Novel methodology for casting process optimization using Gaussian process regression and genetic algorithm

Yao Weixiong, *Yang Yi and Zeng Bin
(School of Manufacturing Sci. and Eng., Sichuan University, Chengdu 610065, China)

Abstract: High pressure die casting (HPDC) is a versatile material processing method for mass-production of metal parts with complex geometries, and this method has been widely used in manufacturing various products of excellent dimensional accuracy and productivity. In order to ensure the quality of the components, a number of variables need to be properly set. A novel methodology for high pressure die casting process optimization was developed, validated and applied to selection of optimal parameters, which incorporate design of experiment (DOE), Gaussian process (GP) regression technique and genetic algorithms (GA). This new approach was applied to process optimization for cast magnesium alloy notebook shell. After being trained, using data generated by PROCAST (FEM-based simulation software), the GP model approximated well with the simulation by extracting useful information from the simulation results. With the help of MATLAB, the GP/GA based approach has achieved the optimum solution of die casting process condition settings.

Key words: high pressure die casting; process optimization; numerical simulation; Gaussian process; genetic algorithm

CLC number: TG249.2/TP391.9 Document code: A Article ID: 1672-6421(2009)03-232-09

High pressure die casting (HPDC) has been widely used to manufacture a large variety of products with well-controlled dimensional accuracy and productivity, for example, the production of automotive components. However, many processing parameters are involved in the complex process, such as melt temperature, coolant temperature and cooling time, etc. A number of variables need to be properly set to ensure the quality of the components. As previously demonstrated [1,2], it is difficult to control and predict the complex process which is nonlinear and dynamic. It has been far from being optimized.

During the process design, various process parameters are selected by experienced engineers based on prior experience and/or reference handbooks, and later fine-tuned by trial-and-error approach. This method highly depends on experimental operation and, is costly and time consuming, especially when coming to new applications. At present, with the advancement in computer aided engineering technique, engineers can virtually simulate the processes to gain process insights, pinpoint potential design problems, and make rational decisions. Today, a number of commercial 3D simulation softwares have been developed for casting process design, for example PROCAST. It has significantly facilitated process design and parameters setting.

Generally speaking, it is time consuming to perform process optimization with simulations over an extensive combination of alternative process parameters. Furthermore, it would be a daunting task to go through the vast amount of simulation-generated data such as velocity field, pressure and temperature distribution, shear stress, just to name a few, to understand the complex, nonlinear interaction among the process variables, not to mention the setting for optimal process conditions. Jeong-Soo and Jongwoo (2002), Wang et al.(2008) have used nonlinear statistical regression techniques to create surrogate models [3,4], which approximate the simulations by extracting useful information from simulated results, to capture the characteristics of process simulation. Correlation between conditions and objectives can be obtained from such models, which can, in turn, be used for prediction. Furthermore, engineers can easily perform current global optimization methods, such as genetic algorithm (GA), on these models, rather than searching the optima randomly.

Several regression approaches, as described below, have been used in die casting processes optimization to build surrogate models with satisfying performance. They are easily to build and validate execution of further tasks such as prediction or optimization. As a linear regression technique, response surface methodology (RSM) has been used in many process optimization [4,5]. Since this regression method is problem-dependent, inconsistent performance may arise when come to such complex task as high pressure die casting process...
optimization, which is nonlinear and involves a large number of variables. Alternately, artificial neural networks (ANN) technique has been commonly adopted as an efficient nonlinear modeling approach [6-8]. Instant execution of predications can be done by well-trained neural networks. Unfortunately, it can not avoid the problems of local optimization. Initially developed for machine learning, support vector regression (SVR) has recently been accepted as a new function approximator, which can perform nonlinear regression through high-dimensional space with a sparse training dataset [9].

Since the surrogate models have been built based on a set of previously collected data points, the corresponding output can be easily predicted in response to a new set of inputs. To fulfill such task, the Gaussian process (GP) regression, which employs a Bayesian statistics approach, can also be adopted as a highly nonlinear regression technique. Unlike the previously discussed models, GP regression assumes a distribution for the output rather than thinking the output as a point value and then seeks to determine the parameters of the distribution. What should be taken into account is the availability of the data and the assumption that the associated error is normally distributed. Furthermore, GP regression predicts the most likely value as well as the variance for the desired response directly through statistical reasoning [10]. In this paper, GP regression technique is employed for HPDC process optimization.

Once the established model can perform well with satisfactory accuracy, further predication or optimization can be done. In this paper, a hybrid optimization algorithm (GA) is used to evaluate the surrogate model to search the global optimal solutions.

1 General theory

1.1 Mathematical model

Technically speaking, the optimal design of filling and solidification process is an optimization problem with objective function subjected to a set of partial differential equations, i.e., momentum conservation equation, mass conservation equation, energy conservation equation, equation of volume function, mass transfer continuum equation, and heat transfer equation on boundary [11]. The optimization objectives vary from temperature uniformity, reduced residual stress to minimal part shrinkage and warpage, or some combinations of them, depending on user’s requirement. In order to minimize or maximize the objectives, the input process condition variables i.e. melt temperature, injection speed, injection pressure, should be optimized. The optimization objectives and process condition parameters are defined by user, while the mould filling and solidification process are governed by the following equation [12].

Equation (1): Momentum conservation Navier-Stokes equation:

$$\rho \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = -\frac{\partial P}{\partial x} + \rho g + \mu \nabla^2 u$$

Equation (2): Mass conservation-continuity equation:

$$\frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} = 0$$

Equation (3): Energy conservation equation:

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} + w \frac{\partial T}{\partial z} = \frac{\lambda}{\rho C_p} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) + \frac{Q}{\rho C_p}$$

Equation (4): Equation of volume function:

$$\frac{\partial F}{\partial t} + u \frac{\partial F}{\partial x} + v \frac{\partial F}{\partial y} + w \frac{\partial F}{\partial z} = 0$$

Equation (5): Mass transfer continuum equation:

$$\frac{\partial \rho}{\partial t} + (\rho u \cdot \nabla)u = 0$$

Equation (6): Heat transfer equation on boundary:

$$-k \nabla T \cdot n = h \times f(t) \times g(T) \times [T - T_a]$$
over traditional methods, and it covers a wider range in the experimental domain with fewer sample points. This means more chance of locating the actual optimal process parameters instead of just the best within the limited domain investigated in traditional methods [Jack, 2008]. To select the appropriate sample points for training the Gaussian process regression, two important criteria should be taken into consideration: First, the samples must portray nonlinearity. Secondly, samples must spread over the input design space proportionally. To enable the model to conduct a sound prediction, it has to be trained using data sample over the entire input design space [13].

1.3 Theory of Gaussian process regression approach

Given a data set $D$ consisting of $N$ pairs of $L$-dimensional inputs $x_n$ and scalar outputs $t_n$ for $n=1…N$, a GP prediction model is concerned with evaluating the probability $P(t_{N+1}|D,x_{N+1})$, where $x_{N+1}$ is the new input vector and the corresponding output is $t_{N+1}$. In this process, the $P(t_{N}|C_N,x_N)$ is assumed to follow a Gaussian distribution as given by Equation (7)[14]:

$$P(t_N|C_N,x_N) = \frac{1}{\sqrt{(2\pi)^N|C_N|}}\exp\left[-\frac{1}{2}(t-u)^T C_N^{-1}(t-u)\right]$$ (7)

Where: $t_n=(t_1(x_1), t_2(x_2), ..., t_N(x_N))$, $C_N$ is covariance matrix for $P(t_{n}|x_{n})$, and $u$ is the mean, which will be zero for properly normalized data:

$$P(t_N|C_N,x_N) = \frac{1}{\sqrt{(2\pi)^N|C_N|}}\exp\left[-\frac{1}{2}t^T C_N^{-1}t_N\right]$$ (8)

If a new set of predictor and target value is included, a similar distribution results:

$$P(t_{N+1}|C_{N+1}, x_N, x_{N+1}) = \frac{1}{\sqrt{(2\pi)^{N+1}|C_{N+1}|}}\exp\left[-\frac{1}{2}t^T_{N+1}C_{N+1}^{-1}t_{N+1}\right]$$ (9)

Since the primary interest is in the predictive probability for the new target given the known data, this corresponds to the following conditional distribution:

$$P(t_{N+1}|x_N, x_{N+1}, C_N) = \frac{P(t_{N+1}|C_{N+1}, x_N, x_{N+1})}{P(t_{N+1}|C_N, x_N)}$$

$$= \frac{1}{\sqrt{(2\pi)^N|C_N|}}\exp\left[-\frac{1}{2}(t_{N+1}^TC_N^{-1}t_{N+1}-t_{N+1}^TC_N^{-1}t_{N+1})\right]$$ (10)

Hence this conditional distribution can be used to make predictions about $t_{N+1}$. The covariance matrix is the covariance between the data points and targets, and $C_N$ is contained in $C_{N+1}$ as shown:

$$C_{N+1} = \begin{pmatrix} C_N & k' \\ k & k' \end{pmatrix}$$ (11)

Where: $k'$ and $k$ are defined as:

$$k'=(C(x_1,x_{N+1}), C(x_2,x_{N+1}), ..., C(x_N,x_{N+1}))$$ (12)

$$k=C(x_{N+1}, x_{N+1})$$ (13)

After some matrix manipulation, the expression for the probability of the target becomes:

$$P(t_{N+1}|t_N, x_N, x_{N+1}, C_N)$$

$$= \frac{1}{\sqrt{(2\pi)|C_N|}}\exp\left[-\frac{(t_{N+1}-\hat{t}_{N+1})^2}{2\sigma_{N+1}^2}\right]$$ (14)

Where:

$$\hat{t}_{N+1} = k^TC_N^{-1}t_N$$ (15)

is mean for the predicted target value;

$$\sigma_{N+1}^2 = k - k^TC_N^{-1}k$$ (16)

is covariance of the predicted target value.

They are very important results, and when using GP, the prediction of the target comes coherently with estimation of the accuracy of the prediction. This is especially useful in engineering application since the accuracy as well as applicability should be concerned. There is still one more thing needed: Some type of covariance function should be given to evaluate the covariance. With an example of covariance function [14], the covariance matrix is expressed as:

$$C_N = \begin{pmatrix} C(x_1,x_1) & C(x_1,x_2) & ... & C(x_1,x_N) \\ C(x_2,x_1) & C(x_2,x_2) & ... & C(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ C(x_N,x_1) & C(x_N,x_2) & ... & C(x_N,x_N) \end{pmatrix}$$ (17)

Where, $\theta_1$ controls the overall vertical scale of the variance relative to the mean of GP, $\theta_2$ sets bias of the correction, $\theta_3$ sets the noise level along the diagonal and, $r$ allows a different distance measure for each input dimension.

Define hyperparameter:

$$\theta = (\theta_1, \theta_2, \theta_3, r)$$ (18)

According to Bayes’ theorem, the posterior probability distribution of the hyperparameters becomes:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$ (19)

The evidence term is independent of $\theta$; and the two other terms, the likelihood and the prior $\theta$, will be considered in terms of their logs, then the log likelihood $L$ for the GP is:

$$L = \ln P(t_N|x_N, \theta) + \ln P(\theta|t_N) - \ln P(t_N|x_N)$$

$$= \ln \left[ \frac{1}{\sqrt{(2\pi)^N|C_N|}}\exp\left[-\frac{1}{2}t^T C_N^{-1}t_N\right] \right] + \ln P(\theta) + \text{const}$$ (20)

And the partial derivative with respect to the hyperparameters $\theta$:

$$\frac{\partial L}{\partial \theta} = -\frac{1}{2} \text{trace}\left[ C_N^{-1} \frac{\partial C_N}{\partial \theta} \right] + \frac{1}{2} t^T C_N^{-1} \frac{\partial C_N}{\partial \theta} t_N + \frac{\partial \ln P(\theta)}{\partial \theta}$$ (21)

To find the optimal value of hyperparameters, it is easy to perform likelihood optimization of hyperparameters.
with common optimization algorithm. The optimal value of hyperparameters is used to construct the covariance matrix, which, in return, is inversed to predict the new target \( t_{n+1} \) as well as its covariance according to equations (15) and (16). A general procedure of GP is shown in Fig.1.

**Fig.1: Procedure of Gaussian Process (GP)**

1. 4 Genetic algorithms

Once the GP model can perform well with acceptable accuracy, by extracting useful information from simulated results, not only can it predict the un-simulated but also convert the searching for optima into an efficient optimization procedure. The GP regression based models are non-differentiable, nonlinear, and may possess multiple stationary points, thus traditional gradient-based approaches are not suitable in further optimization. In this case, it needs to use a specific technique to find the optimal solution. Genetic algorithms (GA) provides one of these methods. Genetic algorithm differs from conventional optimization techniques in following ways:

(a) GA operates with coded versions of the problem parameters rather than parameters themselves.

(b) Almost all conventional optimization techniques search from a single point, while GA always operates on a whole population of points (strings). This plays a major role in the robustness of genetic algorithms.

(c) GA uses fitness function for evaluation rather than derivatives. As a result, they can be applied to any kind of continuous or discrete optimization problem. The key point to be performed here is to identify and specify a meaningful decoding function.

(d) GA uses probabilistic transition operations while conventional methods for continuous optimization apply deterministic transition operations.

Genetic algorithms have been applied to metal forming (MF) process. A number of engineers have already solved sophisticate engineering problems using genetic algorithms. Jitender et al., Tsoukalas, Vijian and Arunachalam, and Krimpenis have applied GA in die casting process optimization.

The use of a GA requires the determination of six fundamental issues:

- chromosome representation, selection function, genetic operators, creation of the initial population, termination criteria, and the evaluation function. The genetic algorithms (GAs) loop over an iteration process to make the population evolve. Each of the iteration consists of the following steps:

  (a) **SELECTION**: Selection is done randomly with a probability depending on the relative fitness of the individuals so that the best ones are often chosen for reproduction rather than the poor ones.

  (b) **REPRODUCTION**: In the second step, offspring are bred by the selected individuals.

  For generating new chromosomes, the algorithm can use both recombination and mutation.

  (c) **EVALUATION**: Fitness of the new chromosomes is evaluated.

  (d) **REPLACEMENT**: Individuals from the old population are replaced by the new ones.

The algorithm is stopped when the population converges toward the optimal solution. A flow chart of GA is given in Fig.2.

**Fig.2: Flow chart of GA**

1. 5 Methodology of die casting process optimization with GP and GA

Base on GP method, a novel procedure of die casting process optimization can be illustrated in Fig.3.
In this procedure, one should first determine inputs (process conditions) and outputs (objective values). The optimization objectives depend on user’s requirement and, in order to simply the problem, only those inputs which have a major influence on the objective need to be selected to define the optimization problem. Then, design of experiments DOE for simulation is used to generate samples needed for initial training of the GP model. Each set of input parameters generated by (DOE) will be analyzed by CAE simulation, i.e. PROCAST. Collect the results and extract useful data from the simulation for building the GP regression model. After testing the validity of the GP model with spare samples, when accepted, the hybrid GA will approach to the near-optimal solution.

2 A case study: validation and process optimization for magnesium alloy shell of a notebook

2.1 Validated GP model

In this study, the proposed Gaussian regression model will be trained with the simulation results of high pressure die casting magnesium alloy shell of a notebook as shown in Fig. 4.

Then, the validation of the trained GP model will be tested by the spare simulation data. At last, the GA will be applied to optimize the five process conditions in order to minimize the temperature difference at the end of filling process \( \Delta T_{end} \). The details of the experiment setup, procedure, and result analysis can be found in [19]. The simulation results are given in Table 1. By defining the temperature difference at the end of filling process, \( \Delta T_{end} \), as the objective function and, the melt temperature, mold temperature, filling speed, location of filling port, size of filling port as process conditions, the optimization problem can be expressed below together with the process ranges:

\[
\text{Minimize: } \Delta T_{end} \\
\text{Subject to: } 655 \degree C \leq T_{melt} \leq 730 \degree C \\
200 \degree C \leq T_{mold} \leq 240 \degree C \\
0.5 \text{ m/s} \leq v_{fill} \leq 3.2 \text{ m/s} \\
1.5 \text{ mm} \leq \phi_{fill} \leq 5 \text{ mm} \\
80 \text{ mm} \leq L_{fill} \leq 175 \text{ mm}
\]

Where, \( T_{melt} \) is melt temperature, \( T_{mold} \) mold temperature, \( v_{fill} \) filling speed, \( \phi_{fill} \) diameter of filling port, \( L_{fill} \) location of filling port.

In the validation process, 17 samples were divided into a training set and a testing set randomly, as tabulated in Table 1. The testing set was put aside, the created GP regression model using the training set would subsequently be validated with the testing data to assess the model’s accuracy. Setups of GP regression modeling see Table 2. Result of prediction with testing set is shown in Table 3.

From the table above, the predicted values \( \Delta T_{end} \) based on GP model agree well with the simulation results. Considering the limited sample points, the model was considered to be acceptable. After testing, the GP model was improved using a merged data set comprised of training set and testing set. At last, the predictions of temperature difference \( \Delta T_{end} \) based on GP regression model were compared with the simulation results, as shown in Table 4.
### Table 1: Simulation results for training and testing

<table>
<thead>
<tr>
<th>Standard order</th>
<th>( T_{\text{melt}} \degree C )</th>
<th>( T_{\text{mold}} \degree C )</th>
<th>( v_{\text{fill}} ) m/s</th>
<th>( \Phi_{\text{fill}} ) mm</th>
<th>( L_{\text{fill}} ) mm</th>
<th>( \Delta T_{\text{end}} ) °C</th>
<th>( \Delta T_{\text{pred}} ) °C</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>175.00</td>
<td>9.004</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>122.00</td>
<td>9.225</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>680</td>
<td>200</td>
<td>1.0</td>
<td>2.0</td>
<td>51.00</td>
<td>29.027</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>690</td>
<td>235</td>
<td>0.5</td>
<td>3.0</td>
<td>86.50</td>
<td>24.981</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>670</td>
<td>205</td>
<td>2.3</td>
<td>5.0</td>
<td>62.50</td>
<td>6.513</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>710</td>
<td>230</td>
<td>1.5</td>
<td>2.5</td>
<td>73.25</td>
<td>14.643</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>700</td>
<td>225</td>
<td>2.8</td>
<td>3.5</td>
<td>99.75</td>
<td>8.214</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>720</td>
<td>240</td>
<td>2.5</td>
<td>4.5</td>
<td>140.25</td>
<td>6.388</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>680</td>
<td>215</td>
<td>3.0</td>
<td>1.5</td>
<td>152.75</td>
<td>16.642</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>670</td>
<td>200</td>
<td>2.0</td>
<td>1.5</td>
<td>160.00</td>
<td>22.563</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>720</td>
<td>220</td>
<td>2.7</td>
<td>4.0</td>
<td>140.00</td>
<td>7.371</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>700</td>
<td>230</td>
<td>3.2</td>
<td>1.0</td>
<td>170.00</td>
<td>33.107</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>710</td>
<td>235</td>
<td>1.0</td>
<td>5.0</td>
<td>130.00</td>
<td>10.650</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>690</td>
<td>240</td>
<td>2.2</td>
<td>2.5</td>
<td>110.00</td>
<td>12.351</td>
<td>training</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>655</td>
<td>210</td>
<td>3.2</td>
<td>5.0</td>
<td>164.50</td>
<td>4.701</td>
<td>testing</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>670</td>
<td>230</td>
<td>3.0</td>
<td>4.0</td>
<td>82.00</td>
<td>7.369</td>
<td>testing</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>115.00</td>
<td>9.182</td>
<td>testing</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Setups of GP regression modeling

<table>
<thead>
<tr>
<th>GP parameters</th>
<th>Setups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input ( x_{\text{y}} )</td>
<td>( T_{\text{melt}}, T_{\text{mold}}, v_{\text{fill}}, \Phi_{\text{fill}}, L_{\text{fill}} )</td>
</tr>
<tr>
<td>Output ( t_{\text{e}} )</td>
<td>Temperature difference ( \Delta T_{\text{end}} )</td>
</tr>
<tr>
<td>Covariance function</td>
<td>( C(x_{i,j}, x_{i,j}) = \theta_1 \exp \left[ -\frac{1}{2 \theta_2} \sum_{j=1}^{n} (x_{i,j} - x_{i,j}^{(i)})^2 / r_j^2 \right] + \theta_2 )</td>
</tr>
<tr>
<td>Hyperparameters optimization algorithm</td>
<td>Hybrid GA</td>
</tr>
<tr>
<td>Software implementation</td>
<td>MATLAB</td>
</tr>
</tbody>
</table>

### Table 3: Result of prediction with testing set

<table>
<thead>
<tr>
<th>Test order</th>
<th>( T_{\text{melt}} \degree C )</th>
<th>( T_{\text{mold}} \degree C )</th>
<th>( v_{\text{fill}} ) m/s</th>
<th>( \Phi_{\text{fill}} ) mm</th>
<th>( L_{\text{fill}} ) mm</th>
<th>( \Delta T_{\text{end}} ) °C</th>
<th>( \Delta T_{\text{pred}} ) °C</th>
<th>Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>655</td>
<td>210</td>
<td>3.2</td>
<td>5.0</td>
<td>164.50</td>
<td>4.701</td>
<td>9.120</td>
<td>1.29%</td>
</tr>
<tr>
<td>Test 2</td>
<td>670</td>
<td>230</td>
<td>3.0</td>
<td>4.0</td>
<td>82.00</td>
<td>7.369</td>
<td>9.182</td>
<td>8.27%</td>
</tr>
<tr>
<td>Test 3</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>115.00</td>
<td>9.182</td>
<td>9.348</td>
<td>3.32%</td>
</tr>
</tbody>
</table>

### Table 4: Result of prediction with all sample points

<table>
<thead>
<tr>
<th>Standard order</th>
<th>( T_{\text{melt}} \degree C )</th>
<th>( T_{\text{mold}} \degree C )</th>
<th>( v_{\text{fill}} ) m/s</th>
<th>( \Phi_{\text{fill}} ) mm</th>
<th>( L_{\text{fill}} ) mm</th>
<th>( \Delta T_{\text{end}} ) °C</th>
<th>( \Delta T_{\text{pred}} ) °C</th>
<th>Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>175.00</td>
<td>9.004</td>
<td>9.120</td>
<td>1.29%</td>
</tr>
<tr>
<td>2</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>122.00</td>
<td>9.225</td>
<td>9.145</td>
<td>0.86%</td>
</tr>
<tr>
<td>3</td>
<td>680</td>
<td>200</td>
<td>1.0</td>
<td>2.0</td>
<td>51.00</td>
<td>29.027</td>
<td>29.302</td>
<td>0.98%</td>
</tr>
<tr>
<td>4</td>
<td>690</td>
<td>235</td>
<td>0.5</td>
<td>3.0</td>
<td>86.50</td>
<td>24.981</td>
<td>23.936</td>
<td>4.18%</td>
</tr>
<tr>
<td>5</td>
<td>670</td>
<td>205</td>
<td>2.3</td>
<td>5.0</td>
<td>62.50</td>
<td>6.513</td>
<td>6.873</td>
<td>5.52%</td>
</tr>
<tr>
<td>6</td>
<td>710</td>
<td>230</td>
<td>1.5</td>
<td>2.5</td>
<td>73.25</td>
<td>14.643</td>
<td>14.164</td>
<td>3.27%</td>
</tr>
<tr>
<td>7</td>
<td>700</td>
<td>225</td>
<td>2.8</td>
<td>3.5</td>
<td>99.75</td>
<td>8.214</td>
<td>7.940</td>
<td>3.33%</td>
</tr>
<tr>
<td>8</td>
<td>720</td>
<td>240</td>
<td>2.5</td>
<td>4.5</td>
<td>140.25</td>
<td>6.388</td>
<td>6.754</td>
<td>7.13%</td>
</tr>
<tr>
<td>9</td>
<td>660</td>
<td>215</td>
<td>3.0</td>
<td>1.5</td>
<td>152.75</td>
<td>16.642</td>
<td>16.086</td>
<td>3.36%</td>
</tr>
<tr>
<td>10</td>
<td>670</td>
<td>200</td>
<td>2.0</td>
<td>1.5</td>
<td>160.00</td>
<td>22.563</td>
<td>22.174</td>
<td>1.72%</td>
</tr>
<tr>
<td>11</td>
<td>720</td>
<td>220</td>
<td>2.7</td>
<td>4.0</td>
<td>140.00</td>
<td>7.371</td>
<td>7.970</td>
<td>8.14%</td>
</tr>
<tr>
<td>12</td>
<td>700</td>
<td>230</td>
<td>3.2</td>
<td>1.0</td>
<td>170.00</td>
<td>33.107</td>
<td>32.853</td>
<td>1.06%</td>
</tr>
<tr>
<td>13</td>
<td>710</td>
<td>235</td>
<td>1.0</td>
<td>5.0</td>
<td>130.00</td>
<td>10.650</td>
<td>11.020</td>
<td>3.47%</td>
</tr>
<tr>
<td>14</td>
<td>690</td>
<td>240</td>
<td>2.2</td>
<td>2.5</td>
<td>110.00</td>
<td>12.351</td>
<td>12.890</td>
<td>4.37%</td>
</tr>
<tr>
<td>15</td>
<td>655</td>
<td>210</td>
<td>3.2</td>
<td>5.0</td>
<td>164.50</td>
<td>4.701</td>
<td>4.320</td>
<td>8.08%</td>
</tr>
<tr>
<td>16</td>
<td>670</td>
<td>230</td>
<td>3.0</td>
<td>4.0</td>
<td>82.00</td>
<td>7.369</td>
<td>7.853</td>
<td>6.07%</td>
</tr>
<tr>
<td>17</td>
<td>730</td>
<td>220</td>
<td>2.0</td>
<td>4.0</td>
<td>115.00</td>
<td>9.182</td>
<td>9.348</td>
<td>1.81%</td>
</tr>
</tbody>
</table>
The averaged relative difference between the simulation results and prediction values is 3.80% with the largest relative difference of 8.14% (Trail No.11) and the least at 0.86% (Trail No.2).

For further analysis on the capability of GP regression technique, the prediction profiles generated by GP model are shown in Fig.5 and Fig.6. By defining temperature difference $\Delta T_{end}$ as the output, in Fig.5, melt temperature $T_{melt}$ and location of filling port $L_{fill}$ were defined as input variables with other conditions kept as constant, while in Fig.6, the filling speed $v_{fill}$ and location of filling port $L_{fill}$ were defined as inputs with other conditions as constant.

From Fig.5 and Fig.6, it can be seen that, as $L_{fill}$ increases, which means placing the filling port toward central while keeping other process conditions constant, the temperature difference $\Delta T_{end}$ decreases. This can be easily explained by computer simulation and physical experiment results due to the geometry symmetry of the casting shell. Another conclusion can be drawn from Fig.5: when the filling port is enlarged, the $\Delta T_{end}$ generally goes down. This indicates that, with the larger filling port while other conditions are set, the temperature distributes normally over the filled cavity. As and with larger filling port, the whole cavity tends to be filled simultaneously, leading to temperature equilibrium.

Thus, the GP model approximated well with the simulation by extracting useful information from simulated results. One can execute optimization of five process parameters to minimize the objective function in terms of the temperature difference $\Delta T_{end}$.

### 2.2 Process optimization with real-coded GA based on validated GP model

Since the established GP has been tested to be well-performed, hybrid GA is used to optimize the process parameters to minimize temperature difference $\Delta T_{end}$. In this paper a real-coded GA is applied to the problem, in which individual chromosome is represented by vector of float point numbers with its value within process ranges. As the coding method determines the genetic operators being used, they must fulfill the requirement.$^{[19,20]}$

**Selection**: Measured by fitness function, those successful individuals have better chance to be selected as ‘parents’. In this paper, the roulette wheel selection is used. For each individual $i$, a selection probability $p_i$ must be evaluated first in the selection as defined$^{[15]}$:

$$p_i = \frac{f_i}{\sum_{i=1}^{N_{pop}} f_i} \quad (22)$$

Where, $f_i$ is fitness of individual $i$, and $N_{pop}$ the population size.

**Crossover**: After the selection process, the population is enriched with better individuals. Crossover operator is applied to the mating pool with the hope that it creates a better offspring. For the real-coded GA, crossover operator includes simple crossover, arithmetic crossover, and heuristic crossover. In this paper, the arithmetic crossover is used, assuming two parent chromosomes as:

$$X^i = (x^i_1, \ldots, x^i_{a_i}, \ldots, x^i_{b_i}) \quad (23)$$

$$X^j = (x^j_1, \ldots, x^j_{a_j}, \ldots, x^j_{b_j}) \quad (24)$$

And two offspring chromosomes are built:

$$Y^1 = rX^i + (1-r)X^j \quad (25)$$

$$= \left(r x^i_1 + (1-r)x^j_1, \ldots, r x^i_{a_i} + (1-r)x^j_{a_i}, \ldots, r x^i_{b_i} + (1-r)x^j_{b_i}\right)$$

$$Y^2 = rX^j + (1-r)X^i \quad (26)$$

Where, $r$ is a random number from interval, which remains constant for uniform arithmetic crossover or varies with regard to the number of generations for non-uniform arithmetic crossover.$^{[20,21]}$

**Mutation**: Mutation prevents the algorithm to be trapped in a local minimum, this is quite important for die casting process optimization. For real-coded GA, the mutation operators include uniform mutation, non-uniform mutation, multi-non-uniform mutation, boundary mutation, etc.$^{[20,22]}$

Assume chromosome and $X = (x_1, \ldots, x_i, \ldots, x_n)$ gene $x_i \in [a_i, b_i]$ to be mutated, after the non-uniform mutation, new
gene will be:
\[
\hat{x}_i = \begin{cases} 
    x_i + (b_i - a_i) \varphi(G) & \text{if } r_0 \leq p_m, r_1 \leq 0.5 \\
    x_i + (x_i - a_i) \varphi(G) & \text{if } r_0 \leq p_m, r_1 > 0.5 \\
    x_i & \text{if } r_0 > p_m
\end{cases}
\]

Where \( \varphi(G) = \left( r_2 \left( \frac{G}{G_{\text{max}}} \right) \right) \); \( p_m \) is the mutation probability; \( a_i \) and \( b_i \) are the lower and upper bounds of each variable; \( G \), \( G_{\text{max}} \) are the number of current generation and the maximum number of generations. And \( r_0 \), \( r_1 \) and \( r_2 \) are three random numbers between \((0 \text{ and } 1)\); \( b \) is a user defined parameter which indicates the dependency on the number of iterations.

With the help of MATLAB, the GP/GA based model has settled the process optimization problem mentioned above. From Fig.7 we can see how the objective value evolves with respect to generations. Details of GA setups and optimum solution are tabulated bellow. The optimum solution, as given in the Table 5, can be tested by computer simulation or experiment. In this case, the location of the filling port is used as an index to estimate the accuracy of result. Due to the geometry symmetry of the casting shell, to get an even temperature distribution, the filling port is expected to be fixed at the central point A, as shown in Fig.4, which is 175 mm from the left edge of the mold. The value of \( L_{\text{fill}} \) in optimum solution is 173.40, with an acceptable relative error of 0.91%.

<table>
<thead>
<tr>
<th>GA setup</th>
<th>Fitness function ( f )</th>
<th>Population No. ( N_{\text{pop}} )</th>
<th>Mutation probability ( p_m )</th>
<th>Max iteration No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( T_{\text{end}} ) °C</td>
<td>( T_{\text{melt}} ) °C</td>
<td>( \nu_{\text{fi}} ) m/s</td>
<td>( \Phi_{\text{fi}} ) mm</td>
</tr>
<tr>
<td>Optimum solution</td>
<td>685.70</td>
<td>242.43</td>
<td>3.016</td>
<td>4.78</td>
</tr>
</tbody>
</table>

3 Summary and future work

Optimization methodologies of high pressure die casting process optimization based on numerical simulation are promising and practical research. The methodology using surrogate model building technique, such as artificial neural network (ANN) or support vector regression (SVR) promises future research. In this paper, the methodology using Gaussian process regression (GP) technique is presented with GA, and used to optimize the casting process of a magnesium part with optimum solution. The design of experiment (DOE) for simulation is employed to select the most characteristic process conditions to enrich the information for training the GP model, while to minimize the time for sample acquisition. From quantitative and qualitative analyses of the model validation, it shows that GP regression technique is an efficient nonlinear modeling method. It predicts the most likely value as well as the variance for the desired response. This is quite useful when the GP model needs improving, as it gives information for where additional training points could be added. Since the model meets the acceptance criterion, a real-coded GA is well suit to this kind of nonlinear optimization. In spite of achieved success mentioned above, further improvements for the model over high pressure die casting process optimization are recommended bellow:

1. Gaussian process approach is a continuous dynamic process in which the ever-built regression model should be adaptively updated and improved with new data. As mentioned above, the prediction of variance can be used as guidance to where additional training points could be added to improve the preliminary model.

2. In this paper, only single-objective was discussed while selecting the process condition is usually treated as multi-objective optimization problem. The methodology discussed above should be further employed to multi-objective optimization problem.

3. Furthermore, design of experiment (DOE), Gaussian process (GP), genetic algorithm (GA) should be integrated with computer simulation to develop system level optimization system for high pressure die casting process.

References


