$$\hat{\beta} = (\mathbf{J}^T \mathbf{R}_D^{-1} \mathbf{J})^{-1} (\mathbf{J}^T \mathbf{R}_D^{-1} y). \tag{7.3.4}$$

where J is a vector of ones and R_D is the $n \times n$ matrix of correlations $R(x_i, x_j)$.

The generalized least squares estimator, and the MLE, of σ^2 is

$$\hat{\sigma}^2 = \frac{(y - \mathbf{J}\hat{\beta})^T \mathbf{R}_D^{-1} (y - \mathbf{J}\hat{\beta})}{n}. \tag{7.3.5}$$

The best linear unbiased predictor (BLUP) at an untried x is

$$\hat{y}(x) = \hat{\beta} + \mathbf{r}^T(x) \hat{\mathbf{R}}_D^{-1} (y - \underline{J\hat{\beta}}) \tag{7.3.6}$$

where $r(x) = [R(x_1, x), \ldots, R(x_n, x)]^T$ is the vector of correlations between Z's at the design points and at the new point x. The BLUP interpolates the observed output at sites x that are in the training data.

Example 7.3 We can continue with the result from Example 7.2. We used a Latin hypercube design to define 14 settings for the piston simulator. The result of the simulation is given in Table 7.2.

The Python package `pylibkriging` implements kriging models for a larger number of predictors.

```
import pylibkriging as lk
outcome = 'seconds'
predictors = ['m', 's', 'v0', 'k', 'p0', 't', 't0']

model = lk.Kriging(mean_result[outcome], mean_result[predictors], 'gauss')
```

We can assess the goodness of fit of the kriging model using leave-one-out cross-validation. Leave-one-out cross-validation removes an observation from the dataset, builds a model with the remaining data and predicts the left out data point. This is repeated for all observations. The following Python code determines the leave-one-out predictions for each data point.

```
def looValidation(data):
    jackknife  = []
    for i, row in data.iterrows():
        subset = data.drop(i)
        model = lk.Kriging(subset[outcome], subset[predictors], 'gauss')
        y_pred = model.predict([row[predictors]], True, False, False)
        jackknife.append({
            'actual': row[outcome],
            'predicted': y_pred[0],
        })
    return pd.DataFrame(jackknife)
validation = looValidation(mean_result)
```

Using the calculated predicted and actual values, we get the following leave-one-out performance metrics.

```
from sklearn import metrics
MAE = metrics.mean_absolute_error(validation['actual'], validation['predicted'])
```
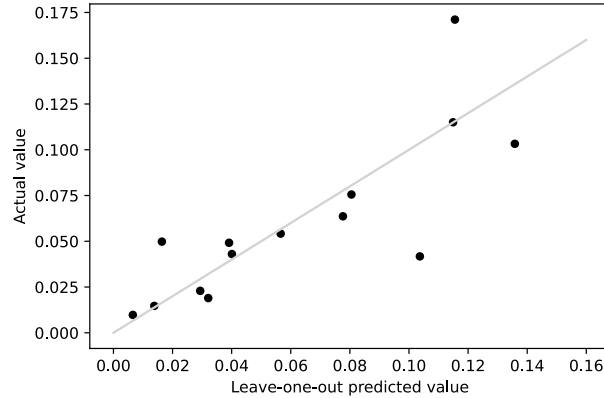
Fig. 7.3: Leave-one-out cross-validation of kriging model

```
R2 = metrics.r2_score(validation['actual'], validation['predicted'])
print(f'MAE = {MAE:.4f}')
print(f'r2 {R2:.3f}', )
```

```
MAE = 0.0173
r2 0.628
```

Both metrics show that the model performs well. In Fig. 7.3, we compare the observed value with the value predicted from the leave-one-out models. Points lying close to the line of equality indicate a good fit of the model since the observed data points are well predicted by the model.

We can derive marginal effect of factors on cycle time using the kriging model. The result of this analysis is shown in Fig. 7.4. It confirms again that only *v0* and *s* have a strong effect on the average cycle time; all other factors have only little effect on cycle time. Figure 7.5 shows the dependence of the cycle time across the full range of *v0* and *s*.   ∎

7.4 Stochastic Emulators

Traditional engineering practice augments deterministic design system predictions with factors of safety or design margins to provide some assurance of meeting requirements in the presence of uncertainty and variability in modeling assumptions, boundary conditions, manufacturing, materials, and customer usage. Modern engineering practice is implementing quality by design methods to account for probability distributions of component or system performance characteristics. Chapter 4 provided several such examples, including the robust design approach developed by Genichi Taguchi in Japan. At Pratt and Whitney, in the USA, Grant Reinman and his team developed a methodology labeled design for variation (DFV) that incorporates
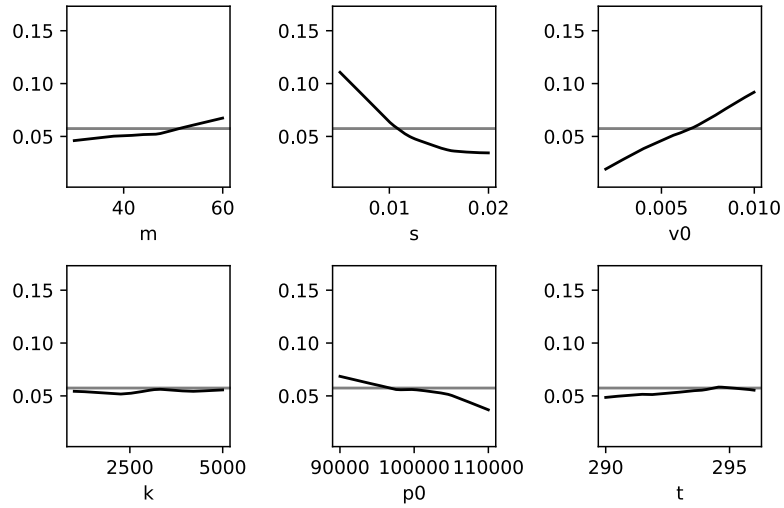
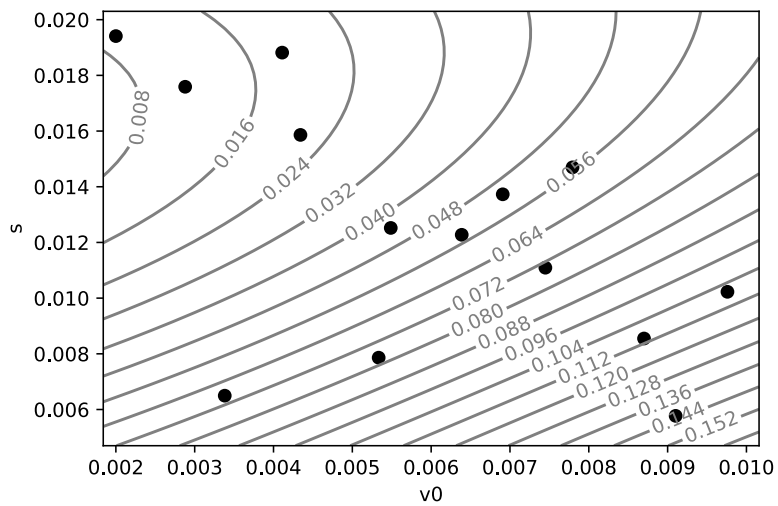Fig. 7.4: Marginal effect of factors on cycle time derived from the kriging model

Fig. 7.5: Latin hypercube design for piston simulator