

# 熱延におけるルーパー制御ゲイン調整の 人間モデルとその学習

今城 周也\*・小西 正躬\*<sup>2</sup>・今井 純\*<sup>2</sup>・西 竜志\*<sup>2</sup>

Human Model for Gain Tuning of Looper Control in Hot Strip Rolling

Shuya IMAJO, Masami KONISHI, Jun IMAI and Tatsushi NISHI

**Synopsis :** In recent years, many control systems have been developed and applied to hot rolling in order to improve the quality of the finished products. Due to the plant characteristics and the control system performance variations over time, human experts intervene and modulate control gains to maintain and to improve the control system performance. Through a breakthrough, we can expect to attain the fully automated modulation of control gains without human intervention. A neural network model representing gain modulating actions by human was developed for a looper height controller in hot strip mills. The developed neural network model is a recurrent type neural network (RNN) which calculates the appropriate PID gains of the looper height controller based on the modification data of human operations as training data. Further, a learning algorithm for the RNN model was developed to accelerate convergence of the gain modification process and to stabilize the looper movement. The neural gain tuning model was applied to the inter-stands looper height controller in hot strip mills. The usefulness of the developed model was checked through numerical experiments. From the experimental results, it was verified that the tuning actions by humans can be realized by the model. Through its learning mechanism, the model could also cope with disturbances such as changes in roll gap. This may lead to the stabilization of threading operations of hot strip mills.

**Key words:** neural networks; learning; gain tuning; identification model; PID controller; hot strip mills; loop controller.

## 1. はじめに

近年、各種の制御技術が開発、適用されプラント制御の自動化がますます進んでいる。しかし、機械の性能や製品種類の変化があるため、人間がプラントのコントローラを調整し、介入して制御性能を改善・維持している場合が多い。熱間圧延においても現代制御理論に基づく新技術が開発されており、スタンド間ルーパーも高度に自動制御化されている<sup>1-3)</sup>。しかし、時間の経過と共にプラントの特性や操業条件の変化があるため、コントローラは必要に応じて調整される。すなわち、熟練者が制御システムに介入し、ロール隙の開閉を行ったり、コントローラゲインを調整することで制御性能の改善を行っている。現在のところ、人間に代ってコントローラのパラメータを自動調整することは極めて困難と考えられる。一方、対象の入出力データのみから、その対象モデルの構造を学習できるニューラルネットワーク<sup>4)</sup>は人間によるコントローラのゲイン調整過程を近似的に表現することができる手法の一つであると考えられる。そこで、本研究では、ニューラルネットワークを用いてPIDゲイン調整を自動化することを目的として、リカレントニューラルネットワーク (RNN) を用いたゲイ

ン調整モデル<sup>5)</sup>を提案する。人間は制御結果を多面的に評価しつつ、過去の経験に基づいてゲイン調整を行っている。提案するRNNモデルもまた、過去に出力したゲインを記憶し出力に反映することができる。以下では、ルーパー高さ制御のコントローラのゲイン調整のRNNモデルについて述べる。さらに、操業データを加味したRNNモデルの更新のための学習アルゴリズムを述べる。最後に、これらの提案の効果を数値実験により確認する。

## 2. RNNを用いた人間の介入モデル

### 2.1 人間による制御系のゲイン調整

本研究では、RNNモデルを用い熱間圧延のスタンド間ルーパー制御において人間が制御ゲインの修正を行う動作のモデル化を検討する。Fig. 1にはPID制御によるプラント制御系が示されている。人間は、制御量 $y$ と目標値 $r$ を比較し、その偏差 $e$ の挙動から制御系の性能を評価し、必要に応じてPID制御ゲイン $k_p$ ,  $k_i$ , および $k_d$ を調整している。本研究では、熱間圧延のスタンド間ルーパー制御のPIDゲイン調整を対象として、RNNによる人間の介入動作のモデル化を検討する。

## 2.2 RNNモデルによるゲイン調整

### 2.2.1 RNNモデル同定用教師データの生成

Fig. 2には、圧延スタンドとスタンド間ルーパーの構成を示している。本研究では、熱延のルーパー制御系を対象に制御ゲインの調整モデルを構築した。ニューラルネットモデルの同定を行うためには、教師データとなる操業データが必要であるが、本研究では、シミュレータによるデータを操業データの代わりに用いている。まず、圧延機3機とルーパー2基からなる熱延ミル上流側ルーパー制御系のシミュレータを作成した。次に、このシミュレータを用いて、人間がルーパー挙動を観察して、次の圧延でのPIDゲインを調整することを繰り返し、RNNモデルの教師データ（調整実績データ）を生成した。シミュレーションの方法として、板厚制御が動作していない条件、かつマスフロー則で決定したロール回転速度を基準として、スタンド#1のロール回転速度に $-0.05\text{ m/s}$ の外乱を加えた条件下で行われた。この外乱が起こると、1基目のルーパーが持ち上がりにくくなる。人間は1基目のルーパー制御系に対して、同一圧延条件における試行錯誤により16回のPIDゲイン調整を行った。この様子を、Fig. 3(a)に示した。圧延回数1回目においては、ルーパーが持ち上がっていなかったため、各ゲインを大きくしていき、その結果、ルーパーのオーバーシュートが大きくなると、ゲインを小さくするという調整過程と

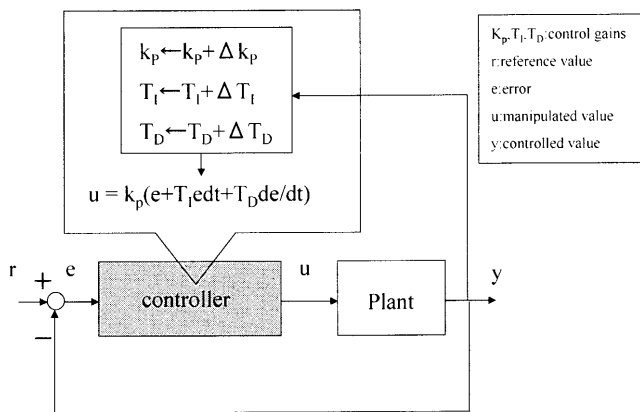
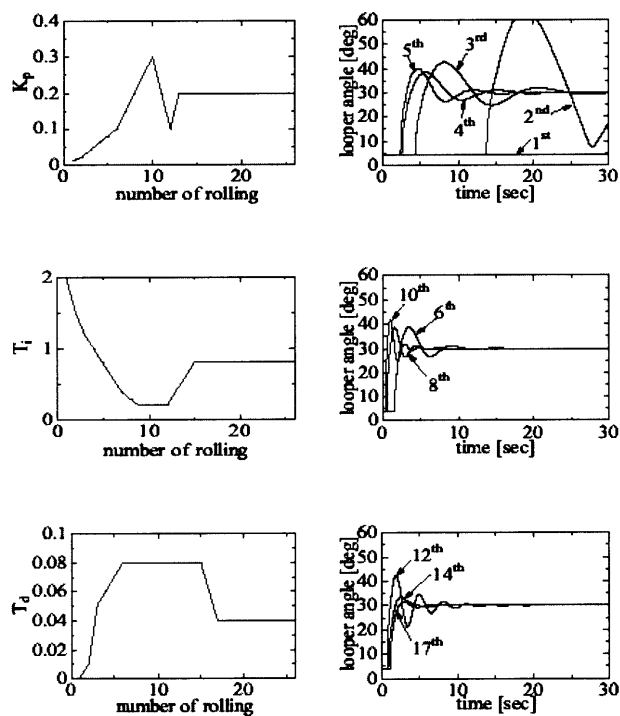


Fig. 1. Human intervention to control system.

なっている。このゲイン調整の結果、Fig. 3(b)に示すように、ルーパー挙動は理想的な状態に収束した。これらのデータを教師データとして蓄積し、RNNモデルの同定に使用した。さらに、同定されたRNNモデルを用いて、Fig. 4に示すように1基目と2基目のルーパー制御系のゲイン調整を行った。以下では、RNNモデルと具体的な入出力情報について説明する。

### 2.2.2 RNNモデル

人間が調整を行うとき、制御対象の挙動に対して、ある評価基準を持ち、その制御対象が理想的な挙動になるように調整を行う。またその際、過去の記憶も次の調整に利用すると考えられる。したがって、本研究ではリカレント型のニューラルネットモデルを利用したが、その構造をFig.



(a) PID gains

(b) looper angle

Fig. 3. Tuning results of PID gains by human.

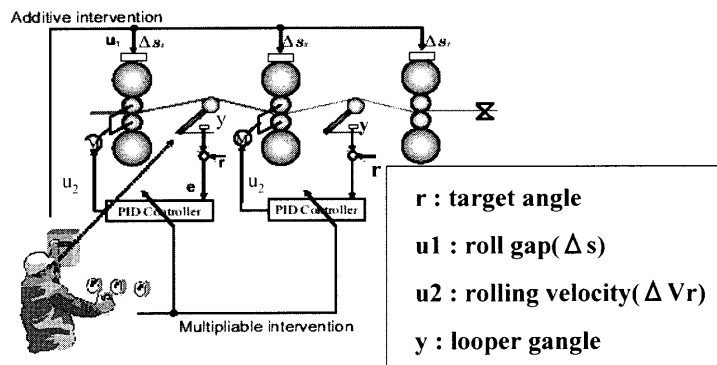


Fig. 2. Human intervention operation.

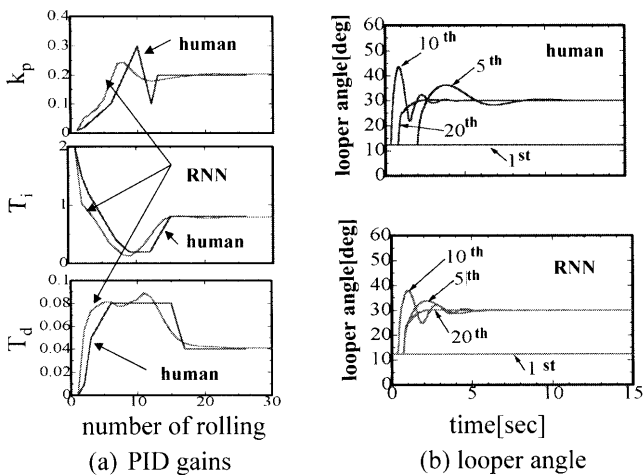


Fig. 4. Identified Results by RNN Model.

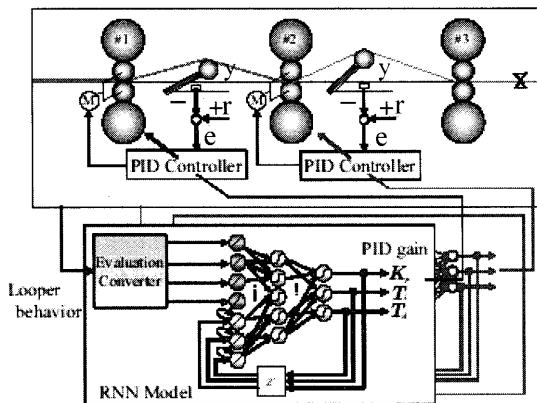


Fig. 5. Three stand hot strip mills with loop control.

5に示す。Fig. 5に示すように、RNNはn回目のループ挙動の評価値からn+1回目のPIDゲインを出力する。ネットワークは、3つの層から構成されている。まず入力層は、2つの部分から構成されている。1つは、外部からの入力を受け持つプラン層であり、もう1つは過去の制御ゲインを入力とする状態層である。入力数はニューラルネットの規模の増大を防ぐため、学習の面からは少数であるほうが良いと考えられる。しかしながら、入力数が少なすぎると種々のループ波形を誤認識する可能性が高くなる。本研究で提案するRNNモデルでは、入力数を少なくすることと、汎用性を持たせるため、以下に示すI<sub>1</sub>からI<sub>4</sub>のループ挙動の評価値をプラン層の入力としている。まず、ループ角度の目標値に対するオーバーシュート量、すなわち安定性を表す評価関数としてI<sub>1</sub>を考える。ループ角度の最大値をα<sub>max</sub>、目標値をα<sup>0</sup>とするとI<sub>1</sub>は、

$$I_1 = \left| \frac{\alpha_{\max}}{\alpha^0} - 1 \right| \dots\dots\dots(1)$$

で示される。ループが持ち上がらないときは、最大値が目標値を超えない場合も存在するので絶対値をとり正の値にする。つぎに、精度を表す評価関数としてI<sub>2</sub>とI<sub>3</sub>を考え

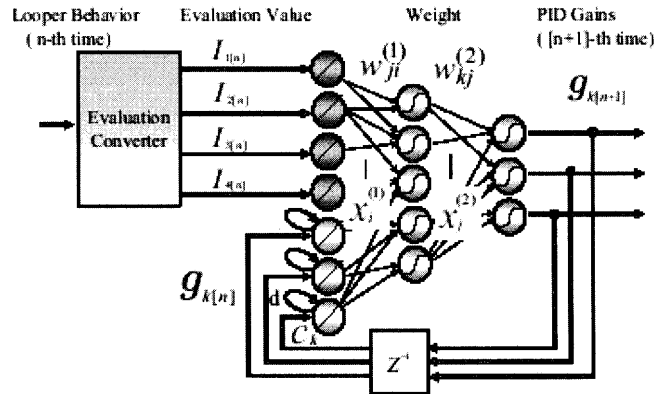


Fig. 6. Structure of the proposed RNN model.

る。I<sub>2</sub>は偏差の絶対値の積分として次式で定義する。

$$I_2 = \frac{1}{T} \int_0^T \left| \frac{\alpha(t)}{\alpha^0} - 1 \right| dt \dots\dots\dots(2)$$

ここで、α(t)は時刻tにおけるループ角度、Tは制御効果の評価時間である。この評価関数は、ループ角度が目標値との偏差が大きいほど、また偏差が存在する時間が長いほど、その値が大きくなる。I<sub>3</sub>は偏差の絶対値の時間重みつき積分として、以下の評価関数を考える。

$$I_3 = \frac{1}{T^2} \int_0^T t \left| \frac{\alpha(t)}{\alpha^0} - 1 \right| dt \dots\dots\dots(3)$$

この評価関数は、I<sub>2</sub>の評価関数と類似の特性を持つが、時間重みがかかっているため、偏差の存在する時間が後になればなるほど、その値は大きくなる。最後に、評価する時間Tの期間中における、ループ角度が最大になる時刻t<sub>p</sub>を考え、ループの速応性の評価関数I<sub>4</sub>として、次式を用いた。

$$I_4 = \frac{t_p}{T} \dots\dots\dots(4)$$

ただし、ループが一度も持ち上がらないときはt<sub>p</sub>=Tとする。この評価関数は、ループの立ち上がり時間の速さを評価する。なお、これらの評価関数は、目標角度や制御評価の時間に依存しないようにするため、またニューラルネットの学習の高速性の向上させるため、ループ角度の目標値α<sup>0</sup>および制御効果の時間Tにより正規化されている。人間が調整を行ったときのこれら評価関数の推移をFig. 6に示す。ループ挙動が理想状態に近づくにつれ、これらの評価関数値が減少する傾向を示すことがわかる。一方、評価関数I<sub>1</sub>とI<sub>2</sub>, I<sub>3</sub>, I<sub>4</sub>の間にはトレードオフの関係もあることがわかる。最終的には、4つの評価関数はすべて小さな値に落ち着いている。これらのことから、これら4つの評価関数を同時に考慮すれば、人間に近い調整ができるものと考えられる。Fig. 5に示すように、入力層のもう一方の部

分は状態層である。状態層は以下のように表され、過去に出力したゲインを入力  $c_k$  ( $k=1\sim 3$ ) として RNN モデルに蓄積する。

$$c_k = \sum_{m=1}^n d^{n-m} g_{k[m]}; \quad k=1\sim 3 \quad \dots\dots\dots (5)$$

$$g_1: K_p, g_2: T_i, g_3: T_d$$

ここで、 $d$  は0以上、1未満の正の定数で、忘却率を表す。その値が1に近ければ、RNNモデルに過去の履歴が反映するため、調整が収束しにくくなる。また、0に近ければ、以前に出力した値を忘却し、調整値を評価関数値のみに依存して大きく変化させてしまう。これらのことを考慮しつつ本研究では、忘却率  $d$  を0.6に設定した。入力層  $i$  番目のニューロン出力  $x_i^{(1)}$  は以下のように表される。

$$x_i^{(1)} = \begin{cases} p_i & 1 \leq i \leq 4 \\ c_{i-4} & 5 \leq i \leq 7 \end{cases} \quad \dots\dots\dots (6)$$

$p_i$  は以下のように、評価値  $I_i$  を線形変換した値である。

$$p_i = \frac{I_i - I_{i,\min}}{I_{i,\max} - I_{i,\min}}; \quad i=1\sim 4 \quad \dots\dots\dots (7)$$

ここで、 $I_{i,\max}$ ,  $I_{i,\min}$  は前もって設定される値である。

中間層  $j$  番目のニューロンの入力  $u_j^{(2)}$ 、および出力  $x_j^{(2)}$  はそれぞれ以下のように表される。

$$u_j^{(2)} = \sum_{i=0}^7 w_{ji}^{(1)} x_i^{(1)}, \quad x_0^{(1)} = -1; \quad j=1\sim 5 \quad \dots\dots\dots (8)$$

$$x_j^{(2)} = f(s_j^{(2)}) = \frac{1}{1 + e^{-s_j^{(2)}}}; \quad j=1\sim 5 \quad \dots\dots\dots (9)$$

ここで、 $w_{ji}^{(1)}$  は入力層の  $i$  番目のニューロンと中間層  $j$  番目のニューロン間の結合の重み係数である。

出力層の  $k$  番目のニューロンへの入力  $u_k^{(3)}$ 、および出力  $g_k$  はそれぞれ以下のように表される。

$$u_k^{(3)} = \sum_{j=0}^5 w_{kj}^{(2)} x_j^{(2)}, \quad x_0^{(2)} = -1; \quad k=1\sim 3 \quad \dots\dots\dots (10)$$

$$g_k = f(u_k^{(3)}) = \frac{1}{1 + e^{-u_k^{(3)}}}; \quad k=1\sim 3 \quad \dots\dots\dots (11)$$

ここで、 $w_{kj}^{(2)}$  は中間層  $j$  番目のニューロンと出力層  $k$  番目のニューロン間の結合の重み係数である。最終的に、RNNモデルからの出力  $g_k$  は以下のように後処理され、適切な値に変換される。

$$G_k = (g_{k,\min} - g_{k,\max}) g_{k,[n+1]} + g_{k,\max} \quad \dots\dots\dots (12)$$

ここで、 $g_{k,\max}$ ,  $g_{k,\min}$  は前もって設定される値である。

RNNモデルの重みの決定は、以下に示すバックプロパ

ゲーション (BP) 法を用いて行った。

2.3 RNNモデルの同定

以上に述べたRNNモデルを構築するためには、モデルに含まれる重み係数  $w_{ji}^{(1)}$  と  $w_{kj}^{(2)}$  を決定する必要がある。このモデル同定は、バックプロパゲーション (BP) 法<sup>6)</sup> を用いておこなった。BP法では、モデルの出力が教師データに近くなるように繰り返し計算を行いながら、重み係数を決定する。本研究では、人間によるPID制御ゲインの介入を行った結果得られるループ角度の挙動を教師データとしている。前節で述べたように、ループ挙動の良否は4種類の評価関数  $I_1$  から  $I_4$  で評価している。教師データとして、人間が調整した数十回の圧延データに対応する、これら4種類の評価関数値と、ゲイン設定値を与える。 $I_1$  から  $I_4$  の値は、Fig. 3(b) に示すループ角度の時間変化データを用いて計算する。まず、人間によるゲイン調整データを用いて得られるループ評価関数値を、 $I_1^0(l) \sim I_4^0(l)$ 、(ただし、 $l$  は圧延回数) とする。つぎに、RNNモデルから出力されるPIDゲインを用いて圧延のシミュレーションを行う。この結果得られた圧延のループの挙動データを用いて評価関数値を計算し、 $I_1(l) \sim I_4(l)$ 、とする。このようにすると、 $l$  回目の圧延について、人間の設定ゲインに対応する評価関数と、RNNの設定ゲインに対応する評価関数値の比較ができるので、以下の誤差関数  $E(l)$  を計算する。

$$E(l) = \sum_{i=1}^5 \frac{1}{2} \rho_i (I_i(l) - I_i^0(l)) \quad \dots\dots\dots (13)$$

ここで、 $\rho_i$  は各評価に対する重み係数である。BP法では、圧延の度に、上記の誤差関数  $E$  が小さくなるようRNNモデルの係数を繰り返し更新する。すなわち、重み係数にたいする誤差関数  $E$  の勾配をもとめ、次式を用いて、結合の重み係数  $w_{ji}^{(1)}$ ,  $w_{kj}^{(2)}$  を更新する。

$$w_{ji}^{(1)} \leftarrow w_{ji}^{(1)} + \Delta w_{ji}^{(1)} \quad \dots\dots\dots (14)$$

$$w_{kj}^{(2)} \leftarrow w_{kj}^{(2)} + \Delta w_{kj}^{(2)} \quad \dots\dots\dots (15)$$

$$\Delta w_{ji}^{(1)} = -\varepsilon_j(l) \frac{\partial E(l)}{\partial w_{ji}^{(1)}} \quad \dots\dots\dots (16)$$

$$\Delta w_{kj}^{(2)} = -\varepsilon_k(l) \frac{\partial E(l)}{\partial w_{kj}^{(2)}} \quad \dots\dots\dots (17)$$

Fig. 7には、上記の方法で行ったモデル同定結果を示しており、RNNモデルによるゲイン調整結果が人間の調整データと比較して示されている。この図から分かるように、RNNモデルにより人間の調整過程を再現できていることが分かる。

2.4 RNNモデルの学習

構築されたRNNモデルを実際にゲイン調整に適用するためには、操業条件の変化に応じてRNNモデルを学習さ

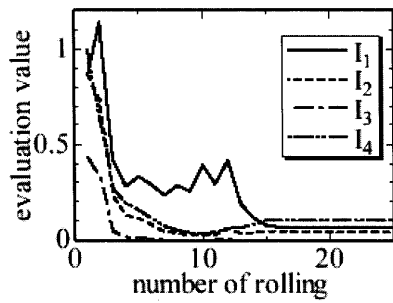


Fig. 7. Transition of evaluation value.

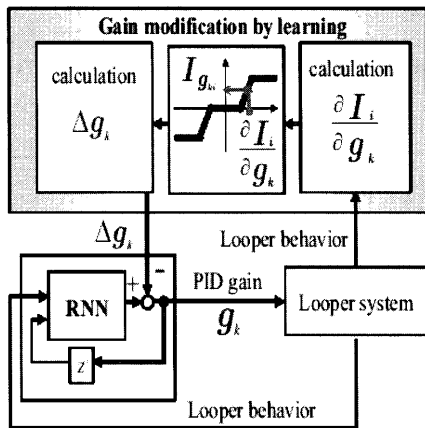


Fig. 8. Gain modification by learning function.

せ、モデルを更新する必要がある。以下では、RNNモデルの性能を向上するための2つの学習法（学習方法1、および、学習方法2）について述べる。

2.4.1 ゲイン修正機能（学習方法1）

学習によるゲイン修正機能の構成を Fig. 8 に示す。RNNモデルから出力されたゲインは、以下の手順で修正される。まず、過去2回の圧延からゲイン  $g_k$  の変化に対する評価  $I_i$  の勾配  $\partial I_i / \partial g_k$  を計算し、その値を  $I_{gk}$  に変換する。ここで、 $I_i$  はRNNモデルの入力としている評価関数値、 $g_k$  はRNNモデルの出力であるPIDゲインの値である。勾配の計算には後退差分  $(I_{i[l]} - I_{i[l-1]}) / (g_{k[l]} - g_{k[l-1]})$  が用いられる。また、 $I_{gk}$  の計算には図に示しているように、不感帯により  $\partial I_i / \partial g_k$  の絶対値がある値より小さいときは0とする。また飽和要素により  $\partial I_i / \partial g_k$  の値がある値以上になったときは1、ある値より小さくなったときは-1とする。そして、ゲイン修正量  $\Delta g_k$  が以下のように計算される。

$$\Delta g_k = \sum_{i=1}^4 \eta_{ki} I_{gki}; \quad k=1 \sim 3 \dots\dots\dots(18)$$

ここで、 $\eta_{ki}$  は正の定数で、 $I_{gki}$  に対する重みである。この  $\Delta g_k$  の値を用いて、RNNモデルから出力されたゲイン  $g_k$  を次式で修正する。

$$g_k \leftarrow g_k + \Delta g_k; \quad k=1 \sim 3 \dots\dots\dots(19)$$

修正後のゲインがRNNモデルの入力側にフィードバック

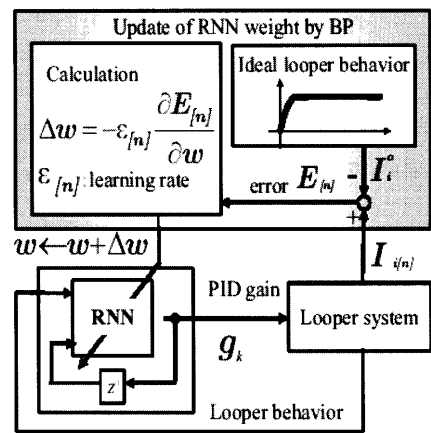


Fig. 9. Update method of RNN weight by BP method.

され、蓄積される。なお、RNNモデルによるゲイン調整を収束させるために、前回と今回のゲインの変化量がある値  $\delta g$  よりも小さければ、このゲイン修正を終了する。

2.4.2 RNNモデルの重みの更新（学習方法2）

前述した学習アルゴリズムにより、RNNモデルの出力に修正値が追加され、適切なゲイン値を出力することが期待される。しかしながら、設備環境や操業条件の変化に対応するためには、RNNモデルの構造自体を更新する必要がある。しかし、RNNモデルの出力に対する教師データが存在しない条件下では、最適なゲインの調整値は分からない。一方、理想的なルーパー挙動の評価関数値は既知であることから、本研究では、Fig. 9 に示すようにルーパー挙動に関する評価関数値から RNNモデルの構造を更新する方法を考案した。以下では、ルーパー挙動の評価を用いて、RNNモデルの重みをBP法により更新するアルゴリズムについて述べる。

まず、現在のルーパー挙動と理想状態との差を表す指標として誤差関数  $E_{[l]}$  を以下のように定義する。

$$E_{[l]} = \sum_{i=1}^5 \frac{1}{2} \rho_i (I_{i[l]} - I_i^o)^2 \dots\dots\dots(20)$$

ここで、添え字  $[l]$  は学習回数（圧延回数に対応）を表し、 $\rho_i$  は各評価に対する重み係数である。また、 $I_i^o$  は理想的なルーパー挙動の評価値であり、目標とするルーパー挙動を決定すれば、前もって計算することができる。本研究では、Fig. 9 に示しているように、オーバーシュートがなく、整定時間が短い波形をルーパーの理想挙動と考える。上で定義した誤差関数は、RNNモデルへ入力している評価関数  $I_{1[l]} \sim I_{4[l]}$  を含む。さらに、(20)式に示すように学習の精度の向上のために、ルーパーの振動に関する評価として、新たに以下の評価関数  $I_5$  を追加している。

$$I_5 = \frac{1}{T} \int_0^T \left| \frac{d(\alpha(t))}{dt} \right| dt \dots\dots\dots(21)$$

学習アルゴリズムは誤差関数  $E$  を小さくするように、BP 法により RNN モデルの重みを更新するものとし、以下のように重みを更新する。

まず、中間層  $j$  番目のニューロンと出力層  $k$  番目のニューロン間の重み  $w_{kj}^{(2)}$  は、次式で更新される。

$$w_{kj}^{(2)} \leftarrow w_{kj}^{(2)} + \Delta w_{kj}^{(2)} \dots \dots \dots (22)$$

ここで、重み更新量  $\Delta w_{kj}^{(2)}$  は次式で計算される。

$$\Delta w_{kj}^{(2)} = -\varepsilon_{k[l]} \frac{\partial E_{[l]}}{\partial w_{kj}^{(2)}} \dots \dots \dots (23)$$

過去2回の圧延結果（ゲイン  $g_{k[l-1]}$  のときのルーパ挙動の評価関数値  $I_{i[l-1]}$ 、ゲイン  $g_{k[l]}$  のときのルーパ挙動の評価関数値  $I_{i[l]}$ ）を用いて、重み更新量  $\Delta w_{kj}^{(2)}$  は次式のように導出される。

$$\begin{aligned} \Delta w_{kj}^{(2)} &= \varepsilon_{k[l]} \delta_k^{(2)} x_{j[l]}^{(2)} \\ &= -\varepsilon_{k[l]} \left\{ \sum_{i=1}^5 \rho_{ki} (I_{i[l]} - I_i^o) \frac{I_{i[l]} - I_{i[l-1]}}{g_{k[l]} - g_{k[l-1]}} \right\} \\ &\quad \times g_{k[l]} (1 - g_{k[l]}) x_{j[l]}^{(2)} \dots \dots \dots (24) \end{aligned}$$

ここで、 $x_{j[l]}^{(2)}$  は学習回数  $l$  回目における中間層の  $j$  番目のニューロンの出力である。この更新式を見てもわかるように、ルーパ挙動が理想状態に近づくとき ( $I_{i[l]} - I_i^o$ ) の値も0に近づき、重みの更新値が小さくなる。また、 $\varepsilon_{k[l]}$  は出力層  $k$  番目のニューロンに対する学習回数  $l$  回目における学習率であり、学習を必ず収束させるために次式で表される。

$$\varepsilon_{k[l]} = \varepsilon_{k0} \left( 1 - \frac{1}{T_L} \right) \dots \dots \dots (25)$$

ここで、 $\varepsilon_{k0}$  は学習率の初期値、 $T_L$  は学習期間の回数である。次に、入力層と中間層との間の重みの更新値  $\Delta w_{ji}^{(1)}$  も同様に求めてられ、次式で表される。

$$\Delta w_{ji}^{(1)} = \varepsilon_{[l]} \left( \sum_k \delta_k^{(2)} w_{kj}^{(2)} \right) x_{j[l]}^{(2)} (1 - x_{j[l]}^{(2)}) x_{i[l]}^{(1)} \dots \dots \dots (26)$$

ここで、 $x_{i[l]}^{(1)}$  は学習回数  $l$  回目における入力層の  $i$  番目のニューロンの出力である。また、 $\varepsilon_{[l]}$  は (20) 式と同様に単調減少する学習率である。以上の式により設定された学習期間  $T_L$  中は、過去2回の圧延結果を用いて圧延が終了するごとに、RNN モデルの重みを1回更新する。そして、重みを更新した後にRNNモデルにより次回圧延のゲインを出力する。

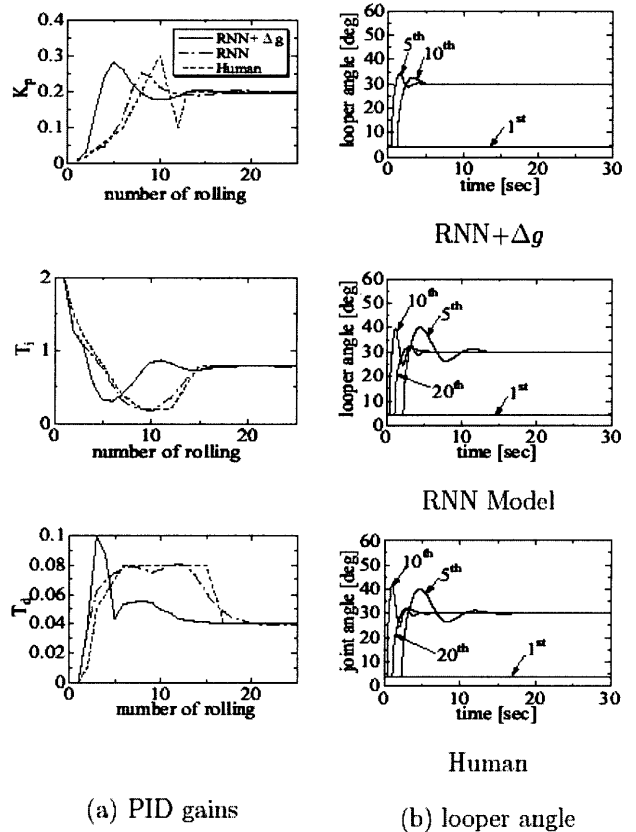


Fig. 10. Tuning results of PID gains by RNN model.

### 3. 数値実験結果

#### 3.1 RNNモデルの性能と学習方法1の適用結果

同定されたRNNモデルの性能を確認するために、熱延のルーパ高さ制御系に対してゲイン調整の数値実験を行った。同定されたRNNモデルにゲイン修正機能(学習方法1)を組み合わせたときのゲイン調整の数値実験結果を、人間の調整結果と合わせ比較して Fig.10 に示す。Fig.10(a) は、圧延回数に対する比例ゲイン  $K_p$ 、積分時間  $T_i$ 、微分時間  $T_d$  の調整過程について人間 (human)、RNN モデル (RNN)、RNN モデルとゲイン修正機能 (RNN+ $\Delta g$ ) の3者を比較している。また、Fig.10(b) は、その際の各圧延回数における人間 (human)、RNN モデル (RNN)、RNN モデルとゲイン修正機能 (RNN+ $\Delta g$ ) の3者についてルーパの時間応答を比較している。なお、圧延条件はルーパ角度の目標値  $\alpha = 30 \text{ deg}$ 、評価時間  $T = 30 \text{ s}$  とし、スタンド #1 のロール回転速度に  $\Delta V_{R1} = -0.05 \text{ m/s}$  の外乱を加えた。

RNNモデルのゲイン調整結果 (Fig.10(a)) をみると、人間の調整過程に比して同等な調整結果が得られている。また、そのときのルーパの時間応答波形 (Fig.10(b)) を見ても、各圧延回数で同程度のルーパ挙動が得られている。また、学習方法1 (RNN+ $\Delta g$ ) を用いることで、RNNモデルのゲイン調整過程の収束性が向上され、人間よりも調整回数で5回分程度速い調整ができていくことがわかる (Fig.10(a))。

Table 1. Experimental conditions.

Case	No. of Rollings	$\Delta s_1$ [mm]	$\Delta V_{R2}$ [m/s]
	1~20	-1	0
	21~40	1	0.05
	41~60	-1	-0.05

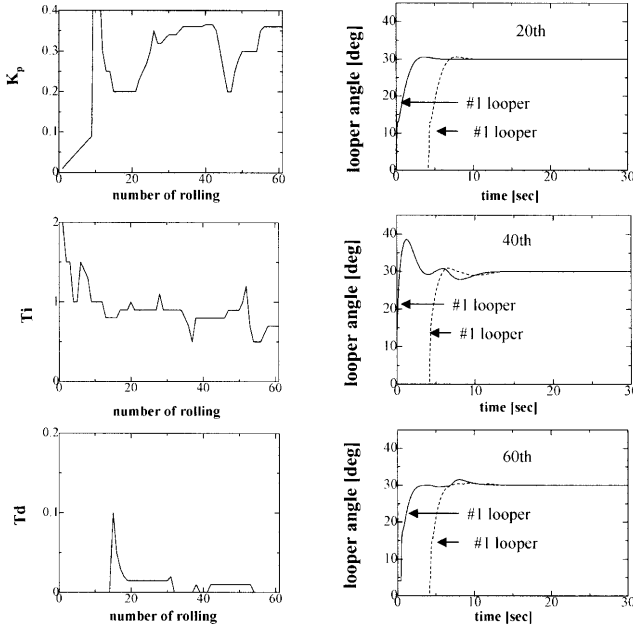


Fig.11(a) PID gains

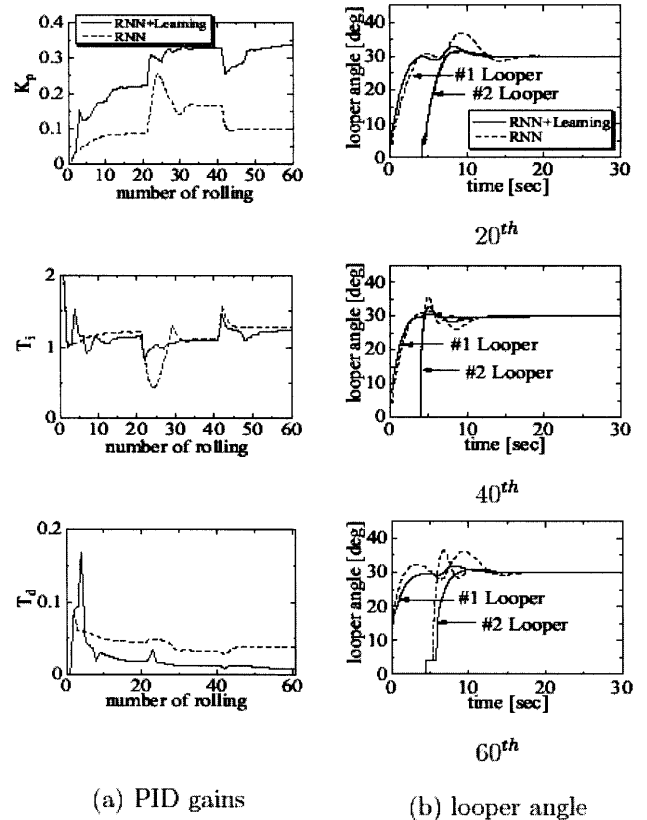
Fig.11(b) looper angle

Fig. 11. Tuning results of PID gains by human.

ルーバ時間応答波形 (Fig.10(b)) をみると、RNNモデルのみの場合はルーバが理想状態になるまで20回程度調整が行われているが、RNNモデルに学習方法1を付加した場合は、調整回数10回程度でルーバの挙動が理想状態になっているのがわかる。

### 3.2 学習方法1と学習方法2の同時適用テスト

学習方法1と学習方法2の同時適用効果を確認するため、教師データ作成時とは異なる条件下で数値実験を行った。Table 1に示すように、Case I, Case II, Case IIIの3つの条件で圧延回数20回ごとに圧延条件（スタンド#1のロール隙  $s_1$ 、スタンド#2のロール回転速度  $V_{R2}$ ）を変化させた。この条件変更により板厚制御の影響がルーバ挙動に現れる、また、教師データではルーバが持ち上がりにくい条件であったがルーバが持ち上がり易い条件も含まれる。まず、Fig.11に、人間の操作結果を示している。Fig.11(a)に示すように、人間は操業条件の変化に対して時間経過と共に適切なゲインを得るが、この例では試行錯誤の調整の様子が見られる。この結果、Fig.11(b)に示すようにルーバの立ち上がり時にオーバーシュートが見られる。つぎに、Fig.12に、RNNモデルのみの結果および、RNNモデルに学習方法1と学習方法2（RNNモデルの重み更新機能）を同時に追加した結果を示している。その際、学習方法1（ゲイン



(a) PID gains

(b) looper angle

Fig. 12. Tuning results of PID gains by RNN model with learning.

修正機能)は、圧延回数2回目から60回目まで行い、学習方法2については、学習回数  $T_L=10$  とし、圧延条件が変化するごとに、調整回数5回目から15回目まで重みの更新を行っている。

Fig.12(a)には、1基目のルーバについてRNNモデルの学習効果を示している。図には、各圧延に対しRNNモデルのみで調整を行ったとき(RNN)と、学習方法1と学習方法2(RNN+learning)を付加したときに得られた、比例ゲイン  $K_p$ 、積分時間  $T_i$ 、微分時間  $T_d$ の3つのゲインの調整過程を示している。この結果から分かるように、学習により適正なゲイン調整値が得られる様子が見られる。また、Fig.12(b)は、圧延回数20回目、40回目、60回目における調整後の1基目と2基目のルーバ角度の時間応答を示している。学習によりゲイン調整値が適切なものとなり、ルーバ挙動はオーバーシュートが小さくなり、より理想的な状態に落ち着いていることが確認できる。

## 4. おわりに

本研究では、熱延ルーバ制御系を対象に人間のゲイン調整過程を再現することを目的に、制御性能を表す複数の評価値と過去のゲインを入力とするRNNモデルを提案した。提案したモデルは、人間のゲイン調整データを教師データとして、そのモデル構造を同定した。また、ルーバ特性に

関する複数の評価関数値を入力することにより、人間の多様な判断を模擬することを試みている。このようにして提案したRNNモデルを熱延ループ高さ制御系のゲイン調整に適用し、その性能を確認した。さらに、RNNモデルの性能向上のために2種類の学習機能を提案した。すなわち、RNNモデルが出力したゲインによる制御性能を評価し、評価値が改善される方向へ制御ゲインを修正しゲイン調整の高速化を図る機能（学習方法1）と、評価値が目標値に合致するようにRNNモデルの重みを逐次更新する機能（学習方法2）の2つである。これらの学習機能によりRNNモデルが改善できることを数値実験により示した。本研究で提案した方法は、人間のゲイン調整機能の1つの表現法であるが、熱延ループ制御系に限定されるものでは

なく、他のプラント制御にも適用される可能性がある。これらの追試が今後の課題である。

## 文 献

- 1) H.Asada, A.Kitamura, S.Nishino and M.Konishi: *ISIJ Int.*, **43** (2003), 358.
- 2) Y.Iijima, F.Wan, H.Mizuno, A.Kido and K.Arishima: *CAMP-ISIJ*, **14** (2001), 1001.
- 3) H.Ikeda, N.Kubo, K.Yano, N.Inami and N.Wakamiya: *CAMP-ISIJ*, **14** (2001), 1002.
- 4) I.M.Mujtaba and M.A.Hussain: *Application of Neural Networks and Other Learning Technologies in Process Engineering*, Imperial College Press, London, (2001), 99.
- 5) S.Imajo, M.Konishi, J.Imai and T.Nishi: *CAMP-ISIJ*, **15** (2002), 931.
- 6) C.H.Dagli: *Artificial Neural Network for Intelligent Manufacturing*, Chapman & Hill, London, (1994), 39.