

技術論文

機械学習を活用した連続鋳造機の浸漬ノズル形状最適化

Surrogate-based Shape Optimization of Immersion Nozzle in Continuous Casting

難波 時永*
Tokinaga NAMBA岡田 信宏
Nobuhiro OKADA

抄 錄

連続鋳造では、タンディッシュから浸漬ノズルを介して鋳型内に溶鋼が供給される。浸漬ノズルにおいては、溶鋼中に存在するアルミナ等を主成分とする介在物が付着堆積し、連続鋳造の連続数を制限する原因となる。このノズル閉塞を抑制するために、浸漬ノズルにArガスが吹き込まれる。しかし、Ar気泡が溶鋼とともに鋳型内に流入し、鋳片に捕捉されると気泡性欠陥の原因となる。気泡性欠陥を抑制するには、Ar気泡を凝固界面から遠ざけるか、凝固界面へ付着した気泡を溶鋼で洗い流すことが有効である。鋳型内の溶鋼流動は浸漬ノズルの形状に大きく影響される。本稿では、凝固界面に捕捉されたAr気泡を低減するための浸漬ノズル形状の最適化を検討する。鋳型内における溶鋼流動及び伝熱凝固の数値解析モデルを最適化手法と組み合わせた。最適化プロセスでは、計算速度向上のため凝固界面に捕捉されたAr気泡量をニューラルネットワークで評価する。本手法を用いた浸漬ノズル形状の探索結果を報告するとともに、得られたノズル形状が気泡性欠陥低減に有効であることを論じる。

Abstract

In continuous casting, molten steel is fed from the tundish into the mold through the immersion nozzle. In the immersion nozzle, inclusions mainly composed of alumina present in the molten steel adhere and accumulate, causing limitation of continuous castings. To prevent the nozzle clogging, Ar gas is blown into the immersion nozzle. However, Ar bubbles flow into the mold along with the molten steel and become trapped in the solidifying shell, causing bubbling defects of the slab. To suppress bubbling defects, it is effective to keep Ar bubbles away from the solidification interface or to use molten steel to wash away Ar bubbles that have adhered to the solidification interface. The molten steel flow in the mold is greatly affected by the shape of the immersion nozzle. In this paper, we consider the optimization of the shape of the immersion nozzle to reduce Ar bubbles trapped in the solidifying shell. A numerical model of molten steel flow and heat transfer solidification in the mold is combined with an optimization method. In the optimization process, Ar bubbles trapped in the solidifying shell are evaluated by a neural network to improve the calculation speed. The application of this method to the search for immersion nozzle shape is also reported, and the effectiveness of the obtained nozzle shape in reducing Ar bubbles is discussed.

1. 緒 言

連続鋳造では、タンディッシュから浸漬ノズルを介して鋳型内に溶鋼が供給される。浸漬ノズルにおいては、溶鋼中に存在するアルミナ等を主成分とする介在物が付着堆積し、連続鋳造の連続数を制限する原因となる。このノズル閉塞を抑制するために、浸漬ノズルにArガスが吹き込まれる。Arガスを吹き込むことにより、連々数を確保できるが、Ar気泡が溶鋼とともに鋳型内に流入し、鋳片に捕捉されると気泡性欠陥の原因となる。この気泡性欠陥は、めっき不

良の原因となるため、特に自動車用外装材に用いられるIF鋼で問題となる。鋳片の表層近くに捕捉されたAr気泡は、鋳片においてピンホール欠陥と呼ばれ、圧延後は引き延ばされたスジ疵となる。また、鋳片の表皮下数mmの位置に捕捉されたAr気泡は、圧延後の鋼板表面に膨らみが生じる欠陥(フクレ疵)となる。フクレ疵の原因となる表皮下の気泡性欠陥を鋳片のブローホール欠陥と呼ぶ。これらの気泡性欠陥を防ぐためにはAr気泡を凝固界面に近づけないか、凝固界面に付着したAr気泡を溶鋼により洗い流すことが有効である。よって、鋳型内溶鋼流動の制御が重要と

なる。

これまでに、溶鋼流動を制御し凝固シェルへのAr気泡の捕捉を防止する技術が開発されている。電磁力を利用した電磁攪拌(Electromagnetic Stirring: EMS)や電磁ブレーキ(Electromagnetic Brake: EMBr)がその代表例である。鋳片表層の気泡性欠陥を抑制するには、EMSを鋳型上部に設置し、湯面近傍を攪拌することが効果的である。一方、EMBrは鋳型中央部及び浸漬ノズル先端付近に設置されており、下部に向かう溶鋼流動に制動を加える装置である。特に浸漬ノズルの吐出流の下部に向かう流れを抑制することで、介在物やAr気泡を浮上分離させやすくなる。

鋳型内の溶鋼流動は浸漬ノズルの形状にも大きく影響するため、浸漬ノズルについてもこれまで検討がなされてきた¹⁴⁾。しかしながら、ノズル詰まり、耐火物強度やコストなどの総合的なバランスから、スラブ鋳造においては、左右に1対の吐出孔を有する伝統的な2孔ノズルが用いられることが多く、近年は議論されることも少なくなった。一方、近年、鋳造から熱延までを一貫処理する薄スラブ連続鋳造機の導入が増加している。薄スラブ連続鋳機では、引き抜き速度が高速となるため、浸漬ノズルの形状と鋳型内流動に関する研究が報告されている⁵⁻⁹⁾。これらの報告を鑑みると、従来のスラブ連続鋳造機の浸漬ノズルに関しては未だ改良の余地は残されていると考えられる。そこで本稿では、従来の連続鋳造機において、凝固シェルへ捕捉されるAr気泡を低減する浸漬ノズルの形状を求めるこを考へる。特に、鋳片の表層～2mmに捕捉されたAr気泡をピンホール、表皮下2mm～15mmに捕捉されたAr気泡をプローホールと呼び、これらを低減することを目的とする。

ピンホールとプローホールを算出するにあたり、これまでに開発、検討が重ねられてきた鋳型内の溶鋼流動及び伝熱凝固の数値解析モデル^{10, 11)}を用いる。本研究では、数値解析モデルと機械学習を使用した最適化技術の枠組みを合わせることにより、ピンホールとプローホールの最小化を目的とした浸漬ノズル形状の最適化を行う。このような数値流体力学(Computational Fluid Dynamics: CFD)と最適化技術を組み合わせた形状最適化技術は、航空機の翼設計¹²⁾や船の設計¹³⁾、高層ビルなどの構造物の設計¹⁴⁾において技術開発及び適用例が見られるが、製鉄プロセスには見られない。CFDを用いた形状最適化における課題は、時間のコストが高く実用性に欠けることである。一般に最適化の過程においては数千回～数万回のシミュレーションを必要とするが、CFD計算は多くの計算時間を要するため、直接的に最適化技術を適用することは難しい。計算コストの削減と最適化の効率を図るために、近年では、代理モデルの研究が活発に行われている。代理モデルは、最適化過程において、複雑で時間のかかる計算を代替する近似モデルの一種である。クリギング¹⁴⁾や動径基底関数¹⁵⁾、(深層)ニューラルネットワーク¹⁶⁾などの機械学習モデルの適用事例が報

告されている¹⁷⁾。

本稿では、ピンホールとプローホールは、溶鋼流動及び伝熱凝固による複雑な現象の結果得られることを考慮し、CFD計算の代わりにニューラルネットワークモデルでピンホールとプローホールを求めることする。すなわち、実施可能な回数のCFD計算で得られたピンホールとプローホールの評価値とその計算条件をデータセットとしてニューラルネットワークを学習し、学習済みのモデルで所与のノズル条件におけるピンホールとプローホールを求める。これにより理想とする最小のピンホールとプローホールとの誤差を求めることができるため、この誤差を目的関数として最適化アルゴリズムを適用してピンホールとプローホールを低減するノズル形状を求める。ニューラルネットワークによる予測は数秒程度であるため、高速な最適化が可能である。

最適化の枠組みには、反復的に最適化を行いながら最適解を求める逐次近似最適化^{18, 19)}を適用する。本枠組みでは、反復過程で得られる計算条件とピンホール及びプローホールからなるデータを段階的にデータセットに追加し、ニューラルネットワークモデルを更新して最適解を求めていく。ニューラルネットワークの予測精度はデータセットに強く依存する。データ数が少ない場合は予測精度が十分でない可能性があり、CFD計算の結果と乖離するため、1度の最適化の実行で最適解が得られる保証はない。一方、逐次近似最適化では、データセットが拡充されることでニューラルネットワークの予測精度が向上されるため、反復的に最適解を更新でき、ピンホールとプローホールを低減するこができる。

最適化アルゴリズムには粒子群最適化²⁰⁾を使用する。これは、ランダムに位置する複数の粒子(探索点)が、決められた規則に則って探索空間を移動しながら最適解を探していく方法である。局所最適解に陥りにくく多峰性関数のような複雑な目的関数に対しても大域最適解を求める能力が高いことが特徴である。

以上に加えて、動径基底関数ネットワークによる密度関数を用いたデータ点の追加方法²¹⁾も取り入れる。逐次近似最適化の過程において、最適なノズル条件のみをデータセットに追加した場合、ニューラルネットワークの精度は最適解の周りは高まるがデータ点の無いところは不十分となる。この状態で再度最適化を行うと最初に得られた最適解周りに新しい最適解が得られるため、本来データ点の無いところに最適解があったとしても見つけることはできない。よって、データ密度の小さい場所のデータ点を求めてデータセットに追加することが有効となる。動径基底関数ネットワークは、既存のデータ点上に基底関数を置き、すべての点について和を取ったもので、これによりデータ密度を測ることができる。基底関数として、本稿ではガウス関数を適用する。動径基底関数ネットワークの最小点を

データセットに追加すれば、探索空間の全様をニューラルネットワークに学習させることができる。

以降では、鋳型内の溶鋼流動及び伝熱凝固の数値解析モデルである数学モデルと浸漬ノズルの形状最適化の方法を述べ、浸漬ノズル形状最適化の実施例を報告する。

2. 数学モデル

2.1 支配方程式

鋳型内の溶鋼流動及び伝熱凝固の数値解析には、高谷らによる数学モデル^{10,11)}を用いる。支配方程式は、流体計算を非定常乱流モデルLES (Large Eddy Simulation)、伝熱凝固を非定常のエネルギー方程式としている。本モデルにおける仮定を下記に示す。

- (1) 流体は非圧縮であり、液相と固相の密度と比熱は同じく定数とする。
- (2) 固液共存相には流体抵抗としてDarcy則を適用する。
- (3) 固相は剛体として作用する(固相の速度は鋳造引き抜き速度と等しい)とし、固相の体積率が0.8より大きな領域は剛体とする。
- (4) ガス気泡は非圧縮かつ分散相であり、形状は球体、直径は変わらないとする。
- (5) 気相の運動量方程式はBasset-Boussinesq-Oseen-Tchen方程式²²⁾とする。ただし、Basset項は無視できるとする。
- (6) エネルギーバランスにおいて、気相のエンタルピーは無視できるとする。
- (7) 乱流モデルはLESを適用する。
- (8) 乱流Prandtle数は1とする。
- (9) 凝固温度はてこの原理で計算する²³⁾。

支配方程式は次の通りである。

$$f_g + f_l + f_s = 1 \quad (1)$$

$$\frac{\partial f_g}{\partial t} + \nabla \cdot (f_g \mathbf{u}_g) = 0 \quad (2)$$

$$\frac{\partial f_l}{\partial t} + \nabla \cdot (f_l \mathbf{u}_l) = -R_v \quad (3)$$

$$\frac{\partial f_s}{\partial t} + \nabla \cdot (f_s \mathbf{u}_s) = R_v \quad (4)$$

$$\rho_g \frac{D \mathbf{u}_g}{Dt} = \rho_l \frac{D \mathbf{u}_l}{Dt} + \frac{1}{2} \rho_l \left[\frac{D \mathbf{u}_l}{Dt} - \frac{D \mathbf{u}_g}{Dt} \right] - \frac{3 \mu_l}{4 d_g^2} \text{Re}_g C_{Dg} (\mathbf{u}_g - \mathbf{u}_l) + (\rho_l - \rho_g) \mathbf{g} \quad (5)$$

$$\frac{\partial (f_l \rho_l \mathbf{u}_l)}{\partial t} + \nabla \cdot (f_l \rho_l \mathbf{u}_l \mathbf{u}_l) = -f_l \nabla p + \nabla \cdot [f_l (\mu_l + \mu_v) (\nabla \mathbf{u}_l + \nabla \mathbf{u}_l^T)] + f_l [\beta (T - T_0) \rho_l \mathbf{g} + \gamma (\mathbf{u}_s - \mathbf{u}_l) + \mathbf{F}_{lg}] \quad (6)$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (f_l \mathbf{u}_l T + f_g \mathbf{u}_g T) = \nabla \cdot [(\alpha + \alpha_v) \nabla T] + \frac{1}{C_p} R_v (-\Delta H) \quad (7)$$

$$T = \frac{(T_i - T)/m_l}{(T - T_s)/m_s + (T_i - T)/m_l} \quad (8)$$

ここで t は時間、 \mathbf{u} は流速、 f は体積率、 p は圧力、 T は温度、 ρ は密度、 μ は粘性係数、 C_p は比熱、 α は熱伝導率、 β は熱膨張率、 \mathbf{g} は重力加速度、 R_v は溶鋼の凝固率、 ΔH は潜熱、 m_l は液相線の勾配、 m_s は固相線の勾配を表す。下付き添

え字の l, s, g はそれぞれ液相、固相、気相である。係数 Re_g 及び C_{Dg} は式(9)と式(10)で与える。

$$\text{Re}_g = d_g |\mathbf{u}_l - \mathbf{u}_g| \frac{\rho_l}{\mu_l} \quad (9)$$

$$C_{Dg} = 0.4 + \frac{24}{\text{Re}_g} + \frac{6}{1 + \sqrt{\text{Re}_s}} \quad (10)$$

また、 μ_l, α_l は乱流粘性係数及び乱流熱伝導率を表わす。乱流粘性係数は式(11)で与える。

$$\mu_l = \rho_l (C_s \Delta^{1/3})^2 \sqrt{2|\mathbf{S}||\mathbf{S}|} \quad (11)$$

ここで係数 $C_s = 0.1$ であり、 Δ はセルの体積、 \mathbf{S} は式(12)で定義されるひずみ速度テンソルである。

$$\mathbf{S} = \frac{1}{2} (\nabla \mathbf{u}_l + \nabla \mathbf{u}_l^T) \quad (12)$$

式(6)中の右辺第3項は温度による浮力であり、 T_0 は液相の平均温度である。第4項は固液共存相の流体抵抗を表し、係数 γ を式(13)で与える。

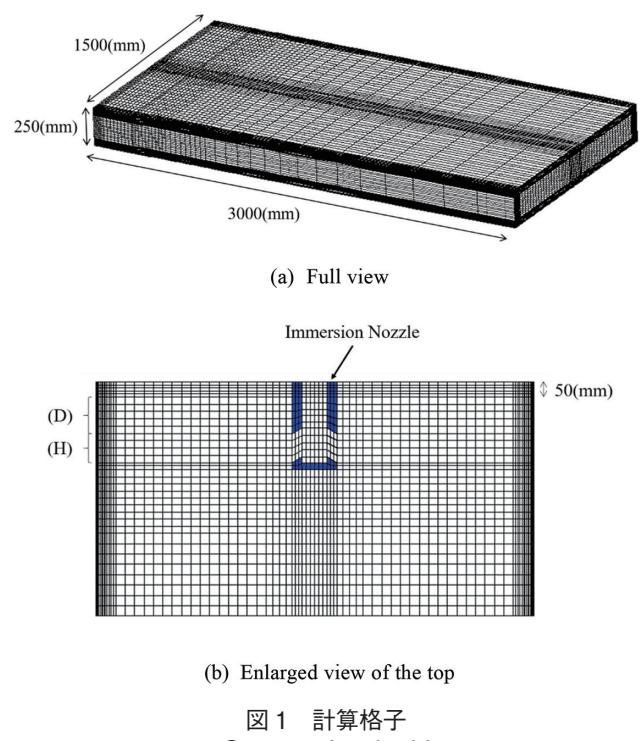
$$\gamma = \frac{\mu_l}{\rho_l} \frac{150(1-f_l)^2}{D_2^2 f_l^3} \quad (13)$$

ここで D_2 は2次デンドライトアーム間隔である。第5項は液相と気相の運動量交換であり、 \mathbf{F}_{lg} を式(14)で与える。

$$\mathbf{F}_{lg} = f_g \frac{3\mu_l}{4d_g^2} \text{Re}_g C_{Dg} (\mathbf{u}_g - \mathbf{u}_l) \quad (14)$$

2.2 計算条件及び計算方法

計算領域は図1に示すように250mm×1500mm×3000mm(厚さ×幅×長さ)の直方体領域とした。湯面から900mmの範囲を鋳型として熱伝達係数を800W/m²Kとし、それ以降は2次冷却と仮定して熱伝達係数は400W/m²Kとした。浸漬ノズルの形状は図2に示す通りとし、浸漬深さ d (mm)、吐出孔高さ h (mm)、ノズル角度 θ (degree)を連続



(b) Enlarged view of the top

図1 計算格子
Computational grid

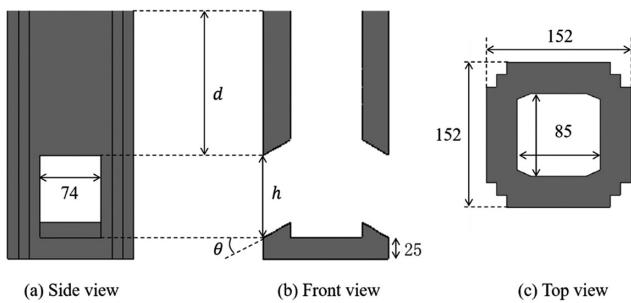


図2 浸漬ノズルの寸法

Geometry of the immersion nozzle. All dimensions are in mm, except angle degree.

なパラメータとして最適化により求める変数とした。計算領域の分割数は、長さ方向に 516mm 以降の領域については $36 \times 80 \times 23$ (厚さ×幅×長さ) で固定とした。516mm までの領域については、 36×80 (厚さ×幅) は固定であるが、長さ方向には浸漬深さと吐出孔高さの値に応じて、それぞれ図1(b)における領域(D)及び(H)の箇所の分割数を変更した。領域(D)については、格子間隔が 25mm 未満となる最小の数を分割数とし、領域(H)については、25mm 未満となる最小の数を分割数とした。ただし、領域(H)の分割数が 4 未満の場合の分割数は 4 とした。

スタッガード格子上に離散化し SOLA 法²⁴⁾を基にした任意形状に対応した解法を用いて計算を行った。移流項には 2 次の中心差分スキーム、拡散項は 2 次の中心差分を用いて、移流項は Euler 法により時間発展し、拡散項は陰的に計算を行った。

鋳造引き抜き速度は 1.4m/min、浸漬ノズルから鋳型へ流入する Ar 気泡の量は文献 25)を参考に、0.5mm 気泡径を 2.668 Nl/min、1.0mm 気泡径を 0.944 Nl/min、1.5mm 気泡径を 0.493 Nl/min とした。なお、最適化の目的であるピンホールとブローホールはいずれも存在量が最も多い 0.5mm 気泡径を対象とした。

2.3 気泡捕捉のメカニズム

気泡が凝固界面に捕捉されるメカニズムとして、速度勾配による Saffman 力²⁶⁾、界面活性元素の濃度勾配に起因する界面張力差^{27,28)}、電磁力²⁹⁾などが作用していると考えられているが、その詳細は不明である。例えば、硫黄濃度が高いほど、ピンホールが発生しやすいことが報告されており³⁰⁾、界面活性元素である硫黄の濃度勾配による界面張力差が大きく影響していると考えられる。式(15)は、界面張力差により溶鋼中の気泡に作用する力 \mathbf{F}_i を示す²⁷⁾。

$$\mathbf{F}_i = \frac{8}{3} \pi R^2 C_0 \frac{\partial \sigma}{\partial C_L} \frac{V_s}{D_L} (1 - K_e) \exp\left(-\frac{V_s(x - \delta)}{D_L}\right) \quad (15)$$

ここで C_L は境界層中溶質濃度、 C_0 は凝固前の液相の溶質濃度、 R は気泡半径、 σ は界面張力、 K_e は実効分配係数、 V_s は凝固界面の成長速度、 x は凝固界面からの距離、 δ は濃度境界層厚み、 D_L は溶質の拡散係数である。実効分配

係数 K_e は、式(16)として表される。

$$K_e = \frac{K_0}{K_0 + (1 - K_0) \exp(-V_s \delta / D_L)} \quad (16)$$

ここで K_0 は平衡分配係数である。

式(16)による界面張力差による気泡への作用力を連続鋳造機の数値解析において考慮するためには、硫黄濃度分布と凝固による偏析を考慮した計算を行えばよいが、0.01~1mm 程度の濃度境界層厚みを考慮することは現在の計算能力では困難である。一方、この界面張力差による作用力は、濃度境界層厚みを気泡径よりも十分に小さくすれば、その効果がなくなる。濃度境界層厚みは、Burton らが式(17)の関係を報告している³¹⁾。

$$\delta = 0.285 D_L^{1/3} v^{1/6} \omega^{-1/2} \quad (17)$$

ここで v は動粘性係数であり、 ω は液相の回転による角速度である。すなわち、濃度境界層厚みは、凝固界面全面の流速のみの関数となり、流速が速いほど境界層厚みは薄くなる。

以上から、連続鋳造機の数値解析におけるピンホールの評価方法として、凝固シェルに捕捉される気泡個数を溶鋼流速に依存するとしてモデル化することが適切である。気泡の凝固シェルへ捕捉される速度 η_e を式(18)に示すように仮定した。

$$\eta_e = n_e R_s P_e(|\mathbf{u}|) \quad (18)$$

ここで n_e は気泡体積率、 R_s は凝固速度であり、 $P_e(|\mathbf{u}|)$ は溶鋼流速に依存した気泡の捕捉確率である。凝固シェルに捕捉された気泡の体積率 n_e は、式(19)に示す通り、凝固シェル速度 \mathbf{u}_s によって輸送する。

$$\frac{\partial n_e}{\partial t} + \nabla \cdot (\mathbf{u}_s n_e) = -\eta_e \quad (19)$$

気泡の捕捉流速の閾値として、0.05 m/s³²⁾ や 0.15 m/s³³⁾ の値が報告されているが、凝固界面位置の定義や流速の評価位置が異なるため、正確な値は不明である。さらに、閾値を 0.05 m/s とした場合は気泡が捕捉され、0.051 m/s の場合は全く捕捉されないというモデルは不自然と考えられる。よって、捕捉確率は速度の関数として、速度が増加するほど確率を低下させる式(20)に示す指指数関数型のモデルとした。

$$P_e(|\mathbf{u}|) = \exp(-C_0 |\mathbf{u}|) \quad (20)$$

指指数関数型モデルでは、0.15 m/s の場合の捕捉確率が、 1×10^{-8} 程度となるように、式(20)の係数を $C_0 = 100$ と調整した。また、CFDにおいて、凝固が開始し固相率を有する要素の流速は、Darcy 則により流速が大きく低下するため、凝固界面流速として固相率は 0.1 の位置の流速を用いた。

2.4 ピンホールとブローホールの算出方法

ピンホールとブローホールは、CFD 計算開始から実時間換算で 1000 秒経過し、定常に達した後、100 秒間の時間平均化を行った値を評価した。具体的に、ピンホール y_p と

ブローホール y_B は式(21)と式(22)で求めた。

$$y_P = \left(\sum_{n=1}^N \sum_{i=1}^{M_p} \frac{C_{i,P}^n V_{i,P}}{100} \delta t \right) \left/ \left(V_c \sum_{i=1}^{M_p} V_{i,P} \right) \right. \quad (21)$$

$$y_B = \left(\sum_{n=1}^N \sum_{i=1}^{M_B} \frac{C_{i,B}^n V_{i,B}}{100} \delta t \right) \left/ \left(\sum_{i=1}^{M_B} V_{i,B} \right) \right. \quad (22)$$

ここで、 N は時間の最大ステップ数、 δt は時間ステップ幅、 M_p と M_B はそれぞれシェル側面の内、表層から 2mm までのセルの総数と 2mm ~ 15mm のセルの総数、 $V_{i,P}$ と $V_{i,B}$ はそれぞれ当該領域における i 番目のセルの体積、 $C_{i,P}^n$ と $C_{i,B}^n$ はそれぞれ時間 n ステップ目の当該領域における i 番目のセル内に捕捉された Ar 気泡の体積率を表す。また、 V_c は直径 0.5mm の球の体積、すなわち $V_c = \pi(0.5 \times 10^{-3})^3/6$ である。したがって、ピンホールは個数密度、ブローホールは体積率として評価している。

3. 浸漬ノズル形状最適化の方法

3.1 問題設定

本稿で考える浸漬ノズル形状最適化では、浸漬深さ d 、吐出孔高さ h 、ノズル角度 θ を設計変数とするため、探索空間を式(23)のように設定する。

$$S = \{(d, h, \theta)^T \in \mathbb{R}^3 : d_{lb} \leq d \leq d_{ub}, h_{lb} \leq h \leq h_{ub}, \theta_{lb} \leq \theta \leq \theta_{ub}\} \quad (23)$$

設計限界があるため、探索空間には下限値 d_{lb} 、 h_{lb} 、 θ_{lb} と上限値 d_{ub} 、 h_{ub} 、 θ_{ub} を設けた。今回用いた上下限値を表1に示す。

浸漬ノズルの上記 3 変数が決まれば CFD によりピンホールとブローホールが計算できるので、この工程を関数 F として式(24)のように表す。

$$(y_P, y_B)^T = F(d, h, \theta) \quad (24)$$

このとき、ピンホールとブローホールを低減するための浸漬ノズル形状最適化問題は式(25)のように定式化できる。

$$\text{Find}(d^*, h^*, \theta^*)^T = \underset{(d, h, \theta)^T \in S}{\text{argmin}} L(F(d, h, \theta)) \quad (25)$$

ここで $L = L(y_P, y_B)$ は目的関数を表している。ピンホールとブローホールを同時に最小化することが本研究の目的であるため、問題は多目的最適化問題に分類される。これを単純化するために、本稿では、式(26)のように 2 つの目的を平均で定義される目的関数に落とし込むことで単一最適化問題とした。

$$L(y_P, y_B) = \frac{1}{2} |y_P| + \frac{1}{2} |y_B| \quad (26)$$

一般に最適化は目的関数を評価するために複数回モデル F を呼び出すため、計算コストの観点から問題(26)を直接解くことは現実的ではない。そこで、本稿では、 F を使わず、

表1 探索空間の上下限値
Upper and lower bounds of the search space

d_{lb} (mm)	h_{lb} (mm)	θ_{lb} (degree)
125	60	15
d_{ub} (mm)	h_{ub} (mm)	θ_{ub} (degree)
225	120	40

F を近似する計算コストの低いモデル \tilde{F} としてニューラルネットワークを使用した。このとき、最適化問題(25)は、式(27)で表される近似問題となる。

$$\text{Find}(\tilde{d}^*, \tilde{h}^*, \tilde{\theta}^*)^T = \underset{(d, h, \theta)^T \in S}{\text{argmin}} L(\tilde{F}(d, h, \theta)) \quad (27)$$

さらに、最適解の精度を高めるために、データ点を段階的に追加して、ニューラルネットワークの精度を高めながら、反復的に最適解の更新を行う逐次近似最適化^{18, 19)}の枠組みを採った。結局、式(28)で表される問題を解くこととした。

$$\text{Find}(\tilde{d}^k, \tilde{h}^k, \tilde{\theta}^k)^T = \underset{(d, h, \theta)^T \in S}{\text{argmin}} L(\tilde{F}^k(d, h, \theta)), \quad k = 1, 2, \dots \quad (28)$$

3.2 ニューラルネットワークによるピンホールとブローホールの予測

本ニューラルネットワークモデルは、浸漬ノズルの設計変数 h, d, θ を入力として、ピンホールとブローホール y_P, y_B を出力とする。教師あり学習を利用してモデルを作成した。データセット $\Omega = \{(x_j, y_j) = (h_j, d_j, \theta_j, y_{P,j}, y_{B,j}) | j = 1, \dots, m\}$ が与えられたとき、まずデータセット Ω を学習用のデータセット Ω_{train} とテスト用のデータセット Ω_{test} に 7:3 の割合で分割した。その後、 Ω_{train} の入出力変数を式(29)で標準化したデータセット Ω'_{train} を作成した。

$$\mathbf{x}' = \frac{\mathbf{x} - \mu}{\sigma} \quad (29)$$

ここで \mathbf{x}' は \mathbf{x} を標準化した値、 μ と σ は各変数の平均と分散である。また、これらの平均と分散を用いて Ω_{test} の入出力変数を式(29)で変換したデータセット Ω'_{test} を作成した。

ニューラルネットワーク \tilde{F} は、式(30)のように複数の関数 \hat{F}_i が入れ子構造または層構造になった関数である。

$$\tilde{F}(\mathbf{x}) = \hat{F}_N(\hat{F}_{N-1}(\dots(\hat{F}_1(\mathbf{x})))) \quad (30)$$

関数 \hat{F}_i は、より具体的に式(31)で表される。

$$\hat{F}_i(\hat{\mathbf{y}}'_{i-1}) = a(\mathbf{W}_i \hat{\mathbf{y}}'_{i-1}) \quad (31)$$

ここで \mathbf{W}_i は重み行列、 $\hat{\mathbf{y}}'_{i-1}$ は式(32)で定義される $i-1$ 層の出力ベクトル $\hat{\mathbf{y}}_{i-1}$ の一行目に 1 を挿入したベクトルである。

$$\hat{\mathbf{y}}'_{i-1} = \begin{cases} \hat{F}_{i-1}(\hat{F}_{i-2}(\dots(\hat{F}_1(\mathbf{x})))) & \text{if } i \geq 2 \\ \mathbf{x} & \text{if } i = 1 \end{cases} \quad (32)$$

関数 a は活性化関数と呼ばれ、双曲線正接関数やランプ関数(ReLU)を始めとして様々な形状のものが提案されている³⁴⁾。

重みは事前には決定できないため、事前に決めた誤差関数を最小化することでニューラルネットワークが Ω'_{train} の入出力に一致するように重みを更新すなわち学習する。学習では、確率的勾配降下法³⁵⁾を用いた。誤差関数としては式(33)で定義される平均二乗誤差に正則化項を加えたものを採用した。

$$E(\mathbf{y}, \hat{\mathbf{y}}; \mathbf{W}) = \frac{1}{m} \sum_{j=1}^m \{(y_{P,j} - \hat{y}_{P,j})^2 + (y_{B,j} - \hat{y}_{B,j})^2\} + \lambda \left(\sum_{i,j} |w_{i,j}|^p \right)^{\frac{1}{p}} \quad (33)$$

ここで \mathbf{y} と $\hat{\mathbf{y}}$ はそれぞれ与えられた出力データと予測値であり、 $w_{i,j}$ は重み行列の成分、 λ は正則化の程度を表すパラメータを表している。ただし正則化項の和はニューラルネットワークすべての重みについて取られるものであり、

また $p \geq 1$ である。

正則化項は過学習を防止することを目的としたものであるが、加えて、ドロップアウトも使用した。ドロップアウトは各層の出力ベクトルの成分を一定の割合で間引き、次層の入力変数の次元を落とす方法である³⁶⁾。これにより、確率的勾配降下法で局所解へ落ち込むことを防ぐ効果が得られる。さらに、学習を早期に完了させるため、重みの初期値を文献³⁷⁾に倣って設定した。以上のニューラルネットワークの作成においては Python のライブラリである TensorFlow³⁸⁾ を用いた。

ニューラルネットワークの構造を決定するパラメータとして、層数 N 、各層の入力変数の次元 n_{unit} 、ドロップアウトの割合 r 、活性化関数の種類、正則化項における p 及び λ 、確率的勾配降下法における学習率 η があり、これらは学習より前に決定しておく必要がある。本稿では、ベイズ最適化を用いて各パラメータの最適化を行った。表 2 に調整対象のパラメータとその探索範囲を示す。実装では、Python のライブラリである Optuna³⁹⁾ を用いた。各パラメータの最適候補値の決定と学習及びテストデータ Ω'_{test} を用いた予測を繰り返し、最もテストデータの予測精度が良いものを最終的なニューラルネットワークモデルとした。なお、ベイズ最適化における試行回数は 1000 回とし、テストデータの予測精度は決定係数 R^2 で測った。

3.3 粒子群最適化

粒子群最適化^{20,40)} は、ランダムに位置する複数の粒子（探索点）が、決められた規則に則って探索空間を移動しながら最適解を探していく方法である。ここではニューラルネットワークによるピンホール及びブローホールの予測を組み込んだ方法について述べる。

一つの粒子の位置は浸漬ノズルの設計変数で構成される探索空間内のベクトルによって $\mathbf{x}_i = (d_i, h_i, \theta_i)^T$ と表される。ここで i は粒子の番号である。また、各粒子は位置ベクトルのほかに移動ベクトルも持っており、探索空間の次元のベクトル $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3})^T$ で表される。さらに、各粒子はそれまでの探索で発見したそれぞれの最良解、すなわち目的関数が最も小さくなった時の位置 $\mathbf{lbest}_i = (lbest_{i1}, lbest_{i2}, lbest_{i3})^T$ と、その目的関数の値 $L(\tilde{F}(\mathbf{lbest}_i))$ を記憶している。群れとしては、すべての粒子がこれまでの探索で発見した最良解 $\mathbf{gbest} = (gbest_1, gbest_2, gbest_3)^T$ を記憶している。

図 3 に計算流れ図を示す。まず指定した粒子数だけ探

表 2 ニューラルネットワーク構造のパラメータと探索空間
Parameters of neural network structure and search range

N	n_{unit}	r
2 to 7	1 to 100	0.0 to 0.9
p	λ	η
1 or 2	10^0 to 10^{-3}	10^0 to 10^{-4}

索空間内にランダムに粒子の位置 $\mathbf{x}_i^{(0)}$ と速度 $\mathbf{v}_i^{(0)}$ を生成する。次にニューラルネットワークを用いて各粒子に対する目的関数の値 $L(\tilde{F}(\mathbf{x}_i^{(0)}))$ を計算し、最良解 \mathbf{lbest}_i を記憶する。その後は、設定した最大の反復回数に到達したか収束するまで以降の処理を繰り返す。反復回数を t とするとき、現在の位置 $\mathbf{x}_i^{(t)}$ から、それぞれの最良解へ向かうベクトル $\mathbf{lbest}_i^{(t)} - \mathbf{x}_i^{(t)}$ 、群れ全体の最良解 $\mathbf{gbest}^{(t)} - \mathbf{x}_i^{(t)}$ 、及び前回の移動ベクトル $\mathbf{v}_i^{(t)}$ の重み付き線形結合として新たな移動ベクトル $\mathbf{v}_i^{(t+1)}$ を生成する。 $t+1$ 回目の移動における i 番目の粒子の移動ベクトルは式(34)で与えられる。

$$\mathbf{v}_i^{(t)} = w\mathbf{v}_i^{(t)} + c_1 r_1 (\mathbf{lbest}_i^{(t)} - \mathbf{x}_i^{(t)}) + c_2 r_2 (\mathbf{gbest}^{(t)} - \mathbf{x}_i^{(t)}) \quad (34)$$

ここで r_1 と r_2 は 0 から 1 の間に分布する一様乱数、 w 、 c_1 、 c_2 は事前に設定する重みパラメータである。そして、式(35)によって各粒子の位置を次の位置 $\mathbf{x}_i^{(t+1)}$ に移動する。

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t)} \quad (35)$$

なお、 $\mathbf{x}_i^{(t+1)}$ の j 番目の成分 $x_{ij}^{(t+1)}$ が下限値を下回った場合には $x_{ij}^{(t+1)}$ を下限値とし、同様に、 $x_{ij}^{(t+1)}$ が上限値を上回った場合には $x_{ij}^{(t+1)}$ を上限値とした。

本稿では、終了条件として最大反復回数を与えることとし、表 3 に挙げる条件で最適化を実施した。

3.4 動径基底関数ネットワークによる密度関数を用いたサンプル点の追加

動径基底関数ネットワークは入力層、隠れ層 1 層、出力

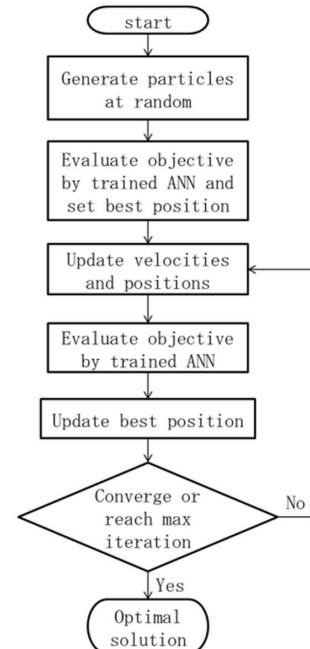


図 3 ニューラルネットワークに基づく粒子群最適化のフロー図
Neural network-based particle swarm optimization process

表 3 粒子群最適化のパラメータ
Particle swarm optimization parameters

Number of particles	w	c_1	c_2	Max iteration
100	0.5	0.14	0.14	500

層の3層からなるニューラルネットワークである。設計変数からなる既存のデータ点を $\{\mathbf{x}_j\}_{j=1,\dots,m}$ とする。ただし、重複はないものとする。この時、動径基底関数ネットワークの出力は式(36)で与えられる。

$$z(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{i=1}^m \alpha_i K(\mathbf{x}, \mathbf{x}_i) \quad (36)$$

ここで $K(\mathbf{x}, \mathbf{x}_i)$ は i 番目のユニットの基底関数、 $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_m)$ は基底関数の重みである。本稿では基底関数として式(37)で定義されるガウス関数を用いる。

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{r_i^2}\right) \quad (37)$$

ここで r_i は i 番目のデータ点の半径である。半径の定義式は様々なものが提案されている^{21, 41, 42}が、本稿では、文献42)に倣い式(38)を用いた。

$$r_i = \frac{d_{i,\max}}{\sqrt{n} \sqrt{m-1}} \quad (38)$$

ここで n は探索空間の次元、 $d_{i,\max}$ は i 番目のデータ点と他のデータ点との距離の内、最大のものを表している。本稿では $d_{i,\max}$ として式(39)に示すようにユークリッド距離を採用した。

$$d_{i,\max} = \max_{j=1,\dots,m} \|\mathbf{x}_j - \mathbf{x}_i\| = \max_{j=1,\dots,m} \sqrt{d_j^2 + |h_j - h_i|^2 + |\theta_j - \theta_i|^2} \quad (39)$$

密度関数は、動径基底関数ネットワークを用いてサンプル点の疎な領域に極小値を生成させる関数である。密度関数を用いたサンプル点の決定方法は、以下の手順で行われる。

(1) すべての成分を 1 とした列ベクトルを用意する。

$$\mathbf{y} = (1, \dots, 1)^T_{m \times 1} \quad (40)$$

(2) 式(41)で係数 $\boldsymbol{\alpha}^* = (\alpha_1^*, \dots, \alpha_m^*)$ を求める。

$$\boldsymbol{\alpha}^* = (\mathbf{K}^T \mathbf{K} + \boldsymbol{\varepsilon})^{-1} \mathbf{K}^T \mathbf{y} \quad (41)$$

ここで \mathbf{K} , $\boldsymbol{\varepsilon}$ はそれぞれ式(42)と式(43)で定義される行列である。式(43)中の $\boldsymbol{\varepsilon}$ は重みパラメータであり、本稿では $\boldsymbol{\varepsilon} = 1.0 \times 10^{-2}$ とした。

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \dots & K(\mathbf{x}_1, \mathbf{x}_m) \\ K(\mathbf{x}_2, \mathbf{x}_1) & K(\mathbf{x}_2, \mathbf{x}_2) & \dots & K(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_m, \mathbf{x}_1) & K(\mathbf{x}_m, \mathbf{x}_2) & \dots & K(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix} \quad (42)$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon & 0 & \dots & 0 \\ 0 & \varepsilon & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varepsilon \end{bmatrix} \quad (43)$$

(3) 次の最小化問題(44)を解き、得られた最小解を追加サンプル点とする。なお、密度関数とは本問題の目的関数を指す。

$$\text{Find } \mathbf{x}^* = \underset{\mathbf{x} \in S}{\operatorname{argmin}} \sum_{i=1}^m \alpha_i^* K(\mathbf{x}, \mathbf{x}_i) \quad (44)$$

3.5 実装方法

図4に全体の計算流れ図を示す。最適化開始時点ではニューラルネットワークの学習に使用するデータがないため、CFD を用いてピンホールとブローホールを計算するこ

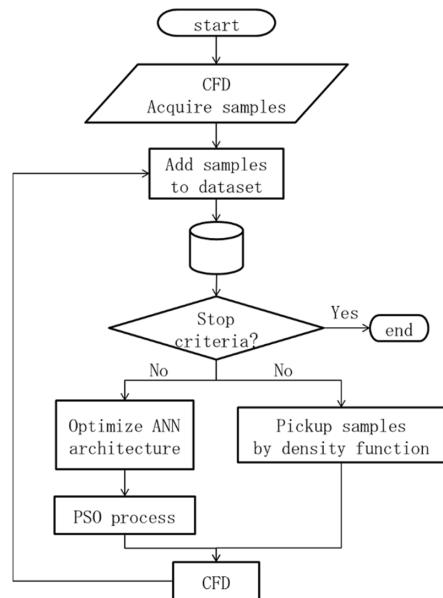


図4 逐次近似最適化のフロー図
Sequential approximate optimization process

とで、その時のノズル条件と合わせてデータセットを作成する。ニューラルネットワークの外挿性能は高くないため、ここで用意するデータセットは探索範囲の境界に相当する条件を中心に数点選択した。具体的には、浸漬深さ(mm)が 60, 100, 120, 吐出孔高さ(mm)が 125, 175, 155, 195, 225, ノズル角度(degree)が 15, 20, 25, 30, 35, 40 である時の組み合わせた条件を計算した。

その後、当該データセットを用いてニューラルネットワークの最適な構造を探すとともに学習を行い、得られたモデルを用いて粒子群最適化を実行する。粒子群最適化により、改善可能なノズルの候補条件が得られる。併せて動径基底関数ネットワークを用いた密度関数により、データがない領域からノズル条件を求めて追加データ点とする。粒子群最適化と密度関数により得られたノズル条件に対して、正確なピンホールとブローホールを得るために CFD を実行する。ここまでで得られたノズル条件とピンホール及びブローホールをまとめてデータセットに追加する。ここで、解候補をできるだけ多く確保してデータセットを拡充することでニューラルネットワークの予測精度を短工期にあげるために、粒子群最適化と密度関数によるサンプル点の追加は3回ずつ繰り返すこととした。以上の処理を反復して行い、所定の回数を終えた時点で全処理を終了とする。

4. 最適化結果と考察

図5に、黒丸として最適化開始前に用意した浸漬ノズル条件の CFD 計算値 (base condition)、黒三角として逐次近似最適化の反復を 10 回行って得られた浸漬ノズル条件でのニューラルネットワークによる予測値 (prediction by neural network)、白丸として同条件並びに動径基底関数ネットワークによる密度関数で求めた浸漬ノズル条件の

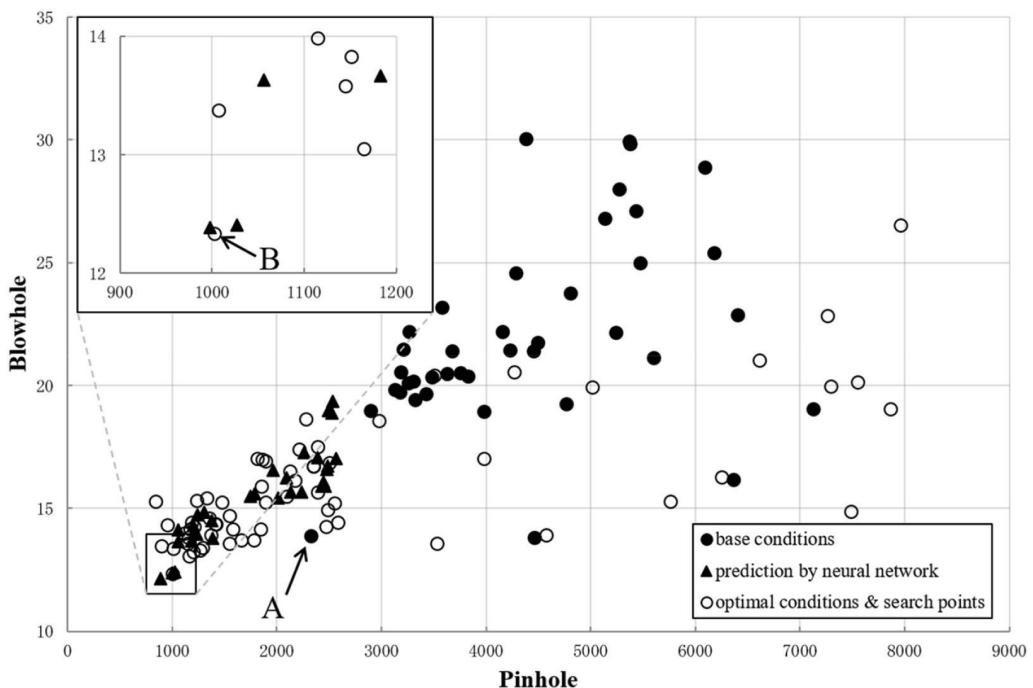


図5 ピンホールとブローホールの散布図
Scatter plot for pinhole and blowhole

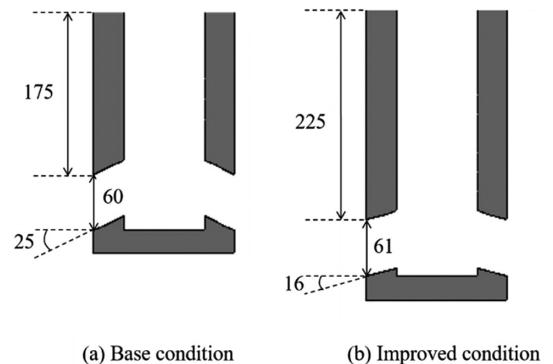
表4 条件別のピンホールとブローホール
Pinholes and blowholes by condition

	Base condition	Improved condition
Pinhole	2322	1002
Blowhole	13.8	12.3

CFD 計算値 (optimal conditions & search points) を示す。本稿では、理想である最小のピンホールとブローホールをいずれも 0 としているため、図の左下ほど良い条件である。ニューラルネットワークによる最適化結果 (黒三角) は、手動での探索結果 (黒丸) よりも良化する条件を予測している。この条件で CFD 計算を実施した結果 (白丸) とニューラルネットワークの予測値 (黒三角) は一致しない場合もあるが、概ね同じ傾向を示すことが確認された。よって、ニューラルネットワークでの予測と確認のための CFD 計算を自動的に繰り返すことにより、効率的かつ短時間で改善可能な条件を見出すことができる。

図5中の条件 A は手動による最良点 (黒丸)、条件 B は CFD 確認計算 (白丸) の最良点を示している。以降、これらのこととを単に基本条件、改善条件と呼ぶ。表4に基本条件と改善条件のピンホールとブローホールを示す。いずれの数値も改善条件の場合が小さくなっている、ピンホールで約 56%, ブローホールで約 10% 低減できた。

図6にそれぞれのノズル形状を示す。基本条件の浸漬ノズルについては、吐出孔高さとノズル角度が今回考えた探索領域における下限値近くの値となっていることから、水平に近く速い吐出流を得ることがピンホールとブローホールを低減する傾向にあることが分かる。図7(a)は基本条件



(a) Base condition (b) Improved condition
図6 浸漬ノズルの寸法比較
Comparison of immersion nozzle shapes

での CFD 計算で得られた流速を示しており、鋳型短辺に衝突した吐出流が上下に分岐している。このような流れはダブルロール流動パターンとして知られている⁴³⁾。ダブルロール流動パターンで生じる上昇流は湯面に到達後、湯面を攪拌することで Ar 気泡が凝固界面に捕捉されることを抑制するため、ピンホールを小さくすることができる。また、吐出流とダブルロール流動パターンによる上昇/下降流により凝固界面に捕捉された Ar 気泡の洗浄効果も生み、ブローホールを小さくすることもできる。

改善条件は、これらの効果をより効果的に得られる結果となっている。浸漬ノズルの形状については、吐出孔高さは基本条件同等であるが、角度をより浅くしていることから、より水平方向への吐出流を生む結果となっている。図7(b)に見て取れるように、この効果によって強い上昇流が得られ、湯面の流速が強くなりピンホール改善につながったと考えられる。浸漬深さが深くなった理由としては、鋳

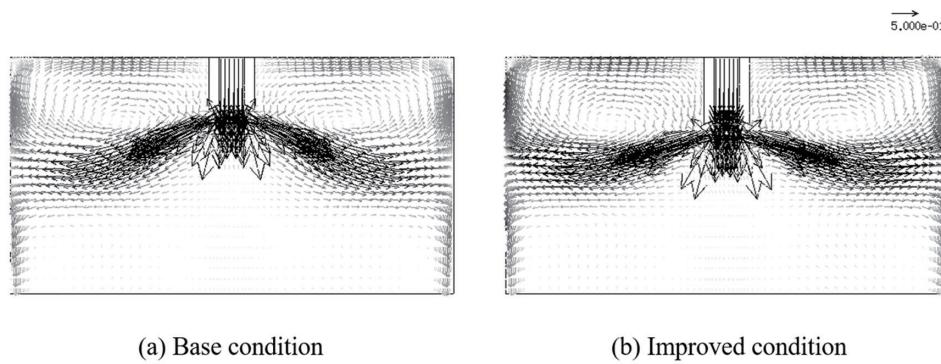


図7 鋳型内の溶鋼流速の比較
Comparison of velocity in the mold

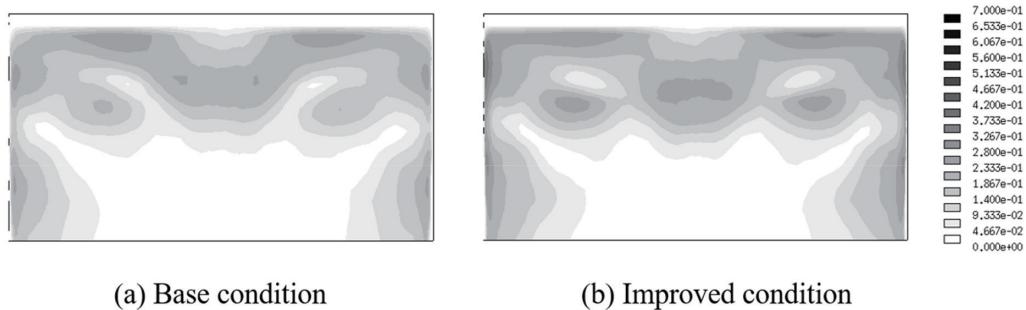


図8 鋳型内の溶鋼流速の大きさの比較
Comparison of velocity magnitude in the mold

型全面に流速を生じさせ、凝固シェルへ捕捉された気泡を流すことでブローホールを低減するためと考えられる。図8に流速の大きさを示す。基本条件に対して改善条件では、全体的に流速が大きくなっていることが確認できる。また、湯面に到達した流れが浸漬ノズルのある鋳型中央部で下方に流れると、同時にAr気泡を運ぶためブローホールが悪化する。よって、浸漬ノズルを深くすることで湯面の中央部に向かう流れを強くしすぎないようにしていると考えられる。

5. 結 言

本稿では連続鋳造機における鋳型内溶鋼流動及び伝熱凝固のCFD計算とニューラルネットワークを用いた最適化法を組み合わせた浸漬ノズル形状最適化方法並びにその適用例について報告した。本法では、CFD計算に鋳型内溶鋼流動と伝熱凝固の数学モデル、最適化の枠組みに逐次近似最適化、CFD計算の代理モデルにニューラルネットワーク、最適化アルゴリズムに粒子群最適化を使用して組み合わせた。本法を、ピンホールとブローホールを低減する浸漬ノズル形状探索に適用した結果、事前にCFD計算により求めたピンホールとブローホールに対して、ピンホールで約56%、ブローホールで約10%低減することができた。また、浸漬深さを深く、吐出孔高さを低く、ノズル角度を浅くしたノズル形状が良い傾向が見られた。これにより、水平に近く速度の速い吐出流と強い上昇流が得られ、湯面がかき混ぜられることでピンホールを低減できたと考えら

れる。また、鋳型内全面にかけて凝固界面に付着したAr気泡を洗い流す効果も得られることでブローホールを低減できたと考えられる。

今後の課題としては、EMSやEMBrを考慮した場合のノズル形状最適化が挙げられる。また、本稿では、ピンホールとブローホールの評価値を平均化することで单一目的最適化として解いたが、多目的最適化とした場合の検討も必要であると考えられる。

参考文献

- 1) Yokotani, S., Haseo, S., Asako, Y., Takagi, S., Ayata, K., Szekely, J., Hara, S.: *Tetsu-to-Hagané*. 82, 581 (1996) (in Japanese), https://doi.org/10.2355/tetsutohagane1955.82.7_581
- 2) Yoshida, J., Iguchi, M., Yokota, S.-I.: *Tetsu-to-Hagané*. 89, 264 (2002) (in Japanese), https://doi.org/10.2355/tetsutohagane1955.88.5_264
- 3) Takatani, K.: *Tetsu-to-Hagané*. 90, 751 (2004) (in Japanese), https://doi.org/10.2355/tetsutohagane1955.90.10_751
- 4) Tsukaguchi, Y., Hayashi, H., Kurimoto, H., Yokoya, S., Marukawa, K., Tanaka, T.: *Tetsu-to-Hagané*. 95, 33 (2009) (in Japanese), <https://doi.org/10.2355/tetsutohagane.95.33>
- 5) Li, B. W., Tian, X. Y., Wang, E. G., He, J. C.: *Acta Metall. Sin-Engl.* 20, 15 (2007), [https://doi.org/10.1016/S1006-7191\(07\)60003-9](https://doi.org/10.1016/S1006-7191(07)60003-9)
- 6) Park, H.-S., Nam, H., Yoon, J. K.: *ISIJ Int.* 41, 974 (2001), <https://doi.org/10.2355/isijinternational.41.974>
- 7) Liu, H., Yang, C., Zhang, H., Zhai, Q., Gan, Y.: *ISIJ Int.* 51, 392

- (2011), <https://doi.org/10.2355/isijinternational.51.392>
- 8) Morales, R.D., Tang, Y., Nitzl, G., Eglsäer, C. Hckl, G.: ISIJ Int. 52, 1607 (2012), <https://doi.org/10.2355/isijinternational.52.1607>
- 9) Xuan, M., Chen, M.: Metals. 11, 1223 (2021), <https://doi.org/10.3390/met11081223>
- 10) Takatani, K., Tanizawa, Y., Mizukami, H., Nishimura, K.: ISIJ Int. 41, 1252 (2001), <https://doi.org/10.2355/isijinternational.41.1252>
- 11) Takatani, K.: ISIJ Int. 43, 915 (2003), <https://doi.org/10.2355/isijinternational.43.915>
- 12) Leifsson, L., Koziel, S.: J. Comput. Sci. 10, 45 (2015), <https://doi.org/10.1016/j.jocs.2015.01.003>
- 13) Yang, C., Huang, F.: J. Hydrom. 28, 947 (2016), [https://doi.org/10.1016/S1001-6058\(16\)60696-0](https://doi.org/10.1016/S1001-6058(16)60696-0)
- 14) Bernardini, E., Spence, S.M.J., Wei, D., Kareem, A.: J. Wind Eng. Ind. Aerodyn. 144, 154 (2015), <https://doi.org/10.1016/j.jweia.2015.03.011>
- 15) Song, X., Lv, L., Sun, W., Zhang, J.: Struct. Multidisc. Optim. 60, 965 (2019), <https://doi.org/10.1007/s00158-019-02248-0>
- 16) Wu, H., Liu, X., An, W., Chen, S., Lyu, H.: Compt. Fluids. 198, 104393 (2020), <https://doi.org/10.1016/j.compfluid.2019.104393>
- 17) Li, J., Du, X., Martins, J.R.R.A.: Prog. Aerosp. Sci. 134, 100849 (2022), <https://doi.org/10.1016/j.paerosci.2022.100849>
- 18) Nakayama, H., Yun, Y., Yoon, M.: Sequential Approximate Multiobjective Optimization Using Computational Intelligence. Springer, Berlin, Heidelberg, 113 (2009), <https://doi.org/10.1007/978-3-540-88910-6>
- 19) Kitayama, S.: Kougaukei No Saiteki Sekkeihou (Kikai Gakusyu Wo Katuyou Shita Riron To Jissen). Kyoritsu Syuppan, Tokyo, (2021) (in Japanese)
- 20) Kennedy, J., Eberhart, R.: Proceedings of ICNN'95 – International Conference on Neural Networks, 4, 1942 (1995), <https://doi.org/10.1109/ICNN.1995.488968>
- 21) Kitayama, S., Arakawa, M., Yamazaki, K.: Optim. Eng. 12, 535 (2011), <https://doi.org/10.1007/s11081-010-9118-y>
- 22) Hinze, J.O.: Turbulence. 2nd ed., McGraw Hill, New York, 460 (1975)
- 23) Wang, Z., Tanaka, M., Inoue, T.: Nihon-Kikai-Gakkai-Ronbunshu A. 53, 1735 (1987)
- 24) Hirt, C.W., Nichols, B.D. Romero, N.C.: SOLA-A Numerical Solution Algorithm for Transient Fluid Flows. Los Alamos Sci. Lab, Los Alamos, LA-5852 (1975)
- 25) Toh, T., Hasegawa, H., Harada, H.: ISIJ Int. 41, 1245 (2001), <https://doi.org/10.2355/isijinternational.41.1245>
- 26) Saffman, P.G.: J. Fluid Mech. 22, 385 (1965), <https://doi.org/10.1017/S0022112065000824>
- 27) Mukai, K., Lin, W.: Tetsu-to-Hagané. 80, 533 (1994) (in Japanese), https://doi.org/10.2355/tetsutohagane1955.80.7_533
- 28) Mukai, K., Zhong, L., Zeze, M.: ISIJ Int. 46, 1810 (2006), <https://doi.org/10.2355/isijinternational.46.1810>
- 29) Taniguchi, S., Brimacombe, J.K.: Tetsu-to-Hagané. 80, 24 (1994), https://doi.org/10.2355/tetsutohagane1955.80.4_312
- 30) Miyake, T., Morishita, M., Nakata, H., Kokita, M.: ISIJ Int. 46, 1817 (2006), <https://doi.org/10.2355/isijinternational.46.1817>
- 31) Burton, J.A., Prim, R.C., Slichter, W.P.: J. Chem. Phys. 21, 1987 (1953), <https://doi.org/10.1063/1.1698728>
- 32) Miki, Y., Ohno, H., Kishimoto, Y., Tanaka, S.: Tetsu-to-Hagané. 97, 423 (2011) (in Japanese), <https://doi.org/10.2355/tetsutohagane.97.423>
- 33) Mollinger, A.M., Nieuwstadt, F.T.M.: J. Fluid Mech. 316, 285 (1996), <https://doi.org/10.1017/S0022112096000547>
- 34) Apicella, A., Donnarumma, F., Isgrò, F., Prevete, R.: Neural Netw. 138, 14-32 (2021), <https://doi.org/10.1016/j.neunet.2021.01.026>
- 35) Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, (2016)
- 36) Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: J. Mach. Learn. Res. 15, 1929-1958 (2014)
- 37) Glorot, X., Bengio, Y.: Proc. 13th Int. Conf. on Artificial Intelligence and Statistics (AISTATS2010). PMLR, 249 (2010)
- 38) TensorFlow, Google LLC: <https://www.tensorflow.org/> (accessed 2021-10-20)
- 39) Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Proc. 25rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. 2623-2631 (2019), <https://doi.org/10.1145/3292500.3330701>
- 40) Aiyoshi, E., Yasuda, K.: MetaHeuristics To Oyou (MetaHeuristics and Applications). IEEJ. Tokyo, 69 (2007) (in Japanese)
- 41) Haykin, S.: Neural Networks (A Comprehensive Foundation). Macmillan College Publishing, New York, USA, (1994)
- 42) Nakayama, H., Arakawa, M., Sasaki, R.: Optim. Eng. 3, 201 (2002), <https://doi.org/10.1023/A:1020971504868>
- 43) Thomas, B.G., Zhang, L.: ISIJ Int. 41, 1181 (2001), <https://doi.org/10.2355/isijinternational.41.1181>



難波時永 Tokinaga NAMBA
先端技術研究所 数理科学研究部
主任研究員
千葉県富津市新富20-1 〒293-8511



岡田信宏 Nobuhiro OKADA
先端技術研究所 数理科学研究部
室長（主席研究員） 工学博士