INFLUENCE OF NETWORK DELAYS IN RESIDUAL COMPUTATION

David Llanos [*] **Joan Colomer** [*] **Joaquim Meléndez** [*]
Marcel Staroswiecki [**]

[*] *eXiT Group, Universitat de Girona. Av. Lluis Santalo s/n E-17071 Girona, Spain. E-mail: {dllanosr, colomer, quimmel}@eia.udg.es*
[**] *Ecole Polytechnique Universitaire de Lille, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex, France. E-mail: marcel.staroswiecki@univ-lille1.fr*

Abstract: A challenging problem that motivates this work is the network delay effects in residual computation. Residuals are assumed to be identically zero in fault-free situations whereas deviations from zero alert the presence of fault in the system. In practice residuals are not identically zero due to various factors (measurements noises, modelling uncertainties, delays, and so on). This work is focused on the study of the availability of data, due to delays in the communication network, to compute residuals. An optimal dynamic alignment of data in a time window (Dynamic Time Warping) is proposed and tested to reduce errors when computing residuals.

Keywords: fault detection and isolation, residues, data acquisition, time delay estimation and dynamic programming.

1. INTRODUCTION

All model-based FDI methods: diagnostic observer (Gertler, 1988), parity space (Isermann, 1993), parameter estimation (Frank *et al.*, 2000) and structural analysis (Blanke *et al.*, 2003), require two steps. In the first step inconsistencies between the actual and expected behavior are generated. Such inconsistencies, also called *residuals*, reflect the potential faults of the system. The second step applies a decision rule for diagnosis.

The check for inconsistency needs some form of redundancy. There are two types of redundancies, hardware or physical redundancy and analytical redundancy. Hardware redundancy requires redundant sensors. It has been applied in nuclear power plant monitoring, aircraft and other safety applications which can justify its high costs and dimension. On the other hand, analytical redundancy is achieved from the functional dependence among the process variables and is usually provided by a set of algebraic or temporal relationships among state, input and the output (Blanke *et al.*, 2003). The essence of analytical redundancy is to check the actual system behavior against the system model for consistency. Any inconsistency, expressed as residuals, can be used for fault detection and isolation purposes.

Residuals are ideally zero in the fault-free case and different from zero, in the faulty case. In practice residuals are not identically zero, due to various imprecision sources (presence of noise in the measurements, modelling uncertainties and delays). A challenging problem that motivates this work is the effect of communication network delays in the computation of residuals.

Indeed, automated systems are more and more complex and spatially distributed, and communication networks have become the backbone of most control architecture. As the systems are required to be more scalable and flexible, they have more sensors, actuators and controllers, often referred to as field devices (Lee *et al.*, 2001; Willing and Wolisz, 2001). Connecting the system components via a communication network such as CAN (Controller Area Network) or PROFIBUS can effectively reduce the complexity of the system. In this work a PROFIBUS network has been used to study communication delays as potential sources of error in residual computation.

Following this analysis it is proposed to use a Dynamic Time Warping (DTW) algorithm (Sakoe and Chiba, 1978; Silverman and Morgan, 1990) for reducing time misalignment with measures and improving residual computation.

The paper is structured as follows: Section 2 illustrates potential sources of time delays in a PROFIBUS network. In section 3, Dynamic Time Warping (DTW) algorithm is summarized and a modification for on-line application is proposed. Section 4 makes an analysis of communication delays in a laboratory plant. Section 5 presents the use of DTW for improving residual computation. Finally, section 6 discusses further work and presents some concluding remarks.

2. DELAYS AND MISALIGNMENTS AFFECTING RESIDUAL COMPUTATION

PROFIBUS is a Fieldbus designed for serial communication between field devices (sensors, actuators, controllers, I/O terminals), PLCs and computers. Based on a real-time asynchronous token bus principle, PROFIBUS defines multi-master and master-slave communication relations, with cyclic or acyclic access, allowing transfer rates up to 500 kbit/s.

Figure 1 represents a typical PROFIBUS network, composed of several controllers, input/output cards and a computer acting as an OPC (OLE for process control) server operating in an Ethernet network. An OPC-client computer performs the monitoring and FDI (Fault Detection and Isolation) tasks. This schema has been used in the example reported in section 4.

PROFIBUS interface modules transmit signal values, coming from the I/O card and controllers to the OPC server, where a timestamp is assigned. The access to process data is performed by supervisory applications also under a client-server strategy with this OPC-server.

Network and bus performances related to speed, availability of devices and parameter configuration of the field bus, location of sensors, conversion speed, sample or actualization rates and so on are factors than can influence the computation of redundancy equations in a real application. Some of these aspects have been intensively addressed in the literature as real-time control problems (resources assignment, scheduling) but they are still unsolved to cope with FDI tasks.

Potential sources of time delay or asynchronous availability of data are:

- Different time response among sensors. For example, pH-meters, temperature sensors or flowmeters have different time constants. This time delay is intrinsic to the measurement process and usually is not taken into account in redundancy relations.
- Field-bus communications. Maximum speed (bit rate) is commonly limited by the length of the bus. Moreover, they have additional limitations due to the token based communication, availability and number of devices in the bus and parameter configuration. In Profibus the major time delay is bounded and results from the server update cycle called T_{TR} Target Rotation Time (Vitturi, 2000). It can vary according to technology.
- Analogue to digital conversion. Converters speed is not a real limitation. Nevertheless, some multichannel devices with a great number of multiplexed inputs could show significant limitations for applications operating at high sample rates.
- Synchronous availability of data in remote applications. There is not a certainty that OPC server can respond to synchronic petitions of clients with the same periodicity. In fact OPC server serves the last stamped data. This can cause the reception of repetitive data in a short period of time or misalignments in multiple-synchronous petitions.

Table 1 summarizes typical time delays (Trevelyan, 2004). In this work it has particulary been considered the time delay resulting from the server update.

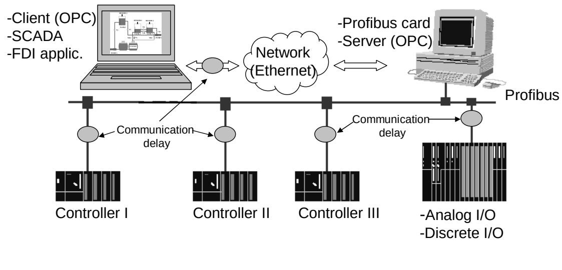

Fig. 1. Data network delays in control systems

Table 1. Typical time delays.

A/D Conversion	1 μs (1MHz sampling rate)
	to 10 ms (1kHz sampling rate)
Memory access	0.1-10 μs (depending on
	bus traffic, memory access delay
O.S. latency	5-50 μs (real time O.S.)
	1-50 ms (Windows)
Commun. latency	1 ms (local LAN) to
(Ethernet)	1 s (between continents)
OPC server update	1 -2 s

In order to cope with misalignments and delays when operating with multiple sensors to compute analytical redundancy relations for fault detection, a dynamic time warping algorithm is proposed. It operates on a time window instead of a single sample time in order to minimize the global distance of misaligned points of two sequences acquired during this time window. This algorithm has been modified and implemented to operate online in a sliding window as described in the next section.

3. DYNAMIC TIME WARPING

There are numerous studies applied to time series that have been carried out in order to compare and classify similar patterns by means of a similarity measure. Most algorithms that operate with data time series use the Euclidean distance or some variant. However, Euclidean distance could produce an incorrect similarity measure because it is very sensitive to small distortions in the time axis. Dynamic Time Warping (DTW) tries to solve this inconvenient. It uses dynamic programming (Sakoe and Chiba, 1978; Silverman and Morgan, 1990) to align time series with a given template so that the total distance measure is minimized (figure 2). DTW has been widely used in word recognition to compensate the temporal distortions related to different speeds of speech. Also, it is a good method to determine the similarity between two temporal sequences due to its capacity to align sequences with different length. It has been also applied for pattern recognition with both numerical or qualitative sequences (Colomer *et al.*, 2002).

Next, a brief notion of *DTW* is described. Given two time series X and Y, of length m and n respectively

$$X = x_1, x_2, ..., x_i, ..., x_m \qquad (1)$$

$$Y = y_1, y_2, ..., y_j, ..., y_n \qquad (2)$$

DTW will align the two sequences by finding a sequence W of k points in a m-by-n matrix where every element (i, j) of the matrix contains the local distance $d(x_i, y_j)$ between the points x_i and y_j. This is illustrated in figure 3. The path W is a

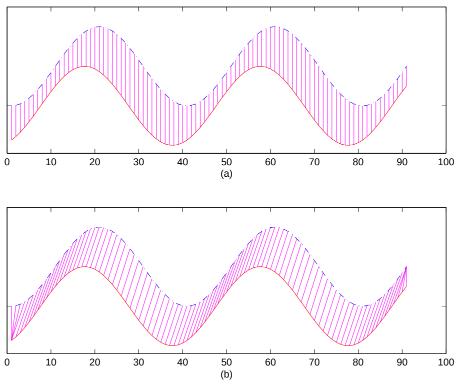

Fig. 2. Two signals with similar shape. a) Euclidean distance b)DTW

contiguous set of matrix elements that minimize the distance between the two sequences.

$$W = w_1, w_2, ..., w_k \quad \max(m, n) \leq k \leq m + n$$
$$w_k = [i_k, j_k] \qquad (3)$$

where i_k and j_k denote the time index of trajectories X and Y respectively. In order to find the best path W, some constraints are widely used to speed up DTW (Sakoe and Chiba, 1978).

The path is extracted by evaluating the cumulative distance $D(i, j)$ as the sum of the local distance $d(x_i, y_j)$ in the current cell and the minimum of the cumulative distances in the previous cells. This can be expressed as:

$$D(i, j) = d(x_i, y_j) + min[D(i-1, j-1), D(i-1, j), D(i, j-1)] \qquad (4)$$

3.1 On-line DTW

Since DTW is a good method to compensate temporal distortions due to communication delays, this paper proposes a slight modification of the algorithm in order to adapt it for on-line application. As main particularities, the two sequences

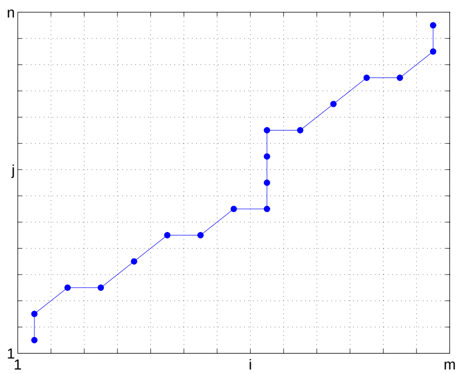

Fig. 3. An example warping path.

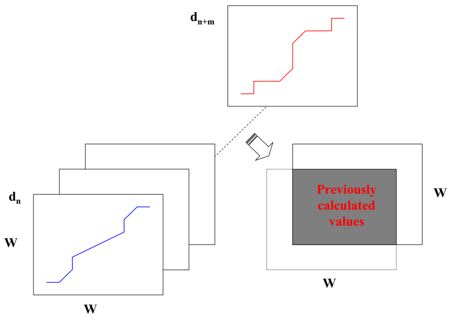

Fig. 4. On-line DTW.

have got the same length and the new algorithm returns a distance value at every sample time. So, it is necessary to obtain a finite sequence from original data to calculate new distances. The algorithm starts at time 2 calculating the local distances for the squared matrix. Later on, the matrix grows up and only local distances for new cells in the matrix are calculated. Next, the matrix reaches a maximum value established according to the process dynamics and it becomes a sliding window. At each sample time oldest cells in the matrix are deleted and local distances are calculated for empty cells corresponding to the new sample (figure 4). A new path must be found for each window and the distance value is obtained calculating the total distance according to this new path.

3.2 Time consuming

Table 2 resumes the time consuming comparison of two sequences. Algorithms implementation were done in a 2.4 GHz Pentium 4 with 512 MB of RAM. What is time consuming in both algorithms is the local distances matrix calculation. On-line DTW is faster than DTW because the first one uses previous calculated values.

Table 2. Time consuming comparison between DTW and On-line DTW

Distance measure	Samples number	Local dist. time	Path time
DTW	50	1.1 s	1 ms
On-line DTW	50	20 ms	1 ms

Important factors to be considered is the window size because it could produce a filtering effect.

4. ANALYSIS OF COMMUNICATION DELAYS IN A LABORATORY PLANT

To illustrate these problems associated with the existence of asynchronous delays in the communication, a laboratory plant has been used. Dynamic time warping strategy has been compared with traditional approach for residual computation.

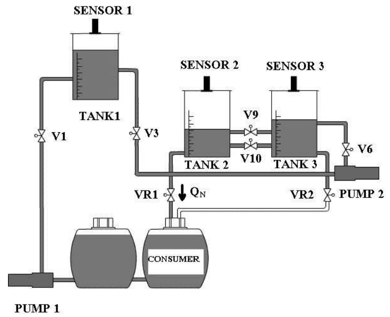

Fig. 5. The laboratory plant.

4.1 Description of the system

The plant of figure 5 provides a constant output flow of tank labelled as TANK 2 due to the control of level in this tank performed by the operation of valves V9 and V10. On the other hand levels in TANK 1 and TANK 3 are regulated by pumps and by PID controllers. Measurements available from the process through Profibus are levels (h_1 -not used in this experiment-, h_2 and h_3), pump control signals, and the input flow into TANK 3 (Q_P).

4.2 Process communication delays

A PID controller and an I/O card are connected (via PROFIBUS in master-slave communication relations) to an OPC sever. The time server update cycle (Time Rotation Time) is 137ms. The OPC client application has been implemented in a remote PC connected via Ethernet to the OPC server. Schema depicted in figure 1 has been used.

The availability of data in the OPC-client computer has been tested under different conditions of refreshing time (Sample time) and number of devices.

Figure 6a) represents the histograms of a signal delays acquired with a sample time of 100ms. Delays were calculated as the difference between

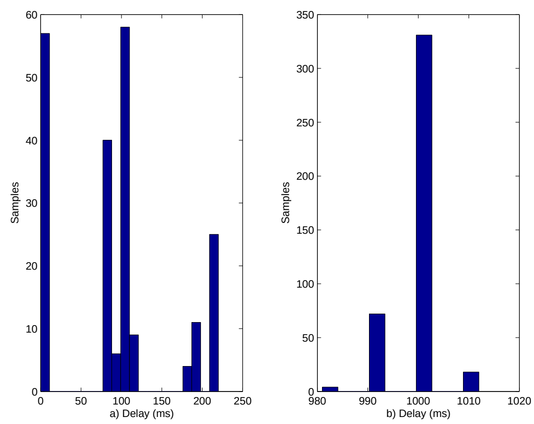

Fig. 6. Histogram of samples delay a)Ts=100ms b)Ts=1000ms.

timestamps labelled by the OPC server. Samples located close to zero denotes repeated data, which means that the server did not have time to update the value. Samples around 100ms represent data arriving on time and samples on the right of the histogram indicate delayed data.

While for histogram of figure 6b), delays are insignificant compared to the sample time (1000ms). In that situation is guaranteed that the refreshing time in the OPC-client computer will coincide with OPC server updates.

It can be concluded that sample time and timestamp not are always coincident and obviously delays and misalignments increase as sample time decreases nearly to the target rotation time. In order to emphasize this source of errors residual computation has been evaluated under these circumstances.

5. DTW FOR IMPROVING RESIDUAL COMPUTATION

According to the design of analytical redundancy relations ARR, that is explained in appendix A, there are three residuals that can be computed for the example described in section 4 (figure 5). This expressions must be evaluated at anytime with values of process variables acquired at the same time. Typically a sampling time is defined to evaluate periodically the consistency of data coming from process. This periodicity also facilitates the computation of derivatives involved in the relations, i.e. ARR2 (eq. A.13). Derivative calculation is done by subtracting the output value at the previous time step from the current value, and dividing by the sample time.

In order to see how misalignments affect residual ARR2, a sample rate of 100ms has been fixed as a periodic interval to compute the redundancy equations. Decision thresholds have been determined as $\bar{x} \pm 3\sigma$, where $\bar{x}$ is the mean and σ is the standard deviation.

Figure 7a) depicts the implementation of eq.A.13 and it can be seen some values that cross the thresholds, which might lead to false alarms.

DTW has to be applied to a couple of signals, for this reason ARR2 was divided in two signals: (Q_{32}) representing the flow between TANK 2 and TANK 3, and the difference between the out put flow (Q_N) and the level variation in TANK 2, that is ($Q_N - A\dot{h}_2$). In that way ARR2 can be computed as follows:

$$DTW(Q_{32}, Q_N - A\dot{h}_2) = 0 \qquad (5)$$

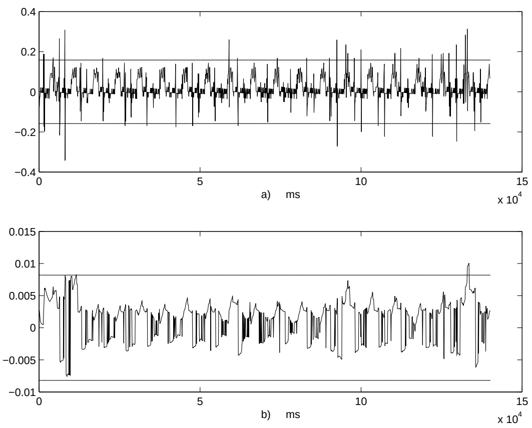

Fig. 7. ARR2: a)eq.A.13 and b)DTW on-line computation.

Figure 7b) shows how DTW reduces the number of false alarms. The window size has been configured in 40 samples.

6. CONCLUSIONS AND FURTHER WORK

In this paper, an approach based on classic DTW was developed to be used online in order to obtain residuals from a laboratory plant. This approach is specially suitable for those errors related with time distortions. Therefore, it will be useful for distributed systems with communication delays and for hybrid systems with on/off sensors or actuators causing misalignments between real and simulated signals.

The results show a hight robustness for on-line DTW. In fact, the results obtained evidenced less false alarms using on-line DTW than a normal implementation.

As restrictions, the new approach continues using dynamic programming and it could be computationally expensive depending on considerations as number of variables, sample time or computer effort.

REFERENCES

Blanke, M., M. Kinnaert, J. Lunze and M. Staroswiecki (2003). *Diagnosis and Fault-Tolerant Control*. Springer Verlag.

Colomer, J., J. Meléndez and F. I. Gamero (2002). Pattern recognition based on episodes and DTW. Application to diagnosis of a level control system. *16th International Workshop on Qualitative Reasoning* pp. 37–43.

Frank, P. M., X. Ding and B. Kppen-Seliger (2000). Current developments in the theory of FDI. *IFAC Safeprocess2000* **1**, 16–27.

Gertler, J. J. (1988). Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine* **8**, 3–11.

Isermann, R. (1993). Fault diagnosis of machines via parameter estimation and knowledge processing. *Automatica* **29**, 815–836.

Lee, D., J. Allan, H. A. Thompson and S. Bennett (2001). Pid control for distributed system with a smart actuator. *Control Engineering Practice* **9**(11), 1235–1244.

Sakoe, H. and S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech, and Signal Proc* **ASSP-26**(1), 43–49.

Silverman, H.F. and D.P. Morgan (1990). The application of dynamic programming to connected speech recognition. *IEEE Signal Processing Magazine* **7**, 6 – 25.

Trevelyan, J. (2004). Industrial process communications and software timing issues. *Mechatronics Design 310: http://www.mech.uwa.edu.au/mechatronics/site-index.html.*

Vitturi, S. (2000). Some features of two feildbuses of the iec 61158 standard. *Computer Standards & Interfaces* **22**, 203–215.

Willing, A. and A. Wolisz (2001). Ring stability of the profibus token-passing protocol over error-prone links. *IEEE Transactions on Industrial Electronics* **48**(5), 1025–1033.

Appendix A. DESIGN OF ANALYTICAL REDUNDANCY RELATIONS

Valves V9 and V10 are located $13cm$ and $7.5cm$, respectively, from the bottom of thanks. Depending on the water levels there exist several different operation modes for the system. As a practical example, level in TANK 3 has been maintained in $20cm$ and TANK 2 in $9cm$. Next equations describe the operation mode for the system:

$$c_1 : Q_L = 0 \tag{A.1}$$

$$c_2 : Q_P = u(t).\bar{Q}_P \tag{A.2}$$

$$c_3 : \dot{h}_3 = \frac{1}{A}(Q_P - Q_L - Q_{32}) \tag{A.3}$$

$$d_4 : \dot{h}_3 = \frac{d}{dt}h_3 \tag{A.4}$$

$$c_5 : Q_{32} = k_1\sqrt{|h_3 - 13|} + k_2\sqrt{|h_3 - h_2|} \tag{A.5}$$

$$d_6 : \dot{h}_2 = \frac{d}{dt}h_2 \tag{A.6}$$

$$c_7 : \dot{h}_2 = \frac{1}{A}(Q_{32} - Q_N) \tag{A.7}$$

$$c_8 : Q_N = k_3\sqrt{h_2} \tag{A.8}$$

$$m_1 : h_3 = h_{3,m_1} \tag{A.9}$$

$$m_2 : h_2 = h_{2,m_2} \tag{A.10}$$

$$m_3 : Q_P = Q_{P,m_3} \tag{A.11}$$

where Q_{32} is the flow between tanks, Q_N is the output flow of the TANK 2, Q_p is the input flow

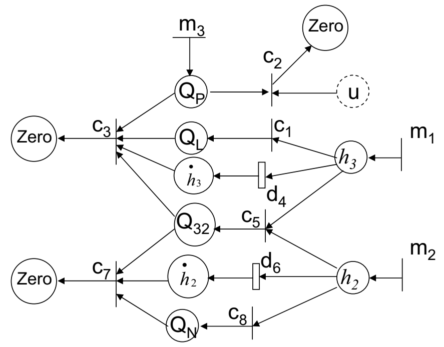

Fig. A.1. Oriented structure graph of the tanks system.

into TANK 3 and Q_L appears when a leakage in TANK 3 occurs. A, k_1, k_2 and k_3 are known parameters. m_1, m_2 and m_3 are additional measurement constraints.

Redundancy relations have been obtained from the following matching table (table A.1) using the ranking algorithm described in (Blanke *et al.*, 2003). The corresponding oriented graph is shown in figure A.1.

Table A.1. Incidence matrix.

↗	Q_L	Q_P	$\dot{h}_3$	h_3	Q_{32}	$\dot{h}_2$	h_2	Q_N
c_1	1							
c_2		1						
c_3	1	1	1		1			
d_4			1	1				
c_5				1	1		1	
d_6						1	1	
c_7					1	1		1
c_8							1	1
m_1				1				
m_2							1	
m_3		1						

Simplifying the operation model equations using the matched variables results in the following redundancy relations:

ARR 1:

$$Q_{P,m_3} - k_1\sqrt{h_3 - 13} - k_2\sqrt{h_3 - h_2} - A\dot{h}_3 = 0 \tag{A.12}$$

ARR 2:

$$k_1\sqrt{h_3 - 13} + k_2\sqrt{h_3 - h_2} - k_3\sqrt{h_2} - A\dot{h}_2 = 0 \tag{A.13}$$

ARR 3:

$$u(t).\bar{Q}_P - Q_{P,m_3} = 0 \tag{A.14}$$