ロバスト制御によるプロセス制御系設計

2000MM041　小島 伸也
指導教員　 高見 勲

1. はじめに

制御対象を数学的にモデル化する際には、モデルの不確かさが避けられない。これは、パラメータの経時変化や、モデル化においてモデルを低次元化した際のモデル誤差などが理由として挙げられる。そこで、そのような不確かさが存在しても制御系を安定化するような制御手法として、ロバスト制御理論がある。[1]

本研究では特性変化の大きいプロセスに対して、ロバスト制御を適用し、その制御性能について考察する。

2. 制御対象

制御対象は、以下のようになる。

$$\tilde{P}(s) = \frac{Ke^{-Ls}}{1+Ts} \qquad \begin{array}{l} K_1 \le K \le K_2 \\ T_1 \le T \le T_2 \\ L_1 \le L \le L_2 \end{array} \qquad (2.1)$$

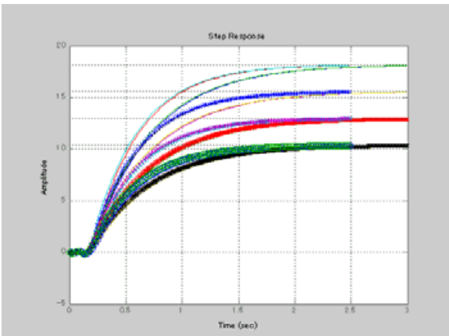

図 2.1 変動する制御対象の開ループステップ応答

以上のことからわかるように、制御対象は 3 つの変動するパラメータをもつ 1 次系とむだ時間を重ね合わせたものであり、変動幅も大きい。

つまり、この制御対象を補償するようなコントローラを設計する際には、図に示したような不確定要素をすべて考慮した上でコントローラを設計する必要がある。

3. 制御手法

3.1. ロバスト安定性制御理論

ロバスト安定性制御理論は、ノミナルプラント $P(s)$ と不確かさの大きさ $Wt(s)$ が与えられたとき、以下の条件式を満たすようなコントローラ $K(s)$ を設計することである。

$$T(s) = \frac{L(s)}{1+L(s)} = \frac{P(s)K(s)}{1+P(s)K(s)} \qquad (3.1)$$

$$|Wt(j\omega)T(j\omega)| < 1, \forall\omega \qquad (3.2)$$

逆にいえば、この式(3.2)を満たすようなコントローラを設計しさえすれば制御系の安定性は保証される。このことは、より一般的には小ゲイン定理として知られている。[1]

3.2. 混合感度法

ロバスト安定性制御理論の条件を満たすことで、安定性が保証される。それとは相対的に

$$S(s) = \frac{1}{1+L(s)} = \frac{1}{1+P(s)K(s)} \qquad (3.3)$$

$$|Ws(j\omega)S(j\omega)| < 1, \forall\omega \qquad (3.4)$$

式(3.4)を満たすことでノミナル性能が保証される。つまり、式(3.4)を満たすようなコントローラ $K(s)$ を決めることで、外乱に強い制御系が設計できるようになる。

$$|Ws(j\omega)S(j\omega)| + |Wt(j\omega)T(j\omega)| < 1, \forall\omega \qquad (3.5)$$

混合感度法は、適当な重み関数 $|Ws(j\omega)|$ と $|Wt(j\omega)|$ を与えて式(3.5)を満たすような感度関数と相補感度関数を成形することにより、外乱抑制とロバスト安定性を確保する設計手法である。[1]

4. プロセスに対するロバスト制御系の設計

4.1. ロバスト安定性制御理論

ロバスト安定性制御理論を適用するために、まず、不確かさを考慮したこの制御対象の伝達関数は式（2.1）となる。

つぎに、制御対象のノミナルモデルを

$$P(s) = \frac{K_0}{1+T_0 s} \qquad (4.1)$$

と定義することで、

$$\left| \frac{\tilde{P}(j\omega)}{P(j\omega)} - 1 \right| \le |Wt(j\omega)| \qquad (4.2)$$

となる $|Wt(j\omega)|$ を選定できる。

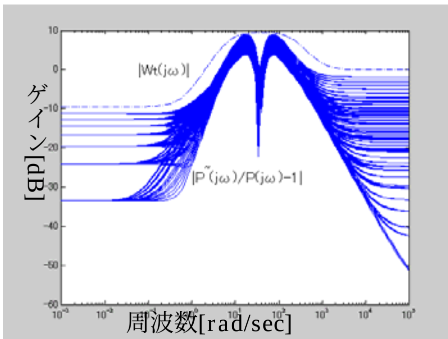

図 4.1 変動幅をカバーする重み関数 $|Wt(j\omega)|$

$$Wt(s) = \frac{0.002s^2 + 1.5s + 1}{0.002s^2 + 0.5s + 3} \qquad (4.3)$$

式(3.2)、式(4.1)よりコントローラ $K(s)$ が決まる。この条件をみたすコントローラは、複数存在するが、その中でも PID の構造を持つようにコントローラを決定する。

$$K(s) = \frac{0.00688(s^2 + 11.63s + 33.8)}{s} \qquad (4.4)$$

ここで、このコントローラを用いたときの安定性の様子を図 4.2、図 4.3 に示す。

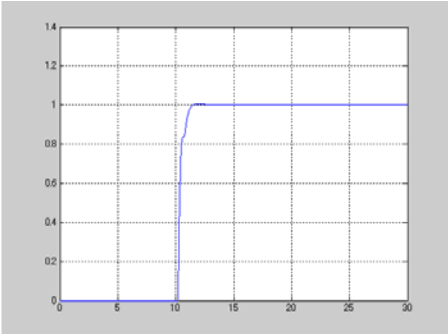

図 4.2 最小モデルのときのステップ応答

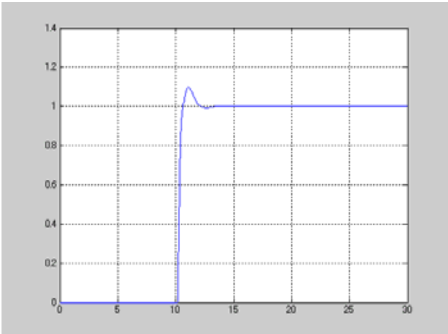

図 4.3 最大モデルのときのステップ応答

4.2. 混合感度法

重み関数を

$$Ws(s)=\frac{10s+1000}{100s+1}\times 0.03 \qquad (4.5)$$

式(4.5)と、式(4.3)と決めたときに、式(4.1)から式(3.5)を満たすようにコントローラ $K(s)$ を決定すると、

$$K(s)=\frac{0.6585s^2+5.501s+9.297}{s^3+21.54s^2+69.93s+0.6971} \qquad (4.6)$$

となる。[2]

このコントローラを用いたときの感度関数、相補感度関数の様子は次の図 4.4 のようになる。この図より感度関数、相補感度関数がともに式(3.2)、式(3.4)を満たすようになっていることが分かる。

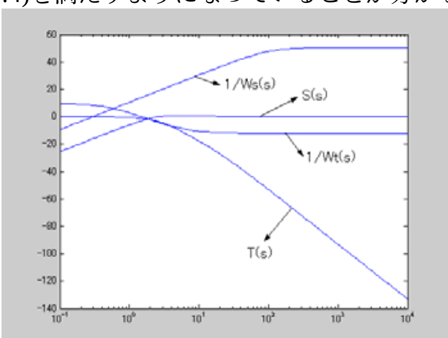

図 4.4 重み関数に対する感度関数、相補感度関数

ここで、このコントローラを用いたときの安定性の様子を図 4.5、図 4.6 に示す。

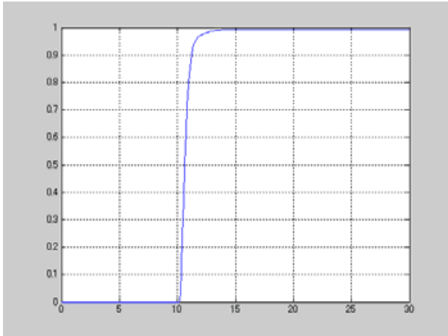

図 4.5 最小モデルのときのステップ応答

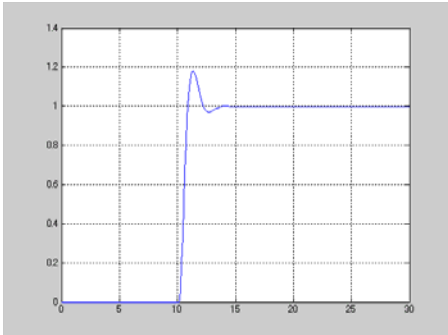

図 4.6 最大モデルのときのステップ応答

5. 考察

ロバスト安定性制御理論を適用するにあたって、重み関数の定義は非常に大きなウエイトを占めている。これは、式(3.2)よりコントローラをノミナルプラントと重み関数の値だけで決定しているからである。今後、ロバスト安定性制御理論を用いる際には、パラメータの定義が、最も注意しなければならない点である。

次に、混合感度法であるが、この手法は制御系のロバスト性とノミナル性能を同時に考慮することができるのが最大の利点である。しかし、制御対象に適した重み関数を定義するためには、いくらかの試行錯誤が必要となり、手間がかかってしまうのと、必ずしも好ましい結果が得られる保証がないところが難点である。

6. おわりに

本研究より、変動幅の大きい制御対象でもロバスト制御を用いることで、適したコントローラが設計できることがわかった。このように良好な結果が得られたのは、コントローラを PID の形に設計したことや、適切な重み関数が設計できた点が良かったと思われる。しかし、混合感度法に関しては、設計したコントローラの低次元化を検討する必要があり、今後の課題として残っている。

参考文献

[1] 杉江俊治,藤田政之:フィードバック制御入門:コロナ社(1999).

[2] 野波健蔵,西村秀和,平田光男:MATLAB による制御系設計:東京電機大学出版(1998).

※ MATLAB/Simulink は米国 the Math Works 社の登録商標