修正することができる。起動画面を図 1 に示す。

†九州工業大学

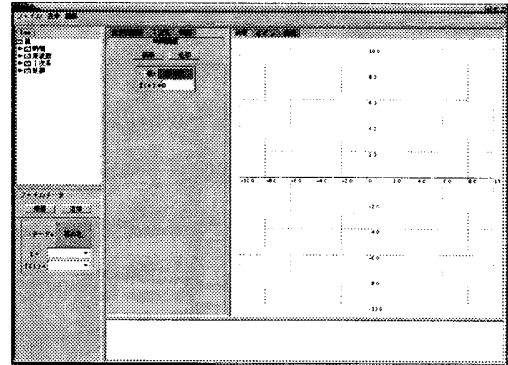


図 1: GUI からの起動画面 (時間)

表 1: 時間応答で使用可能な演算

演算	入力方法	演算	入力方法
+	+	-	-
×	*	÷	/
sin	sin	cos	cos
tan	tan	log _e	ln
√	sqrt	π	Pi
e	exp		

3.1 線の描画

表 1 に示される演算を用いて表現される t についての式を入力することによって時間応答を描画できる。

表 2 に示す伝達関数を入力することによって周波数応答(ボード線図)を描画できる。1 次系はゲイン K , 時定数 T , 2 次系はゲイン K , 減衰係数 ζ , 自然角周波数 ω_n , 位相遅れ・進み補償は K, α, T の値を設定できる。1 次系と 2 次系と積分器の逆システムも簡単に入力できる。

3.2 折れ線近似を用いた線の変形

線の変形を行うために、折れ線近似を利用した。折れ線近似を変形させることにより元の線を変形させる。1 次系、位相遅れ・進み補償の折れ線近似を図 2~5 に示す。実線が折れ線近似した線を、破線が近似した元の線を示す。図 2 のゲイン線図では A 点を移動させると、1

表 2: 周波数応答で使用可能な伝達関数

名前	式	逆システム
1 次系	$\frac{K}{1+Ts}$	$\frac{1+Ts}{K}$
2 次系	$\frac{K\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$	$\frac{s^2+2\zeta\omega_n s+\omega_n^2}{K\omega_n^2}$
積分器	$\frac{1}{s}$	s
位相遅れ補償	$K \frac{\alpha(Ts+1)}{\alpha Ts+1}$	×
位相進み補償	$K \frac{Ts+1}{\alpha Ts+1}$	×

次系の K, T が、図3の位相線図ではA点を移動させると、1次系の T が変化する。図4と図5については、A点を移動すると位相遅れ・進み補償の α が、B点を移動すると K, T が変化する。

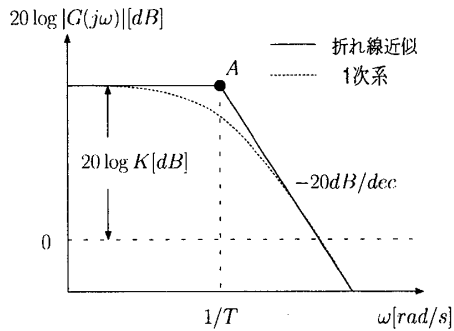


図 2: 1次系の折れ線近似 (ゲイン)

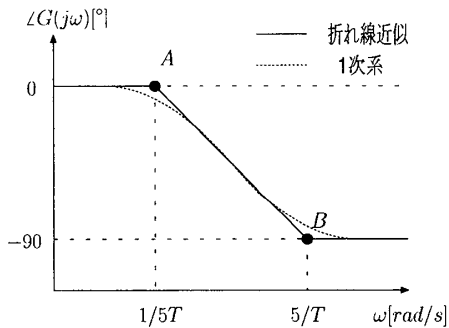


図 3: 1次系の折れ線近似 (位相)

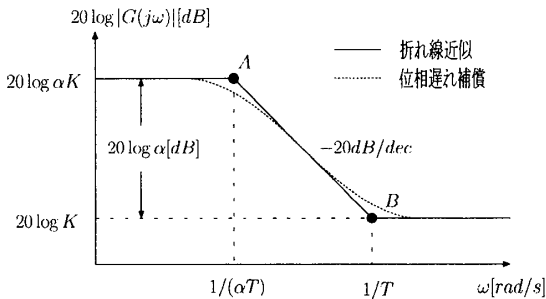


図 4: 位相遅れ補償

4. Jpit による位相遅れ補償の設計

位相遅れ補償 [2] は、定常特性を改善する目的で用いられ、低周波域と高周波域でのゲインがそれぞれ $K(0) = \alpha K, K(\infty) = K$ となる。

位相遅れ補償の一般的な設計手順を以下に示す。

1. 制御対象 $P(s)$ のボード線図を描き、ゲイン交差周波数 ω_{gc} を求める。
2. $KP(s)$ が位相余裕 PM の仕様を満たすゲイン補償 K を決定する。
3. \hat{K}_p, \hat{K}_v を求める。

$$\hat{K}_p = \lim_{s \rightarrow 0} KP(s)$$

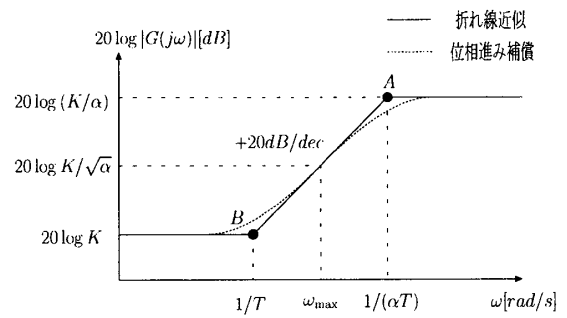


図 5: 位相進み補償

$$\hat{K}_v = \lim_{s \rightarrow 0} sKP(s)$$

4. $\alpha \hat{K}_p, \alpha \hat{K}_v$ が K_p, K_v の仕様を満たす α を決定する。
5. $1/T \leq 0.1\omega_{gc}$ を満たす T を決定する。

以上のように一般的な設計手順では、式を導出してグラフ表示し、グラフからパラメータを読み取り、その値を用いて新たな式を導出するという作業を行う。Jpit を用いると、以下に示すように視覚的に設計を行うことができ、作業効率を上げることができる。

1. 制御対象 $P(s)$ のボード線図を描き、ゲイン交差周波数 ω_{gc} を求める。
2. PM の仕様を満たすように $P(s)$ を構成している伝達関数を垂直移動させ、ゲイン補償 K を決定する。
3. K_p, K_v の仕様を満たすように決定した K の値を用いた $K(s)$ の線を移動させ、 $K(s)P(s), sK(s)P(s)$ を編集して α を決定する。
4. $K(s)P(s)$ が PM の仕様を満たすように $K(s)$ を水平移動させて、 T を決定する。

5. まとめ

設計仕様として与えたグラフをインタラクティブに修正することで制御系設計を効率的に行える制御系設計支援ツールを開発した。

今後、編集できる伝達関数の種類を増やす予定である。周波数応答以外のグラフを設計仕様として与えることができる一般的な問題に対応したフレームワークに発展させたい。

参考文献

- [1] 美多 勉. H_∞ 制御. 昭晃堂, 1994.
- [2] 杉江 俊治 and 藤田 政之. フィードバック制御入門. コロナ社, 2001.
- [3] David M. Geary. グラフィック Java2 vol.1 AWT 編. 株式会社アスキー, 2000.
- [4] 相良 節夫. 基礎自動制御. 森北出版株式会社, 2001.