Advances in Knowledge-Based Technologies

Proceedings of the
Master and PhD Seminar
Winter term 2013/14, part 2

Softwarepark Hagenberg
SCCH, Room 0/2
5 February 2014

Software Competence Center Hagenberg
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz
Softwarepark 21
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

Session 1. Chair: Bernhard Moser

9:00 Jorge Martinez-Gil:

MaF: An Ontology Matching Framework

9:30 Georgios Chasparis:

Design and Implementation of Distributed Resource Management for Time Sensitive Application

Session 2. Chair: Susanne Saminger-Platz

10:15 Werner Reisner:

Models and Analysis of Visual Discomfort Measures for Stereoscopic Images

10:45 Kurt Pichler:

Monitoring System for Reciprocating Compressor Valves (Part 1)

11:15 Francisco Serdio:

Gradient-based Fault Isolation

MaF: An Ontology Matching Framework

Jorge Martinez-Gil, Software Competence Center Hagenberg (Austria)

email: jorge.martinez-gil@scch.at

Keywords: Knowledge Integration, Ontology Matching, Semantic Similarity Measures

Abstract

We present our experience when developing the Matching Framework (MaF), a framework for matching ontologies that allows users to configure their own ontology matching algorithms and it allows developers to perform research on new complex algorithms. The framework can be configured by selecting the simple algorithms which will be used from a set of 136 basic algorithms, indicating exactly how many and how these algorithms will be composed and selecting the thresholds for retrieving the most promising mappings. Output results are provided in a standard format so that they can be used in many existing tools (evaluators, mediators, viewers, and so on) which follow this standard. The main goal of our work is not to improve the existing solutions for ontology matching, but to help research new ways of combining algorithms in order to meet specific needs. In fact, the system can test more than $6 \cdot 136!$ possible combinations of algorithms, but the graphical interface is designed to simplify the matching process.

Design and Implementation of Distributed Resource Management for Time Sensitive Applications*

Georgios Chasparis[†] M. Maggio[‡] E. Bini[§] Karl-Erik Årzén[¶]

June 4, 2013
February 2, 2014 (revised)

Abstract

When several applications are running over the same computing platform, requiring more computing capacity than the available one, two actions should be taken: (i) the applications should reduce their computational demand, possibly degrading the delivered performance (*service level adaptation*), and (ii) the available computing capacity should be properly distributed among the competing applications (*resource adaptation*). The problem of coordinating these actions (namely, *resource management of service level aware applications*) is often solved by centralized optimization schemes. The centralized approach, however, exhibits several problems, e.g., it does not scale with the number of applications and it requires applications to share internal information with the resource manager.

In this paper, we address the problem of distributed convergence to fair allocation of CPU resources for time-sensitive applications. We propose a novel resource management framework where both the applications and the resource manager act independently trying to maximize their own performance measure according to a utility-based adjustment process. Contrary to prior work on centralized optimization schemes, the proposed framework exhibits adaptivity and robustness to changes both in the number and nature of applications, while it assumes minimum information available to both applications and the resource manager. We validate our framework with simulations using the TrueTime toolbox in MATLAB/Simulink.

Keywords: Resource Management; Distributed Optimization; Real-Time Systems

1 Introduction

The current trend in embedded computing demands that the number of applications sharing the same execution platform increases. This is due to the increased capacity of new hardware

*The research leading to these results was supported by the Linneaus Center LCCC, the Swedish VR project n.2011-3635 "Feedback-based resource management for embedded multicore platform", and the Marie Curie Intra European Fellowship within the 7th European Community Framework Programme. An earlier version of parts of this paper appeared in [4] and its implementation framework appeared in [12].

†Software Competence Center Hagenberg GmbH, Softwarepark 21, A-4232 Hagenberg, Austria, georgios.chasparis@scch.at.

‡Department of Automatic Control, Lund University, Sweden, e-mail: martina.maggio@control.lth.se.

§Department of Automatic Control, Lund University, Sweden, e-mail: bini@control.lth.se.

¶Department of Automatic Control, Lund University, Sweden, e-mail: karlerik@control.lth.se.

platforms, e.g., through the use of multi-core techniques. An example includes the move from federated to integrated system architectures in the automotive industry [6].

With a larger number of applications, the need for better mechanisms for controlling the rate of execution of the applications becomes apparent. To this end, virtualization or resource reservation techniques [1, 13] are used. According to these techniques, each reservation is viewed as a *virtual processor* (or *platforms*) executing at a fraction of the speed of the physical processor, i.e., the *bandwidth* of the reservation, while the tasks in the different reservations are temporally isolated from each other.

An orthogonal dimension along which the performance of an application can be tuned is the selection of its *service level*. It is assumed that an application is able to execute at different service levels, where a higher service level implies a higher quality-of-service (QoS). Examples include the adjustable video resolutions, and the adjustable sampling rates of a controller.

Typically this problem is solved by using a *resource manager* (RM), which is in charge of: (a) *assigning virtual processors to the applications*, (b) *monitoring the use of resources*, and (c) *assigning the service level to each application*. The goal of the RM is to maximize the overall delivered QoS. This is often done through *centralized optimization* and the use of *feedback* from the applications.

RM's that are based on the concept of feedback, monitor the progress of the applications and adjust the virtual platforms based on measurements [7, 16]. In these early approaches, however, quality adjustment was not considered. Reference [5] proposed an inner loop to control the resource allocation nested within an outer loop that controls the overall delivered quality.

Optimization-based resource managers have also received considerable attention [11, 14]. These approaches, however, rely on the solution of a centralized optimization that determines *both* the amount of assigned resources and the service levels of all applications [3, 14, 15]. In the context of networking, reference [10] models the service provided by a set of servers to workloads belonging to different classes as a utility maximization problem. However, there is no notion of adjustment of the service level of the applications.

An example of combined use of optimization and feedback was developed in the ACTORS project [2, 3]. In that project, applications provide a table to the RM describing the required amount of CPU resources and the expected QoS achieved at each supported service level [2, 3]. In the multi-core case, applications are partitioned over the cores and the amount of resources is given for each individual partition. Then, the RM decides the service level of all applications and how the partitions should be mapped to physical cores using a combination of Integer Linear Programming (ILP) and first-fit bin-packing.

On-line centralized optimization schemes have several weaknesses. First, the complexity of the solvers used to implement the RM (such as ILP solvers) grows significantly with the number of applications. It is impractical to have a RM that optimally assigns resources at the price of a large consumption of resources by the RM itself. Second, to enable a meaningful formulation of a cost function in such optimization problems, the RM must compare the quality delivered by different applications. This comparison is unnatural because the concept of quality is extremely application dependent. Finally, a proper assignment of service levels requires application knowledge. In particular, applications must inform the RM about the available service levels and the expected consumed resources at each service level, and they may be unable or unwilling to provide such information. Thus, service levels would better be assigned by the applications rather than the RM.

To this end, distributed optimization schemes have recently attracted considerable attention. Reference [17] considered a cooperative game formulation for job allocation to several service providers in grid computing. Reference [18] proposed a non-cooperative game-theoretic formulation to allocate computational resources to a given number of tasks in cloud computing. Tasks have full knowledge of the available resources and try to maximize their own utility function. Similarly, in [8] the load balancing problem is formulated as a non-cooperative game.

In this paper, the problem differs significantly from the grid computing setup of [17] or the load balancing problem of [8, 18] in cloud-computing services. In particular, we are concerned with the problem of allocating CPU resources among several applications, while applications are able to adjust their own service levels. Under the proposed scheme, both applications and the RM act independently trying to maximize their own performance measure (utility) according to a utility-based adjustment process. Naturally, this framework can be interpreted as a strategic-interaction (or game) among the applications and the RM. It is shown analytically that *fair* resource allocation can be achieved in a distributed fashion through the proposed adjustment process when the CPU is overloaded.

The paper builds upon previous contributions [4, 12]. In comparison to [4], this paper extends its theoretical contributions by addressing global convergence and asynchronous updates. We also demonstrate several experiments using the TrueTime MATLAB toolbox [9] in the case of both synchronous and asynchronous applications. Instead, in [12], we discuss the full implementation framework in Linux.

References

[1] L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *Proc. of the* 19^{th} *IEEE Real-Time Systems Symposium*, pages 4–13, Madrid, Spain, Dec. 1998.

[2] K.-E. Årzén, V. Romero Segovia, S. Schorr, and G. Fohler. Adaptive resource management made real. In *Proc. 3rd Workshop on Adaptive and Reconfigurable Embedded Systems*, Chicago, IL, USA, Apr. 2011.

[3] E. Bini, G. C. Buttazzo, J. Eker, S. Schorr, R. Guerra, G. Fohler, K.-E. Årzén, R. Vanessa, and C. Scordino. Resource management on multicore systems: The ACTORS approach. *IEEE Micro*, 31(3):72–81, 2011.

[4] G. Chasparis, M. Maggio, K.-E. Årzén, and E. Bini. Distributed management of CPU resources for time-sensitive applications. In *Proc. of the 2013 American Control Conference*, 2013.

[5] T. Cucinotta, L. Palopoli, L. Abeni, D. Faggioli, and G. Lipari. On the integration of application level and resource level QoS control for real-time applications. *IEEE Transactions on Industrial Informatics*, 6(4):479–491, Nov. 2010.

[6] M. Di Natale and A. Sangiovanni-Vincentelli. Moving from federated to integrated architectures in automotive: The role of standards, methods and tools. *Proc. of the IEEE*, 98(4):603–620, 2010.

[7] J. Eker, P. Hagander, and K.-E. Årzén. A feedback scheduler for real-time controller tasks.

Control Engineering Practice, 8(12):1369–1378, Jan. 2000.

[8] D. Grosu and A. T. Chronopoulos. Noncooperative load balancing in distributed systems. *Journal of Parallel and Distributed Computing*, 65(9):1022–1034, 2005.

[9] D. Henriksson, A. Cervin, and K.-E. Årzén. TrueTime: Simulation of control loops under shared computer resources. In *Proc. of the* 15^{th} *IFAC World Congress on Automatic Control*, Barcelona, Spain, July 2002.

[10] B. Johansson, C. Adam, M. Johansson, and R. Stadler. Distributed resource allocation strategies for achieving quality of service in server clusters. In *Proc. of the* 45^{th} *IEEE Conference on Decision and Control*, pages 1990–1995, Dec. 2006.

[11] C. Lee, J. P. Lehoczky, D. Sieworek, R. Rajkumar, and J. Hansen. A scalable solution to the multi-resource QoS problem. In *Proc. of the* 20^{th} *IEEE Real-Time Systems Symposium*, pages 315–326, Phoenix, AZ, Dec. 1999.

[12] M. Maggio, E. Bini, G. Chasparis, and K.-E. Årzén. A game-theoretic resource manager for RT applications. In *25th Euromicro Conference on Real-Time Systems, ECRTS13*, Paris, France, July 2013.

[13] C. W. Mercer, S. Savage, and H. Tokuda. Processor capacity reserves: Operating system support for multimedia applications. In *Proc. of IEEE International Conference on Multimedia Computing and Systems*, pages 90–99, Boston, MA, U.S.A., May 1994.

[14] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. A resource allocation model for QoS management. In *Proc. of the IEEE Real Time System Symposium*, 1997.

[15] M. Sojka, P. Píša, D. Faggioli, T. Cucinotta, F. Checconi, Z. Hanzálek, and G. Lipari. Modular software architecture for flexible reservation mechanisms on heterogeneous resources. *Journal of Systems Architecture*, 57(4):366–382, 2011.

[16] D. C. Steere, A. Goel, J. Gruenberg, D. McNamee, C. Pu, and J. Walpole. A feedback-driven proportion allocator for real-rate scheduling. In *Proc. of the* 3^{rd} *Symposium on Operating Systems Design and Implementation*, Feb. 1999.

[17] R. Subrata, A. Y. Zomaya, and B. Landfeldt. A cooperative game framework for QoS guided job allocation schemes in grids. *IEEE Transactions on Computers*, 57(10):1413–1422, Oct. 2008.

[18] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *The Journal of Supercomputing*, 54(2):252–269, Nov. 2010.

Models and Analysis of Visual Discomfort Measures for Stereoscopic Images

Werner Reisner
Software Competence Center Hagenberg
{werner.reisner} @ scch.at

Abstract — Looking at stereoscopic three-dimensional (3D) pictures and watching 3D movies can provide immersive viewing experience. 3D contents have not only the benefit of the enhanced depth perception, but could also enhance the perception of sharpness, sense of presence and naturalness [6].

The large availability of 3D displaying devices such as 3D-TVs as well as the upcoming trend towards glasses-free 3D mobiles and holographic 3D projection systems drastically raise the demand for good 3D film content. In contrast to 2D film production, 3D film acquisition requires two cameras for generating two stereoscopic views. The higher complexity of a stereoscopic 3D film production workflow typically results in more time- and cost-intensive production and post-production processes. One goal of the complicated post-production steps is to guarantee best quality of experience (QoE) for the viewer. This includes also aspects of safety and health, since some viewers experience visual discomfort (VDC) when looking at stereoscopic displays.

Eye strain, headache, and nausea are only some physical symptoms reported from people who watched stereoscopic 3D contents [4]. In additional bad-quality 3D stereoscopic content can cause permanent damage to the visual system of children [2].

Many research has been done to identify the reasons of visual discomfort. Because several authors have acknowledged that it is "essential that stereoscopic content must be evaluated in terms of evaluation metrics that reflect the full extent of viewers experience" [1], VC-measures have been developed which can predict the level of discomfort associated with stereoscopic contents [1], [5], [3]. Based on research in computer vision, machine learning and subjective assessment methods VC-measures can facilitate the post-production process of 3D-content production and can help to provide good quality stereoscopic contents. Therefore it is important to develop visual discomfort predicting systems which make use of VC-measures. Such systems should be based on state-of-the-art research to guarantee best prediction performance.

To identify the main modules of VC-measures it is important to define a clear structure of the main parts of VC-measures as well as to compare them on a very basic level. This work attempts to identify the main modules of state-of-the-art VC-measures and compares some of them using methods from computer vision, machine learning and statistics. The proposed structures and methods should help new researchers to identify algorithms which can be used for developing stereoscopic contents which cause a low level of visual discomfort. The revue of some explicit measures also gives a small overview of current research in that field.

References

[1] Marc Lambooij et al. "Visual discomfort of 3D TV: Assessment methods and modeling". In: *Displays* 32.4 (2011), pp. 209–218.

[2] David A Goss and Huifang Zhai. "Clinical and laboratory investigations of the relationship of accommodation and convergence function with refractive error". In: *Documenta ophthalmologica* 86.4 (1994), pp. 349–380.

[3] Donghyun Kim and Kwanghoon Sohn. "Visual Fatigue Prediction for Stereoscopic Image". In: *IEEE Trans. Circuits and Systems for Video Technology* 21.2 (2011), pp. 231–236.

[4] Marc Lambooij et al. "Visual Discomfort and Visual Fatigue of Stereoscopic Displays: A Review". In: *Journal of Imaging Science and Technology* 53.3 (2009).

[5] Hosik Sohn et al. "Predicting Visual Discomfort Using Object Size and Disparity Information in Stereoscopic Images". In: *IEEE Trans. Broadcasting* 59.1 (2013), pp. 28–37.

[6] Wa James Tam et al. "Stereoscopic 3D-TV: Visual Comfort". In: *IEEE Trans. Broadcasting* 57.2 (2011), pp. 335–346.

Monitoring System for Reciprocating Compressor Valves – A data-driven Approach

Kurt Pichler, PhD-Thesis

Reciprocating compressors are heavily used in modern industry, for instance for gas transportation and storage. In many cases, compressors run at high capacity and without backup. Hence unexpected shutdowns lead to large losses in productivity. Furthermore, there is an economic trend towards saving labor costs by reducing the frequency of on-site inspection. Such considerations mean that compressors are run by remote control stations and monitored by automated technical systems. In this case, the system must be able to retrieve and evaluate relevant information automatically to detect faulty behavior.

The state of the art solutions for reciprocating compressor valve fault detection are designed for constant load and pressure conditions. For changing conditions or different valve types, operators adapt the threshold values manually. Since modern reciprocating compressors are controlled by reverse flow control systems, changing load and pressure are not unusual, and the fault detection methods have to cope with that fact. In this thesis, two independent novel approaches are presented: one evaluates measurements of accelerometers, the other one cylinder pressure measurements. Both methods can handle the tasks mentioned above, i.e. varying load and pressure conditions and different valve types. Two different approaches were developed because existing compressors are equipped with different sensing systems. Hence it is easier to upgrade the monitoring system without mounting additional sensors.

The first method evaluates time-frequency representations (spectrograms) of accelerometer measurements at the valve covers. Based on previous publications, we know that a cracked or broken valve influences the amplitudes of the power spectrum in certain frequency bands. Furthermore, it is obvious that the load control system changes the timing of the valve events. Of course, both factors are reflected in the spectrogram. Keeping that knowledge in mind, we have a look at the point-wise difference of a faultless reference spectrogram and a test spectrogram. Depending on the fault state of the valve and the load levels, it shows specifically shaped structures (Fig. 1 and Fig. 2). The positions of the structures within the spectrogram are varying unpredictably with the valve type and the load. Hence, an automated detection would be hard to realize. Additionally, measurement noise makes the detection even more difficult. Both problems can be solved by applying two-dimensional autocorrelation to the point-wise spectrogram difference: the significant structures are centered and the noise effects are reduced. Thus makes it easier to define features that characterize the specific patterns. For example, Fig. 3 shows the autocorrelation for a faultless test spectrogram, but with changing load. In contrast, Fig. 4 shows the autocorrelation for a test spectrogram measured from a valve with a fissure. The different shapes can be seen clearly.

We tested the method with numerous real world test measurements. The measurements were recorded with different valve types, different accelerometers and with constant as well as varying load conditions. All of the tests proved the ability of the method to detect cracked and broken valves, cross validation using SVM classification shows very high classification accuracy.

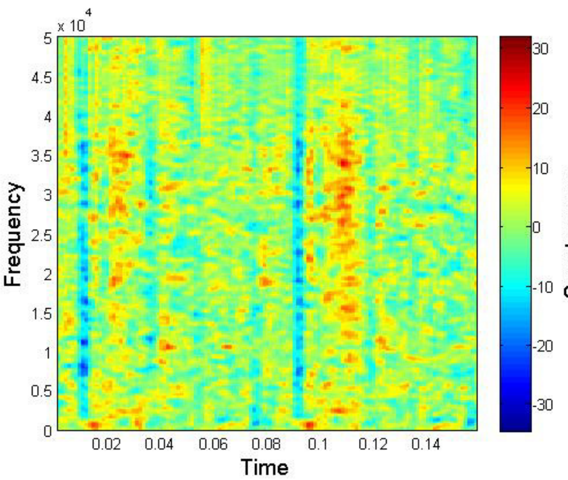

Fig. 1: Point-wise spectrogram difference for a test spectrogram from a faultless valve

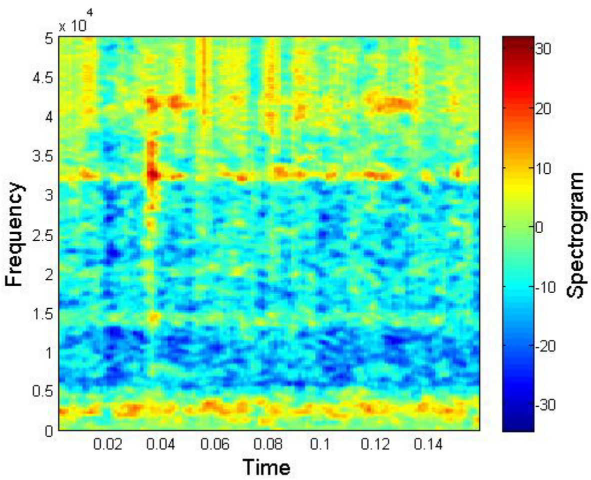

Fig. 2: Point-wise spectrogram difference for a test spectrogram from a cracked valve

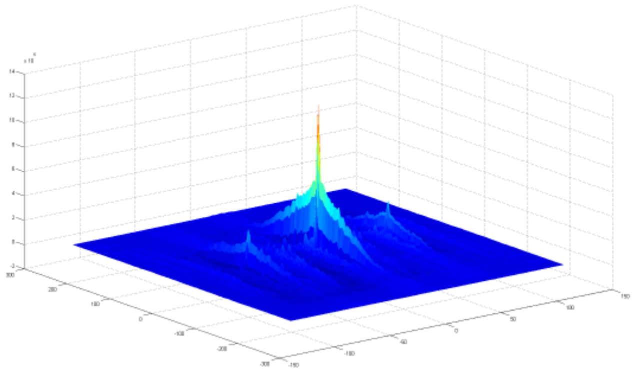

Fig. 3: Autocorrelation representation for a test measurement from a faultless valve

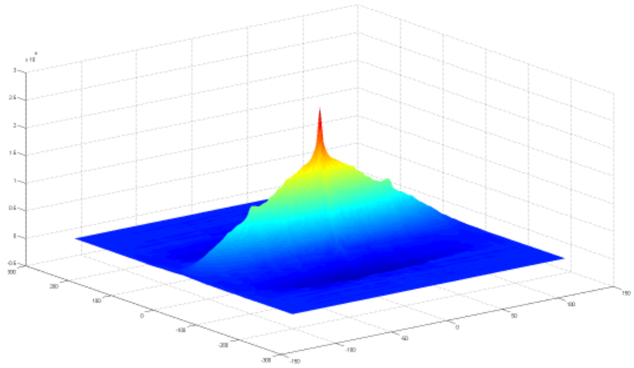

Fig. 4: Autocorrelation representation for a test measurement from a cracked valve

The second method evaluates cylinder pressure measurements in the shape of pV diagrams. When a valve of a compressor breaks, there is a leak and gas can flow through the closed valve. Of course, this affects the shape of the pV diagram of a compression cycle significantly. The pV diagram is used to describe corresponding changes in volume and pressure in a system. As the load control affects mainly the compression phase of a compression cycle, we concentrate on the evaluation of the expansion phase. This leads to a load independent method.

In the case of a broken discharge valve, the pressure in the cylinder decreases slower during expansion than in the faultless case. The reason for that is that gas flows through the closed valve from the discharge chamber into the cylinder. In the case of a broken suction valve, gas flows through the suction valve from the cylinder into the suction chamber resulting in a faster decreasing cylinder pressure. To quantify this difference, we linearize the pV diagram by switching to logarithmic scales. Then we can easily use the gradient of the expansion phase as an indicator for pressure reduction velocity in the compression cylinder during expansion (Figure 5). Since the pressure reduction velocity is also affected by the pressure conditions (suction and discharge pressures) we have to consider the pressure conditions in the feature space as well (Figure 6). But even faultless valves are not 100% leak tight. Depending on the valve type (design and material), they have different leakage factors. This is reflected in an offset in the feature space. The features can be classified using SVM classification. We validated the proposed method with real world data from a reciprocating compressor test bench. Compared with another feature extraction method from pV diagrams proposed in literature (F. Wang, L. Song, L. Zhan and H. Li, 2010, "Fault diagnosis for reciprocating air compressor valve using p-V indicator diagram and SVM"), our features show higher validation accuracy, especially in the case of small faults.

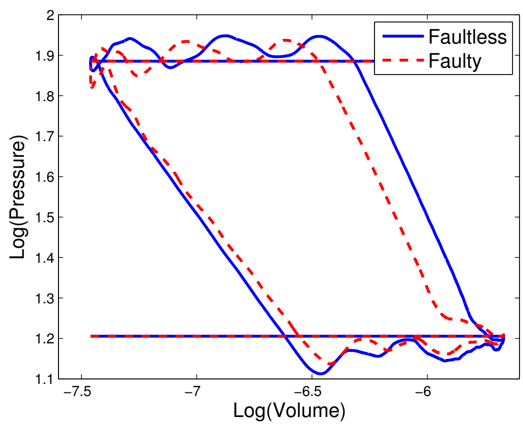

Figure 5. Logarithmic pV diagrams of measurements from faultless and faulty valves

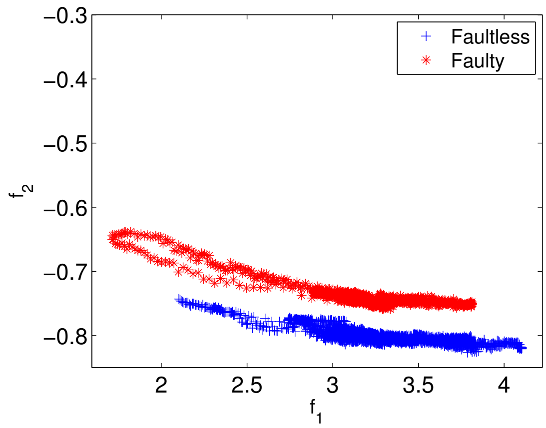

Figure 6. Features extracted from the logarithmic pV diagrams

Gradient-based Fault Isolation

Francisco Serdio[1], Edwin Lughofer[1]

[a] *Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria*

Abstract

Together with the great importance of Fault Detection (FD) -determine whether a system is experienced a fault condition- it is the Fault Isolation (FI) -determine who is responsible for the fault. In cases where there is no expert knowledge, there is still room to perform Fault Isolation. That is the case of our Gradient-based Fault Isolation technique, where an incoming fault can be isolated by means of analyzing how the models are impacted by such a fault. This impacts, reflected along the warning models by its gradients, will allow to revert back to the original variables, thus determining who is the root cause of the incoming fault, meaning fault isolation. Gradients, obtained using partial derivatives of the system models, will determine the relative importance of the corresponding variables (channels) on that function. A proper aggregation of these gradients lead to a voting system where a variable or a set of variables are signaled as the candidate variables experimenting the fault in the system. Since partial derivatives can be computed either by the analytical expression of the models or using numerical tricks, we have introduced a technique applicable to fault detection residual-based systems, where the isolation can be performed regardless the models used during the FD stage. We also introduce tools to measure the Fault Isolation capabilities numerically and graphically, allowing to compare Fault Isolation against Fault Detection capabilities as well.

Advances in Knowledge-Based Technologies *February 5, 2014*