IFAC Workshop on Mutual Impact of Computing Power and Control Theory, Prague, Sept. 1-2, 1992, Preprints 235-240

1643

Robust PID control

Pavel Kovanic, Josef Böhm

Institute of Information Theory and Automation
Czechoslovak Academy of Sciences
Pod vodárenskou věží 4, 18208, Prague 8
Czechoslovakia, Phone: (422)8152274,

Summary The recent progress of microelectronics enables the creation of intelligent sensors as integrated units including both a sensor and a microcontroller capable of performing many decentralized monitoring and/or control system functions. This paper shows how the robustness of such a device applied for the PID-control can be substantially increased by the gnostical filter. The high filtering efficiency of such an algorithm enables the use of fixed optimum digital operators for estimation of signal derivatives according to the method of 'static programming' and yields considerable savings in operation time and memory requirements, so that the task is manageable by a single-chip computer. An example demonstrates practical realizability of this approach by currently available technology.

Key words. PID control, filtering.

1 Intelligent sensors

It is well-known that in spite of the impressive progress of automatic control theory, the prevailing application have automatic control systems of the PID-type. The main reasons for broad usage of such 'conservative' technology can be probably found in the following:

- most of practical control problems are simple enough to be solved by an experimentally tuned PID-controller,
- functions of a PID-controlled systems are fully understandable for an experienced practician without the necessity of references to a theory,
- there exists natural resistance to application of techniques more complicated than actually necessary,
- simple techniques are believed to be less expensive than the complicated ones.

The recent progress of microelectronics represents a powerful factor which cannot be without influence on these considerations. A microcontroller may be not only more universal in application but also cheaper than a simple specialized control device. A single-chip computer is reliable and flexible enough to be considered as a black-box easilly programmable for desired functions from the outside without a deep understanding of its inside. There are also further impacts of progress of microelectronics.

Starting with computers of the second generation, the typical approach to control and supervising complex technological processes has been the application of large central main-frame computers. This central 'brain' of the process concentrated all process information from a mass of simple peripheral sensors via centralized multiplexors and analog-to-digital convertors. Such a centralization – although advantageous in many aspects – also involved serious problems:

- Danger of technical faults: A break-down of the central computer can result in a serious outage of the whole system. Application of a stand-by computer involves further complex electronic equipment, increasing operating and software problems.
- Danger of software faults: The complexity of the system and of its functions requires support by extremely complicated programs. Such software cannot be reliably verified in all its functions. Practical operation of the system can thus create unexpected conditions under which the software fails.
- Vulnerability of the system: A single centre of all monitoring, emergency and control functions can be easily subjected to damage or an attack with fatal consequences for the system.

This is why centralization gave rise to an inverse tendency towards decentralization. The recent development of integrated electronics enabled the decentralization of not only some simple monitoring functions but also the more sophisticated, 'intelligent' functions of the main-frame computer. It is possible now to create *intelligent sensors* as compact units integrating a sensor with a microprocessor, taking over many important functions of the central 'brain' of the system as well as some new functions:

- reliable self-diagnostics of the particular sensor,
- robust filtering of the process level, of its rate of change or even its acceleration,

- reliable classification of states of the observed object ('everything O.K.'/'significant trend'/'fast transient'/'emergency condition of i-th class')
- efficient automatic control of a local subsystem,
- activation of local warning or emergency systems,
- 'post-mortem' record (registration of the process development under emergency conditions),
- on-line estimation of probabilities of some events or states.

Decentralization of intelligence enables a significant simplification of the central supervising computer and its programs. However, such a distribution of both functions and computational power can bring a positive effect only if the reliability of the distributed system will be actually better than that of the centralized one. The reliability requirements of each of the local devices should be therefore enormous. It can be therefore expected that requirements for intelligent sensors will include following conditions:

1. the hardware should be limited by the capacity of single-chip microcontrollers,
2. although performing complex functions, software should be both simple and robust.

There are two techniques application of which can be useful in meeting these requirements, namely *gnostical robust filtering* and *static programming*. Gnostical filtering protects against gross disturbances of monitored variable and reduces the effects of noise without relying on *a priori* assumptions related to some statistical models. The problem of robustness can thus be solved and reliability of functions increased. The method of static programming applied to data already smoothed by the gnostical filter enables the realization of minimum variance estimation of such important quantities as rate of change and acceleration of the observed variable, etc. with minimal time and memory requirements. The questions to be answered below are whether existing microelectric technology can in fact realize such intelligent sensors suitable at least for PID-control of a real system, and what performance characteristics would such sensors exhibit.

2 Gnostical robust filter

There are two main elements of gnostical filtering algorithm, on-line estimation of the scale parameter and the recursive weighted filtration.

2.1 Scale parameter estimation

In gnostical formulae the scale parameter is playing the role of a data spread characteristic. There is no universal optimum of this characteristic, it depends on the circumstances of the application. The on-line filters need a scale parameter that reflects the behaviour of the last few data, not of all data observed. The experience has shown that the scale parameter estimation procedure based on the balanced entropy principle described in [1] serves well for this purpose. The idea of this procedure is to choose a scale parameter value that makes the summed entropy of data equal the mean entropy obtained by integration using the data distribution estimate dependent on the scale parameter. This procedure is organized as a recursive algorithm to run quickly in real-time applications.

2.2 Weighted filtration

One feels intuitively that to obtain good filtration effect, it is necessary to give 'good' data greater weight than 'bad' data. However, a variable data weight can be provided only by a nonlinear filter. The problem is how to choose the nonlinearity that achieves the 'best' filtration effect. The gnostical theory develops formulae for information loss and entropy drop caused by data uncertainty. Extremizing these important characteristics, we may get robust and efficient filters especially suitable for heavy-duty applications and treatment of strongly dispersed data. The general approach is described in detail in [2]: The data model is represented by a linear or nonlinear function of a given structure with unknown parameters. The criterion function for evaluating the goodness-of-fit of measured data by the model is a gnostical uncertainty characteristic e.g. the information loss due to uncertainty. This criterion is extremized using the steepest-descent algorithm. As shown in [2], resulting iterative algorithm has a form remininiscent of the least-squares solution but in which each data portion is given a variable rather than a constant weight. The solution for the filter of a quasi-constant input disturbed by an unknown noise have been obtained as a special case of formulae from [2].

2.3 Example of the dynamics of the gnostical filter

A comparison of the gnostical filter (G) with a linear filter (L) is illustrated in Fig. 1. Data (denoted by crosses, D) are composed by a constant (1_f), by approximately normally distributed random components and by random gross errors uniformly distributed between 0 and 5. Time intervals between gross errors are also random. To show both steady state and transients, there is a change of data level in the middle part of Fig. 1 caused by multiplication of all data by 1.5. The appearance of gross errors (outliers) is shown by arrows on the top of Fig. 1. The high sensitivity of the linear filter to outliers is manifested by high bursts of its output (L) reacting to gross errors. By contrast, the gnostical filter ignores the outliers to such a degree that even its first and second derivatives (d^1 and d^2) remain undisturbed. Such stability is not caused by a large time constant, as seen from transients: after a step change of the signal level, the gnostical filter (G) reaches the new value as quickly as the linear filter. A small 'conservative hesitation' of the gnostical filter after a sudden signal change is later compensated by a considerably faster reaction. This effect is especially clear from the reaction to the second, negative step of signal level. It can be also observed that both derivatives exhibit a high signal-to-noise ratio. Both derivatives are delayed about 2 sampling intervals but this drawback does not play a role for slower signal changes. Moreover, a faster sampling would diminish this delay.

It can thus be concluded that the gnostical filter outperforms the linear filter of comparable time constant in both steady-state and transient behaviour. This positive experience can be complemented by the report on a successful implementation of the gnostical filter into the read-only-memory of a single-chip microcontroller. One can thus expect that positive features of the gnostical filters will be available for practical applications of intelligent sensors within distributed monitoring and control systems.

3 Static Programming

The main idea of this old method (survey of which is in [3]) is to use optimum digital operators to avoid unnecessary repetition of calculations. This method is suitable for a broad class of problems connected with minimization of quadratic criteria. Solutions of these problems are linear with respect to data. Under certain conditions, this linearity enables the algorithm to split mass data processing into two stages:

1. Computation of optimum digital operators in the form of numerical vectors or matrices ('smart matrices'):
   (a) calculating the base of the linear space containing the useful, informative data component,
   (b) evaluating the covariance matrices of both useful and noisy random data components,
   (c) calculating the pseudoinverse covariance matrices,
   (d) representing as a numerical matrix the required type of operations to be performed with data,
   (e) making decision on a compromise between bias and efficiency of results of data processing,
   (f) calculating all pseudoinverses necessary for the formula of optimum digital operators.
2. Repeated application of optimum digital operators to different data.

It is obvious that the first stage is difficult, its requirements with respect to time and memory may be enormous. However, it is sufficient to realize it only once and forever as a permanent 'prolog', no matter how many times the resulting operators are applied. The second stage represents only multiplication of the (variable) data vector or matrix by the (fixed) matrix operator. It is not necessary that the first stage be performed in real-time or by the computer realizing the fast cycles of the second stage.

3.1 Numerical example of optimum linear operators

Let us consider a continuous time-dependent signal $y(t)$ consisting of an informative component $y_x(t)$ and of noise $\varepsilon(t)$,

$$y(t) = y_x(t) + \varepsilon(t). \qquad (1)$$

An N-tuple of the observed value $y(t)$ for $t = t_1, .., t_N$ can be written as (column) vector $Y \in R^N$. It is supposed that the corresponding informative vector Y_x composed by $y_x(t_1), .., y_x(t_N)$ can be expressed using a full-rank $N \times M$ constant matrix X and a $M \times 1$ vector A as

$$Y_x = XA \qquad (2)$$

where X is a basis of a subspace of the vector space R^N, $M < N$. Denoting the vector of required results of linear operations on $y_x(t)$ by L and supposing that noise $\varepsilon(t)$ is uncorrelated and stationary, we obtain optimum digital operator for unbiased estimation as a special case of (18) from [3] in the form

$$W = L(X^T X)^{-1} X^T. \qquad (3)$$

Choosing a basis of polynomial type and using symmetrical coordinates to write elements of the basis X as

$$x_{i,j} = ((i-(N+1)/2)\Delta)^{j-1} \quad (i = 1, .., N,\ j = 1, .., M) \qquad (4)$$

(where Δ is the sampling period) we can show that in matrix $X^T X$ as well as in its inverse denoted $K := (X^T X)^{-1}$, all elements for which the sum of indexes i and j is odd equal zero. Moreover, restricting oneself to $M = 4$ (smoothing by polynomials up to third order) and denoting

$$C = N(N^2-1)(N^2-4)(N^2-9)\Delta^4 \qquad (5)$$

we obtain

$$K(1,1) = \frac{3(3N^2-7)}{4N(N^2-4)}, \qquad (6)$$

$$K(1,3) = K(3,1) = -\frac{15}{N(N^2-4)\Delta^2}, \qquad (7)$$

$$K(2,2) = 25(3N^4 - 18N^2 + 31)/C, \qquad (8)$$

$$K(2,4) = K(4,2) = -140(3N^2-7)/C, \qquad (9)$$

$$K(3,3) = 180(N^2-9)/C, \qquad (10)$$

$$K(4,4) = 2800/(C\Delta^2), \qquad (11)$$

$$K(1,2) = K(2,1) = K(1,4) = K(4,1) = K(2,3) = = K(3,2) = K(4,3) = K(3,4) = 0. \qquad (12)$$

For estimation of the value e.g. of the first derivative at the point t_* we substitute

$$L_j = (J-1)\Delta((t_* - (N+1)/2)\Delta)^{J-2}. \qquad (13)$$

Let $t_* = 0$, then the result of estimation will be the first derivative at the center of the 'mooving window', data $(y_{t-N+1}, .., y_t)$ of which are multiplied by $W_1, .., W_N$. Using all mentioned formulae for $N = 5$, we may obtain the optimum digital operator of first derivative as

$$W = (0.08333,\ -0.66667,\ 0,\ 0.66667,\ -0.08333). \qquad (14)$$

For a more general case (such as shown in [3]), the calculations of optimum digital operators are more complicated than for our simple example. However, even these simple results may be practically applicable to save both time and memory e.g. of a computer used for robust PID-control, as in our practical example.

4 Experimental verification

The gnostical filter described above complemented with linear operators provides a robustly filtered signal and its first and second derivative. The behaviour of the closed loop system with the gnostical filter as controller was tested in application of automatic control of the system known as *ball and beam*. This nonlinear system consists of a double integrator describing the motion of the ball on an inclined beam and of a second order lag system representing the servo inclining the beam. It is known that this system is very sensitive to values of the PID-controller's parameters. Nonlinearities due to friction, limited servo velocity and finiteness of the beam length make the control even more difficult. The dominant role is played by the derivative component of the controller. The system output (ball position) may exhibit noisy behaviour because of the ball rolling upon resistance wire.

The experiments were carried out by simulation of the complete nonlinear model. In cases simulating the measurement noise of weak (realistic) intensity both unfiltered PID and gnostical controller exhibited similar behaviour as demonstrated in Fig. 2 and 3. Control parameters were adjusted by trial and error method in both cases. In the case of gnostical filter, the quality of behaviour was limited by the lag introduced by the filter, while the limiting factor in the case of the unfiltered PID-controller appeared to be the saturation of inputs. Due to the frequency response of the double integrator the fast changes of input signal cannot affect the output, however, they are dangerous for the servo. Inputs for both cases are shown in Fig. 4. revealing the superiority of gnostical filter.

In the case of increasing level of noise (for this particular system even to unrealistic intensity) the unfiltered PID is deteriorating its behaviour until the control is completely lost (see Fig. 5) while the gnostical PID-controller maintains its quite satisfactory control (Fig. 6).

The lag of gnostical filter could be easily removed by a higher sampling rate, but the quality of control would be still limited by the mentioned nonlinearities and difficult dynamics of the controlled object.

5 Conclusions

Compact PID-controllers based on intelligent sensors representing integrated units consisting of microcontrollers and sensors are realizable. Gnostical robust filters provide such units with high robustness with respect to gross measuring errors and enable their operation under strong noise and disturbances. Application of optimum digital operators saves operation time and memory requirements for real-time calculations.

References

[1] Kovanic P., A New Theoretical and Algorithmical Basis for Estimation, Identification and Information, IX-th World Congress IFAC '84, Budapest (1984), Vol.XI, 122-131

[2] Kovanic P., A New Theoretical and Algorithmical Basis for Estimation, Identification and Control, Automatica IFAC 22 (1986), 6, 657-674

[3] Kovanic P., Smart Matrices, IFAC Workshop MICC, Prague, Sept. 1 – 2 (1992).

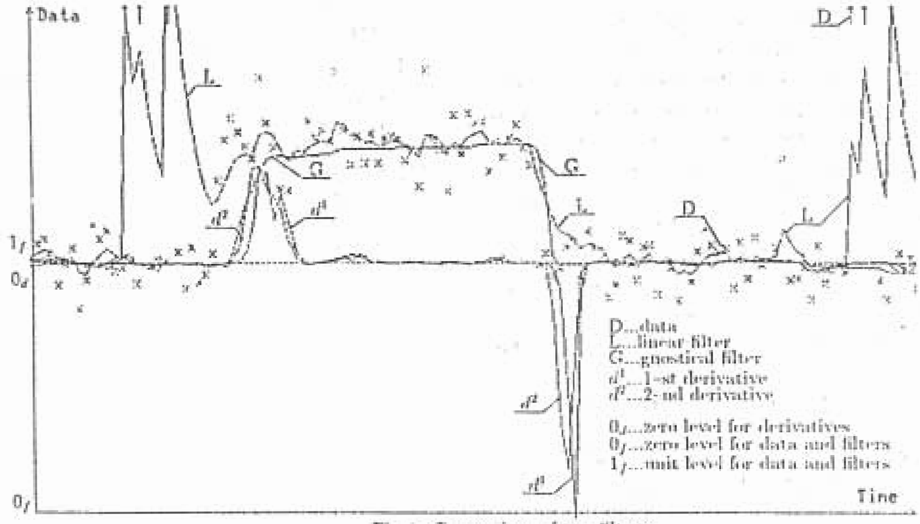

Fig. 1. Comparison of two filters

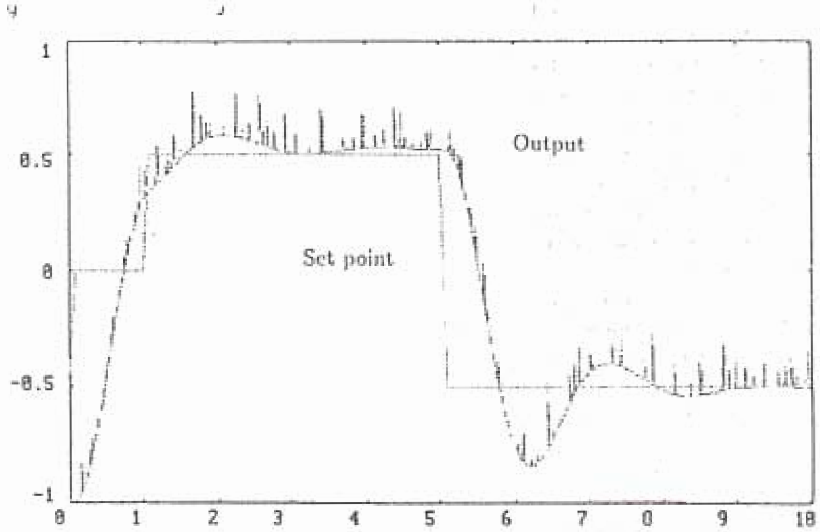

Fig.2. The system controlled by PID controller

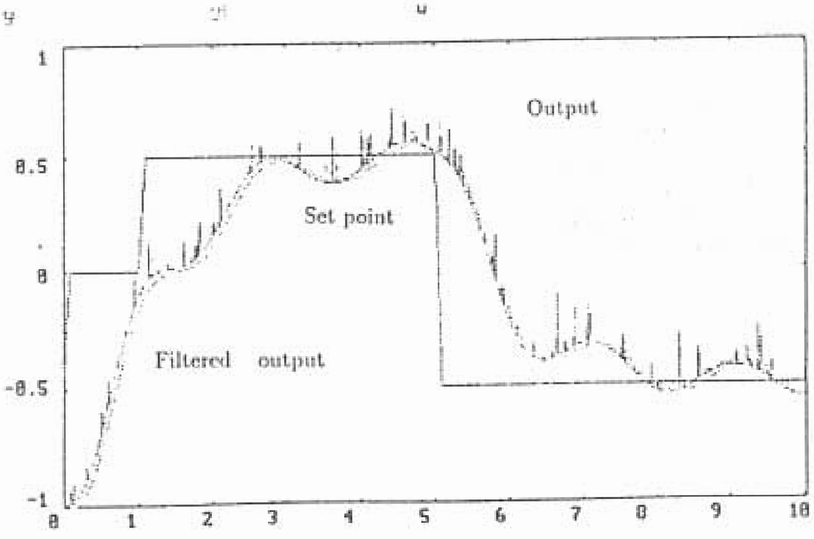

Fig.3. The system controlled by gnostical controller

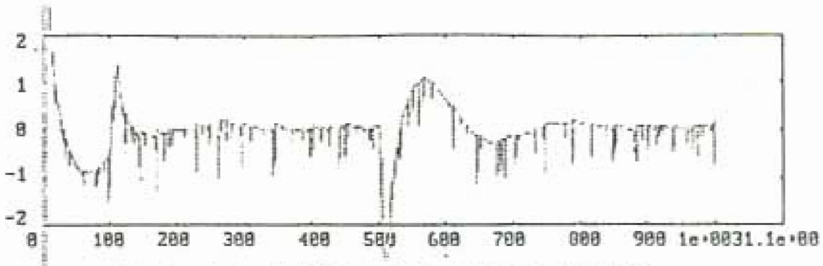

Fig.4a. Output of PID controller P=.05,I=.001;D=5

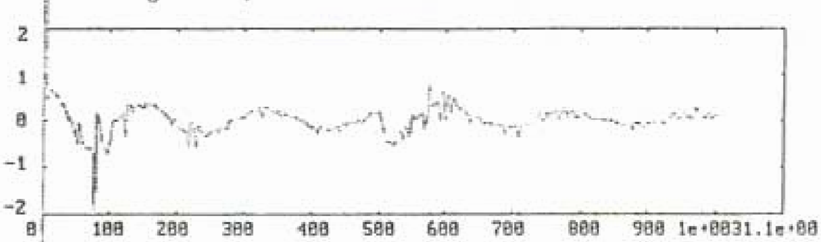

Fig.4b. Output of gnostical controller P=.7,D=12,DD=40

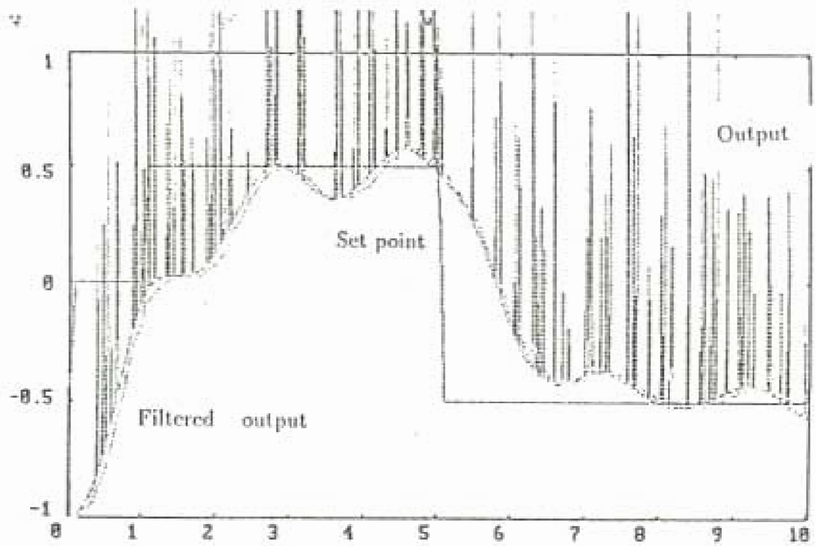

Fig.5. System with higher output noise controlled by gnostical controller

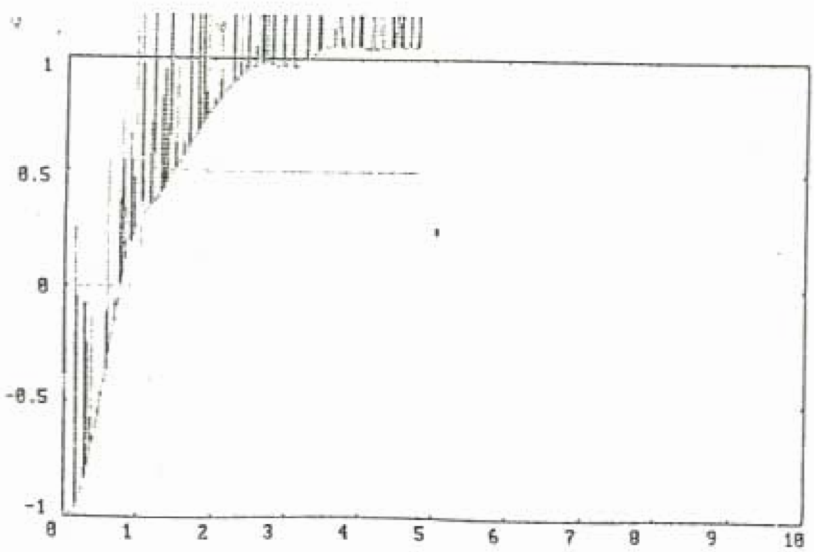

Fig.6. System with higher output noise controlled by PID controller