MODELING, IDENTIFICATION AND CONTROL, 1995, VOL. 16, NO. 3, 145–154
doi:10.4173/mic.1995.3.3

Automatic programming of grinding robot restoration of contours

ARE WILLERSRUD†, FRED GODTLIEBSEN†
and TRYGVE THOMESSEN‡

Keywords: *Filtering, robot, grinding, automatic contour tracking*

A new programming method has been developed for grinding robots. Instead of using the conventional jog-and-teach method, the workpiece contour is automatically tracked by the robot. During the tracking, the robot position is stored in the robot control system every 8th millisecond. After filtering and reducing this contour data, a robot program is automatically generated.

The contour data contain different types of noise due to small vibrations and overshoot when the robot is tracking the workpiece. To remove the noise, a new filtering algorithm is proposed. The Look Ahead filter removes outliers from the data. After that, the data is smoothed by the Nadaraya–Watson estimator.

The system is tested out by a Multicraft 560 robot. Experimental results show that the filtering algorithm efficiently removes the noise without reducing the accuracy of the contour data. This makes it easy to automatically restore the contour by a robot program.

1. Introduction

During production of parts in sand mouldings, burrs often appear on the castings. The burrs are normally removed by manual grinding. in addition to being hard work, grinding is also unhealthy due to the vibrations and the noisy and dusty surroundings. A better solution is for industrial robots to perform the grinding.

Unfortunately, the programming of grinding robots has traditionally been time consuming because the workpiece geometry is often complicated. Consequently, the industrial robots have required large batch sizes to be profitable in grinding applications.

To reduce the programming time, Thomessen *et al.* (1993), developed a system for automatic programming of the grinding robot. First, the workpiece contour was automatically tracked by the robot using the automatic contour tracking algorithm developed by De Scutter (1986). During this tracking, the robot position was stored in the control system every 8th millisecond. Because the amount of contour data was very large, it had to be reduced. This was done by the deviation height method.

But before the data reduction, the noise had to be removed from the data. This was done using the nonlinear Gaussian filter (Godtliebsen and Spjøtvoll 1991). Unfortunately, the Gaussian filter had a tendency to smooth the contour too much along sharp curves. This reduced the accuracy of the robot program.

In this paper, a new filtering algorithm is presented. The filter removes efficiently

Received 25 August 1994.

† The Norwegian Institute of Technology, Department of Mathematical Sciences, N-7034 Trondheim, Norway.

‡ SINTEF Production Engineering, N-7034 Trondheim, Norway.

the noise from the contour data without reducing the accuracy of the contour along sharp curves.

The paper is organized as follows: Section 2 describes briefly the automatic tracking system and the data processing system. In Section 3, the different kinds of noise and the filtering problem are discussed. Section 4 describes the new filtering algorithm based on outlier removal and data smoothing. Some experimental results obtained from the method used in Thomessen *et al.* (1993) are presented in Section 5. Furthermore, we compare these results to those obtained from our new proposal. Finally, we give a discussion of our new method and draw some conclusions in Section 6.

This paper is based upon results from a project running at the Research Institute SINTEF in cooperation with the Norwegian Institute of Technology.

2. The automatic programming system

The common way of programming grinding robots is the jog-and-teach approach. The robot is moved to characteristic locations on the workpiece contour using joy-sticks. During the execution of the program, the path is generated by interpolation between these points.

The advantage of the jog-and-teach approach is that it is simple to use. On the other hand, the approach is time consuming. If the path is curved, many points have to be programmed.

A more simple and rapid way of programming the grinding robot is the automatic programming approach (Thomessen *et al.* 1993). Using joy-sticks, the robot is moved with the desired orientation to the starting point on the contour to be tracked. Then, the robot tracks the contour using the automatic contour tracking algorithm (Fig. 1).

To generate a robot program, the contour data is processed through three steps; noise reduction, point reduction and robot program generation (Fig. 2). Briefly explained, the noise reduction removes outliers and smooths the path. Then, the point reduction algorithm approximates the contour by a minimum number of straight line segments. Finally, a program on a standard format for the industrial robot is generated.

3. The filtering problem

A typical set of contour data after tracking is shown in Fig. 3. Regions 1 and 2 contain the abrupt changes of the contour that are most difficult to restore. Note that there are mainly two kinds of noise; outliers around sharp curves and a pure noise component which can be reasonably modeled as independent are identically distributed (iid) Gaussian noise.

Figure 4 illustrates that the points are not in a consecutive order. This is caused by small vibrations of the robot arm when tracking the workpiece. From Fig. 4 (*b*) it can be seen that the distortion around the corner is given as a loop which should be excluded from the curve. This observation is utilized in our proposed method described in Section 4.

This concludes that the filtering should go through two steps; first removing the outliers and then removing the pure noise component.

4. The filtering algorithm

Thomessen *et al.* (1993) processed the x-coordinates and the y-coordinates independently by using the non-linear Gaussian filter. More precisely, the underlying

Figure 1. Automatic programming using the Multicraft 560 robot.

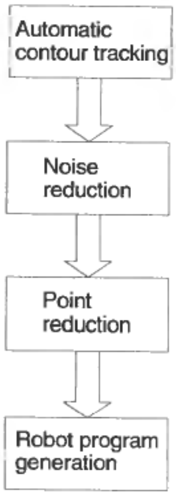


Figure 2. Description of automatic path generation.

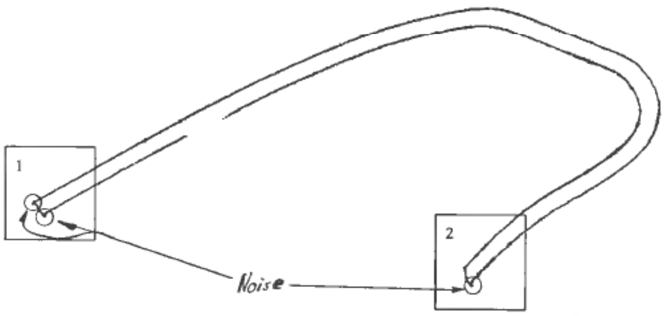


Figure 3. An observed contour with distortions.

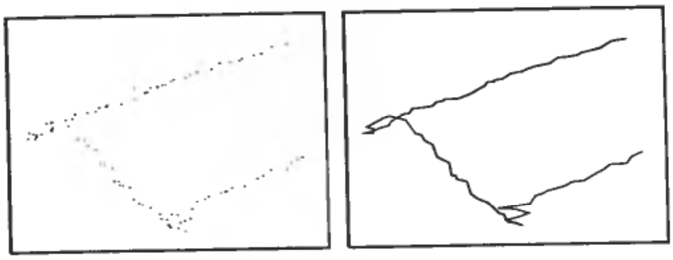


Figure 4. A detailed illustration of the distortion around the corner.

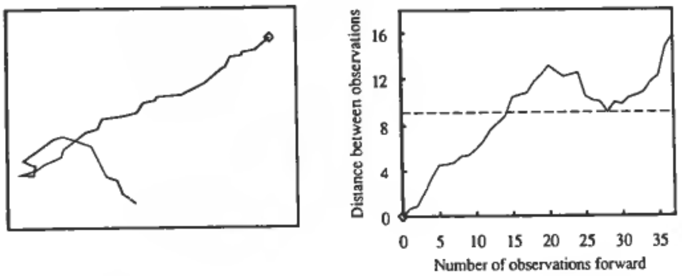


Figure 5. A section of the workpiece with a loop around the corner.

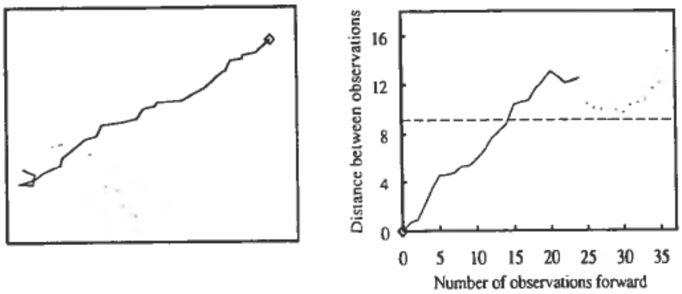


Figure 6. An example where important information is hidden.

true value for x at point i, which we shall denote u_i, was estimated by

$$\hat{\mu}_i = \frac{1}{v_i} \sum_{j \in D_i} \exp\left(-\frac{1}{2\tau^2}(x_i - x_j)^2\right) x_j \tag{1}$$

where

$$v_i = \sum_{j \in D_i} \exp\left(-\frac{1}{2\tau^2}(x_i - x_j)^2\right), \tag{2}$$

D_i means all the points within a chosen window size, and τ^2 is a smoothing parameter. The y-coordinates are filtered correspondingly. This filter is successful in situations where the underlying true values are disturbed by random Gaussian noise, see e.g. Godtliebsen (1991). In the present situation, however, the distortion also contains some outliers which reduces the accuracy of the fitted contour after using the nonlinear Gaussian filter.

Note further that since this filter is nonlinear, the parameter estimation is a difficult task. To overcome this difficulty, the Nadaraya–Watson (NW) estimator (Watson 1964) is useful. Then the true value at point i is estimated by

$$\hat{\mu}_i = \frac{1}{w_i} \sum_{j \in D_i} \exp\left(-\frac{1}{2h^2} (i-j)^2 \right) x_j \tag{3}$$

where

$$w_i = \sum_{j \in D_i} \exp\left(-\frac{1}{2h^2} (i-j)^2 \right).$$

This estimator is linear and the smoothing parameter h can be found easily e.g. by minimizing a risk function, illustrated in Mallows (1973). Unfortunately, also this estimator fails to give a reasonable estimate of the underlying true curve. As before, this is caused by the outliers around the corners.

Recall from Fig. 4 (*b*) that the observed contour typically makes a loop at corners. This information can be utilized by looking ahead a fixed number of points from the present point. Then, the distances from the selected point to each of the subsequent points are calculated. As an example the distances from one point to a fixed number of subsequent points are calculated and illustrated in Fig. 5. The idea is to use the distances to locate and remove the remaining outliers. The plot of distances will differ for different initial points at the contour of the workpiece. There are two possible forms in the plot of distances. The distance can increase for all the following points or there can be one or several local minima at the curve. An example of this is shown in Fig. 5.

The data will be treated in two different ways depending on the distances. The first alternative is that the distance increases for all the following points or a local minimum in the distances is located in the second half of the fixed interval. If this is the case the selected point will be accepted and the next point is studied. Based on this new point, the calculation of the distances starts over again. The reason for ignoring local minima in the second half of the interval, when the distances in the first interval are linearly increasing, is that the behaviour of the future points is unknown. There may be hidden information. i.e. a minimum beyond the last point in the fixed interval. Hidden minima with shorter distances than the detected minima have importance because they help us to remove outliers. An example of this is shown in Fig. 6. Here the minimum is an outlier which we want to remove. In this situation we move just one point ahead when we pick the next point. The local minima that arise in the second half of the interval will be used for initial points that appear later.

The second alternative is that there are one or several local minima in the first half of the interval. Then we consider the whole interval for using as much of the information as possible. The distance to the point where the smallest minimum arises, is now essential. All the points that follow the selected one with a shorter distance than to the minimum will be accepted. Then we skip the points with a larger distance than the one to the minimum and then take the point of the minimum as the next point. Note that this means that the points in the loop are considered to be outliers and hence removed. With basis in the minimum point the distances to the subsequent points are calculated again. We shall denote this procedure the Look Ahead filter.

One of these alternatives is used for all the points in the data set. The only thing remaining is to find the optimal number of points, m, to look ahead. We will choose different values of m in our experiments. This is commented on further in Section 6.

It turns out that the Look Ahead filter is quite successful in removing outliers appearing at corners. A final estimate of the underlying true curve can now be obtained by applying the NW estimator to the Look Ahead filtered data. For the x-coordinates we can estimate the smoothing parameter h by minimizing an estimator for the risk function

$$R(h) = E\left[\sum_{i=1}^{N} (\hat{\mu}_i - \mu_i)^2\right] = \sum_{i=1}^{N} [E(\hat{\mu}_i^2) - 2\mu_i E(\hat{\mu}_i) + \mu_i^2].$$

An unbiased estimator for $R(h)$ can be found by first noting that $\hat{\mu}_i^2$ is an unbiased estimator for $E(\hat{\mu}_i^2)$. If we suppose that σ^2 is known or can be estimated by an unbiased estimator $\hat{\sigma}^2$, it follows that an unbiased estimator for μ_i^2 is given by $x_i^2 - \hat{\sigma}^2$. Finally, an unbiased estimator for $-2\mu_i E(\hat{\mu}_i)$ is given by

$$-\frac{2}{A}\left[\sum_{j \neq i} K\left(\frac{i-j}{h}\right) x_j x_i + (x_i^2 - \hat{\sigma}^2)\right]$$

where

$$K\left(\frac{i-j}{h}\right) = \exp\left(-\frac{1}{2h^2}(i-j)^2\right)$$

$$A = \sum_{j=1}^{N} K\left(\frac{i-j}{h}\right).$$

Hence,

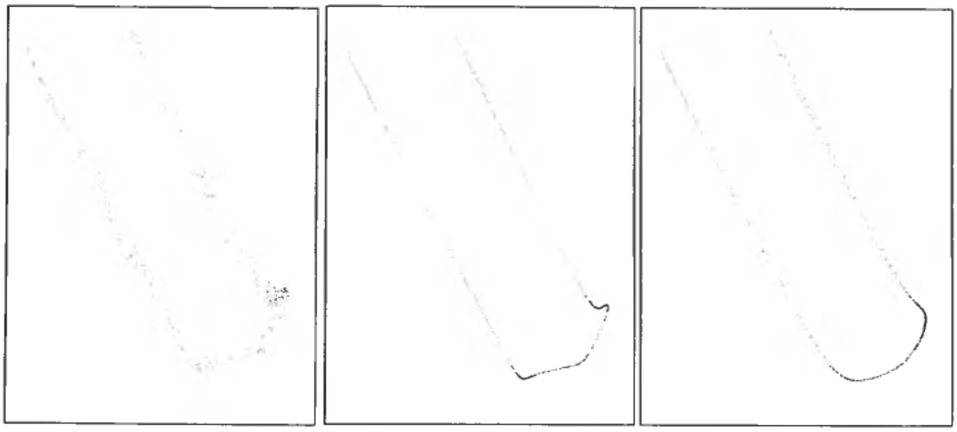
$$\hat{R}(h) = \sum_{i=1}^{N}\left\{\hat{\mu}_i^2 - \frac{2}{A}\left[\sum_{j \neq i} K\left(\frac{i-j}{h}\right) x_j x_i + (x_i^2 - \hat{\sigma}^2)\right] + (x_i^2 - \hat{\sigma}^2)\right\} \qquad (4)$$

is an unbiased estimator for $R(h)$. In eqn. 4, N denotes the number of observations. Estimation of the smoothing parameter for the y-coordinates is performed analogously.

5. Experimental results

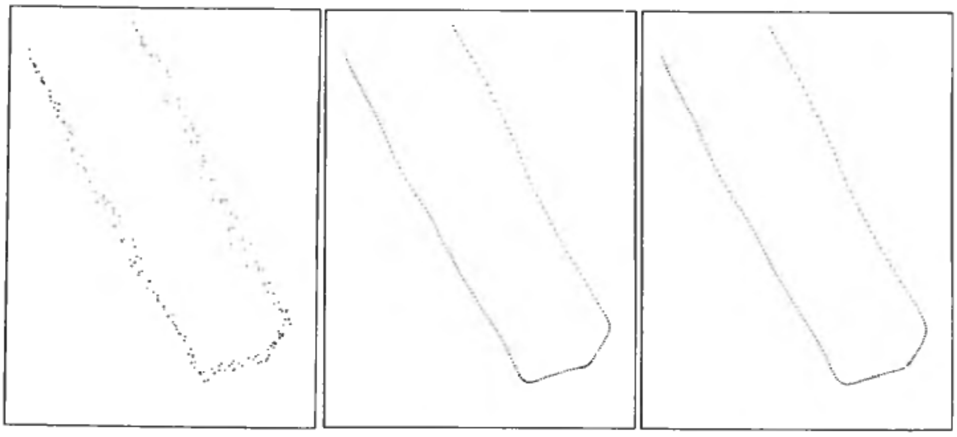
Most of the results shown here are obtained by applying the different methods to the workpiece in Fig. 3. Typical results by applying the nonlinear Gaussian filter to the data in Region 2 are given in Fig. 7. Note that if we want to remove the outliers at the corner, then the resulting estimate of the curve becomes too smooth. A small value of the smoothing parameter will on the other hand give too little smoothing. The same type of results are obtained by applying the NW estimator to the data. By applying the Look Ahead filter to the noisy contour in Fig. 3, we get a new data set from which we have plotted Region 2 in Fig. 8.

In Fig. 8, we have also plotted two smoothed curves for different values of h^2. As can be seen from these plots, the first curve is not enough smoothed while the other gives a very reasonable estimate of the underlying true curve. Figure 9 gives the result for the whole workpiece by applying the Look Ahead filter succeeded by the NW-estimator. As we can see the restoration is quite successful. An estimate of this quality would not be possible to obtain by only applying the nonlinear Gaussian filter on the NW estimator. It turns out that the parameters in the Look Ahead filter and the NW estimator is important but not crucial within reasonable limits.



(*a*) (*b*) (*c*)

Figure 7. Region 2 processed by the nonlinear Gaussian filter. (*a*) The observed data. (*b*) Processed contour with $\tau^2 = 5$. (*c*) Processed contour with $\tau^2 = 100$.



(*a*) (*b*) (*c*)

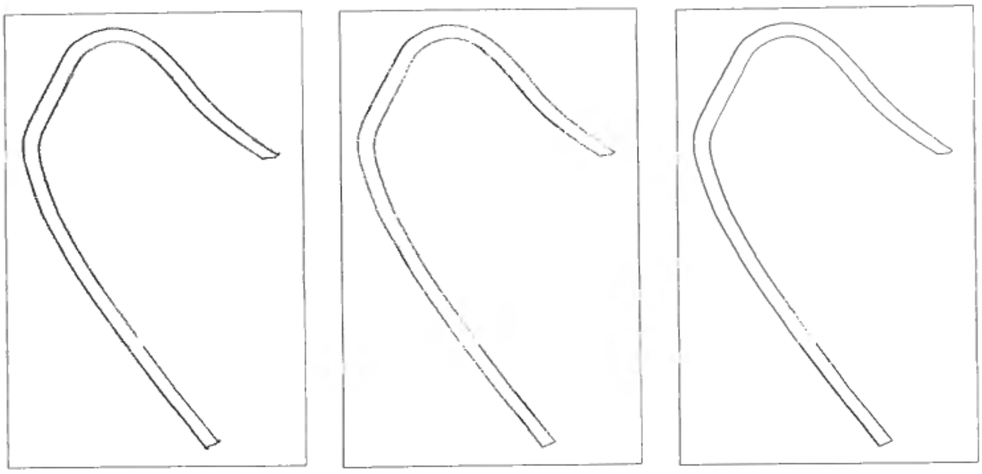
Figure 8. Region 2 processed by the Look Ahead filter and the NW estimator. (*a*) The Look Ahead filtered data. (*b*) Result after applying the NW estimator with $h^2 = 5$ to (*a*). (*c*) Result after applying the NW estimator with $h^2 = 15$ to (*a*).

6. Discussion and conclusion

We have presented a successful method for restoring a contour tracked by a robot. The method is able to cope with both outliers at corners and random noise.

According to our investigations it turns out that the restoration becomes more difficult as the speed increases. This is caused by more vibrations in the robot and, hence, more degraded data. Thomessen *et al.* (1993) used the observed contact force against the workpiece to remove outliers around acute angles. We started out by using this procedure too, but it turned out that it was not sufficient to get rid of most of the outliers.

For complicated structures with acute angles it seems difficult to obtain a perfect



(*a*) (*b*) (*c*)

Figure 9. The workpiece under study. (*a*) The observed data. (*b*) The Look Ahead filtered data. (*c*) Final estimate.

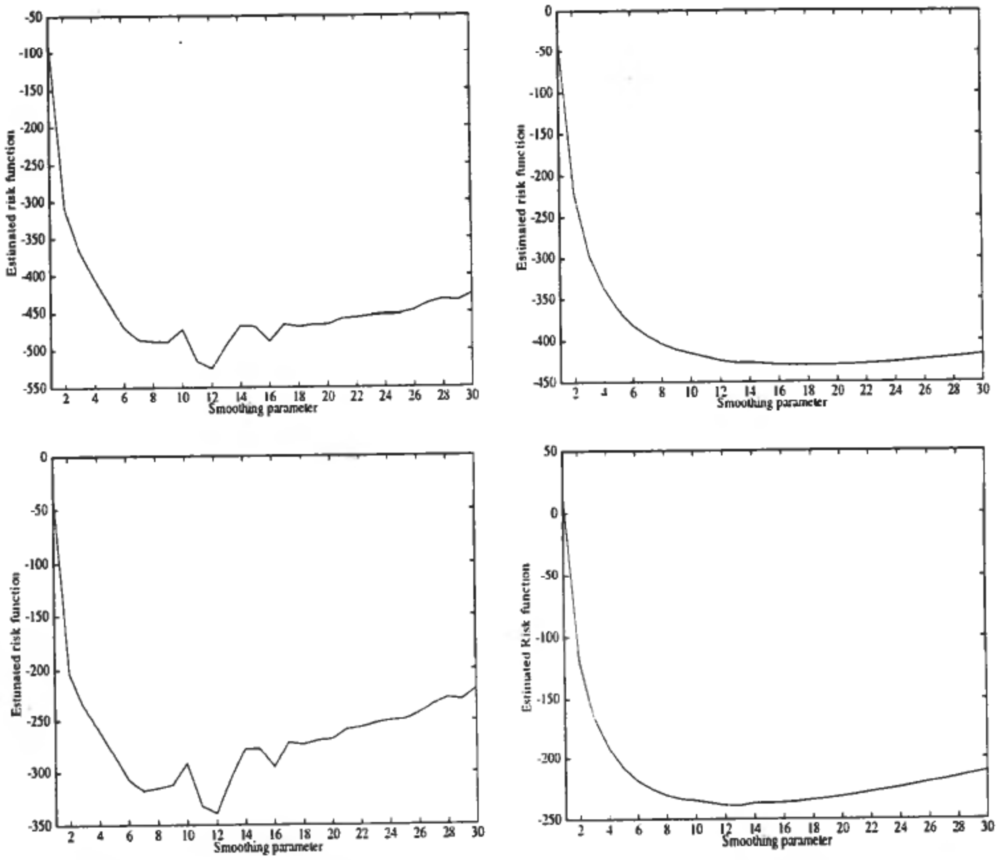


Figure 10. Estimated risk functions. To the left estimated risk function in x-direction for different h^2. Corresponding curve for y-direction to the right.

restoration of the underlying true curve. However, this is not important since, in practice, tracking angles of 90° will not be done by the robot due to the dynamic limitations. Therefore, we can conclude that our method will be able to restore the geometry for typical workpieces used in the industry.

The most important novelty in our method compared to the nonlinear Gaussian filter applied by Thomessen *et al.* (1993), is that we can remove two kinds of distortions, i.e. outliers and random noise. Recall that the Look Ahead filter contains the parameter m which denotes the number of points the filter looks ahead. In our experiments we have used different choices. Obviously, the number of points must be larger than the number of outlier points located around a corner. If m is smaller, some of the outliers will be accepted. Moreover, the upper limit for m will depend on the geometry of the workpiece. If the workpiece has sharp corners the Look Ahead filter will cut corners if m is too large. For the workpieces we have studied we found $m = 50$ to be a reasonable choice. An alternative to the method mentioned above would be to look fewer points ahead and instead use the filter several times. But then we also need to decide how many times the filter must be repeated. This is difficult to find a general rule for and, therefore, we have performed the filter only once.

The final step in our method is the application of the NW estimator. Here we need to decide a suitable value for the smoothing parameter h. Recall from Section 4 that this can be done by minimizing an estimator, $\hat{R}$(h), for the risk function $R(h)$. Before $\hat{R}(h)$ can be minimized with respect to h an estimator for σ^2 must be found. We did this by estimating a smooth curve with the NW estimator in an area where the curve is essentially linear. More precisely, we estimate a linear part of the curve by using a very big value of h, say $h = 100$. Then we estimate σ^2 by

$$\hat{\sigma}^2 = \frac{1}{M-2} \sum_i \{(x_i - \hat{\mu}_i)^2 + (y_i - \hat{\nu}_i)^2\}. \tag{5}$$

In eqn. (5) $\hat{\mu}_i$ and $\hat{\nu}_i$ denote NW estimators for the true x- and y-coordinates at point i. The parameter M denotes the number of points we use for estimating σ^2. Observe that this is an average of the quadratic differences between the smoothed curve and our observations. A natural question would be whether there should be one estimator for σ^2 in the x-direction and another in the y-direction. We tried this too but the estimated values for h were generally underestimated by this procedure. For the workpiece in Fig. 9 we found $\hat{\sigma}^2 = 0{\cdot}30$. In Fig. 10 this estimated variance is used in the calculations of $\hat{R}(h)$. For the x-direction we see that $h^2 = 12$ is a good estimate of the smoothing parameter. For the y-direction the smoothing looks relatively robust with a smoothing parameter between 14 and 20.

Our proposed filtering method was applied to other workpieces as well. From our results we conclude that the method is relatively robust for moderate fluctuations in m and h. For the workpieces we have studied a reasonable choice for m is 50 while a value for h^2 around 12 to 15 is reasonable in both the x- and y-direction.

REFERENCES

DE SCUTTER, J. (1986). Compliant Robot Motion. PhD. Thesis, Katholieke Universiteit Leuven.

GODTLIEBSEN, F. (1991). Noise Reduction Using Markov Random Fields. *Journal of Magnetic Resonance*, **92,** 104–114.

GODTLIEBSEN, F. and SPJØTVOLL, E. (1991). A Gaussian Window Filter. Department of Mathematical Sciences, Norwegian Institute of Technology.

MALLOWS, C. (1973). Some Comments on C_p. *Technometrics*, **15,** 661–675.
THOMESSEN, T., ELLE, O. J., LARSEN, J. L., ANDERSEN, T., PEDERSEN, J. E. and LIEN, T. K. (1983). Automatic Programming of Grinding Robot. *Modeling, Identification and Control*, **14,** 93–105.
WATSON, G. S. (1964). Smooth Regression Analysis. *Sankhya* A, **26,** 359–372.