Gaussian Process Modelling of Austenite Formation in Steel

C.A.L. Bailer-Jones[1,2,3], H.K.D.H. Bhadeshia[2], D.J.C. MacKay[1]

[1] Cavendish Laboratory, Department of Physics, University of Cambridge,
Madingley Road, Cambridge, CB3 0HE
[2] Department of Materials Science and Metallurgy, University of Cambridge,
Pembroke Street, Cambridge, CB2 3QZ
[3] Present address: Max-Planck-Institut für Astronomie Königstuhl 17,
D-69117 Heidelberg, Germany. email: calj@mpia-hd.mpg.de

Submitted to *Materials Science and Technology*, 26 May 1998

Abstract

We introduce the Gaussian process model for the empirical modelling of the formation of austenite during the continuous heating of steels. A previous paper has examined the application of neural networks to this problem, but the Gaussian process model is a more general probabilistic model which avoids some of the arbitrariness of neural networks, and is somewhat more amenable to interpretation. We demonstrate that the model leads to an improvement in the significance of the trends of the Ac_1 and Ac_3 temperatures as a function of the chemical composition and heating rate. In some cases, these predicted trends are more plausible than those obtained with the neural network analysis. Additionally, we show that many of the trace alloying elements present in steels are irrelevant in determining the austenite formation temperatures.

1 Introduction

The formation of austenite is an important component in the heat treatment of steels. The temperature at which austenite begins to form during the continuous heating of steel is called the Ac_1 temperature. That at which the steel becomes fully austenitic is the Ac_3 temperature. The corresponding equilibrium temperatures (i.e. those for an infinitesimally small heating rate) are Ae_1 and Ae_3 respectively, with $Ac_1 \geq Ae_1$ and $Ac_3 \geq Ae_3$. In previous work [1] (hereafter referred to as GBMS) a neural network was used to model the variation in these transformation start and finish temperatures as a function of the steel chemical composition and the heating rate (the "inputs"). The analysis was conducted on a large dataset compiled from the published literature, taking into account a total of 21 different alloying elements.

Generally speaking, the trained neural network was demonstrated to be largely consistent with phase transformation theory, and its predicted trends (i.e. the variation of Ac_1 and Ac_3 with the inputs) agreed with established metallurgical understanding. However, in some cases the trends were found to be uncertain (large error bars) and hence difficult to interpret, and in a few cases the trends differed considerably from what was expected. Subsequent use of the model has also revealed that the analysis may have been over ambitious in the number of composition terms included as input variables. It turns out that many of the trace elements showed little or no significant variation within the dataset used to develop the network. Thus the model was unable to learn the dependence of Ac_1 or Ac_3 on these trace elements, and as a consequence the model tended, not surprisingly, to give uncertain predictions when the the trace element concentrations were varied.

The present work has two purposes. The first is to repeat the analysis using a reduced set of variables by eliminating those variables which (a) show very little variation, (b) we believe not to significantly influence the austenite formation process at the concentrations involved, or (c) are likely to have been inaccurately determined.

The second purpose is to introduce a more general method of data modelling, the Gaussian process model [2] [3]. The neural network method in the original work involved the creation of a large set of models, each with a different level of complexity. Those models which were too simple — and hence unable to capture the trends in the data — were rejected, as were over complex models which did not generalize well. Thus considerable effort was expended in determining the required complexity of the function (network) describing the relationship between the input and output variables. The Gaussian process model, on the other hand, avoids the explicit parameterisation of the input–output function. One of the advantages of this is that it does not require us to make the often ad hoc decision regarding the complexity of our model (i.e. the number of hidden nodes and layers in the network). It has been shown that the Gaussian process is a generalisation of many standard interpolation methods, including neural networks and splines [4]. In metallurgy, Gaussian Processes have been applied to the problem of modelling recrystallisation in Aluminium alloys [5] [6].

In the present work, a Gaussian process model is trained on the reduced dataset, and the results compared with those from the neural network in GBMS. In order to ascertain whether the differences are due to the use of a new type of model, or the reduced dataset, another Gaussian process model is trained on the full dataset (i.e. all 22 input variables) used in GBMS.

2 The Data

The original dataset consisted of 22 input variables, and two output variables, namely the Ac_1 and Ac_3 temperatures which describe the onset and completion (respectively) of austenite formation during continuous heating from ambient temperature. 788 cases (input–output pairs) were used in the analysis.

In addition to the heating rate, the input variables consisted of the elements C, Si, Mn, Cu, Ni, Cr, Mo, Nb, V, W and Co, together with the trace elements S, P, Ti, Al, B, As, Sn, Zr, N and O. However, a close examination of the dataset indicated that the variation in concentration for the trace elements was rather small (zero in the case of As, Sn, Zr and O) and possibly insignificant compared with the estimated precision of the chemical analysis. Furthermore, in the majority of cases the trace elements are in such small concentrations that they are not expected to greatly influence transformation behaviour. They were therefore eliminated from the list of inputs. The reduced dataset is presented in Table 1.

3 Data Modelling

The problem we address is one of obtaining a model of the dependence of an ‘output’ variable, such as the Ac_1 temperature, on several input variables, such as the mass fractions of the different alloying elements. A general approach to this type of problem is to formulate a physically motivated, parameterized model which describes the relationship between the inputs and the output. We can then infer the parameters of the model using a set of measured inputs and outputs (the *training data*), i.e. we interpolate these data. This can be done with standard regression methods such as least squares minimisation. Once the parameters have been determined we can make predictions of the output for any values of the inputs. However, such an approach is limited to those simple problems in which a physical model can be devised which still models the data with sufficient accuracy. To obtain accurate predictions in more complex situations it is often necessary to take an empirical approach, in which the input–output relationship is determined from the data without reference to a simplified physical model.

One such approach is neural network modelling. Neural networks are a flexible approach to data modelling as they can provide an arbitrarily complex, non-linear mapping between one or more inputs and an output. This mapping is parameterized by a set of 'weights', the optimum values of which are determined by training the network.

4 The Gaussian process model

In modelling complex problems empirically, we do not know what the parameterized form of the input–output relationship should be. The Gaussian process model is a way of avoiding having to explicity parameterize this relationship by instead parameterizing a *probability* model over the data [2] [3].

Let the training dataset consist of N input vectors, $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, and the corresponding set of known outputs (or "targets") $\{t_1, t_2, \ldots, t_N\}$. A prediction, t_{N+1}, can then be made at any new input value, $\mathbf{x}_{N+1}$, based on these training data. For brevity, let $\mathbf{X}_N$ represent the set of input vectors and $\mathbf{t}_N$ be the vector of corresponding outputs.

The approach of the Gaussian process model is as follows. Let $P(\mathbf{t}_N|\mathbf{x}_N)$ be the joint probability distribution over the N output values in the training dataset. This is a probability distribution in an N-dimensional space. Similarly, the joint probability distribution of both the N training data points and the single new point is $P(t_{N+1}, \mathbf{t}_N|\mathbf{x}_{N+1}, \mathbf{X}_N)$. In making predictions based on the training data, we want to find $P(t_{N+1}|\mathbf{x}_{N+1}, D)$, that is, the probability distribution over the predicted point given that we know the corresponding input, $\mathbf{x}_{N+1}$, and all of the training data, $D = \{\mathbf{t}_N, \mathbf{X}_N\}$. The relationship between these quantities comes from the simple rule of probability, $P(A, B) = P(A|B)P(B)$, which in this case translates to

$$P(t_{N+1}|\mathbf{x}_{N+1}, D) = \frac{P(t_{N+1}, \mathbf{t}_N|\mathbf{x}_{N+1}, \mathbf{X}_N)}{P(\mathbf{t}_N|\mathbf{X}_N)} \quad . \tag{1}$$

In order to evaluate this, we need to choose the form of these probability distributions. The Gaussian process model specifies that the joint prior probability distribution of any N output values is a multivariate Gaussian,

$$P(\mathbf{t}_N|\mathbf{X}_N, \mathbf{\Theta}) \propto \exp\left(-\frac{1}{2}(\mathbf{t}_N - \boldsymbol{\mu})^T \mathbf{C}_N^{-1}(\mathbf{t}_N - \boldsymbol{\mu})\right) \tag{2}$$

where $\boldsymbol{\mu}$ is the mean, and $\mathbf{C}_N$ is a covariance matrix which is a function of $\mathbf{X}_N$ and $\mathbf{\Theta}$, the latter being a set of parameters which will be discussed later. A similar equation holds for $\mathbf{t}_{N+1}$, where $\mathbf{t}_{N+1} = (\mathbf{t}_N, t_{N+1})$. Thus we see that the numerator and denominator in equation 1 are multivariate Gaussians of dimension $N+1$ and N respectively, in which case it can be shown that $P(t_{N+1}|\mathbf{x}_{N+1}, D)$

is a univariate Gaussian,

$$P(t_{N+1}|\mathbf{x}_{N+1}, D) = \frac{1}{\sqrt{2\pi}\sigma_{\hat{t}}} \exp\left[-\frac{(t_{N+1} - \hat{t})^2}{2\sigma_{\hat{t}}^2}\right] \tag{3}$$

with mean $\hat{t}$ and standard deviation $\sigma_{\hat{t}}$. This is the desired probability distribution over the output value for the given input, $\mathbf{x}_{N+1}$. (Note that we are not saying that the input–output *function* is a Gaussian!) As we have an entire probability distribution, we can not only make a prediction but also assign confidence intervals to this prediction. Typically we would report a prediction as $\hat{t} \pm \sigma_{\hat{t}}$, where $\sigma_{\hat{t}}$ is the 1σ error determined by the model.

The values of $\hat{t}$ and $\sigma_{\hat{t}}$ depend on the covariance matrix, $\mathbf{C}_N$, of the Gaussian process model in equation 2 (see Appendix). The elements of this matrix, C_{ij}, are given by the covariance function, C. The form of the covariance function is central to the Gaussian process model, and embodies our assumptions about the nature of the underlying input–output function we are trying to model. It is through the covariance function that the predictions depend upon the inputs in the training data.

The covariance function we use is

$$C = \theta_1 \exp\left[-\frac{1}{2}\sum_{l=1}^{l=L}\frac{(x_i^{(l)} - x_j^{(l)})^2}{r_l^2}\right] + \theta_2 + \sigma_n^2\delta_{ij} \ . \tag{4}$$

This gives the covariance between any two output values, t_i and t_j, with corresponding L-dimensional input vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. The first term in C specifies our belief that the underlying function we are modelling is smoothly varying: r_l is the characteristic length scale over which the function varies in the l^{th} input dimension. Examining the form of this term, we see that when two inputs are "close" (on the scale of their length scales) the exponent is small, so this term makes a large contribution to the covariance. In other words, if two input vectors are close, their corresponding outputs are highly correlated, making it probable that they have similar values. This form of C places relatively few constraints on the function, and permits us to model non-linear functions.

The second term in equation 4 simply allows functions to have a constant offset, i.e. have a mean different from zero. (This could also be done by setting the mean in equation 2 to $\boldsymbol{\mu} = (1, 1, \ldots, 1)c$, where c is a hyperparameter which must be inferred. We instead choose to set $\boldsymbol{\mu} = \mathbf{0}$ and have all of the hyperparameters in the covariance function.) The final term is the noise model: δ_{ij} is the delta function, so this term only gives a contribution to the covariance when $i = j$. In this case we have a constant (input independent) noise model; the variance of the noise is σ_n^2. Note that this noise model is only for the outputs: we assume that the inputs to be noise free.

The set of parameters r_l ($l = 1 \ldots L$), θ_1, θ_2 and σ_n in equation 4 are called *hyperparameters* because they explicity parameterize a probability distribution over the input–output function rather than the function itself. They will be denoted $\mathbf{\Theta}$ for brevity, as in equation 2. Along with $\mathbf{x}_{N+1}$ and the training data, the hyperparameters completely specify the elements of the covariance matrix and hence the values of $\hat{t}$ and $\sigma_{\hat{t}}$ (see Appendix).

If enough is known about the problem, these hyperparameters can be set by hand. More commonly, however, we must infer their optimum values from the training data. This is done by maximizing $P(\mathbf{\Theta}|D)$, the probability of the hyperparameters given the training data, with respect to $\mathbf{\Theta}$. This is related to

$P(\mathbf{t}_N|\mathbf{X}_N, \mathbf{\Theta})$ in equation 2 via Bayes' Theorem,

$$P(\mathbf{\Theta}|D) = \frac{P(\mathbf{t}_N|\mathbf{X}_N, \mathbf{\Theta})P(\mathbf{\Theta})}{P(\mathbf{t}_N|\mathbf{X}_N)} \ . \tag{5}$$

The optimisation must usually be done numerically. The second term in the numerator of equation 5 is the *prior* probability distribution over the hyperparameters. The hyperparameter priors are an important method for introducing any prior knowledge we may have of the values of the length scales, or the magnitude of the noise variance. This approach of maximizing the evidence for the hyperparameters is a Bayesian one which automatically embodies complexity control, even if the prior on the hyperparameters is uniform (uninformative) [7].

5 Results: model performance

In developing a Gaussian process model, the dataset was randomly divided into two halves, each consisting of 394 input–output pairs. The first half (training data) was used to train the model, and the second half (test data) was then used to test the ability of the model to generalize its predictions. To aid interpretation of the model, each input and output variable was linearly scaled into the range -0.5 to $+0.5$, using the range values in Table 1.

Two separate Gaussian process models were developed in this fashion, one to model each of Ac_1 and Ac_3 as a function of the reduced input dataset (Table 1). The performance of these models on the training and the test data is shown in Fig. 1, by plotting the predicted value, $\hat{t}$, against the target value, T. This plot clearly demonstrates that both models have generalized well, and therefore have captured the underlying relationships in the training data. These plots also show the modelling uncertainties, $\sigma_{\hat{t}}$, determined by the model (equation 3). Given that the size of these uncertainties is commensurate with the scatter of the predictions about the target values, we can be confident that the model is making good predictions of its own error. This can be better seen in Fig. 2, which shows histograms of the z values. The z value is the number of standard deviations from which the predicted value differs from the target value, i.e. $z = (\hat{t} - T)/\sigma_{\hat{t}}$. This figure shows that the model is predicting reasonable uncertainties, with 94% (68%) of both Ac_1 and Ac_3 predicted temperatures lying with $2\sigma_{\hat{t}}$ ($1\sigma_{\hat{t}}$) of the target value.

Fig. 3 shows the inverse of the length scale hyperparameters for the two Gaussian process models. As can be seen from equation 4, the inverse square of the length scale, r_l, gives the scale of the l^{th} input over which the output varies by a significant amount. Thus r_l^{-1} is a measure of the "relevance" of the l^{th} input in determining the output: if r_l is small, the output varies quite a lot as x_l does, so its relevance is large. Note that the relevance is not quite the same as "sensitivity" of the output, t, to an input, x_l. (This latter quantity would be $\partial t/\partial x_l$, which depends on the value of x_l.) Rather the relevance is an overall measure of the significance of that input variable in determining the output. The relevance parameters can be compared directly with the σ_w values in GBMS, and we see similar overall results.

According to Fig. 3, the models predict that carbon explains less of the variation in Ac_1 than in Ac_3. This can be understood physically because the carbon is, in the starting microstructure, present as carbides, with very little of it in solution. The average carbon concentration is therefore of secondary importance for the start of austenite formation.

It is important to note that the length scales reported in Fig. 3 are in units of the *scaled* input variable. As the inputs were scaled to lie in the range -0.5 to $+0.5$, the size of the length scale depends on the range of the inputs. Thus the reported length scale of 1.68 for carbon in the Ac_1 problem corresponds to a length scale in units of concentration of 1.61 wt-%, whereas the almost identical reported length scale of 1.69 for silicon is a concentration length scale of 3.60 wt-%, on account of the larger range of concentrations of silicon in the dataset (Table 1). It is nonetheless useful to report length scales in terms of the scaled variable, as this takes into account the differences in the typical concentration ranges of the different alloying elements.

The standard deviation of the noise in the data, σ_n, is one of the hyperparameters learned by the model during training (see equation 4). This was found to be $\sigma_n = 14.0$ °C and $\sigma_n = 17.5$ °C for the Ac_1 and Ac_3 models repsectively. The noise, along with some additional "fitting uncertainty", comprise the modelling uncertainty, $\sigma_{\hat{t}}$, at a given point (the error bars in Fig. 1). The average value of $\sigma_{\hat{t}}$ for the test data is 17.3 °C and 19.7 °C for Ac_1 and Ac_3 respectively. Assuming that the fitting uncertainty and noise add in quadrature, we see that the noise typically contributes about 85% of the total modelling uncertainty. In other words, the intrinsic noise in the measurements of Ac_1 and Ac_3 is probably the dominant source of error.

6 Results: model predictions

Figs. 4, 5 and 6 show the models' predictions of the effects of varying the concentration of the different alloying elements on the Ac_1 and Ac_3 temperatures. Other than the predicted trends shown in Fig. 4 for carbon and the heating rate, all predictions are for a heating rate of $1\,\mathrm{K\,s^{-1}}$ and for Fe–0.2C (wt-%) steels, i.e. the fraction of all alloying elements other than the one being varied is zero. The predicted effect of the carbon concentration (Fig. 4a) is for a plain carbon steel (i.e. binary Fe–C alloys), again at a heating rate of $1\,\mathrm{K\,s^{-1}}$. The heating rate prediction is (Fig. 4b) for a binary Fe–0.2C (wt-%) steel.

The peak in the transformation temperature (Fig. 4a) was not initially expected, although it can be explained if retained austenite is present in the microstructure. This prediction is consistent with that given by the neural network in GBMS, and the reader is referred to that paper for a discussion. The predictions for carbon (Fig. 4b) are in broad agreement with those obtained in GBMS.

Nickel is an austenite stabilizer, and judging from the phase diagram, both the Ac_1 and Ac_3 temperatures should decrease with increasing nickel concentration. These trends are predicted by the Gaussian process models as illustrated in Fig. 4c. This result is a large improvement over the results of the previous neural network analysis, which not only gave an incorrect trend for the Ac_1 temperature as a function of the nickel concentration, but also indicated very large uncertainties in the calculations.

The predictions for chromium (Fig. 4d) are more interesting and again significantly differ from GBMS. The Ac_3 temperature goes through a minimum at about 5 wt-% of Cr, which is consistent with a minimum found in the equilibrium Ae_3 temperature at about 7 wt-%. On the other hand, the trend of Ac_1 as a function of Cr is opposite to that expected for Ae_1; the reason for this is not understood.

It would be useful to know whether these discrepancies between the Gaussian process model and the neural network model are on account of having used a different model or the reduced dataset. To determine this, we developed Gaussian process models on the full dataset used in GBMS, and used these to make predictions for the same alloys. For brevity, we shall refer to the Gaussian process model trained on the reduced dataset as the 12d model (for 12 input dimensions) and that trained on the full dataset as the 22d model. We find that both models give essentially identical trends with HR, C and Ni for both $Ac_1$and Ac_3, indicating that the improved predictions in Fig. 4 are on account of having used the Gaussian process model rather than due to the removal of the trace elements.

Fig. 5a shows that copper, over the concentration range considered, has little influence on the Ac_3 temperature but depresses the Ac_1 temperature as the concentration approaches the upper limit of about 1 wt-%. The latter effect is expected from the phase diagram, since copper increases the stability of austenite.

The Gaussian process predictions for manganese are similar to those obtained by the neural network, but the larger uncertainties predicted by the Gaussian process seem more reasonable. This is particularly the case when we consider that the model should never predict uncertainties smaller than the inferred noise level. Fig. 5b shows some 'high frequency' detail in the variation of the Ac_1 predictions with manganese, which is not expected and was not present in the neural network predictions. This is reflected by the small length scale (0.09) for the Gaussian process model. In comparison, the 22d Gaussian process model predicts less variation, and has a correspondingly larger length scale (0.22). The discrepancy may be on account of the 12d model slightly overfitting the data in this region of the input space.

The Ac_1 and Ac_3 temperatures show a similar insensitivty to low concentrations of molybdenum as in the previous work (Fig. 5c). As noted in GBMS, the opposite trends of Ac_1 and Ac_3 at higher concentrations are consistent with phase diagram calcuations. The 22d Gaussian process models gives essentially identical predictions.

It is worth mentioning at this point that the Gaussian process model, like the neural network, is an interpolation model. Thus when extrapolating beyond the ranges of the input values in the training dataset the predictions are less well determined by the data, so we would expect poorer predictions. Correspondingly we would expect the model to predict larger uncertainties, and this can be seen in Figs. 4 and 5. As we move away from the range of the training data, the predictions become more and more model-dependent, in this case dependent on the form of the covariance function. Inspection of equation 4 shows that well away from the data (large x_l) the dominant covariance term is θ_2, which is constant, so the extrapolations will asymptote towards a constant value. The predictions in Figs. 4, 5 and 6 are mostly for values within the range of inputs. However, it should be pointed out that the distribution of the inputs is sometimes skewed towards low values, e.g. for molybdenum (Table 1). Thus could model-dependent extrapolation explain the "turn down" in Ac_1 for high Mo concentrations in Fig. 5c? The answer is probably no, as a similar turn down is seen in GBMS from the neural network, which has a different model prior. We are similarly confident that much of the behaviour in the other plots is "real".

An initial reaction may be that extrapolations should follow the last trend in the data, rather than tending towards a constant value. However, we cannot necessarily justify that such trends should influence predictions at very distant parts of the parameter space. The choice of the model prior is thus a somewhat philosophical one which we will not discuss here. We simply warn the reader that the extrapolations from any model will be model-dependent. A different Gaussian process covariance function could be introduced to provide different extrapolation behaviours [8].

The Gaussian process model gives a predicted trend for the Ac_1 temperature in silicon (Fig. 5d) which differs from the neural network predictions. However, given the very large error bars in the latter case, the neural network essentially failed to learn any significant trend. The 22d Gaussian process models give similar predictions and error bars to the 12d Gaussian process models, showing that the reduced uncertainty is a result of using the Gaussian process model.

The $\gamma + \alpha$ phase field in Fe–Co alloys is extremely narrow and the phase boundaries are virtually horizontal on the plot of temperature versus concentration. This is accurately reflected in the predictions for cobalt (Fig. 6a). Vanadium is a very strong carbide–forming element, with limited solubility even in austenite. Hence, it is not surprising that both the Ac_1 and Ac_3 temperatures are insensitive to the vanadium concentration (Fig. 6b). This is similar to the predictions of GBMS, but the Gaussian process model does not predict the sharp increase in both Ac_1 and Ac_3 between concentrations of 0.1 and 1.0 wt-%. The 22d Gaussian process models show very similar behaviour to the 12d Gaussian process models.

7 Model Comparison

The performance of the models discussed above (12d Gaussian process model; 22d Gaussian process model; 22d neural network model from GBMS) can be assessed by comparing rms errors on the test data. These are shown in Table 2. The models give very similar rms errors on the Ac_1 problem. This indicates that removing the 10 trace variables does not decrease the quality of the predictions. It also shows that there is little difference in the *average* performance of the models, although the analysis above has shown that the Gaussian process models give better predictions (as well as more plausible error bars) for some Fe–0.2C (wt-%) steels.

The smaller Ac_3 error for the 12d Gaussian process model as compared with the 22d Gaussian process model is presumably on account of having removed the trace variables. This would seem to confirm our suspicion that retaining the trace variables leads to inferior performance, probably due to the inaccuracy of their measurement. It is interesting, however, that this has not affected the Ac_1 Gaussian process model. It is possible that the trace variables are more significant in determining Ac_1 than Ac_3, but the reduction for Ac_3 is small (2.6°C), so this conclusion may not be that significant.

The rms error does not tell the whole story, as it ignores the predicted error, $\sigma_{\hat{t}}$. Thus even if a prediction which greatly differs from its target value has a correspondingly large predicted uncertainty, it will nonetheless give a large contribution the rms error. This limits the value of the rms error as a measure of model performance. A measure which does take into account the predicted uncertainties is $P(t_{N+1}|\mathbf{x}_{N+1}, D)$ in equation 3, the probability of a prediction given the training data. The product of these probabilities over the K vectors in the test dataset gives the total predicted error

$$\ln L = -\prod_{k=1}^{k=K} P(t_k|\mathbf{x}_k, D) = \sum_{k=1}^{k=K} \left(\frac{(T_k - \hat{t}_k)^2}{2\sigma_k^2} + \ln \sigma_k \right) \tag{6}$$

where T_k, $\hat{t}_k$ and σ_k are the target, prediction and model predicted uncertainty respectively for the k^{th} vector; an additive constant has been dropped. $\ln L$ is a dimensionless error, with more negative values indicating a better model. The values of $\ln L$ for the Gaussian process models are listed in Table 2. While the 22d Gaussian process model gave a larger rms error than the 12d Gaussian process model on the Ac_3 problem, they have very similar $\ln L$ values. In terms of the consistency of their predictions with their reported uncertainties, therefore, the two models are equally good.

The neural network gives slightly lower average errors than the Gaussian process for the 22d problems. However, it should be noted that many neural networks with different degrees of complexity and different initial weights were tried before selecting the best one, i.e. that which gave the smallest error on the *test* dataset [1]. No such selection was done (or is necessary) with the Gaussian process, so the comparison is not entirely fair.

8 Conclusions

The Gaussian process model has been introduced for empirically modelling the relationship between a set of input variables and an output variable. This model has been applied to the problem of predicting the temperature at which austenite starts to form (Ac_1) and the temperature at which austenite formation completes (Ac_3), during the continuous heating of a steel alloy. In contrast to previous work (GBMS), a reduced dataset has been used by removing those trace elements believed to be largely irrelevant in determining Ac_1 and Ac_3 at the concentrations involved. Their irrelevance has been confirmed by our analyses, as Gaussian process models trained on the original full dataset (22 inputs) give very similar results to Gaussian process models trained on the reduced dataset (12 inputs).

The Gaussian process model has the advantage over the neural network model that we avoid having to make a decision regarding the number of hidden nodes and layers in the network. The neural network architecture must often be optimised by training a range of networks and comparing their performances on a separate validation dataset. No such validation dataset is required when training the Gaussian process, allowing all of the data to be used for training. Another advantage of the Gaussian process model is that its hyperparameters are more interpretable than the weights in a neural network. Our results have demonstrated that the Gaussian process model performs at least as well as the neural network model of GBMS [1], and in many cases produced better predictions (e.g. Ni) and smaller error bars (e.g. Si).

9 Acknowledgement

The authors would like to thank Mark Gibbs for use of his Gaussian Process software.

References

[1] L. Gavard, H.K.D.H. Bhadeshia, D.J.C. MacKay and S. Suzuki: *Mater. Sci. Technol.*, June 1996, **12**, 453–463. (GBMS)

[2] C.K.I. Williams and C.E. Rasmussen: in 'Neural Information Processing Systems 8', (ed. D.S. Touretzky, M.C. Mozer and M.E. Hasselmo); 1996, Boston, MA, MIT Press.

[3] M.N. Gibbs: 'Bayesian Gaussian processes for regression and classification' PhD thesis, University of Cambridge; 1998.

[4] R.M. Neal: 'Bayesian Learning for Neural Networks', No. 118 in Lecture Notes in Statistics; 1996, New York, Springer.

[5] C.A.L. Bailer-Jones, T.J. Sabin, D.J.C. MacKay and P.J. Withers: in 'Proceedings of the Australasia Pacific Forum on Intelligent Processing and Manufacturing of Materials', (eds. T. Chandra et al.), **2**, 913–919; 1997, Watson Ferguson & Co.

[6] C.A.L. Bailer-Jones, D.J.C. MacKay, T.J. Sabin and P.J. Withers: *Australian Journal on Intelligent Information Processing Systems*, 1998, accepted.

[7] D.J.C. MacKay: *Neural Computation*, 1992, **4**, 415–447.

[8] A. O'Hagan: *Journal of the Royal Statistical Society*, 1978, B **40**, 1–42.

[9] R. von Mises: 'Mathematical Theory of Probability and Statistics'; 1964, New York, Academic Press.

Appendix

The predictive probability distribution from the Gaussian process model (equation 2) is a univariate Gaussian with mean $\hat{t}$ and standard deviation $\sigma_{\hat{t}}$ (equation 3). These are given by (e.g. [9])

$$\hat{t} = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \ , \tag{7}$$

$$\sigma_{\hat{t}}^2 = k - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \ , \tag{8}$$

where

$$\mathbf{k} = [C(\mathbf{x}_1, \mathbf{x}_{N+1}), C(\mathbf{x}_2, \mathbf{x}_{N+1}), \ldots, C(\mathbf{x}_N, \mathbf{x}_{N+1})] \ , \tag{9}$$

$$k = C(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) \ . \tag{10}$$

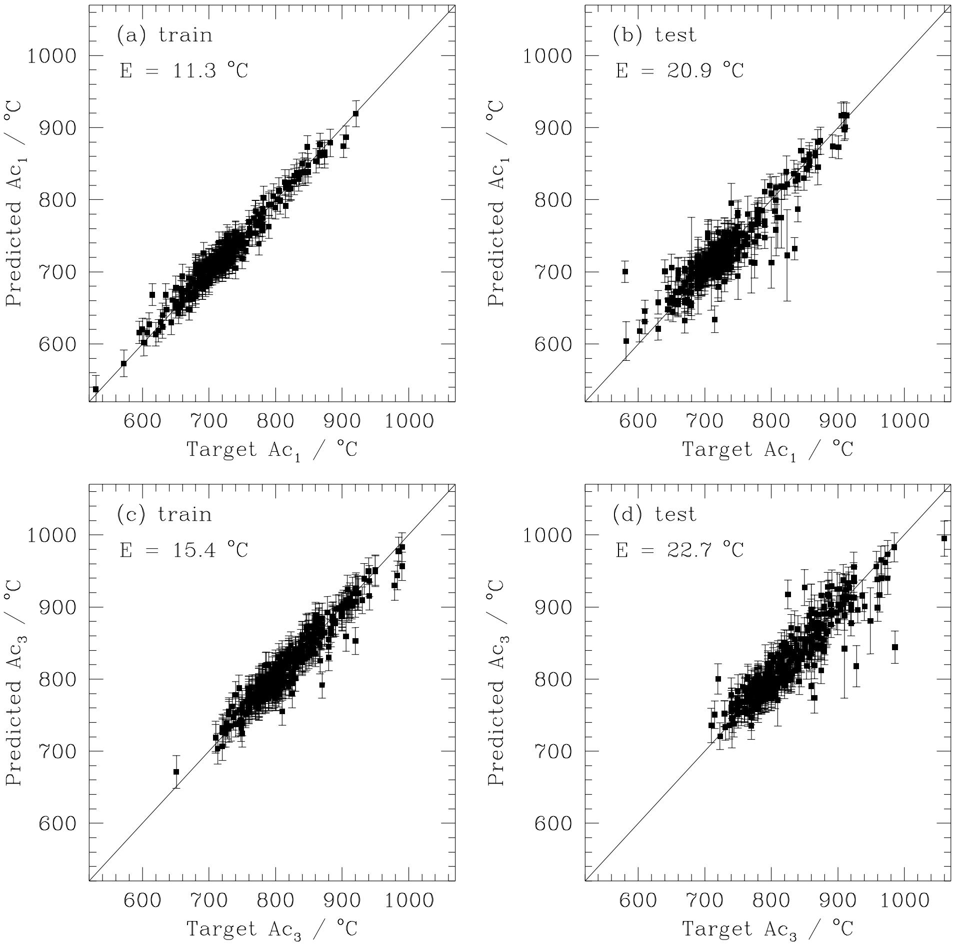
C is the covariance function (equation 4). $\mathbf{C}_N$ is the $N \times N$ covariance matrix formed from the N training data points, the elements of which are $C_{ij} = C$. $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are the N input vectors in the training dataset corresponding to the N outputs, $t_1, t_2, \ldots, t_N$. $\mathbf{x}_{N+1}$ is the input at which we wish to make a prediction of t.

Table 1: The reduced dataset. The 12 input variables consist of 11 element concentrations (in wt-%) and the heating rate. The two output variables are the Ac_1 and Ac_3 temperatures.

Variable	Range	Mean	Standard Deviation
C	0–0.96	0.30	0.17
Si	0–2.13	0.39	0.41
Mn	0–3.06	0.82	0.38
Cu	0–2.01	0.05	0.13
Ni	0–9.12	1.01	1.48
Cr	0–17.98	1.23	2.38
Mo	0–4.80	0.32	0.37
Nb	0–0.17	0.003	0.013
V	0–2.45	0.05	0.13
W	0–8.59	0.06	0.48
Co	0–4.07	0.06	0.42
Heating Rate / $\mathrm{K\,s^{-1}}$	0.03–50	1.0	11.6
Ac_1/ °C	530–921	724	52
Ac_3/ °C	651–1060	819	55

Table 2: A comparison of the rms errors, E, and log predicted error, $\ln L$, on the test data obtained by three different models. (L is defined in the text.) The neural network model is from GBMS. In all cases one of the inputs is the heating rate; the other inputs are the alloying elements.

Model	$E(Ac_1)$/ °C	$E(Ac_3)$/ °C	$\ln L(Ac_1)$	$\ln L(Ac_3)$
Neural network (22 input dimensions)	20.2	21.8	—	—
Gaussian process (22 input dimensions)	21.0	25.3	971	954
Gaussian process (12 input dimensions)	20.9	22.7	967	964



(a) Ac_1 model predictions of the training data; (b) Ac_1 model predictions of the test data; (c) Ac_3 model predictions of the training data; (d) Ac_3 model predictions of the test data.

Figure 1: Predicted vs. target outputs for the Ac_1 and Ac_3 Gaussian process models trained on the reduced dataset. E is the rms value of the predictions in each case (i.e. the scatter of points about the overplotted predicted = target line). The error bars, $\sigma_{\hat{t}}$, are the modelling uncertainties predicted by the Gaussian process model (equation 3).

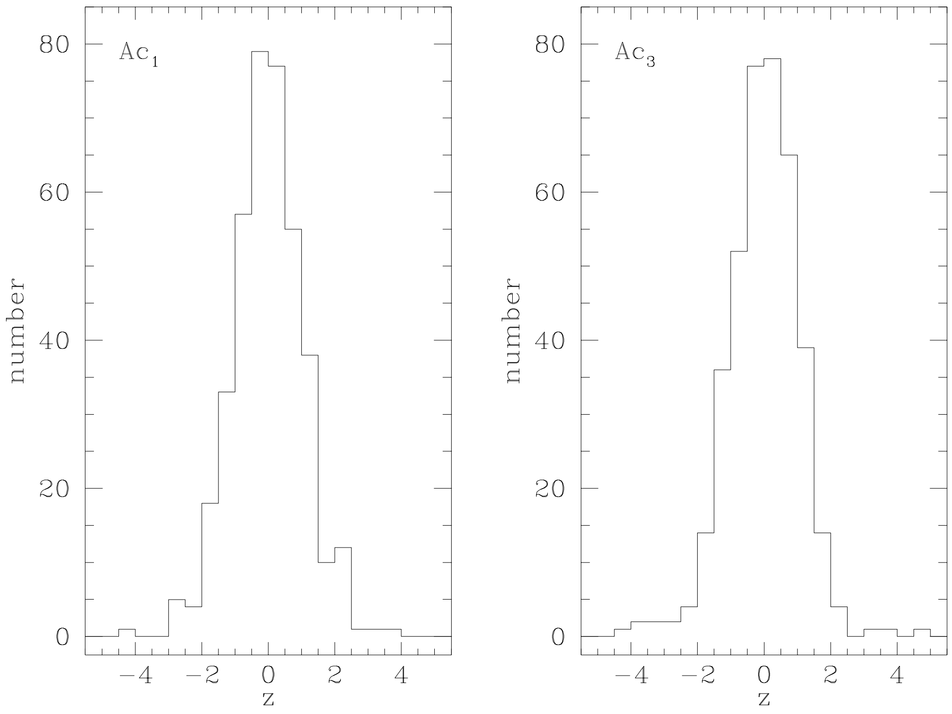


Figure 2: Histograms of $z = (\hat{t} - T)/\sigma_{\hat{t}}$ for Ac_1 and Ac_3.

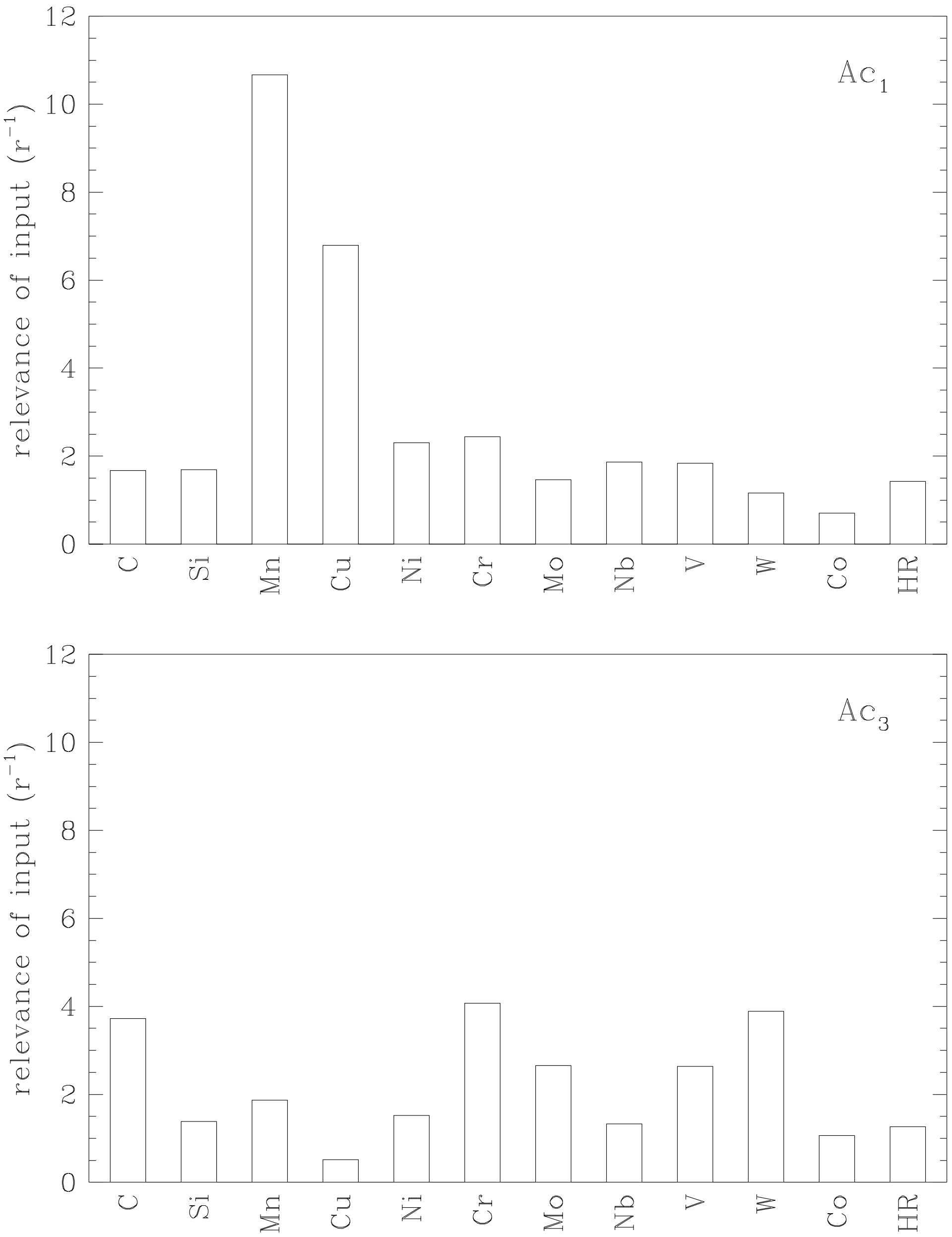


Figure 3: Model inferred relevance of the 12 inputs for the Ac_1 and Ac_3 Gaussian process models. HR is the heating rate. The relevances are the inverses of the length scales, r_l.

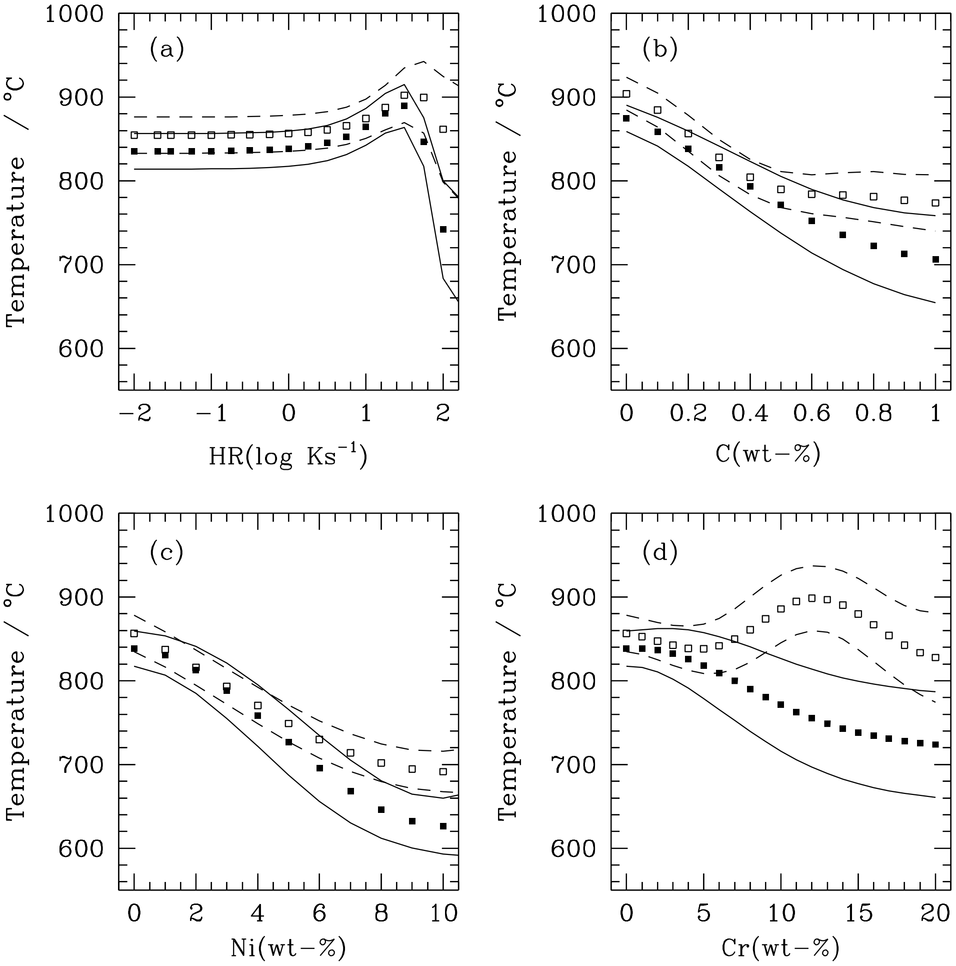


Figure 4: Gaussian process model predictions for HR, C, Ni and Cr. The filled points are the Ac_1 predictions, and the solid lines the corresponding $\pm 1\sigma$ modelling uncertainties. The open points are the Ac_3 predictions, and the dashed lines the corresponding $\pm 1\sigma$ modelling uncertainties. For the predictions against heating rate (HR), the alloy is a binary Fe–0.2C (wt-%) steel. The predictions against carbon are for plain carbon steels, and for Ni and Cr the alloys are Fe–0.2C (wt-%) steels. In these latter three cases the heating rate is $1\,\mathrm{K\,s^{-1}}$.

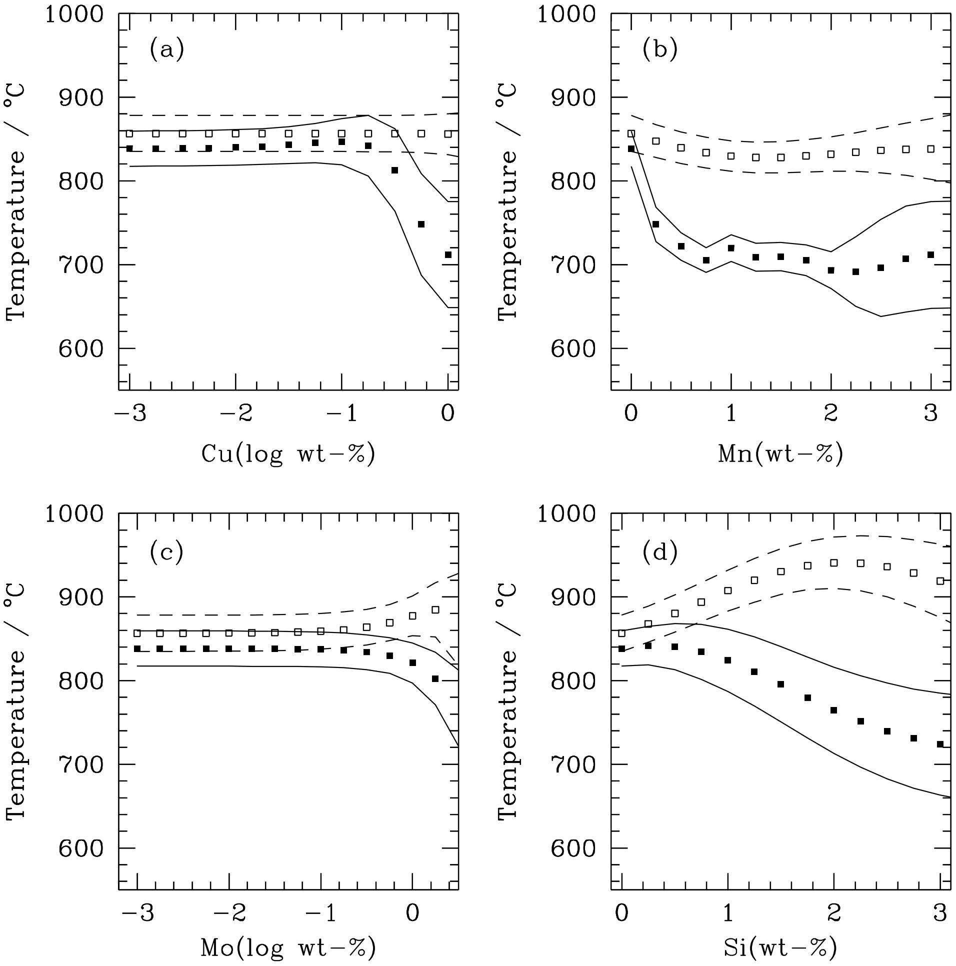


Figure 5: Gaussian process model predictions for Cu, Mn, Mo and Si. The filled points are the Ac_1 predictions, and the solid lines the corresponding $\pm 1\sigma$ modelling uncertainties. The open points are the Ac_3 predictions, and the dashed lines the corresponding $\pm 1\sigma$ modelling uncertainties. The alloys are Fe–0.2C (wt-%) steels; the heating rate is $1\,\mathrm{K\,s^{-1}}$

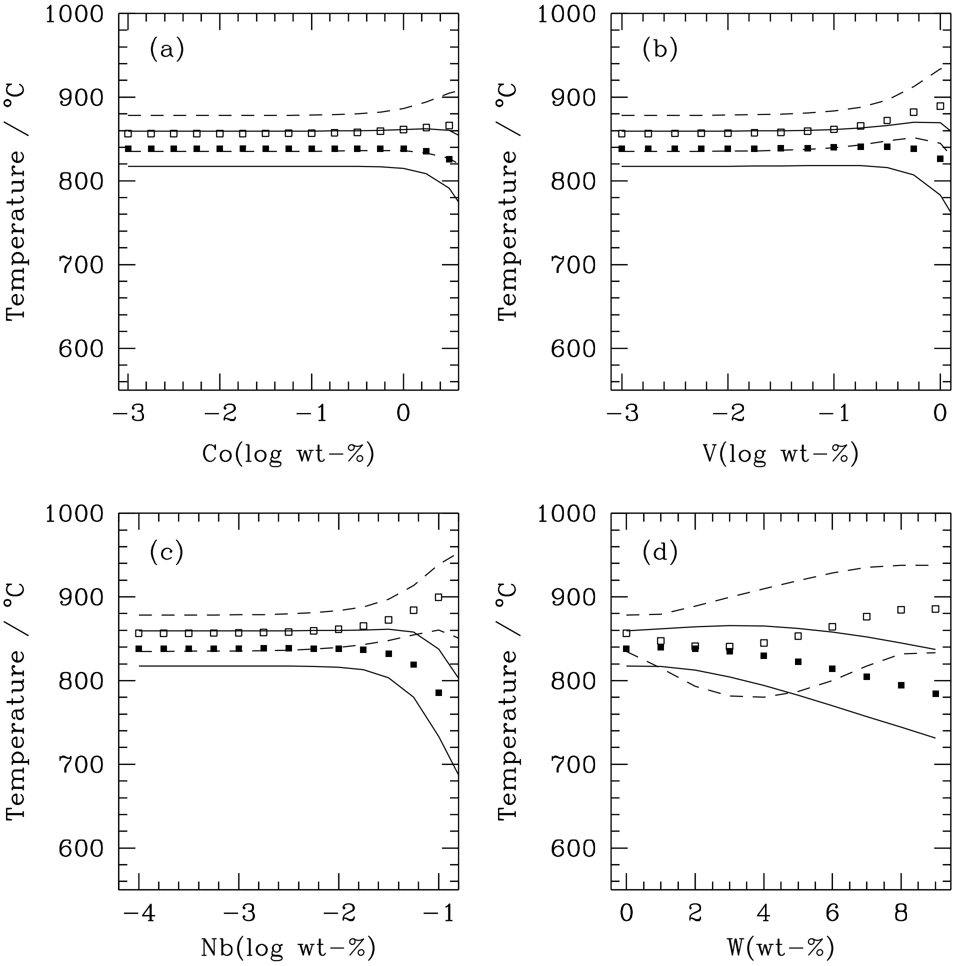


Figure 6: Gaussian process model predictions for Co, V, Nb and W. The filled points are the Ac_1 predictions, and the solid lines the corresponding $\pm 1\sigma$ modelling uncertainties. The open points are the Ac_3 predictions, and the dashed lines the corresponding $\pm 1\sigma$ modelling uncertainties. The alloys are Fe–0.2C (wt-%) steels; the heating rate is $1\,\mathrm{K\,s^{-1}}$.