

ダイナミックシステムシミュレータの開発と応用

Development of Dynamic System Simulator and Its Application

今野 雄 介^{*(1)} 塩谷 政典^{*(2)} 植山 高次^{*(3)}
 Yuusuke KONNO Masanori SHIOYA Takatsugu UEYAMA

抄 録

産業システムソリューション事業におけるプロセス制御系開発ツールとして、製鉄事業で培った制御・シミュレーション技術を生かし、汎用ダイナミックシステムシミュレータを開発した。本シミュレータは、常微分方程式、差分方程式及び代数方程式で表される一般の動的システムを解析することが可能である。更に、強力な数値解析機能を有するだけでなく、ブロック図表現の分かりやすいユーザインターフェースを用いており、容易に動的システムのシミュレーションを行うことが可能である。本稿では、シミュレータの特徴と構成及び応用としての熱間圧延プロセスシミュレータについて述べた。

Abstract

A general purpose dynamic system simulator has been developed as a tool for developing the process control system in an industrial system solution business, by making the most use of control and simulation techniques fostered in steelmaking business. This simulator can solve general dynamic systems represented by ordinary differential equations, difference equations and algebraic equations. This simulator has, not only a powerful numerical analysis function, but an easy-to-use graphic user interface which readily enables to simulate dynamic systems. This paper presents the features of the simulator and steel rolling process simulator as its application.

1. 緒 言

プロセスにおける様々な制御系を開発する際には、プロセスの挙動について十分に理解をする必要がある。しかし、操業条件やコスト等の問題により現場での実験を行うのは容易ではない。例え、パイロットプラントを用いて実験する機会に恵まれたとしても、検証できるケースは限られてしまうのが現状である。

このような状況の下で、

- 1) プロセスの動きを理解し、
- 2) 考えついた制御方法の有効性を確認したり、
- 3) 発想を転換したりしながら研究開発を進めたり、
- 4) 若手技術者にプロセス制御の疑似体験をさせながらその育成を図るにも、

計算機上に構築されたプロセスシミュレーターは不可欠である。

従来、個々のプロセスについてシミュレーションプログラムがFORTRAN等の汎用プログラム言語で作成されてきたが、これらの保守、改造は多くの時間とマンパワーを要していた。また、制御技術者は本来のシミュレーションに取り掛かる前に、コンピュータプログラミングを学ばなければならず、シミュレータを気軽に扱うわけには行かなかった。

筆者らは、上述のような従来型シミュレータの問題を解決する産

業ソリューション事業の有力なツールとして、対話型ダイナミックシステムシミュレーションツールを独自に開発してきた。

本稿では、シミュレータの概要及びその応用例として熱間圧延(以下熱延と略す)プロセスシミュレータを示す。

2. シミュレータの概要

2.1 設計思想

制御系設計において、新しいアイデアを創出するためには、現場のプロセスメカニズムをよく理解することがスタートとなる。プロセスで起こっている物理現象を正確に把握し、創造力を豊かにして、頭の中にダイナミックなイメージを描けることが重要である。しかしながら、実プロセスの観測のみでは、(1)内部で起こっている現象をほとんど観測できない、(2)思うように操作できない、(3)自由に実験ができない、(4)定常的な操業では問題点を見出しにくい、などの問題があり、本質的な理解を得ることはできない。これらの問題を解決するには、コンピュータ上でプロセス及び制御系を動的に解析するダイナミックシミュレータが有力な武器となる。

対象とするプロセスに多くの条件・操作を与えてシミュレーション実験を繰り返すことにより、プロセスの過渡的な挙動を微視的に知ることができる。このことでプロセス制御をイメージでき、新たな発想を生み出す想像力を養うことができる。また、シミュレータ開

*⁽¹⁾ 技術開発本部 プロセス技術研究所 計測・制御研究部 研究員

*⁽²⁾ 技術開発本部 プロセス技術研究所 計測・制御研究部

主任研究員

*⁽³⁾ 技術開発本部 プロセス技術研究所 計測・制御研究部 部長

発を行うことによっても、開発者の頭の中に数式モデル間の関連が整理されるため、プロセスメカニズムの理解に大きな助けとなる。

このように、プロセスシミュレーションは、現場実験の代替になるだけでなく、“プロセスの理解”を助け、イメージを深めることにより、性能の飛躍的向上を生む制御アイデアへの“発想転換”を促すという大きな役割を持つ。プロセスシミュレータが以上のような役割を果たすためには、次のような条件を満たすべきである。

2.1.1 高いビジュアル性と疑似体験性

シミュレーションにより、現実のプロセスで起こっている現象をイメージできるためには、実際のプロセスを運転していると思わせる疑似体験性が得られることが必要である。このためには、美しい画像により現象を視覚的に確認できるビジュアル性を持ち、かつシミュレータへの介入が自由にできる仕組みが望ましい。

2.1.2 容易に理解できるシミュレータ構造

コンピュータ実験を目的とするシミュレータでは、発想されたアイデアをシミュレータ上で時間を置かず簡単に実現できることが望ましい。そのためには、シミュレータ構造の理解が容易にでき、簡単に改造できる必要がある。このことは、従来のソフトシミュレータがその複雑で理解しにくい構造のために、開発者が居なくなると使われなくなったり、メンテナンスされなくなったりすることに対する解決策にもなる。

2.1.3 制御系設計が容易な構造

通常対象とするプロセスは、非線形でかつ時間的に特性が変化している。一方、制御理論は通常線形時不変モデルを対象に組み立てられており、シミュレータを用いて直ちに制御系設計をすることは非常に困難で、手作業による線形化により、多大な労力と時間を要している。このことは、思考の中断と開発の非効率さを招いている。発想したアイデアを実現するコントローラをシミュレータを用いて簡単に設計でき、そのコントローラを組み込んだプロセスの挙動を直ちにシミュレータで確認できることが望ましい。

2.2 本シミュレータの特徴

以上の目標に沿って開発したシミュレータの特徴を以下に挙げる。

2.2.1 ビジュアル・インタラクティブ操作

図1は本シミュレータによるシミュレーション画面の例である。

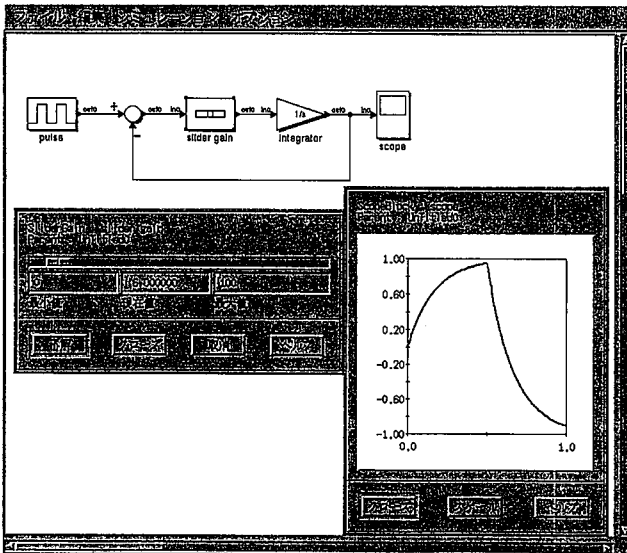


図1 ポリュームとグラフ

ここでは、ボリュームとグラフが表示されており、ボリュームの動きによる系の変化は直ちにグラフに反映される。このようなグラフやメータにより、時々刻々の計算結果が画面以上にビジュアルに描画される。

更に、マウスやキーボードでボリュームやスイッチを操作し、手動介入やコントローラの切替え、制御系の調整などをシミュレータの動作を逐次観察しながら行うことができる。このインタラクティブ操作環境により、ユーザは高いビジュアル性と疑似体験性を得ることができる。

2.2.2 モジュール型構造

シミュレータの内部構造を容易に理解できることを目的とし、本システム上で開発されるシミュレータでは、個々の装置（モデル）を出力端子を持ったブロックで表現し、装置間の関連はそれらを結ぶ接続線で表現した。また、複数のブロックを1つのモジュールとして表せ、大規模で複雑なシミュレータでも階層的に理解しやすい構造で構築できるようにした。図2は、枠で囲んだブロック群を右下のような一つのモジュールで表せることを示している。

このように、シミュレータがプログラム言語ではなくブロック（モジュール）と接続線で表現されるため、個々の装置とそれらの関連が視覚的に理解でき、開発者以外の人でもその構成を理解しやすい。また、個々のブロックは独立しているため、一つのブロックの修正や追加がシミュレータ全体の修正に及ぶことが無く、部分的な改造・機能追加が容易であり、他のシミュレータへの再利用性が高い。更に、加算器、積分器など利用頻度の高いブロックは図3に示すような組み込みブロックとしてあらかじめ用意されており、部品の標準化が図られている。

2.2.3 制御系設計機能

シミュレーション実行中に任意の時点でシミュレータを停止し、自動的にこの状態での線形時不変モデルを計算する線形モデル自動生成機能を開発した。このモデルを当部開発の制御系設計ツールに取り込むことにより、コントローラの設計が容易に行える。

そして、設計したコントローラを自動的にシミュレータに取り込み、実行することも可能である。この機能により、制御系設計とシミュレーション実行を遅滞なく行え、思考を中断せずにコンピュータ実験を続けることができる。

2.2.4 拡張性

図3の組込みブロックでは機能が不足する場合、ユーザはCまたはC++で新しいブロックを記述することができる。新規ブロックはコンパイルの後、ダイナミックリンクライブラリとしてシミュレ-

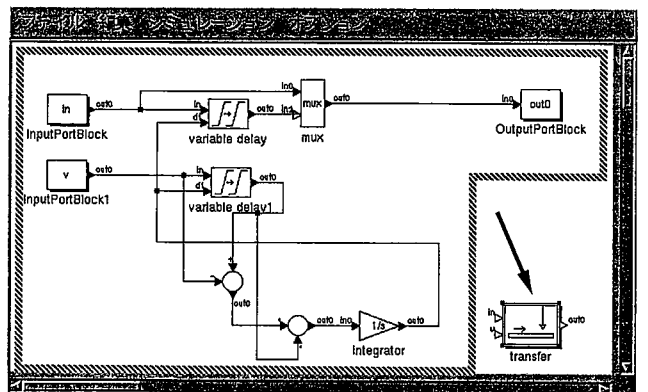


図2 モジュール化機能

ション実行時に動的に結合される。そのため、通常のプログラムのようにリンクしなおす必要はなく、ブロックの開発が容易な構成となっている。更に、シミュレータ本体の製作に当たっては、オブジェクト指向によるソフトウェア開発技術を用いているため、シミュレータ自身の拡張性・再利用性が高いソフトウェアとなっている。

2.3 数値解析手法

本シミュレータが対象とするシステムは、一般に次のような形で記述される。

$$\frac{dx_c}{dt} = F(x_c, x_d, x_a, t) \quad \dots\dots(1)$$

$$x_d(i+1) = G(x_d(i), x_c, x_a, t) \quad \dots\dots(2)$$

$$H(x_c, x_d, x_a) = 0 \quad \dots\dots(3)$$

(1)式は微分方程式、(2)式は差分方程式、(3)式は代数方程式であり、この3式を同時に解く必要がある。これらのうち、微分方程式と代数方程式は何らかの数値解法を用いて近似的に解く必要があり、以下に解法の概要を述べる。

2.4 微分方程式

微分方程式部分(1)式を解くに当たってはいろいろな解法があるが、それらは陽解法と陰解法に大別される。両者には、それぞれ次のような特徴がある。

陽解法：ある時刻の値から次の時刻の値が直ちに計算できるので、プログラム作成が容易。大きな時間刻みでは数値的な不安定性を起しやす。

例：前進オイラー法 $x_{i+1} = x_i + hf(x_i)$

(x_i ：現時刻の解, x_{i+1} ：次の時刻の解, h ：時間刻み幅, $f(x)$ ：被積分関数)

陰解法：大きな時間刻みでも不安定になりにくい性質を持つ。各ステップごとに方程式の解を求める必要があるため、1ステップ当

りの計算量は増加する。

例：後退オイラー法 $x_{i+1} = x_i + hf(x_{i+1})$

本シミュレータでは陽解法3種類、陰解法3種類からユーザが積分法を選択できる。

陽解法では安定性の制約から、また陰解法でも積分誤差の観点から、時間刻み幅を適切に選択することが重要である。しかし、問題に応じた適切な時間刻みをユーザが決めるのは困難であり、自動的に決めるのが望ましい。また、解があまり変化しないところでは、時間刻み幅は粗くてもよいが、解が急激に変化するところでは、刻み幅を細くするべきである。これらの理由から、すべての積分法について、時間刻み幅の自動調整を行う。ユーザは、刻み幅の最小・最大値と許容誤差を入力することで、許容誤差を満足する範囲で最大の刻み幅となるように時間刻み幅が自動調整される。

ステップ幅自動調節の例として(4)式で表される Van der Pol の非線型微分方程式¹⁾を解いた例を示す。

$$\dot{x} = 50(1-x^2)\dot{x} - x, \quad x(0) = 1, \quad \dot{x}(0) = 1 \quad \dots\dots(4)$$

この微分方程式は、陽解法ではステップ幅を小さくしないと解けないことで知られている。

ここでは、利用可能な積分法のうち、陽解法の代表として Runge Kutta Fehlberg 4次5次(RKF45)²⁾、陰解法の代表として4次の後退微分公式(BDF4)²⁾の二つを最大刻み幅2秒、許容誤差0.001の条件で比較した。

図4において両者の解は一致しているが、この結果を得るために要した計算ステップ数は、RKF45は2727ステップ、BDF4は712ステップでBDF4のほうが格段に効率が良い。

このことは図5において、BDF4のほうが大きな時間刻み幅で計算

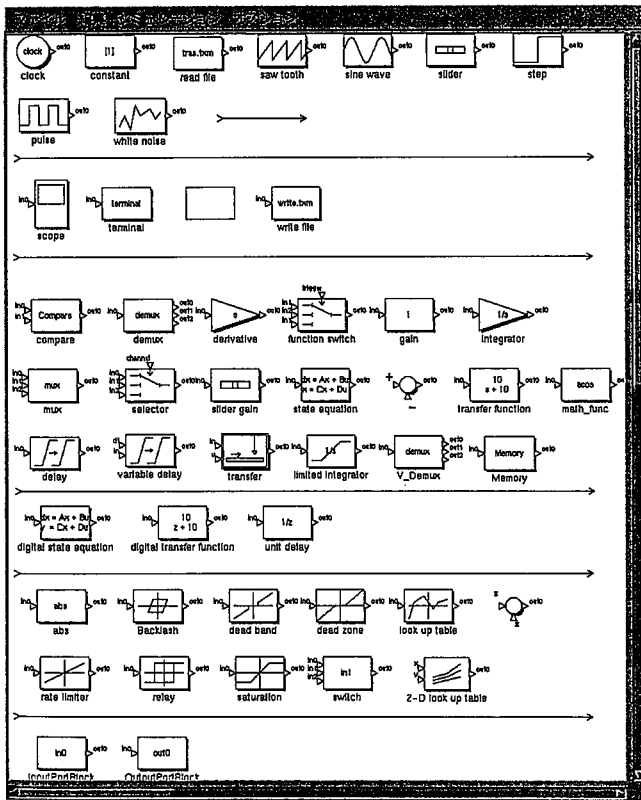
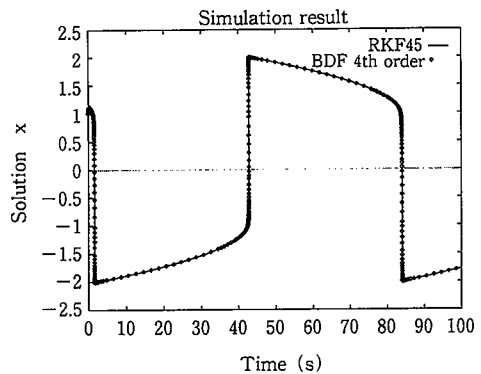


図3 組み込みブロック一覧



町4 Van der Pol方程式の解

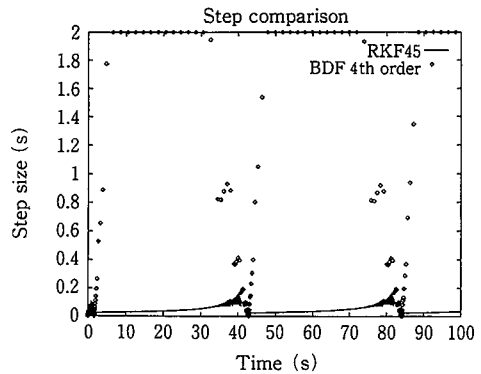


図5 時間刻み幅の比較

を行っていることから明らかである。また、両者共に解の変化が急なところでは、時間刻み幅が小さくなり、刻み幅自動調整の効果が出ていることが分かる。

2.5 代数方程式

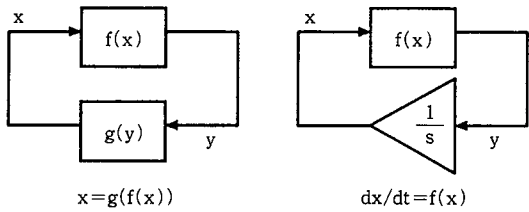
シミュレーション中に解くべき代数方程式は、シミュレーションモデル自体に含まれる場合(代数ループ：図6参照)と、前述の微分方程式の陰解法に伴うものがある。解法として、単純反復法あるいはNewton法が考えられるが、収束性の良いNewton法及びその変種を採用している。Newton法の反復過程で現れる連立一次方程式は、直接法(LU分解)で解いている。ユーザは、最大反復回数、収束判定条件等を変更することが可能である。

3. シミュレータの構築例(熱延シミュレータ)

現在までに、本シミュレータの性能検証のため、熱延仕上げスタンドのタンデムシミュレータを製作し、このシミュレータを用いて様々な制御系を開発してきた^{3,4)}。以下に、熱延シミュレータの構造と、制御系開発の一例として周波数整形型AGCを説明する。

3.1 熱延シミュレータの構造

熱延シミュレータは、図7に示すような機能毎のモジュールにより構成されている。モジュールをコピーすることで、多スタンド化が容易に行える構成となっている。各モジュールの機能は大別して



With algebraic loop Without algebraic loop
図6 代数ループ

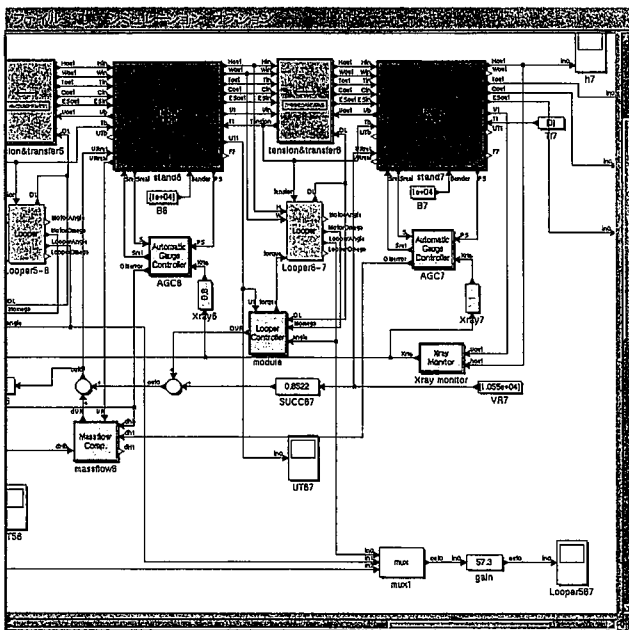


図7 熱間圧延7スタンドシミュレータの一部

以下の8種である。

- ・ミルモデル
- ・ルーパーモデル
- ・ルーパー制御系
- ・張力モデル
- ・スタンド間クリープモデル
- ・板厚制御系
- ・移送モデル
- ・その他制御機能

この熱延シミュレータは、被圧延鋼板中心部の長手方向の変数の挙動を時間の経過と共に、1)板厚、2)板幅、3)エッジより特定点での板クラウン及び形状(伸び歪み差)、4)スタンド間単位張力、5)鋼板温度、6)鋼板速度、7)ロール周速度、8)压下位置、9)ルーパー角度の挙動などを確認できる。また、操業時のデータを取り込んだシミュレーションの実行も可能である。モデルの詳細については別文献⁵⁾に譲る。

3.2 制御系の開発

周波数整形型AGC^{4,6)}は、熱延仕上げ工程の板厚偏差要因である1)スキッドマーク、2)ロール偏芯の、2大外乱の板厚への影響を同時に低減することを目的として考案された。図9のような従来のBISRAやGauge Meter方式のAGCは、チューニングファクタと称する唯一のパラメータを調整することで板厚制御を行ってきた。しかし、圧延反力を基に、板厚偏差を検出して制御する方法では、両外乱に関して、

- ・スキッドマーク：圧延反力増加→板厚増加
- ・ロール偏芯：圧延反力増加→板厚減少

という性質があり、いずれか一方の影響を低減しようとすると、他方の影響を助長する結果となる。

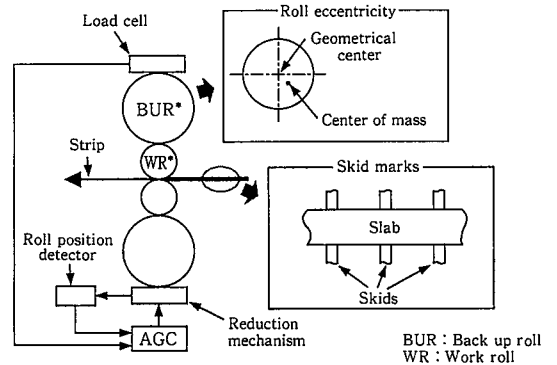


図8 熱間圧延プロセスにおける板厚外乱

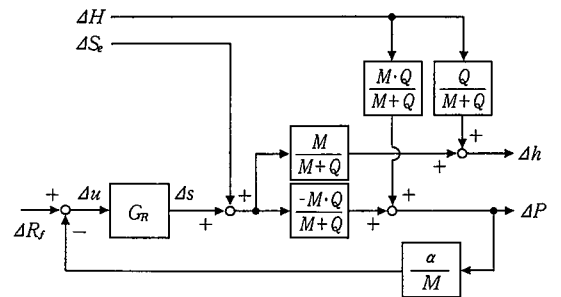


図9 BISRA型AGC

ところが、ロール偏心の周波数帯域が主に10 rad/s 以上であるのに対して、スキッドマークの周波数帯域は1 rad/s 程度であり、両外乱の周波数帯域は分離している。

この点に着目し、それぞれの周波数帯域でその周波数の外乱に応じた制御を行うのが周波数整形型 AGC である。周波数整形型 AGC のブロックは、図10で表され、 α_1 のループで速いロール偏心を除去し、 α_2 のループで遅いスキッドマークを除去する。

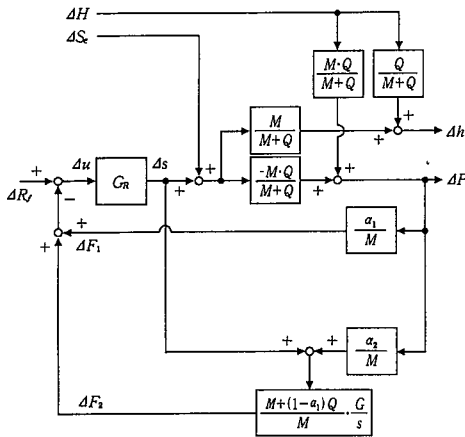


図10 周波数整形型 AGC

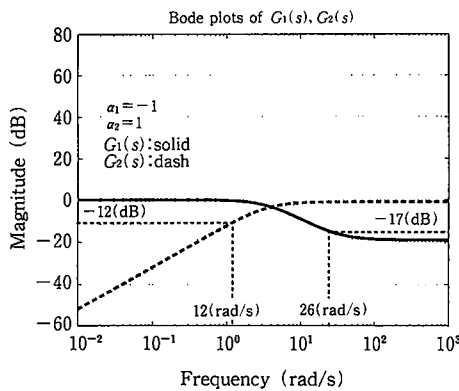


図11 $G_1(s)$, $G_2(s)$ の周波数特性

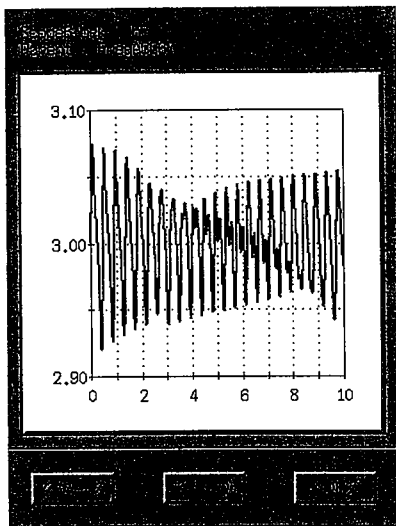


図12 BISRA-AGCの場合

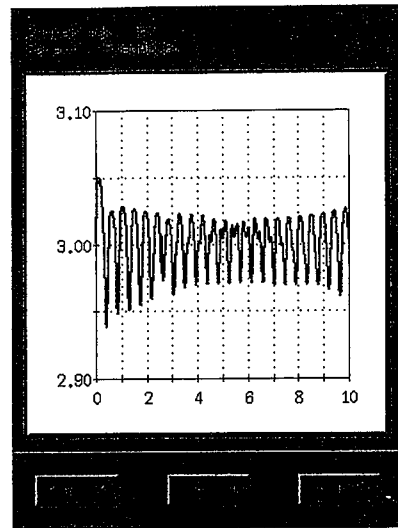


図13 周波数整形型 AGC の場合

この方式において $G_1(s)$:スキッドマーク外乱 (ΔH) から出側板厚偏差 (Δh) までの伝達関数、 $G_2(s)$:ロール偏芯外乱 (ΔS_0) から出側板厚偏差 (Δh) までの伝達関数を示したのが、図11である。

図12は、従来 AGC (BISRA) で、チューニングファクタ α_1 を 1.0 にした場合であり、ロール偏芯の影響が顕著である。

図13は、周波数整形型 AGC を導入し、チューニングファクタ α_1 , α_2 をそれぞれ -1.0, 1.0 にした場合であり、ロール偏芯の影響が約 60% に低減しているのが分かる。

4. 結 言

ダイナミックシステムシミュレータの概要を紹介した。本シミュレータは、プロセスシミュレータを分かりやすく、かつ使いやすく構築するためのツールとして極めて有効である。数値解析の観点からも効率の良いものとなっている。更に、熱延シミュレータと周波数整形型 AGC の開発を通じ、本シミュレータが大規模なプロセスの制御系設計ツールとして非常に有効であることが確認できた。

今後、数値解析機能の強化、高速化を行うと共に、産業システムソリューション事業に対応したプロセスモデルを構築し、ライブラリとして蓄積していく予定である。更に、離散事象システムシミュレーション機能を付け加え、スケジューリングからプロセスシミュレーションまで、工場全体を模擬できるシミュレーションツールへと発展させる予定である。

参 照 文 献

- 1) Hairer, E., Norsett, S.P., Wanner, G.: Solving Ordinary Differential Equations I. Springer-Verlag, 1987
- 2) 三井：数値解析入門。朝倉書店, 1985
- 3) 熱延ルーパー多変数非干渉制御。日本鉄鋼協会共同研究会, 第97回圧延理論部会, 1992.11
- 4) 熱延 2 自由度 AGC。日本鉄鋼協会共同研究会, 第94回圧延理論部会, 1991.9
- 5) 圧延ダイナミックシミュレータの開発。日本鉄鋼協会共同研究会, 第112回制御技術部会, 1994.11
- 6) An Application of Two-Degree-of-Freedom Control to Automatic Gauge Control. 1992, American Control Conference, p.806-810