Available online at www.sciencedirect.com
ScienceDirect
Computers in Industry 58 (2007) 531–538
COMPUTERS IN INDUSTRY
www.elsevier.com/locate/compind

Integration of multiple platforms for real-time remote model-based condition monitoring

John Z. Shi [a,*], Fengshou Gu [b], Peter Goulding [b], Andrew Ball [b]

[a] School of Computing and Engineering, The University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK
[b] School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Oxford Road, Manchester M13 9PL, UK

Received 13 October 2006; accepted 16 December 2006
Available online 23 January 2007

Abstract

Model-based condition monitoring has been demonstrated to have superior performance in process condition monitoring. However, it demands increased computational resources to support its more advanced intelligence. This requirement imposes difficulties in integration into an existing control system that typically has resources dedicated solely to control applications. To overcome this problem, this paper presents an integration of multiple platforms, in which tasks are allocated in different machines with different levels of software platform. Time synchronisation between the local and remote devices is implemented by means of Fieldbus network with published-subscriber architecture. A client–server arrangement is used to deal with the data communication between Matlab and Labwindows. A temporary data buffer provides variable time accuracy while using a small fraction of system resources. This approach has been realised in an electro-hydraulic control system, demonstrating the full use of existing rich software resources and the convenience of configuring the systems.

Keywords: Remote condition monitoring; Model-based approach; Real-time systems; WorldFip Fieldbus; Control systems; Distributed monitoring

1. Introduction

After several decades of development, the model-based approach has been shown to be effective in the condition monitoring of a broad array of control systems. By consulting the model prediction, this approach can be configured to be sensitive to changes due to abnormal behaviour but not to those caused by system inputs, so providing enhanced capability to distinguish between faults and acceptable fluctuations in operating conditions [1,2]. The approach permits detection not only of system faults, such as actuator faults and component faults, but most significantly, sensor faults. It is extensively applied in various industrial fields including power plant [3], jet engine [4], and vehicle systems [5].

Published applications have given good address to the model-based approach, the modelling accuracy and the residual robustness [6–8]. Less attention has been paid, however, to practical issues such as real-time implementation especially when advanced software package is used. Reasons for this circumstance include difficulties from either hardware resource conflict or uncertain time delay in data transmission [9–12]. In fact, these practical issues are sometimes decisive in selection of an appropriate monitoring method. For example, when a system is controlled by a computer, it is preferable to integrate the model-based approach with the control task on the same computer. However, the integration will be restricted under certain common conditions. Sometimes, the control program cannot be changed without the help of its supplier, or its change may be subject to burdensome safety procedural requirements. Even when the controller allows the integration, its CPU loading has to trade off the control application with other tasks [13]. The CPU in a real-time control system must read sensor measurement and process it and then send signals to actuators at periodic intervals [14]. If a complex fault diagnostic algorithm is used and the CPU is loaded heavily by the model prediction and the fault diagnosis, the control command may be delayed to send out and the control accuracy will be affected consequently. This is the case especially when non-linearity exists in the process so that a massive computation is required not only for prediction but also for fault diagnosis. While these objections

* Corresponding author. Tel.: +44 1484 472842.
E-mail address: Z.shi@hud.ac.uk (J.Z. Shi).

0166-3615/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.compind.2006.12.002

can be addressed to some extent by coding directly into a real time operating system, this approach carries a burden of expert coding and the greatly enhanced load of error checking when designing and implementing a system at this low level.

A direct way to solve above problem is to use another machine with appreciated resources in parallel with a remote or distributed implementation. Thus the control task remains undisturbed and the condition-monitoring task is taken by a new CPU in a separate machine. Instead of in real-time, databases are often built up through remote acquisition and network transmission [15,16]. Different networks such as WorldFip [11], Ethernet [13], Internet [17], CANbus [18] are used for this purpose. Although real-time computing and communicating techniques are developed rapidly [19], real-time implementations of remote model-based condition monitoring are seldom found so far. This rarity is the major motivation of the investigation in this paper.

In this paper, Matlab platform with advanced toolboxes is used for data processing and fault diagnosis, whilst Labwindows/CVI, a virtual instrument platform, is used for data acquisition and process control. By communication through WorldFip network, a real-time remote model-based condition monitoring schematic is proposed using the multiple platform approach. Firstly, the requirement of the on-line model-based approach is discussed and hardware resource limitations are analysed. Then, the synchronisation requirements are emphasised for the remote generation of the residuals between the real-time measurement and the remote model prediction. To fulfil these requirements, different Fieldbus networks are compared and the WorldFip protocol is chosen to implement the multiple platform approach. Finally, the proposed approach is demonstrated with local execution of control and measurement programs and remote synchronised execution of both model-based residual generation and condition monitoring of a non-linear electro-hydraulic control system.

2. Integration and synchronisation

2.1. Resource restrictions for integration

Typically, in a real-time computer controlled system [19] the computer collects sensor measurements (M) at periodic intervals and sends control signal to the actuator through a pre-designed signal processing (SP) and control algorithm (Ctr). This executing order is illustrated in Fig. 1(a) and its time scheduling is shown in Fig. 1(b). Because every task within this procedure relies on the previous one, the execution order has to be consecutive and the duration of the whole control cycle is the sum of the execution time of these tasks. The computer's resources are usually designed sufficient to provide a spare time (ST) to execute additional tasks. A model-based approach can be integrated with the control procedure if the model prediction (MP) and the condition monitoring (CM) algorithms satisfy the necessary computation conditions such that the ratio between computing time and its period is less for every task and the sum of the ratio over all tasks is also less than one. However, the model prediction, residual generation, and advanced condition monitoring techniques normally involve a long period of execution time. The utilisation of Matlab often makes the execution even longer due to the loading of necessary toolboxes to maintain its powerful data processing capability. What is the most important is that the generation of residuals requires synchronisation of the measurement and the model prediction. This means that the model prediction and the residual generation cannot be set for lower priority with respect to the control task. Consequently, the whole control cycle has to be prolonged and control accuracy will be affected if the monitoring task (which may be of unpredictable length) takes too long a time. This situation is termed a resource conflict, and is depicted in Fig. 2.

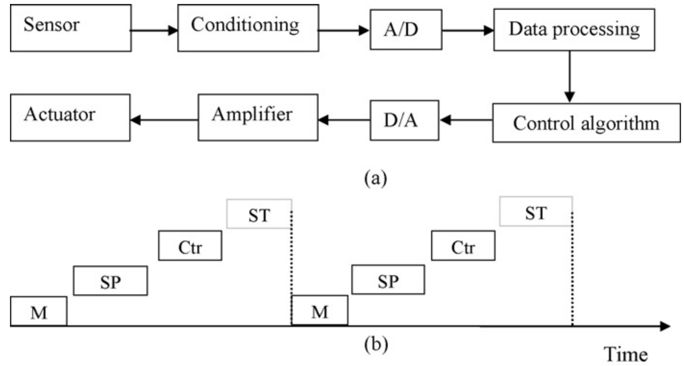

Fig. 1. Real-time computer control: (a) control order and (b) time scheduling.

2.2. Synchronisation requirements

As noted above, remote condition monitoring provides more computation resources and executes tasks within different machine/PCs. These two machines are connected using a communication network, as shown in Fig. 3(a). In the local control machine, control and measurement tasks are executed to ensure a critical time cycle for desired control accuracy. The measurement is processed and then fed back into the control algorithm. At the same time, the measured data is also uploaded onto the Fieldbus network, which transmits the data to the remote machine. The remote machine then processes the data and compares it to the model-prediction to generate a residual signal for condition monitoring.

Fig. 3(b) shows the time scheduling that uses the remote arrangement to overcome the conflict as described in Fig. 2. Real-time control and measurement is maintained in the local machine, which devotes a small part of its spare time to the data

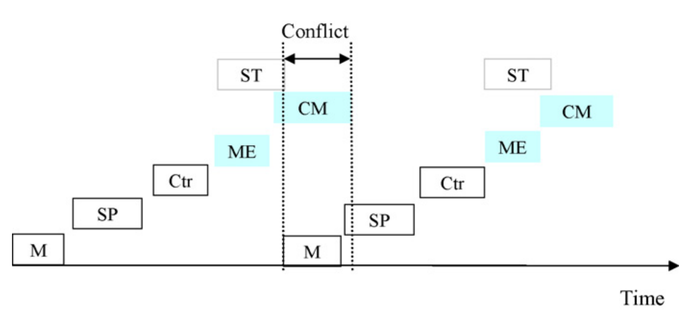

Fig. 2. Time scheduling for integration control and monitoring.