Final Report - Development of a Probabilistic Technique for On-line Parameter and State Estimation in Non-linear Dynamic Systems

DOE Project DE-FG07-98ID13639

Principal Investigator: Tunc Aldemir, The Ohio State University

Co-Principal Investigators: Don W. Miller and Brian K. Hajek, The Ohio State University

Report Period: August 1, 1998 - July 31, 2001

Below is the copy of the paper that was presented at the American Nuclear Society 2001 Annual Meeting, June 17-21, 2001, Milwaukee, Wisconsin whose abstract appeared in *Trans. Am. Nucl. Soc.*, **84**, 109-110 (June 2001). The paper summarizes the project accomplishments which fulfilled all the project objectives. It was indicated by the session organizer Kathy McCarthy that the paper can also serve as the final project report. An interactive version of the software that is the final deliverable can be accessed at <http://rclsgi.eng.ohio-state.edu/courses/Aldemir/DSDInteractive/index2.htm>

Parameter and State Estimation Using DSD

Tunc Aldemir, Peng Wang, Don W. Miller

The Ohio State University, Nuclear Engineering Program,
206 West 18th Avenue, Columbus, Ohio 43210

1. INTRODUCTION

The DSD (Dynamic System Doctor) is a system-independent, interactive software under development for on-line state/parameter estimation in dynamic systems [1], partially supported through a Nuclear Engineering Education (NEER) grant during 1998 - 2001. This paper summarizes the recent accomplishments in improving the user-friendliness and computational capability of DSD.

2. BACKGROUND

The DSD estimation algorithm uses a recursive technique based on the representation of the system dynamics in terms of transition probabilities between user specified cells that partition the system parameter/state space during user specified time intervals $k \leq t \leq (k+1)T$ ($k = 0, 1, 2, \dots$). The DSD obtains these probabilities from the user supplied system model and the data obtained from the process computer at each kT . The distinguishing features of DSD are the following:

1. *The DSD does not use linearization.* The transition probabilities between user specified cells can be generated from a variety of model types including differential equations, difference equations, neural nets and hard-wired codes that may use tabular data. In contrast, most of the methods proposed for state and parameter estimation in non-linear systems use a system model linearized about an operating point which may have a limited validity range and may not converge in the presence of a large amount of noise.
2. *The DSD is naturally suited to accommodate noise and modeling uncertainties.* The DSD represents system evolution in terms of transition probabilities between user specified cells which can be chosen to envelope the noise in the dynamical variables or the uncertainties in the system parameters in general. Among the variety of existing parameter estimation methods, the least-square estimation (LSE) seems to be one of the most used to handle noise. However, the existing LSE approaches may: a) be limited to low noise-to-signal ratios, b) contain discontinuities at the measurement points, c) require preprocess filtering, d) not converge when correlated noise is present, e) require large memory and/or lead to high computational load, and, f) not be able to handle modeling uncertainties.
3. *The DSD estimation algorithm is system independent.* This aspect of DSD is perhaps its most practically useful feature, since the estimation algorithm is transparent to the user. Conventional approaches to non-linear state/parameter estimation generally require system specific algorithms that need to be designed by the user. However, the choice of the

optimum partitioning scheme requires system-specific experimentation and hence the DSD has to be "tuned" for a given system.

4. The DSD yields both the ranges and probability distributions of the uncertainty in the estimated parameters/variables. This feature of DSD allows calculating the expected value of the estimated variable/parameter in addition to the uncertainty range. Such a capability has not been encountered in the literature for any other parameter/state estimation technique.

The DSD has been developed in C++ to run under NT due to the widespread use of NT and consists of a parent process and a number of child processes. Each child process (task) corresponds to a system whose parameters/unmonitored state variables are being tracked. The DSD is capable of handling several systems simultaneously. To date, the DSD algorithm has been successfully applied for fault detection and identification in automotive engines [2], estimation of time varying reactivity in the presence of noise [3], reactivity ramp rate estimation [4], state/parameter estimation in boiling water reactor dynamics [5], real time xenon estimation in nuclear power plants [6,7] and for state/parameter estimation in a number of mechanical and control systems [8,9]. The system model types used in these applications include differential equations [2-6,8,9], difference equations [2] and neural nets [7].

3. RECENT ACCOMPLISHMENTS

During the past 3 years several accomplishments have been made in improving the user-friendliness and computational capability of DSD. The major ones are described in Sections 3.1 - 3.4 below.

3.1 Development of User Interface Module for Model and Partitioning Data Input

The user interface module is the entry point into DSD from NT. This module sets up the frame window and the client window that the child windows reside in and also accepts the user's instructions via a clickable menu and converts them into a series of actions. The menu items allow the user to: a) edit an existing system model or create a new system model (currently assumed to consist of a set of first order differential equations), b) define the partitioning scheme to be used, c) create a new task by compiling the system model and the partitioning data along with the appropriate library functions and linking it with the estimation algorithm, and, d) start, stop or pause the execution of a task. The user interface module also has a simulator menu that facilitates testing of the estimation algorithm for a given model and partitioning scheme.

3.2 Development of Multi-Threading Capability

The multi-threading capability of DSD allows the execution of multiple tasks. Once a child process (i.e. task) is set up and activated by DSD, it runs independently of all the other child processes. The communication module is responsible for establishing this dynamic data exchange and other data exchanges between the parent process and the child processes. Each child process is controlled by its own timer signal, generated at each kT . Each child process has a data collection child window (DCCW) and an estimation child window (ECW) that are also run as separate threads. Upon the receipt of the timer signal at each kT , the DCCW transfers the monitored data

from the process computer to a buffer. The ECW; a) reads the monitored data at $t = (k + 1)\Delta t$ from the buffer, b) runs the estimation algorithm to obtain the probability distribution of the parameters/unmonitored variables, and, c) pipes the results to the communication module. The communication module passes the results to the graphics module which can display at each time point k : a) the magnitudes of the monitored variables, b) the estimated ranges for the unmonitored parameters/system variables, and, c) the probability distribution of the unmonitored parameters/system variables over these ranges.

3.3 Dynamic Partitioning

Substantial memory savings have been obtained by carrying out estimation process is the only in the space of the unmonitored variables/parameters rather than the whole state space. If the system has L state variables out of which $l = 1, 2, \dots, i$ have to be kept track of and if the range of interest for each variable is partitioned into $J_l (l = 1, 2, \dots, L)$ intervals, then only $\prod_{l=1}^i J_l$ cells need to be used with dynamic partitioning instead of $\prod_{l=1}^L J_l$ cells. For the xenon estimation problem of [6], such a dynamic partitioning leads to a 400-fold reduction in memory requirements.

3.4 Recursive Partitioning

Another way to reduce memory requirements and computation time is to start with a coarse partitioning scheme and progressively refine the partitioning scheme upon convergence until the desired refinement in the parameters/variable is reached (e.g. based on the noise in the monitored data), rather than to use this final level of refinement from the beginning. The recursive scheme also avoids the need to "tune" DSD for the system of interest by the user (see Item 3 in Section 1), increasing the user-friendliness of DSD.

The dynamic and recursive partitioning schemes have been tested on the xenon estimation problem of [6] but not yet have been integrated into the DSD software. Other accomplishments in improving the user-friendliness and computational capability of DSD include development of: a) self-installment and de-installment capability, b) pop-up windows for exception handling, and, c) adaptive integration scheme for the generation of cell-to-cell transition probabilities. The DSD was also extensively tested for software reliability.

4. CONCLUSION

The DSD is currently capable of tracking several parameters/state variables per task (or system) for on-line monitoring and multiple tasks can be run. The algorithm is transparent to the user. The model and data input are accomplished through dialogue windows. The software is highly transportable due to its self-installment/de-installment capability. A current practical implementation is global xenon estimation in nuclear reactors in real-time. Future applications will include in-core power monitoring in nuclear reactors using constant-temperature power sensors [10].

REFERENCES

1. X.M. Chen, P. Wang, T. Aldemir, "DSD: A Parameter/State Estimation Tool for Model-based Fault Diagnosis in Non-linear Dynamic Systems", *Dynamic Reliability : Future Directions*, C. Smidts, J. Devooght, P. E. Labeuau (Eds), 251-276, International Workshop Series on Advanced Topics in Reliability and Risk Analysis, Center for Reliability Engineering, University of Maryland(October 2000)
2. L. Dinca, T. Aldemir, G. Rizzoni, "A Model-Based Probabilistic Approach for Fault Detection and Identification with Application to the Diagnosis of Automotive Engines", *IEEE Trans. Automatic Control*, **44**, 2200-2205 (November 1999)
3. L. Dinca, T. Aldemir, "Estimation of Time-Varying Reactivity in the Presence of Noise", *Trans. Am. Nucl. Soc.* , **76** , 164-165(June 1997)
4. P. Wang, T. Aldemir, D. W. Miller, "Reactivity Ramp Rate Estimation Using A Probabilistic Approach", *Trans. Am. Nucl. So c.* , **78** , 121-123 (June 1998)
5. L. Dinca, T. Aldemir, "State/Parameter Estimation in BWR Dynamics", *Trans. Am. Nucl. Soc.* , **80** , 124-126 (June 1999)
6. P. Wang, T. Aldemir, "Real Time Xenon Estimation in Nuclear Power Plants", *Trans. Am. Nucl. Soc.* , **81** , 154-156 (November 1999)
7. M. Gadioli, T. Aldemir, E. Zio, M. Marseguerra, "Accelerating DSD Using A Neural Net System Model", *Trans. Am. Nucl. Soc.* , **82** , 137-139 (June 2000)
8. L. Dinca, T. Aldemir, "Parameter Estimation Towards Fault Diagnosis in Non-Linear Systems Using a Markov Model of System Dynamics", *Nucl. Sci. Eng.* , **127** , 199-219 (October 1997)
9. T. Aldemir, P. Wang, "The Use of the Cell-to-Cell Mapping Technique as a Model-Based Diagnostic Tool", *M&C'99 - Madrid* , J. M. Aragonés, C. Ahnert, O. Cabellos (Eds.), 1330-1338, Senda Editorial, S.A., Madrid, Spain (1999) (invited paper)
10. I. Munteanu, H.B. Zhou, T. D. Radcliff, T. Aldemir, D. W. Miller, "In-Core Power Detection Using DSD", *Trans. Am. Nucl. Soc.* , **83** , 279-281 (November 2000)