

PID 型加速器補正を有する Extremum Seeking 制御

高田 等* 上原 幸樹** 八野 知博*

Extremum Seeking Control Modified by PID-type Accelerator

Hitoshi TAKATA*, Kouki UEHARA** and Tomohiro HACHINO*

In this paper we consider a modification of the standard extremum seeking control for nonlinear systems. This method modifies amount of control by replacing an integrator with an accelerator of PID-type. The proposed method is applied to Monod model of bioreactor. Simulation results show that this enables its operation to reach the maximum point swiftly.

Keywords: Nonlinear control, Extremum seeking control, PID-type accelerator, Monod model

1. まえがき

我々の身の回りには様々なシステムが存在しているが、そのほとんどが非線形システムである。プロセス系などにおける制御対象は、一般に複雑な特性を持ち非線形性が強く、最適な制御をすることは容易ではない。この問題に対し、最適運転点をオンラインで探索しながらシステムを運転する適応制御の一つとして、Extremum Seeking (極値探索) 制御がある。本制御法の適用範囲としては、化学や生物の攪拌プロセス、ガス炉燃焼プロセス、粉碎プロセス、自動車エンジンの発火角度制御器、磁気浮遊フライホイール制御器、太陽電池や電波望遠鏡のアンテナ調節器、タービンや風車の羽根調節器など多岐である。

本報告では、標準型 Extremum Seeking 制御¹⁾に対し出力応答の速応性の改良を目的とし、自動整合型 PID を基とした PID 型加速器補正の改良型 Extremum

Seeking 制御法^{2)~5)}について考察した。本手法を Monod model に適用し、数値シミュレーション実験を行った結果、従来の標準型 Extremum Seeking 制御に対して、本手法の速応性の改良が確認された。

2. 標準型 Extremum Seeking 制御

2.1 概要

標準型 Extremum Seeking 制御は、非線形問題として生成物反応器の運転の最適化などで研究されてきた¹⁾。本節では、この標準型 Extremum Seeking 制御について述べる。

2.2 問題設定

制御対象 (プラント) として未知パラメータを含む次の 1 入力 1 出力の非線形システムを考える。

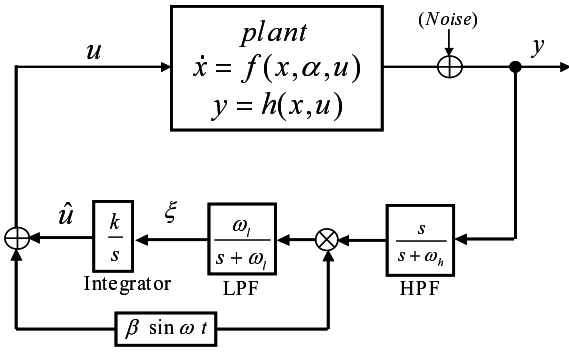
$$\dot{x} = f(x(t), \alpha(t), u(t)) \quad (1)$$

$$y(t) = h(x(t), u(t)) \quad (2)$$

2009 年 7 月 10 日受理

* 電気電子工学専攻

** 博士前期課程電気電子工学専攻



図－１ 標準型 Extremum Seeking 制御

ただし、 t :時刻、 $x \in R^n$:状態ベクトル、 $u \in R$:入力、 $y \in R$:出力、 f と h :未知の非線形関数、 $\alpha \in R$:未知パラメータ。
評価関数は(1)式の平衡点状態 $\{z: f(z, \alpha, u) = 0\}$ に対し

$$J(u) = h(z, u) \quad (3)$$

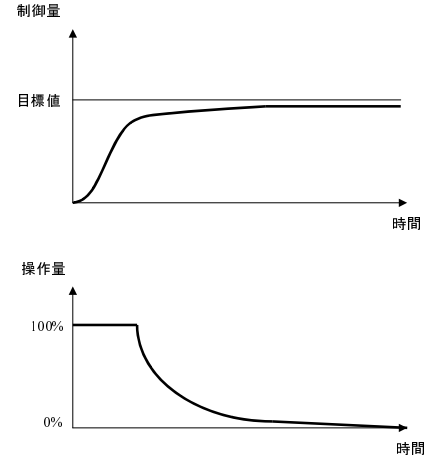
で与えられる。図－１は標準型 Extremum Seeking 制御のフィードバック構成図を示す。ただし、 $k > 0$ 、 $\beta > 0$ 、 $\omega_l \leq \omega_h$ である。ここで制御量は、

$$u(t) = \hat{u}(t) + \beta \sin \omega t \quad (4)$$

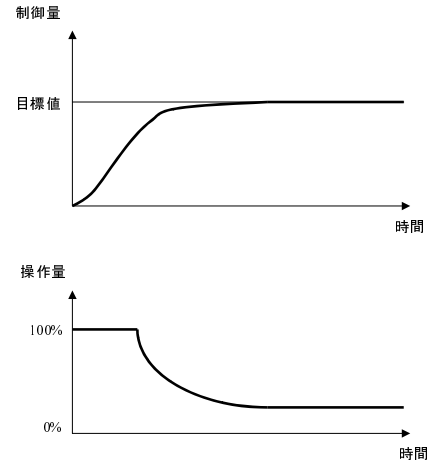
である。強制振動項 $\beta \sin \omega t$ は、 f や h のシステム変化の場合にも、常に $J(u) \leq J(u^*)$ となる最適値 u^* へ制御量 u を移行するための常時監視機構である。 $s/(s + \omega_h)$ は高域フィルタで、 y の直流成分を消去する。次に、 $\{s/(s + \omega_h)\} \cdot y$ に含まれる $\sin \omega t$ の項に、 $\beta \sin \omega t$ が乗じられると直流成分 ξ が発生し、これを低域フィルタ $\omega_l/(s + \omega_l)$ で抽出する。積分器 k/s によって、 $\hat{u} = (k/s)\xi$ の傾きが決定され、 \hat{u} が最適値 u^* の方向へ移動する。標準型 Extremum Seeking 制御は構造が簡単で実用的であるが、制御量 u の最適値 u^* への移行に時間がかかる。これを短縮するため次にPID型加速器補正を導入した改良型 Extremum Seeking 制御を考える。

3. PID 動作

PID動作は比例動作と積分動作と微分動作を組み合わせたものである^{6)～11)}。比例動作は操作量を最適値と現在値との差に比例した大きさとなるようにして、



図－２ P 動作の制御量とその操作量

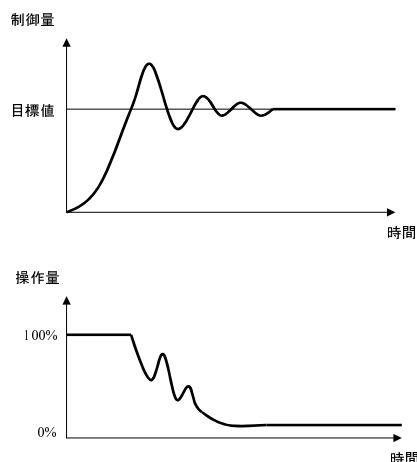


図－３ PI 動作の制御量とその操作量

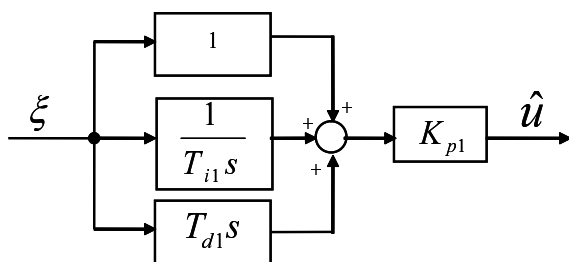
徐々に調節する制御である(図－２参照)。

しかし、実際には制御量が最適値へ近づくと操作量が小さくなりすぎ、最適値に極めて近い制御量の状態で安定するため、それ以上細かく制御出来ない。つまり、制御量が最適値へ完全には収束せず、わずかな誤差である残留偏差が残る。この残留偏差をなくすために適用されるのが積分動作である。残留偏差を時間的に累積し、ある大きさになった時点で操作量を増加させ、偏差を無くすよう動作させる(図－３参照)。このように、比例動作に積分動作を加えたものをPI動作という。

このとき、積分制御は偏差を足していくため時間がかかる。このため現在値が設定値からずれた場合、PI動作のみだとある程度時間が経ってからでないとシステムが反応しない。この欠点を補うために微分制御は



図－4 PID 動作の制御量とその操作量



図－5 PID 動作の基本構成

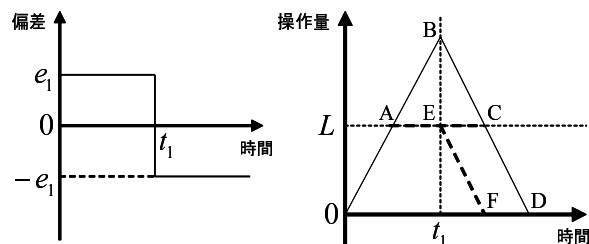
偏差の変化率、つまり偏差が変化する速度を捉え、これに比例した操作量をだすのが微分動作である。このように PI 動作に微分動作を加えたものが PID 動作である (図－4 参照)。

図－5 に、PID 動作の基本構成図を示す。

4. PID 型加速器補正

4.1 概要

標準型 Extremum Seeking 制御は最適運転点探索の際に、オーバーシュートを原因とする最適運転点移行時間の遅延がみられる。また図－5 に示した PID の基本形は I 動作によるオーバーシュートが起りやすい。そこで本節ではオーバーシュート対策の為に PID の改良型の一つである自動整合型 PID⁹⁾ を導入する。

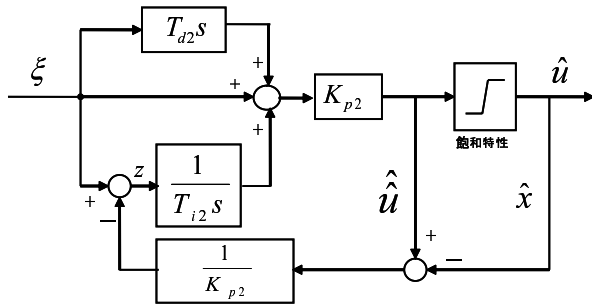


図－6 リセットwindアップ

4.2 リセットwindアップ

現実の操作量には上下制限が課せられることが多い。例えば、弁の全開と全閉などである。偏差の積分量がこの上下制限を越えたときに生じるのがリセットwindアップと呼ばれる問題である。開ループで積分動作だけを考え、入力である偏差が図－6 のように変化する場合を考える。上下制限がなければ、操作量は OABCD と推移する。当然ながら、偏差の積分量が増加するときには、それに比例して操作量も増加し、偏差の積分量が減少するときには、それに比例して操作量も減少するという、積分動作の本来の機能が果たされている。操作量が図の L という値で飽和する場合、積分器の出力は依然として OABCD と変化するが、実際の操作量は OAECD と推移することになる。このとき、AE 間では、偏差の積分量の増加に比例して操作量も増加するという機能が失われているが、それは制約のためやむを得ない。一方、EC 間では、偏差の積分量が減少するにもかかわらず、操作量は減少しない。それは、積分器出力が AB と巻き上げ (windアップ) られているため、BC と巻き戻されるまでは、偏差の積分量の減少に比例して操作量も減少するという機能が回復しないためである。

開ループの場合について説明したが、フィードバック制御系に組み込まれた場合も同様で、巻き戻しの期間だけ本来の機能の回復が遅れ、そのために偏差の整定が遅れ、オーバーシュートも大きくなる傾向がある。もちろん、制約条件内に戻る方向の積分は継続する。図－6 でいえば、A で積分が停止され、積分器出力は操作量と同じく AE と推移する。偏差の減少が始まる E で本来の機能を回復して、その後、操作量は EF という軌跡をたどることになる。



図－7 自動整合型 PID の構成図

4.3 自動整合型 PID

リセットワインドアップ対策として、操作量が上下限制約の範囲内であるときには通常の PID として機能し、操作量が上下限制約値に達した時点でそれを越える方向の積分機能を停止する自動整合型 PID を用いる。自動整合型 PID の構成図を図－7 に示す。図－7 において操作量が上下限制約の範囲内であるときは $\hat{u} = \hat{x}$ である。

5. PID 型加速器補正を有する Extremum Seeking 制御

標準型 Extremum Seeking 制御系に PID 動作を付加することで、最適運転点探索時間の短縮を試みる。そこで本節では、PID 型加速器補正を有する改良型 Extremum Seeking 制御を提案する。

標準型 Extremum Seeking 制御における図－1 の積分器の代わりに PID 動作器を設置する。その構成図を図－8 に示す。なお、その他の強制振動項、高域フィルタ、低域フィルタのパラメータは標準型 Extremum Seeking 制御のものと同じである。

6. 数値シミュレーション

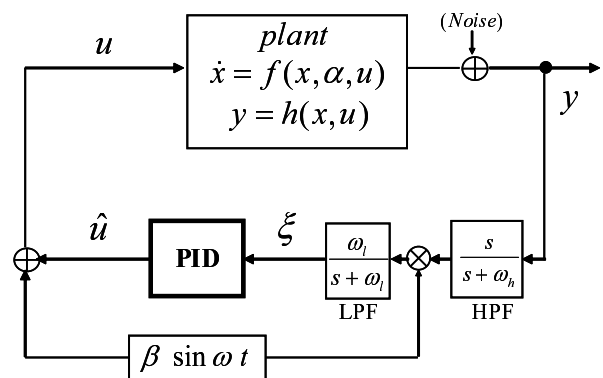
6.1 制御対象

連続攪拌生成物反応器内の生物生産最大化問題を考える。次式で示される Monod model 型非線形システム^{1)～5)}を考える。

$$\dot{x}_1 = f_1(x, \alpha, u) = x_1 \left(\frac{x_2}{(\alpha + x_2)} - u \right) \quad (5)$$

$$\dot{x}_2 = f_2(x, \alpha, u) = u(1 - x_2) - \frac{x_1 x_2}{(\alpha + x_2)} \quad (6)$$

$$y = h(x, u) = x_1 \cdot u \quad (7)$$



図－8 PID 型加速器付 Extremum Seeking 制御

ただし、 x_1 : 生物濃度、 x_2 : 基質濃度、 u : 希釈率、 α : 飽和定数、 y : 生物生産率である。ここで、シミュレーション実験として飽和定数 α を次のように設定した。

<ステップ入力タイプ>

$$\begin{aligned} \alpha_1 &= 0.02 & (0 \leq t < 500) \\ &= 0.1 & (500 \leq t < 1000) \\ &= 0.02 & (1000 \leq t < 1500) \\ &= 0.1 & (1500 \leq t < 2000) \\ &= 0.02 & (2000 \leq t < 2500) \end{aligned}$$

<ランプ入力包含タイプ>

$$\begin{aligned} \alpha_2 &= 0.02 & (0 \leq t < 500) \\ &= 0.1 & (500 \leq t < 1000) \\ &= 0.00016t + 0.26 & (1000 \leq t < 1500) \\ &= 0.02 & (1500 \leq t < 2500) \end{aligned}$$

パラメータを以下に設定した。 $\beta = 0.05$, $\omega = 0.19$, $\omega_h = 0.2$, $\omega_l = 0.1$, $k = 0.6$, $K_{p2} = 1.85$, $T_{i2} = 0.95$, $T_{d2} = 2.5$, 飽和特性 $7.2 < \hat{u} < 8.6$, 初期値 $\hat{u}(0) = 0.6$, $Noise\ power = 0$ or $Noise\ power = 0.0001$ また、 $Noise\ power$ とは白色雑音の強度を示し、パワースペクトル密度 (PSD) の高さを表している。

6.2 シミュレーション結果

シミュレーションにおける、飽和定数 α_1 のステップ入力タイプの結果を図－9、図－10、図－11、図－12 に示す。また、飽和定数 α_2 のランプ入力包含タイプの結果を図－13、図－14、図－15、図－16 に示す。

6.3 シミュレーション結果の考察

図－9, 図－10 は、ステップ入力タイプにおいて *Noise* を付加していない Extremum Seeking 制御の制御量 u と出力 y において従来の標準型 Extremum Seeking 制御との比較を示している。どちらの図においても、0 秒付近において従来の標準型が最適運転点移行に時間がかかっているのに対し、本手法は素早く最適運転点に移行していることが分かる。500 秒, 1500 秒付近において標準型はオーバーシュートして最適運転点移行に余計な時間がかかっているのに対し、本手法はほぼオーバーシュートしていないことが分かる。また、1000 秒, 1500 秒付近では標準型がなだらかに最適運転点に移行するため非最適運転時間が長いのに対し、本手法は素早く最適運転点に移行した。

図－11, 図－12 は、ステップ入力タイプにおいて *Noise* を付加した Extremum Seeking 制御の制御量 u と出力 y において標準型 Extremum Seeking 制御との比較を示している。*Noise* 環境下においてもステップ入力に対する本手法の優越性が確認できた。

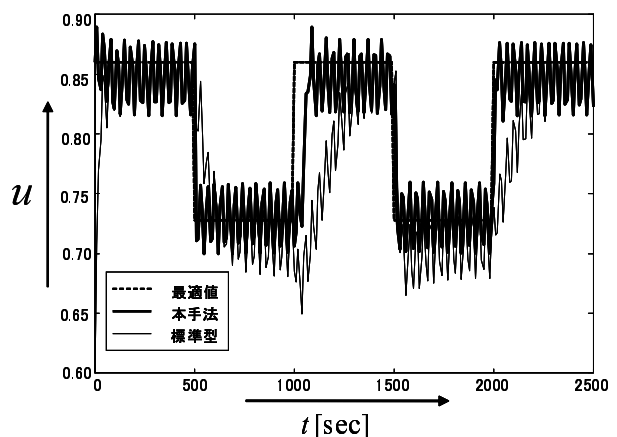
図－13, 図－14 は、ランプ入力包含タイプにおいて *Noise* を付加していない Extremum Seeking 制御の制御量 u と出力 y において従来の標準型 Extremum Seeking 制御との比較を示している。0 秒から 1000 秒まではステップ入力と同じであるため、先に述べたような本手法の優越性が確認出来る。また、1000 秒から 1500 秒のランプ入力に対しても、本手法は標準型より最適運転点に近づいていることが分かる。

図－15, 図－16 は、ランプ入力包含タイプにおいて *Noise* を付加した Extremum Seeking 制御の制御量 u と出力 y において従来の標準型 Extremum Seeking 制御との比較を示している。*Noise* 環境下においてもランプ入力に対する本手法の優越性が確認できた。

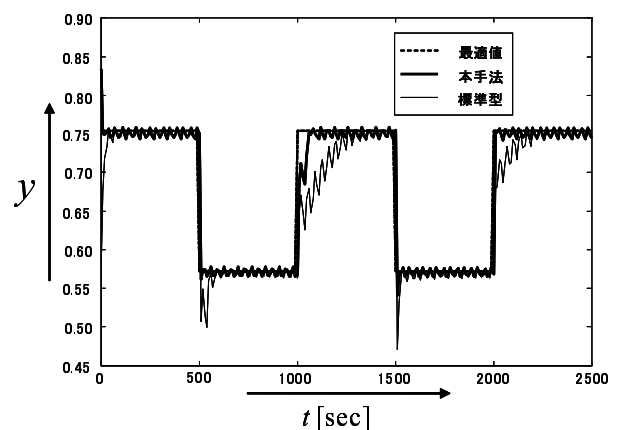
7. おわりに

本報告は標準型 Extremum Seeking 制御の改良型を提案した。非線形システムに対して未知の最適運転点探索時間短縮のため、PID 制御においてオーバーシュート対策を施した自動整合型 PID を PID 型加速器補正として標準型 Extremum Seeking 制御系に合成した構造である。

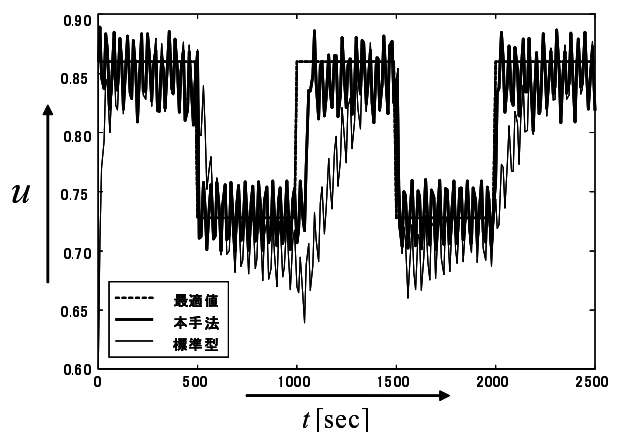
シミュレーション結果より、本手法は、標準型 Extremum Seeking 制御より最適運転点探索時間が短縮され、オーバーシュートも軽減されていることを確認



図－9 α_1 に対する u の時間変化 (*Noise* 無)



図－10 α_1 に対する y の時間変化 (*Noise* 無)



図－11 α_1 に対する u の時間変化 (*Noise* 有)

した。今後の課題として、PID 動作のパラメータの適切な選定や、更なる最適運転点探索時間短縮、他のシステムへの適用などがあげられる。

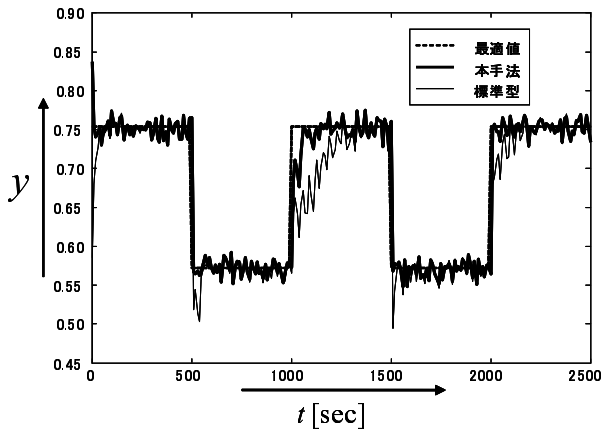


図-12 α_1 に対する y の時間変化 (Noise 有)

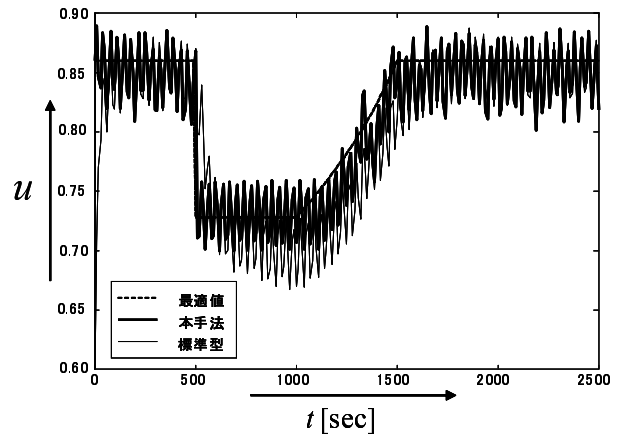


図-15 α_2 に対する u の時間変化 (Noise 有)

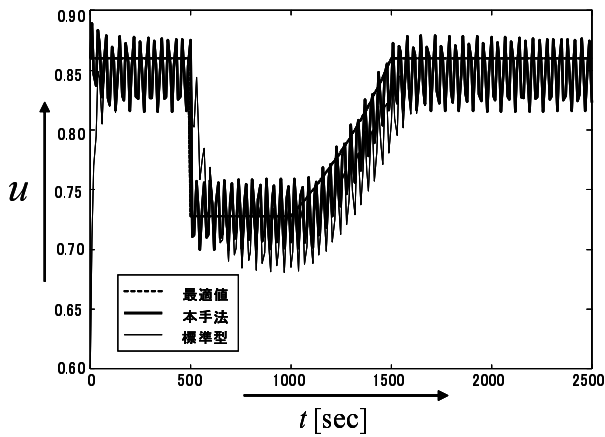


図-13 α_2 に対する u の時間変化 (Noise 無)

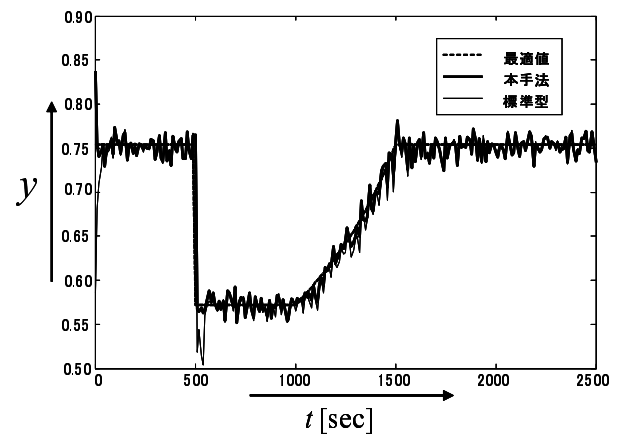


図-16 α_2 に対する y の時間変化 (Noise 有)

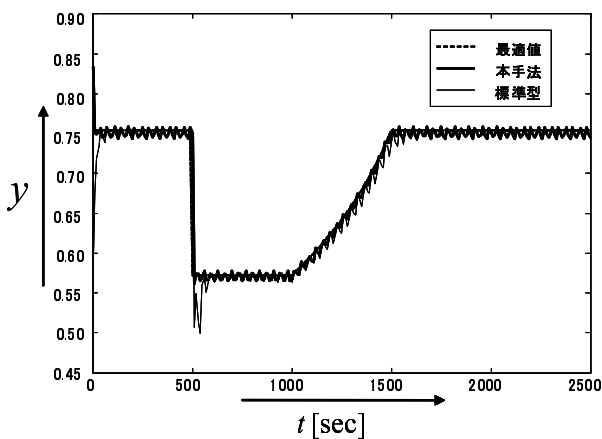


図-14 α_2 に対する y の時間変化 (Noise 無)

参考文献

- 1) K. B. Ariyur, M. Krstić : Real-Time Optimization by Extremum Seeking Control, A John Wiley & Sons Inc. Publication, pp.99-117 (2003).
- 2) 高田、上原、八野:PI型補正を有する Extremum Seeking 制御, 第 26 回計測自動制御学会学術講演会予稿集, pp.175-176 (2007).
- 3) 高田、上原、八野:PID型補正を有する Extremum Seeking 制御, 第 61 回電気関係学会九州支部連合大会講演論文集, 12-2P-03 (2008).
- 4) 高田、田村、八野:加速器付加による Krstić 型 Extremum Seeking 制御の改善, 第 57 回電気関係学会九州支部連合大会, p.382 (2004).
- 5) 高田、田村、菊永、八野:Krstić 型 Extremum Seeking 制御の改良形について, 第 48 回システム制御情報学会, pp.283-284 (2004).

- 6) 山本 重彦、加藤 尚武：PID 制御の基礎と応用（第2版），朝倉書店（2005）.
- 7) 須田 信英：PID 制御，朝倉書店（1992）.
- 8) 相良 節夫：基礎自動制御，森北出版（2004）.
- 9) 加納 学：MATLAB を利用した制御系設計—発展編—，京都大学大学院工学研究科化学工学専攻（2001）.
- 10) The Math Works：MATLAB The Language of Technical Computing，サイバネットシステム株式会社（2004）.
- 11) The Math Works：SIMULINK Dynamic System Simulation for MATLAB，サイバネットシステム株式会社（1997）.