On Preprocessing Multi-Channel Sensor Data for Online Process Monitoring

Holger Schöner, Bernhard Moser
Software Competence Center Hagenberg
A-4232 Hagenberg, Austria
{holger.schoener,bernhard.moser}@scch.at

Edwin Lughofer
Fuzzy Logic Laboratorium Linz-Hagenberg
Johannes Kepler University Linz
A-4040 Linz, Austria
edwin.lughofer@jku.at

Abstract

This paper discusses online monitoring production processes based on multi-channel sensor data. Particularly the problem of transient and anomaly detection is addressed for which a processing framework consisting of a preprocessing module and a reasoning engine is outlined. While there is much theory available in the literature for the reasoning engine this is not the case for the preprocessing which massively depends on the physical interpretation and semantics of the data. The paper addresses these problems and proposes new normalization concepts based on regularization especially for making transients of multi-channel data comparable and adequate for further processing by a reasoning engine. A proof of concept is demonstrated by means of real data from an injection moulding process.

1 Motivation

The motivation for this study is the idea to check the quality of products produced by means of an injection moulding process indirectly by monitoring sensor data from the moulding machine rather than measuring various relevant parameters of the produced product directly which in this case would be too expensive. The mechanisms leading to anomalies are for example defects in heating, parts sticking to the moulding form, changes in production parameters like the target temperature of certain machine parts, and changing environment conditions like humidity and draft.

In this application context it is crucial to detect deviations from a steady state of the machine in order to get information about possible negative impacts on the quality of the resulting products. For classifying the actual status of the machine, whether the process shows a steady normal state, a transition from one normal state to another one or whether the machine behaves in an abnormal relevant manner, a reasoning engine is necessary which takes all the available sensor data together with its statistical features into account and aggregates all this information to a scalar valued degree of instability. There is an extensive literature available concerning extracting statistical features from time series including outlier detection, see e.g., [1, 2, 3, 4, 5, 6, 7] on the one hand and just as well for the design of reasoning engines in the context of process monitoring, see e.g. [8, 9] and related literature in the context of approximate reasoning and knowledge-based systems, see e.g. [10]. The mentioned literature mainly deals with single channel problems or is restricted to post hoc analysis of the time series which would not allow an online classification of the actual status of the process.

Especially for online transient classification a concept for evaluating gradients, let call it *normalized gradient measure* is needed that is at least invariant with respect to linear transforms of the time series. This means that $\mathcal{G}(f(.),t) = \mathcal{G}(\alpha + \beta f(.),t)$, where $\mathcal{G}(f(.),t)$ models the normalized gradient measure of the signal f at time t.

Such invariant properties are necessary to keep the complexity of the reasoning engine in a reasonable range, otherwise in the worst case for any combination of transform-parameters of a channel thresholds and other parameters in the reasoning engine have accordingly to be adapted which might lead to a combinatorial complexity. A further crucial invariance property is comparability across different channels which might be modeled by means of an evaluation measure

$$\mu_{\text{trans}} : \mathcal{F} \times T \rightarrow [0,1]$$

that reflects to which degree a sensor signal $f \in \mathcal{F}$ at time $t \in T$ is in a transient status. Then this comparability property can be formulated by

$$\mathcal{G}(f,t_f) = \mathcal{G}(g,t_g) \Rightarrow \mu_{\text{trans}}(f,t_f) = \mu_{\text{trans}}(g,t_g). \quad (1)$$

Obviously, the standard gradient does not meet these invariant conditions. After recalling some basic related work in Section 2, the paper focuses on proposing a concept for such a normalized gradient measure $\mathcal{G}$, which is discussed in Section 3. With Section 4 the paper concludes with an exper-

imental proof of concept for sensor data from a moulding process.

2 Model based anomaly detection

This section recalls two standard approaches for detecting untypical signal behaviors like jumps and peaks in a time series by looking for significant deviations from the predictions of a model.

2.1 Sliding Regression with error bars

Sliding Regression denotes an algorithm, which allows a polynomial curve fit to be efficiently updated online, i.e. for each new sample available for a channel, while older points have an exponentially decaying weight. This model also computes a confidence band around its predictions, and then determines, whether a new point lies within this confidence band.

The modelling procedure The basis for sliding regression models is the linear regression of the sampled channel values $\vec{y} = \big(y_1, \ldots, y_N\big)^T$ on the time variable t, with $t = N$ being the last sampled time. The function fitted is the polynomial

$$f(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + ... + \beta_n t^n. \tag{2}$$

The fit is performed by choosing the parameters $\vec{\beta}$ to minimize the squared error

$$\sum_{t=1}^{N} \big(y_t - f(t)\big)^2 \tag{3}$$

The solution for this optimization problem is, cf. [11]:

$$\vec{\beta} = (X^T X)^{-1} X^T \vec{y} \tag{4}$$

where X is the $N \times n$ regression matrix:

$$X = \begin{bmatrix} 1 & 1 & 1^2 & \ldots & 1^n \\ 1 & 2 & 2^2 & \ldots & 2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & N & N^2 & \ldots & N^n \end{bmatrix}.$$

Each column j corresponds to one parameter β_{j-1}, and each row i to one of the available sample times $i \in \{1, \ldots, N\}$.

For the online case it is more efficient to update the regression model with new incoming samples, rather than to re-estimate with the complete amount of data. Furthermore, we are actually interested in local models fitting only the last sampled values and not the whole time series, to model the current trend in the data. When using an exponential decay of the weight of older samples, both ideas can be combined. Letting $\lambda < 1$ denote the factor for the exponential decay, the cost in (3) is modified to

$$\sum_{t=1}^{N} \lambda^{N-t} \big(y_t - f(t)\big)^2 \tag{5}$$

The online version of (4), for updating the parameter vector $\vec{\beta}$ (for each newly arriving sample $N+1$), is, cf. [12, 13]:

$$\vec{\beta} \longleftarrow \vec{\beta} + \vec{\gamma}_N \big(f(N+1) - \vec{r}_{N+1}^T \vec{\beta}\big), \tag{6}$$

with the new row $\vec{r}_{N+1}$ in X

$$\vec{r}_{N+1} = \big(1, N+1, (N+1)^2, \ldots (N+1)^n\big)^T, \tag{7}$$

the correction vector

$$\vec{\gamma}_N = \frac{P_N \vec{r}_{N+1}}{\lambda + \vec{r}_{N+1}^T P_N \vec{r}_{N+1}}, \tag{8}$$

and the update of the matrix $P = (X^T X)^{-1}$:

$$P_N = \frac{1}{\lambda} \big(I - \vec{\gamma}_{N-1} \vec{r}_N^T\big) P_{N-1}. \tag{9}$$

To achieve good convergence to the optimal parameter values, the starting values for P_0 should be αI, with α being a large constant, and $\vec{\beta}$ should be initialized with zeros. For two reasons, the degree n of the polynomials was set to a low value. One is, that polynomials of low degree can be fitted faster than those of higher degrees; the other is, that several of the analyzed channels are highly noisy, which could lead to overfitting with low values of λ and higher values of n.

Computation of the confidence band To determine, whether a deviation of a newly observed value from the prediction of the sliding regression model is significant, the confidence band of the model is computed, cf. [14, 15]. For the new sample $N+1$ the upper and lower boundaries are

$$f(N+1) \pm \sqrt{\mathrm{diag}\Big(\mathrm{cov}\big(f(N+1)\big)\Big)}, \tag{10}$$

with the following definitions:

$$\mathrm{cov}\big(f(N+1)\big) = \vec{r}_{N+1} \mathrm{cov}(\vec{\beta}_N) \vec{r}_{N+1}^T, \tag{11}$$

$$\mathrm{cov}(\vec{\beta}_N) = \hat{\sigma}_N^2 P_N. \tag{12}$$

$\hat{\sigma}_N^2$ can be estimated by:

$$\hat{\sigma}_N^2 = \frac{\sum_{t=1}^{N} \big(y_t - f(t)\big)^2}{N - \deg},$$

where "deg" is the number of parameters, here the polynomial degree plus 1, ie. $n+1$.

The linear parameters $\vec{\beta}$ and the matrix P are adapted according to (6) and (9), while $\hat{\sigma}^2$ can be updated by

$$\hat{\sigma}_N^2 = \frac{(N-1-\text{deg})\hat{\sigma}_{N-1}^2 + \big(y_N - f(N)\big)^2}{N-\text{deg}}. \quad (13)$$

Thus, the confidence band can be updated on-line together with the rest of the sliding regression model.

The decision, whether a newly available sample represents an anomaly, is made based on the size of the deviation of the measured signal value from the predicted signal value, relative to the computed confidence band. The larger the value

$$\frac{|y_{N+1} - f(N+1)|}{\sqrt{\text{diag}\Big(\text{cov}\big(f(N+1)\big)\Big)}} \quad (14)$$

is, the more likely it is, that the new sample is an outlier.

2.2 Online prediction with adaptive AR models

To allow detection of anomalies in time series with dependencies other than learnable by the polynomial model described above, we implemented the same approach, but using AR (auto-regressive) models [16] instead of Sliding Regression models. These models are often used to model time series data, and use a linear combination of previous values to predict new ones:

$$f(t) = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \ldots + \beta_m y_{t-m}, \quad (15)$$

where m denotes the degree of the AR model, ie. the maximal time delay or number of past values used for prediction.

For AR models, the function f is again linear in the model parameters $\vec{\beta}$, and the techniques for Sliding Regression models (recursive least squares, confidence bands) can be used for them as well, when changing the regression matrix X to

$$X = \begin{bmatrix} 1 & y_m & y_{m-1} & \cdots & y_1 \\ 1 & y_{m+1} & x_m & \cdots & y_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & y_N & y_{N-1} & \cdots & y_{N-m} \end{bmatrix},$$

and the regression vector $\vec{r}$ for the recursive adaptation to $\vec{r}_{N+1} = (1, y_{N+1}, y_N, \ldots, y_{N+1-m})^T$. A new prediction is obtained by substituting the last m points into (15) and comparing the predicted value with the measured one by (14).

3 Normalized transient measure

Let us start with a local linear fit f to the last M samples $\vec{y} = \big(y_{N-M+1}, \ldots, y_N\big)^T$ yielding the function

$$f(t) = \beta_0 + \beta_1 t \quad (16)$$

with the parameters β_0 and β_1. Further let us consider the regression matrix X_N according to (4),

$$X_N = \begin{bmatrix} 1 & N-M+1 \\ 1 & N-M+2 \\ \vdots & \vdots \\ 1 & N \end{bmatrix}.$$

Because $X^T X$ may be singular (or badly conditioned), let us modify (4) by a regularization term (see [17], 'ridge regression') which yields

$$\vec{\beta} = (X^T X + \alpha I)^{-1} X^T \vec{y}, \quad (17)$$

where $\vec{\beta} = \beta_0, \beta_1$ and α is the regularization parameter Formula (17) can be looked at as a numerical stable approximation of the gradient which also is not a normalized gradient measure in the sense as outlined in Section 1.

Our proposed normalized transient measure $\mathcal{G}$ is constructed in two steps:

[S1] The gradient β_1 is transformed to $\beta_1' \in [0, \infty)$ taking the quality of fit of the linear model and other parameters into account.

[S2] The range of β_1' is normalized to the unit interval by some monotonic mapping $\iota : [0, \infty) \to [0, 1]$.

Step **[S1]** relies on the following transformation

$$\beta_1' = \sqrt{M}\sqrt{1 + n_{\text{high}}}\ \frac{\big(|\beta_1| \ / \ \sigma_N\big)^2}{1 + \big(e_N \ / \ \sigma_N\big)^w} \quad (18)$$

where M denotes the window size, n_{high} some compensation factor which refers to long continuing transients (see item 'Continuation of Long Transients' below), σ_N denotes some measure for the standard deviation (see item 'Variance' below), e_N denotes a quality measure of the linear fit and w some weighting parameter.

In the following we discuss various aspects being relevant for this proposed normalization concept and give some heuristic arguments for special choices of parameters:

Window size Note that the detection of trends is less sensitive to random fluctuations in the channel when the window size M of the fit is larger. This leads to a multiplication of the gradient value (17) by a factor monotonous in M. We choose as factor $\sqrt{M}$ to reduce the influence of enlarging the window size, if it is already large.

Continuation of long transients In case of long continuing transients we argue that the conditional probability is rather high that the transient will further continue rather than abruptly breaking down. A multiplication of (17) by $\sqrt{1+n_{\text{high}}}$ takes this into account where n_{high} is the number of preceding time steps, in which transients were detected without interruption.

Variance Some measure of the average deviation of new channel values from the mean of recent values can be used for normalization; for channels with highly fluctuating values, an observed small gradient is less probable to be significant, than for a channel with very little noise. This measure of deviation or variance is denoted as σ_N (for time N) in the following, and might be e.g. the standard deviation. We use the exponentially weighted mean absolute deviation from the moving average μ_t instead, to achieve less sensitivity to outliers:

$$\sigma_N = c_\sigma \sigma_{N-1} + (1-c_\sigma)\,|y_N - \mu_{N-1}| \tag{19}$$

$$\mu_{N-1} = c_\mu \mu_{N-2} + (1-c_\mu) y_{N-1}, \text{where} \tag{20}$$

$$\mu_1 = y_1, \quad \sigma_1 = \infty, \quad \sigma_2 = |y_2 - y_1|,$$

with appropriately chosen c_σ and c_μ.

Fit quality Another characteristic used for normalization is the current error of the linear model fit e_N, by dividing the current transient measure value by $(1+e_N)$. This assigns less importance to the transient measure in regions where the underlying model does not fit well. For sake of less sensitivity to outliers we prefer the mean absolute prediction error in the fit window of the last M channel values $\frac{1}{M}\sum_{t=N-M+1}^{N} |f(t)-y_t|$. Additionally we use a parameter w for weighting this fit quality by taking it to the power of w, i.e., e_N^w.

Absolute value In our application, it usually does not matter whether a transient is occurring with rising or falling values; for this reason we only take the absolute value of the transient measure value into account when comparing it to an appropriately chosen threshold.

Determination of threshold In the second step, **[S2]** the so transformed gradient has to be normalized to the unit interval by some function $\iota(.)$ which we model by a function $\iota_{\theta_1,\theta_2}(.)$ (several choices are possible here; our definition of $\iota_{\theta_1,\theta_2}(.)$ is given in the appendix) that is parametrized by threshold parameters θ_1 and θ_2,

$$\theta_1 = c_{\text{thresh}} \ \text{quantile}\big(B, q\big) \tag{21}$$

$$B = \big(\beta_1'(N-M_B), \ldots, \beta_1'(N-1)\big), \tag{22}$$

where $\beta_1'(t)$ denotes the β_1' computed for the tth data point (for $t < 1$ these are initialized with a configured constant β'_{def}).

To allow the transient detection to work for a diverse range of channels, we determine the threshold s for comparison with the normalized transient measure adaptively (cf. the appendix). A list B of the last M_B observed gradient values is kept.[1] Each new gradient value replaces the oldest one in this list. The current threshold is then determined by computing a quantile q of this list, e.g. the median $q = 50\%$. This quantile reflects expectations about the fraction of instabilities during the time represented in B. The lower this value, the more (or longer) instabilities can occur without raising the threshold manifestly. On the other hand this can lead to many overdetections, when in reality there are not as many instabilities as expected. To compensate for this, the threshold determined by the quantile is multiplied by a configured factor c_{thresh}, before comparing it with the actual normalized transient measure values using $\iota(.)$. The so transformed gradients are values from the unit interval and can be aggregated by e.g. fuzzy logical connectives within an reasoning engine.

Due to this transformation of the original gradient the normalized gradient measures $\mathcal{G}$ can further be processed in the reasoning engine on a more abstract and logical level which also allows the integration with other methods like the 'jump' detection provided by the sliding regression method.

4 Experimental results

The system described in the previous sections was developed for instability detection in a variety of injection moulding machines. Data are sampled from the available channels once for each part produced, and contain information about configuration settings, timing, forces and pressures, temperatures, speed, and dimensions, among others. The configuration of the method parameters used for the following plots is given in the appendix. Several channels contain mainly instabilities manifesting themselves as more or less obvious jumps (Panel 3 in Fig. 1); these can be detected quite well using the sliding regression.

Some other channels, containing mostly smooth curves with transients (Panel 2) or even mixtures of transients and single outliers or jumps (Panel 1), are harder to analyze; human experts also have difficulties to tell, which parts should be considered to be irregular. Examples are: (i) After the extreme shift of the channel values between samples 31 and 37 in Panel 1 (which are shown only cropped in the plot), the transient method is reset[2] and it takes some samples until it

[1] And it is initialized with a configured value β'_{def}, chosen manually not to give too many false detections on a variety of different channels.

[2] Which is technically necessary to avoid long phases of overdetection

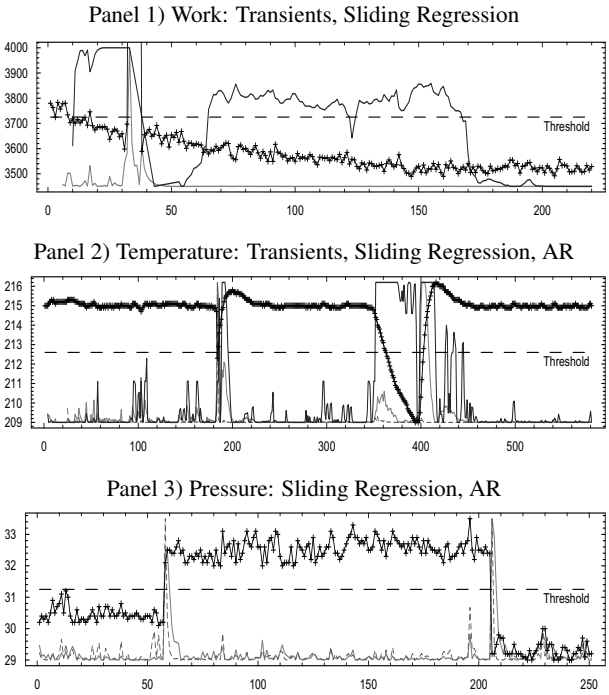

Figure 1. Examples of method performances on injection moulding data. The samples are placed along the horizontal axis, the vertical axis shows the channel values. Instability predictions by the different methods are overlaid, with 0 (no instability) at the bottom and 1 (almost certain instability) at the top of the plot; an example for an alert threshold of 0.5 is also shown. *Black solid lines with "+" marker:* channel data, *Dark solid lines:* transient detection, *Light grey lines:* Sliding regression method, *Light grey dashed lines:* AR method.

can work again (sample 42). (ii) The length of the transient in Panel 1 (especially when considering only past values); is it finished at sample 70, or (as detected by the transient method) around sample 170? (iii) The transient at and after sample 400 in Panel 2; when the signal levels off, even for only a few samples, the detection is often interrupted, although seeing the whole time series, it would probably not be judged so.

5 Conclusion

The intent of our work was to develop a framework of transient detection methods suitable for the injection molding process, which deals explicitly with the challenges of real time prediction, a wide variety of channels with different characteristics, a wide variety of different machine types, and the desire to obtain diagnostic information about instability reasons.

In this paper we proposed a strategy for making gradient measures comparable in order to detect transient states which can be used for online process monitoring. In spite of the heuristic nature of the outlined approach the results for the injection moulding process are very promising. These experimental results should motivate to intensify research in this direction. Particularly, what remains to be investigated in more depth is a comprehensive analytic and theoretic study of the concepts and problems within the framework of stochastic processes.

Appendix

Specifications of the application

As normalization function $\iota_{\theta_1,\theta_2} : [0,\infty) \to [0,1]$ we choose a model that fits $\iota_{\theta_1,\theta_2}(\theta_1) = 0.5$ and $\iota_{\theta_1,\theta_2}(\theta_2) = 1$ as e.g.

$$\iota_{\theta_1,\theta_2}(x) = \begin{cases} a_1 x^2; & |x| < \theta_1 \\ a_2 x^2 + b|x| + c; & \theta_1 \le |x| \le \theta_1 \cdot \theta_2 \\ 1; & |x| > \theta_1 \cdot \theta_2 \end{cases} . \tag{23}$$

For the final assessment whether at time t the process is in a transient state the normalized gradient measures of all the channels are aggregated by means of the $\max$-operator, which is the standard fuzzy logical connective for disjunction. The choice of the $\max$-operator seems reasonable as the final assessment honors the most striking indication for a transient status detected by any of the methods in any of the channels.

The following table specifies ranges of values for the model parameters of Section 4.

Parameter	**Useful Range**	**Values used in Panel**		
		1	2	3
Transients				
M	5–100	50	5	
q	0.5–0.99	0.5	0.5	
c_{thresh}	2.0–8.0	4.0	7.0	
M_B	50–10000	1000	250	
β'_{def}		0.02	0.02	
w	0.1–5.0	1.0	3.0	
Sliding Regression				
λ	0.5–0.99	0.9	0.9	0.9
s	5.0–20.0	12.0	12.0	15.0
Autoregressive Models				
n	2–10			10
s	5.0–50.0			30.0

Some parameters are less important or sensitive; they were

in some circumstances.

set to $c_\sigma = 0.95$, $c_\mu = 0.05$, $s_{\text{scale}} = 10$ (transient detection), $s_{\text{scale}} = 2$ (sliding regression, AR models).

References

[1] J. Zheng, M. Hu, and H. Zhang (PRC), "Usage Behavior Profiling for Anomaly Detection using Vector Quantization," in *Proc. of Communication Systems and Applications (CSA) 2005*, Banff, Alberta, Canada, 2005.

[2] S. Bay, K. Saito, N. Ueda, and P. Langley, "A framework for discovering anomalous regimes in multivariate time-series data with local models," in *Symposium on Machine Learning for Anomaly Detection*, Stanford, U.S.A., 2004.

[3] C.C. Aggarwal and P.S. Yu, "Outlier Detection for High Dimensional Data," *SIGMOD Conference*, 2001.

[4] E. Keogh and S. Lonardi and W. Chiu, "Finding Surprising Patterns in a Time Series Database in Linear Time and Space," *Proc. of the Eighth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 550–556, 2002.

[5] E. Keogh and S. Lonardi and C.A. Ratanamahatana, "Towards Parameter-Free Data Mining," *Proc. of the Tenth ACM SIGKDD Int. Conf. on Knowlede Discovery and Data Mining*, 2004

[6] C. Chen and L. Liu, "Joint Estimation of Model Parameters and Outlier Effects in Time Series," *Journal of the American Statistical Association*, 88:284–297, 1993.

[7] E. Keogh, J. Lin, and A. Fu, "Hot sax: Efficiently finding the most unusual time series subsequence," in *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005)*, Houston, Texas, 2005, pp. 226–233.

[8] J. Korbicz, J. Koscielny, Z. Kowalczuk, and W. Cholewa, *Fault Diagnosis - Models, Artificial Intelligence and Applications*. Berlin Heidelberg: Springer Verlag, 2004.

[9] L. Chiang, E. Russell, and R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. London, Great Britain: Springer Verlag London Berlin Heidelberg, 2001.

[10] R. Kruse, E. Schwecke, and J. Heinsohn, *Uncertainty and Vagueness in Knowledge Based Systems. Numerical Methods*. New York, Berlin, Heidelberg, Germany: Springer Verlag, 1991.

[11] F. Harrel, *Regression Modeling Strategies*. New York, USA: Springer Verlag New York Inc., 2001.

[12] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, New Jersey 07458: Prentice Hall PTR, Prentic Hall Inc., 1999.

[13] K. Aström and B. Wittenmark, *Adaptive Control - Second Edition*. Addison-Wesley ISBN-0-201-55866-1, 1995.

[14] N. Draper and H. Smith, *Applied Regression Analysis. Probability and Mathematical Statistics*. New York: John Wiley & Sons, 1981.

[15] O. Nelles, *Nonlinear System Identification*. Germany: Springer Verlag Berlin, 2001.

[16] G. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco: Holdenday, 1970.

[17] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York, Berlin, Heidelberg, Germany: Springer Verlag, 2001.