(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 April 2008 (10.04.2008)

PCT

(10) International Publication Number
WO 2008/042807 A2

(51) International Patent Classification:
G05B 23/02 (2006.01)

(21) International Application Number:
PCT/US2007/079986

(22) International Filing Date:
28 September 2007 (28.09.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/827,631 29 September 2006 (29.09.2006) US

(71) Applicant *(for all designated States except US)*: **FISHER-ROSEMOUNT SYSTEMS, INC.** [US/US]; 12301 Research Blvd., Research Park Plaza, Bldg. III, Austin, TX 78759 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **MILLER, John, Philip** [US/US]; 8473 Hiawatha Ave., Eden Prairie, MN 55347 (US). **LUNDEBERG, Marcus, Richard** [US/US]; 4531 Heritage Hills Circle, Bloomington, MN 55437 (US).

(74) Agent: **PETERS, Aaron, M.**; Marshall, Gerstein & Borun LLP, 233 S. Wacker Drive, Suite 6300, Sears Tower, Chicago, IL 60606-6357 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

WO 2008/042807 A2

(54) **Title:** STATISTICAL SIGNATURES USED WITH MULTIVARIATE ANALYSIS FOR STEADY-STATE DETEECTION IN A PROCESS

(57) **Abstract:** Methods and systems to detect steady-state operations in a process of a process plant include collecting process data. The collected process data is generated from a plurality of process variables of the process. A multivariate statistical model of the operation of the process is generated using the process data. The multivariate statistical model may be generated from a principal component analysis. The model is executed to generate outputs corresponding to the most significant variations in the process. Statistical measures of the outputs are generated and used to determine whether a steady-state or unsteady- state is related to the process.

STATISTICAL SIGNATURES USED WITH MULTIVARIATE ANALYSIS FOR STEADY-STATE DETECTION IN A PROCESS

Cross–References To Related Applications

[0001] This application claims priority from U.S. Provisional Application Serial No. 60/827,631, which was filed on September 29, 2006, entitled "ON-LINE PRINCIPAL COMPONENT ANANLYSIS IN A DISTRIBUTED CONTROL SYSTEM." The above-referenced provisional patent application is hereby incorporated by reference herein, in its entirety.

Technical Field

[0002] This disclosure relates generally to process plant diagnostics and, more particularly, to monitoring and diagnostic systems involving multivariate statistical techniques.

Description of the Related Art

[0003] Process control systems, like those used in chemical, petroleum or other processes, typically include one or more centralized or decentralized process controllers communicatively coupled to at least one host or operator workstation and to one or more process control and instrumentation devices such as, for example, field devices, via analog, digital or combined analog/digital buses. Field devices, which may be, for example, valves, valve positioners, switches, transmitters, and sensors (e.g., temperature, pressure, and flow rate sensors), are located within the process plant environment, and perform functions within the process such as opening or closing valves, measuring process parameters, increasing or decreasing fluid flow, etc. Smart field devices such as field devices conforming to the well-known FOUNDATION™ Fieldbus (hereinafter "Fieldbus") protocol or the HART® protocol may also perform control calculations, alarming functions, and other control functions commonly implemented within the process controller.

[0004] The process controllers, which are typically located within the process plant environment, receive signals indicative of process measurements or process variables made by or associated with the field devices and/or other information pertaining to the field devices, and execute controller applications. The controller applications implement, for

example, different control modules that make process control decisions, generate control signals based on the received information, and coordinate with the control modules or blocks being performed in the field devices such as HART® and Fieldbus field devices. The control modules in the process controllers send the control signals over the communication lines or signal paths to the field devices, to thereby control the operation of the process.

[0005] Information from the field devices and the process controllers is typically made available to one or more other hardware devices such as operator workstations, maintenance workstations, personal computers, handheld devices, data historians, report generators, centralized databases, etc., to enable an operator or a maintenance person to perform desired functions with respect to the process such as, for example, changing settings of the process control routine, modifying the operation of the control modules within the process controllers or the smart field devices, viewing the current state of the process or of particular devices within the process plant, viewing alarms generated by field devices and process controllers, simulating the operation of the process for the purpose of training personnel or testing the process control software, and diagnosing problems or hardware failures within the process plant.

[0006] As is known, problems frequently arise within a process plant environment, especially a process plant having a large number of field devices and supporting equipment. These problems may take the form of broken or malfunctioning devices, logic elements, such as software routines, residing in improper modes, process control loops being improperly tuned, one or more failures in communications between devices within the process plant, etc. These and other problems, while numerous in nature, generally result in the process operating in an abnormal state (i.e., the process plant being in an abnormal situation) which is usually associated with suboptimal performance of the process plant.

[0007] Many diagnostic tools and applications have been developed to detect and determine the cause of problems within a process plant and to assist an operator or a maintenance person to diagnose and correct the problems, once the problems have occurred and been detected. For example, operator workstations, which are typically connected to the process controllers through communication connections such as a direct or wireless bus, Ethernet, modem, phone line, and the like, have processors and memories that are adapted to run software, such as the DeltaV™ and Ovation® control systems, sold by Emerson Process Management. These control systems have numerous control module and control loop

diagnostic tools. Likewise, maintenance workstations, which may be connected to the process control devices, such as field devices, via the same communication connections as the controller applications, or via different communication connections, such as object linking and embedding (OLE) for process control (OPC) connections, handheld connections, etc., typically include one or more applications designed to view maintenance alarms and alerts generated by field devices within the process plant, to test devices within the process plant and to perform maintenance activities on the field devices and other devices within the process plant. Similar diagnostic applications have been developed to diagnose problems within the supporting equipment within the process plant.

[0008] Thus, for example, software available commercially as the AMS™ Suite: Intelligent Device Manager from Emerson Process Management enables communication with and stores data pertaining to field devices to ascertain and track the operating state of the field devices. See also U.S. Patent Number 5,960,214 entitled "Integrated Communication Network for use in a Field Device Management System." In some instances, the AMS™ software may be used to communicate with a field device to change parameters within the field device, to cause the field device to run applications on itself such as, for example, self-calibration routines or self-diagnostic routines, to obtain information about the status or health of the field device, etc. This information may include, for example, status information (e.g., whether an alarm or other similar event has occurred), device configuration information (e.g., the manner in which the field device is currently or may be configured and the type of measuring units used by the field device), device parameters (e.g., the field device range values and other parameters), etc. Of course, this information may be used by a maintenance person to monitor, maintain, and/or diagnose problems with field devices.

[0009] Similarly, many process plants have included software applications such as, for example, the Machinery Health® application provided by CSI Systems, to monitor, diagnose, and optimize the operating state of various rotating equipment. Maintenance personnel usually use these applications to maintain and oversee the performance of rotating equipment in the plant, to determine problems with the rotating equipment, and to determine when and if the rotating equipment must be repaired or replaced. Similarly, many process plants include power control and diagnostic applications such as those provided by, for example, the Liebert and ASCO companies, to control and maintain the power generation and distribution equipment. It is also known to run control optimization applications such as, for example, real-time optimizers (RTO+), within a process plant to optimize the control

activities of the process plant. Such optimization applications typically use complex algorithms and/or models of the process plant to predict how inputs may be changed to optimize operation of the process plant with respect to some desired optimization variable such as, for example, profit.

[0010] These and other diagnostic and optimization applications are typically implemented on a system-wide basis in one or more of the operator or maintenance workstations, and may provide preconfigured displays to the operator or maintenance personnel regarding the operating state of the process plant, or the devices and equipment within the process plant. Typical displays include alarming displays that receive alarms generated by the process controllers or other devices within the process plant, control displays indicating the operating state of the process controllers and other devices within the process plant, maintenance displays indicating the operating state of the devices within the process plant, etc. Likewise, these and other diagnostic applications may enable an operator or a maintenance person to retune a control loop or to reset other control parameters, to run a test on one or more field devices to determine the current status of those field devices, or to calibrate field devices or other equipment.

[0011] While these various applications and tools are very helpful in identifying and correcting problems within a process plant, these diagnostic applications are generally configured to be used only after a problem has already occurred within a process plant and, therefore, after an abnormal situation already exists within the plant. Unfortunately, an abnormal situation may exist for some time before it is detected, identified and corrected using these tools, resulting in the suboptimal performance of the process plant for the period of time during which the problem is detected, identified and corrected. In many cases, a control operator will first detect that some problem exists based on alarms, alerts or poor performance of the process plant. The operator will then notify the maintenance personnel of the potential problem. The maintenance personnel may or may not detect an actual problem and may need further prompting before actually running tests or other diagnostic applications, or performing other activities needed to identify the actual problem. Once the problem is identified, the maintenance personnel may need to order parts and schedule a maintenance procedure, all of which may result in a significant period of time between the occurrence of a problem and the correction of that problem, during which time the process plant runs in an abnormal situation generally associated with the sub-optimal operation of the plant.

[0012] Additionally, many process plants can experience an abnormal situation which results in significant costs or damage within the plant in a relatively short amount of time. For example, some abnormal situations can cause significant damage to equipment, the loss of raw materials, or significant unexpected downtime within the process plant if these abnormal situations exist for even a short amount of time. Thus, merely detecting a problem within the plant after the problem has occurred, no matter how quickly the problem is corrected, may still result in significant loss or damage within the process plant. As a result, it is desirable to try to prevent abnormal situations from arising in the first place, instead of simply trying to react to and correct problems within the process plant after an abnormal situation arises.

[0013] One technique that may be used to collect data that enables a user to predict the occurrence of certain abnormal situations within a process plant before these abnormal situations actually arise, with the purpose of taking steps to prevent the predicted abnormal situation before any significant loss within the process plant takes place. This procedure is disclosed in U.S. Patent Application Serial No. 09/972,078, now U.S. Patent No. 7,085,610, entitled "Root Cause Diagnostics" (based in part on U.S. Patent Application Serial No. 08/623,569, now U.S. Patent No. 6,017,143). The entire disclosures of both of these applications are hereby incorporated by reference herein. Generally speaking, this technique places statistical data collection and processing blocks or statistical processing monitoring (SPM) blocks, in each of a number of devices, such as field devices, within a process plant. The statistical data collection and processing blocks collect process variable data and determine certain statistical measures associated with the collected data, such as the mean, median, standard deviation, etc. These statistical measures may then be sent to a user and analyzed to recognize patterns suggesting the future occurrence of a known abnormal situation. Once a particular suspected future abnormal situation is detected, steps may be taken to correct the underlying problem, thereby avoiding the abnormal situation in the first place.

[0014] Principal Component Analysis (PCA) is a multivariate data analysis technique that has been used in the process control industry for analysis of multidimensional data sets. PCA techniques generally involve reducing data from a high-dimensional space to a lower-dimensional space, which still explains most of the significant variations in the original data. For example, PCA may be used to reduce the dimensionality of a multivariable data space (e.g., multidimensional data) to a data space of a few dimensions (e.g., 2 or 3

dimensions), such that the multidimensional data is projected (scored) onto a lower dimensional principal component space. Further details regarding the typical implementation of PCA techniques for fault detection in industrial processes may be found in L. H. Chiang, et al., "Fault Detection and Diagnosis in Industrial Systems," Springer-Verlag London Limited, pp. 35-54 (2001), and E. L. Russell, et al., "Data-Driven Techniques for Fault Detection and Diagnosis in Chemical Processes," Springer-Verlag London Limited, (2000).

[0015] A number of software packages provide PCA functionality, but most of these packages utilize offline process data from a database. In other words, PCA has been an off-line multivariate statistical analysis tool. This is useful for an after-the-fact analysis of an abnormal situation in the process data, such as historical data or stored multidimensional data files, but it cannot be used to detect the abnormal situation in real time (e.g., on-line, at-line or in-line data analysis). Some existing PCA software packages may be capable of doing real-time analysis of the data, but only if the software would have access to the process data from the control system, often through an interface such as an OPC server. As a result, plant personnel would unfortunately have the burden of maintaining a software package separate from the control system, including supporting its access to, and interface with, the control system. Further, detection of abnormal process situations should be available at all times and for various abnormal situations (e.g., a continuous process vs. a one-time solution), in order for the abnormal situation prevention tools to exhibit alertness and accuracy at various times for various abnormal situations.

[0016] Traditional PCA techniques also may fail to easily convey information indicative of process dynamics. The results of PCA analysis, i.e., the scores, are often displayed through scatter plots that are static representations of the underlying process data. Regardless of the manner in which score data is recorded or interpreted in the scatter plots, there will generally be a small uncertainty associated with data being characterized incorrectly. As described below, without more information, a process operator often cannot be certain that a given score is normal or abnormal. For example, the first two principal components of a PCA analysis capture the largest variations in the multidimensional data, such that the first two scores from a PCA analysis may be plotted because they represent the most significant information about the behavior of the multidimensional data, including the largest variations in the process.

[0017] For these reasons, traditional PCA techniques have exhibited unreliable performance in the detection of abnormal situations. A PCA score can suggest an abnormal situation when one does not exist (i.e., a false alarm), and conversely can indicate normal operation when the process is behaving abnormally (i.e., missed diagnostics). Further, in order to implement PCA, a PCA model is developed using training data (e.g., data from a "normal" process operation), and the resulting model is applied for detecting abnormal process situations. The training data is used to derive the model in terms of the principal components. As such, the PCA scores are not directly related to the actual process variables, and the physical process limits of the process variables being monitored using PCA are lost in the analysis, thereby requiring a statistical interpretation (e.g., to quantify the "goodness of data").

Summary of the Disclosure

[0018] In accordance with certain aspects of the disclosure, a number of techniques are disclosed to facilitate the detection of a steady-state of a process and any elements thereof. Collected process data is generated from a plurality of process variables of the process. A multivariate statistical model of the operation of the process is generated using a set of the process data. The multivariate statistical model may be generated from principal component analysis (PCA). The model is executed to generate one or more outputs each corresponding to the most significant variations in the process. Statistical measures of the output(s) are generated and used to determine whether a steady-state or unsteady-state is related to the process.

[0019] The steady-state or unsteady-state may be determined from the statistical measure of the output corresponding to the most significant variation alone (e.g., the first principal component). If the statistical measure of the output corresponding to the most significant variation meets steady-state criteria, it may be determined that the whole process is in a steady-state. Alternatively, steady-state or unsteady-state may be determined from the statistical measure of a plurality of outputs each corresponding to a significant variation in the process. If all of the statistical measures of the plurality of outputs meet the steady-state criteria, it may be determined that the system is in a steady-state. If any one of the statistical measures of the plurality of outputs does not meet the steady-state criteria, it may be determined that the process is in an unsteady-state.

[0020] The steady-state criteria may be based on a difference between the means of an output as compared to a significant deviation of the output. The means and standard deviations may be generated from outputs corresponding to different sampling windows of the process data. The standard deviation used in the steady-state criteria may be the standard deviation corresponding to any of the sampling windows, or may be the minimum of the standard deviations of all the sampling windows being used. By monitoring the statistical measures of outputs from the multivariate statistical representation, the steady-state of the process may be determined based on the component(s) of the representation corresponding to the largest variation(s) in the process, and the steady-state operation of the process may be used as the basis for further multivariate statistical analysis to model the process in a normal, on-line state.

Brief Description of the Drawings

[0021] Fig. 1 is an exemplary block diagram of a process plant having a distributed process control system and network including one or more operator and maintenance workstations, controllers, field devices and supporting equipment;

[0022] Fig. 2 is an exemplary block diagram of a portion of the process plant of Fig. 1, illustrating communication interconnections between various components of an abnormal situation prevention system located within different elements of the process plant;

[0023] Fig. 3 is a graphical plot of a process variable over time with operational requirements or control limits for monitoring operation of the process plant;

[0024] Fig. 4 is a graphical, multivariate visualization or plot of a number of process variables relative to operational requirements or control limits, the visualization or plot being suitable for incorporation into a user interface generated in accordance with certain embodiments and aspects of the disclosure for on-line process monitoring;

[0025] Fig. 5 is another view of the visualization plot of Fig. 4 after one of the process variables has exceed the operational requirement or control limit, which may be indicative of an abnormal situation or fault condition;

[0026] Fig. 6 is a representation of an exemplary function block configured for implementing principal component analysis (PCA) techniques and instantiated within a

process control system for training and operation in accordance with one aspect of the disclosure;

[0027] Fig. 7 is a representation of another exemplary PCA function block shown during implementation of the PCA techniques in accordance with another aspect of the disclosure;

[0028] Fig. 8 is a block diagram of a multivariate monitoring and diagnostics module (MMDM) in accordance with another aspect of the disclosure, components of which, in some embodiments, may be implemented in the function blocks of Figs. 6 and 7;

[0029] Figs. 9-12 are exemplary user interface displays generated in accordance with another aspect of the disclosure, which, in some embodiments, may be created or supported by the function blocks of Figs. 6 and 7 or by the MMDM tool(s) of Fig. 8, for multivariate monitoring and fault detection;

[0030] Figs. 13 and 14 are further exemplary user interface displays generated in accordance with another embodiment and created or supported in manners similar to the user interface displays of Figs. 9-12 for multivariate monitoring and fault detection;

[0031] Figs. 15-19 are still further exemplary user interface displays generated in accordance with yet another embodiment and created or supported in manners similar to the user interface displays of Figs. 9-12 for multivariate monitoring and fault detection;

[0032] Figs. 20 and 21 are still further exemplary user interface displays generated in accordance with yet another embodiment and created or supported in manners similar to the user interface displays of Figs. 9-12 for multivariate monitoring and fault detection;

[0033] Figs. 22 and 23 are block diagrams of PCA-based steady-state detection systems and techniques in accordance with another aspect of the disclosure that may utilize the function blocks of Figs. 6 and 7;

[0034] Fig. 24 is a schematic diagram of a fired heater, an exemplary process to which the disclosed techniques and MMDM tools of the present disclosure may be applied for monitoring, diagnostics and fault detection;

[0035] Fig. 25 is a graphical representation of training data scores mapped to a two-dimensional space based on principal components t1 and t2, the training data being used for PCA-based monitoring, diagnostics and fault detection in connection with the fired heater of Fig. 24;

[0036] Figs. 26 and 27 are exemplary user interface displays generated by the MMDM tool in connection with the PCA-based monitoring, diagnostics and fault detection for the fired heater of Fig. 24;

[0037] Fig. 28 is a graphical plot of crude oil flow rate data to be used as validation data in connection with a multivariate statistical technique for detecting or determining dynamic operation in accordance with another aspect of the disclosure;

[0038] Fig. 29 is a graphical plot comparing the actual output temperature data (T_{out}) associated with the flow rates depicted in Fig. 28 with the output temperatures predicted by a PCR model of the multivariate statistical technique for detecting or determining dynamic operation;

[0039] Figs. 30-36 are exemplary user interface displays generated in connection with the implementation of the multivariate statistical technique for detecting or determining dynamic operation;

[0040] Fig. 37 is a flow diagram of the multivariate statistical technique for detecting or determining dynamic operation in accordance with one embodiment;

[0041] Figs. 38 and 39 are graphical plots comparing actual output temperature data for an exemplary fired heater with data predicted by a multivariate statistical model that may be used for detection of coking in accordance with another aspect of the disclosure;

[0042] Figs. 40 and 41 are graphical plots comparing actual and predicted output temperature data in accordance with one embodiment of the coking detection technique based on the multivariate statistical model;

[0043] Fig. 42 is a graphical plot of a PCA model parameter for the output temperature of a fired heater demonstrating another, non-regression-based technique for coking detection in accordance with another aspect of the disclosure; and

[0044] Fig. 43 is a flow diagram of one embodiment of the non-regression-based coking detection technique depicted in the graphical plot of Fig. 42.

Detailed Description

[0045] Referring now to Fig. 1, an exemplary process plant 10 in which an abnormal situation prevention system may be implemented includes a number of control and

maintenance systems interconnected together with supporting equipment via one or more communication networks. In particular, the process plant 10 of Fig. 1 includes one or more process control systems 12 and 14. The process control system 12 may be a traditional process control system such as a PROVOX or RS3 system or any other control system which includes an operator interface 12A coupled to a controller 12B and to input/output (I/O) cards 12C which, in turn, are coupled to various field devices such as analog and Highway Addressable Remote Transmitter (HART)® field devices 15. The process control system 14, which may be a distributed process control system, includes one or more operator interfaces 14A coupled to one or more distributed controllers 14B via a bus, such as an Ethernet bus. The controllers 14B may be, for example, DeltaV™ controllers sold by Emerson Process Management of Austin, Texas or any other desired type of controllers. The controllers 14B are connected via I/O devices to one or more field devices 16, such as for example, HART® or Fieldbus field devices or any other smart or non-smart field devices including, for example, those that use any of the PROFIBUS®, WORLDFIP®, Device-Net®, AS-Interface and CAN protocols. As is known, the field devices 16 may provide analog or digital information to the controllers 14B related to process variables as well as to other device information. The operator interfaces 14A may store and execute tools 17, 19 available to the process control operator for controlling the operation of the process including, for example, control optimizers, diagnostic experts, neural networks, tuners, etc.

[0046] Still further, maintenance systems, such as computers executing the AMS™ application and/or the monitoring, diagnostics and communication applications described below may be connected to the process control systems 12 and 14 or to the individual devices therein to perform maintenance, monitoring, and diagnostics activities. For example, a maintenance computer 18 may be connected to the controller 12B and/or to the devices 15 via any desired communication lines or networks (including wireless or handheld device networks) to communicate with and, in some instances, reconfigure or perform other maintenance activities on the devices 15. Similarly, maintenance applications such as the AMS™ application may be installed in and executed by one or more of the user interfaces 14A associated with the distributed process control system 14 to perform maintenance and monitoring functions, including data collection related to the operating status of the devices 16.

[0047] The process plant 10 also includes various rotating (and other) equipment 20, such as turbines, motors, etc. which are connected to a maintenance computer 22 via

some permanent or temporary communication link (such as a bus, a wireless communication system or hand held devices which are connected to the equipment 20 to take readings and are then removed). The maintenance computer 22 may store and execute any number of monitoring and diagnostic applications 23, including commercially available applications, such as those provided by CSI (an Emerson Process Management Company), as well the applications, modules, and tools described below, to diagnose, monitor and optimize the operating state of the rotating equipment 20 and other equipment in the plant. Maintenance personnel usually use the applications 23 to maintain and oversee the performance of rotating equipment 20 in the plant 10, to determine problems with the rotating equipment 20 and to determine when and if the rotating equipment 20 must be repaired or replaced. In some cases, outside consultants or service organizations may temporarily acquire or measure data pertaining to the equipment 20 and use this data to perform analyses for the equipment 20 to detect problems, poor performance or other issues effecting the equipment 20. In these cases, the computers running the analyses may not be connected to the rest of the system 10 via any communication line or may be connected only temporarily.

[0048] Similarly, a power generation and distribution system 24 having power generating and distribution equipment 25 associated with the plant 10 is connected via, for example, a bus, to another computer 26 which runs and oversees the operation of the power generating and distribution equipment 25 within the plant 10. The computer 26 may execute known power control and diagnostics applications 27 such as those provided by, for example, Liebert and ASCO or other companies to control and maintain the power generation and distribution equipment 25. Again, in many cases, outside consultants or service organizations may use service applications that temporarily acquire or measure data pertaining to the equipment 25 and use this data to perform analyses for the equipment 25 to detect problems, poor performance or other issues effecting the equipment 25. In these cases, the computers (such as the computer 26) running the analyses may not be connected to the rest of the system 10 via any communication line or may be connected only temporarily.

[0049] As illustrated in Fig. 1, a computer system 30 implements at least a portion of an abnormal situation prevention system 35, and in particular, the computer system 30 stores and implements a configuration application 38 and, optionally, an abnormal operation detection system 42, a number of embodiments of which will be described in more detail below. Additionally, the computer system 30 may implement an alert/alarm application 43.

[0050] Generally speaking, the abnormal situation prevention system 35 may communicate with (or include) abnormal operation detection systems, modules or tools (not shown in Fig. 1) optionally located in the field devices 15, 16, the controllers 12B, 14B, the rotating equipment 20 or its supporting computer 22, the power generation equipment 25 or its supporting computer 26, and any other desired devices and equipment within the process plant 10, and/or the abnormal operation detection system 42 in the computer system 30, to configure each of these abnormal operation detection systems and to receive information regarding the operation of the devices or subsystems that they are monitoring. The abnormal situation prevention system 35 may be communicatively connected via a hardwired bus 45 to each of at least some of the computers or devices within the plant 10 or, alternatively, may be connected via any other desired communication connection including, for example, wireless connections, dedicated connections which use OPC (or OLE for process control), intermittent connections, such as ones which rely on handheld devices to collect data, etc. Likewise, the abnormal situation prevention system 35 may obtain data pertaining to the field devices and equipment within the process plant 10 via a LAN or a public connection, such as the Internet, a telephone connection, etc. (illustrated in Fig. 1 as an Internet connection 46) with such data being collected by, for example, a third party service provider. Further, the abnormal situation prevention system 35 may be communicatively coupled to computers/devices in the plant 10 via a variety of techniques and/or protocols including, for example, Ethernet, Modbus, HTML, XML, proprietary techniques/protocols, etc. Thus, although particular examples using OPC to communicatively couple the abnormal situation prevention system 35 to computers/devices in the plant 10 are described herein, one of ordinary skill in the art will recognize that a variety of other methods of coupling the abnormal situation prevention system 35 to computers/devices in the plant 10 can be used as well. In any case, the abnormal situation prevention system 35 may communicate with, and receive process variable data from, any of the computers, devices or other aspects of the plant 10, including, but not limited to a process control system (e.g., DeltaV™, Ovation® or other distributed control systems), devices and computers that conform to various standards or protocols (e.g., FOUNDATION™ Fieldbus, HART®, OPC, Modbus, wireless, etc.), and various transmitters, sensors and actuators which may be implemented with the devices or distributed throughout the process plant 10. As discussed further below, the data received and used by the abnormal situation prevention system 35 may be historical data, such as data from a data historian, but may also be on-line data (e.g., data collected when the process is on-line),

which includes, but is not limited to, data collected on-line, at-line or in-line, as well as data collected in real-time as the process is operating.

[0051] By way of background, OPC is a standard that establishes a mechanism for accessing process data from the plant or process control system. Typically, an OPC server is implemented in a process control system to expose or provide process information from, for example, field devices. An OPC client creates a connection to an OPC server and writes or reads process information to or from a field device. OPC servers use OLE technology (i.e., Component Object Model or COM) to communicate with such clients so that the software applications implemented by the clients can access data from the field devices or other process plant equipment.

[0052] Fig. 2 illustrates a portion 50 of the example process plant 10 of Fig. 1 for the purpose of describing one manner in which the abnormal situation prevention system 35 and/or the alert/alarm application 43 may communicate with various devices in the portion 50 of the example process plant 10. While Fig. 2 illustrates communications between the abnormal situation prevention system 35 and one or more abnormal operation detection systems within HART® and Fieldbus field devices, it will be understood that similar communications can occur between the abnormal situation prevention system 35 and other devices and equipment within the process plant 10, including any of the devices and equipment illustrated in Fig. 1.

[0053] The portion 50 of the process plant 10 illustrated in Fig. 2 includes a distributed process control system 54 having one or more process controllers 60 connected to one or more field devices 64 and 66 via input/output (I/O) cards or devices 68 and 70, which may be any desired types of I/O devices conforming to any desired communication or controller protocol. The field devices 64 are illustrated as HART® field devices and the field devices 66 are illustrated as Fieldbus field devices, although these field devices could use any other desired communication protocols. Additionally, each of the field devices 64 and 66 may be any type of device such as, for example, a sensor, a valve, a transmitter, a positioner, etc., and may conform to any desired open, proprietary or other communication or programming protocol, it being understood that the I/O devices 68 and 70 must be compatible with the desired protocol used by the field devices 64 and 66.

[0054] In any event, one or more user interfaces or computers 72 and 74 (which may be any types of personal computers, workstations, etc.) accessible by plant personnel

such as configuration engineers, process control operators, maintenance personnel, plant managers, supervisors, etc. are coupled to the process controllers 60 via a communication line or bus 76 which may be implemented using any desired hardwired or wireless communication structure, and using any desired or suitable communication protocol such as, for example, an Ethernet protocol. In addition, a database 78 may be connected to the communication bus 76 to operate as a data historian that collects and stores configuration information as well as on-line process variable data, parameter data, status data, and other data associated with the process controllers 60 and field devices 64 and 66 within the process plant 10. Thus, the database 78 may operate as a configuration database to store the current configuration, including process configuration modules, as well as control configuration information for the process control system 54 as downloaded to and stored within the process controllers 60 and the field devices 64 and 66. Likewise, the database 78 may store historical abnormal situation prevention data, including statistical data collected by the field devices 64 and 66 within the process plant 10, statistical data determined from process variables collected by the field devices 64 and 66, and other types of data that will be described below.

[0055] While the process controllers 60, I/O devices 68 and 70, and field devices 64 and 66 are typically located down within and distributed throughout the sometimes harsh plant environment, the workstations 72 and 74, and the database 78 are usually located in control rooms, maintenance rooms or other less harsh environments easily accessible by operators, maintenance personnel, etc.

[0056] Generally speaking, the process controllers 60 store and execute one or more controller applications that implement control strategies using a number of different, independently executed, control modules or blocks. The control modules may each be made up of what are commonly referred to as function blocks, wherein each function block is a part or a subroutine of an overall control routine and operates in conjunction with other function blocks (via communications called links) to implement process control loops within the process plant 10. As is well known, function blocks, which may be objects in an object-oriented programming protocol, typically perform one of an input function, such as that associated with a transmitter, a sensor or other process parameter measurement device, a control function, such as that associated with a control routine that performs PID, fuzzy logic, etc. control, or an output function, which controls the operation of some device, such as a valve, to perform some physical function within the process plant 10. Of course, hybrid and other types of complex function blocks exist, such as model predictive controllers (MPCs),

optimizers, etc. It is to be understood that while the Fieldbus protocol and the DeltaV™ system protocol use control modules and function blocks designed and implemented in an object-oriented programming protocol, the control modules may be designed using any desired control programming scheme including, for example, sequential function blocks, ladder logic, etc., and are not limited to being designed using function blocks or any other particular programming technique.

[0057] As illustrated in Fig. 2, the maintenance workstation 74 includes a processor 74A, a memory 74B and a display device 74C. The memory 74B stores the abnormal situation prevention application 35 and the alert/alarm application 43 discussed with respect to Fig. 1 in a manner that these applications can be implemented on the processor 74A to provide information to a user via the display 74C (or any other display device, such as a printer).

[0058] Each of one or more of the field devices 64 and 66 may include a memory (not shown) for storing routines such as routines for implementing statistical data collection pertaining to one or more process variables sensed by sensing device and/or routines for abnormal operation detection, which will be described below. Each of one or more of the field devices 64 and 66 may also include a processor (not shown) that executes routines such as routines for implementing statistical data collection and/or routines for abnormal operation detection. Statistical data collection and/or abnormal operation detection need not be implemented by software. Rather, one of ordinary skill in the art will recognize that such systems may be implemented by any combination of software, firmware, and/or hardware within one or more field devices and/or other devices.

[0059] As shown in Fig. 2, some (and potentially all) of the field devices 64 and 66 include abnormal operation detection (i.e., abnormal situation prevention) blocks 80 and 82, which will be described in more detail below. While the blocks 80 and 82 of Fig. 2 are illustrated as being located in one of the devices 64 and in one of the devices 66, these or similar blocks could be located in any number of the field devices 64 and 66, could be located in other devices, such as the controller 60, the I/O devices 68, 70 or any of the devices illustrated in Fig. 1. Additionally, the blocks 80 and 82 could be in any subset of the devices 64 and 66.

[0060] Generally speaking, the blocks 80 and 82 or sub-elements of these blocks, collect data, such as process variable data, from the device in which they are located and/or

from other devices. Additionally, the blocks 80 and 82 or sub-elements of these blocks may process the variable data and perform an analysis on the data for any number of reasons. For example, the block 80, which is illustrated as being associated with a valve, may have a stuck valve detection routine which analyzes the valve process variable data to determine if the valve is in a stuck condition. In addition, the block 80 may include a set of one or more statistical process monitoring (SPM) blocks or units such as blocks SPM1 - SPM4 which may collect process variable or other data within the valve and perform one or more statistical calculations on the collected data to determine, for example, a mean, a median, a standard deviation, a root-mean-square (RMS), a rate of change, a range, a minimum, a maximum, etc. of the collected data and/or to detect events such as drift, bias, noise, spikes, etc., in the collected data. Neither the specific statistical data generated, nor the method in which it is generated, is critical. Thus, different types of statistical data can be generated in addition to, or instead of, the specific types described above. Additionally, a variety of techniques, including known techniques, can be used to generate such data. The term statistical process monitoring (SPM) block is used herein to describe functionality that performs statistical process monitoring on at least one process variable or other process parameter, and may be performed by any desired software, firmware or hardware within the device or even outside of a device for which data is collected. It will be understood that, because the SPMs are generally located in the devices where the device data is collected, the SPMs can acquire quantitatively more and qualitatively more accurate process variable data. As a result, the SPM blocks are generally capable of determining better statistical calculations with respect to the collected process variable data than a block located outside of the device in which the process variable data is collected.

[0061] It is to be understood that although the blocks 80 and 82 are shown to include SPM blocks in Fig. 2, the SPM blocks may instead be stand-alone blocks separate from the blocks 80 and 82, and may be located in the same device as the corresponding block 80 or 82 or may be in a different device. The SPM blocks discussed herein may comprise known FOUNDATION™ Fieldbus SPM blocks, or SPM blocks that have different or additional capabilities as compared with known FOUNDATION™ Fieldbus SPM blocks. The term statistical process monitoring (SPM) block is used herein to refer to any type of block or element that collects data, such as process variable data, and performs some statistical processing on this data to determine a statistical measure, such as a mean, a standard deviation, etc. As a result, this term is intended to cover software, firmware,

hardware and/or other elements that perform this function, whether these elements are in the form of function blocks, or other types of blocks, programs, routines or elements and whether or not these elements conform to the FOUNDATION™ Fieldbus protocol, or some other protocol, such as Profibus, HART®, CAN, etc. protocol. If desired, the underlying operation of blocks 80, 82 may be performed or implemented at least partially as described in U.S. Patent No. 6,017,143, which is hereby incorporated by reference herein.

[0062] It is to be understood that although the blocks 80 and 82 are shown to include SPM blocks in Fig. 2, SPM blocks are not required of the blocks 80 and 82. For example, abnormal operation detection routines of the blocks 80 and 82 could operate using process variable data not processed by an SPM block. As another example, the blocks 80 and 82 could each receive and operate on data provided by one or more SPM blocks located in other devices. As yet another example, the process variable data could be processed in a manner that is not provided by many typical SPM blocks. As just one example, the process variable data could be filtered by a finite impulse response (FIR) or infinite impulse response (IIR) filter such as a bandpass filter or some other type of filter. As another example, the process variable data could be trimmed so that it remained in a particular range. Of course, known SPM blocks could be modified to provide such different or additional processing capabilities.

[0063] The block 82 of Fig. 2, which is illustrated as being associated with a transmitter, may have a plugged line detection unit that analyzes the process variable data collected by the transmitter to determine if a line within the plant is plugged. In addition, the block 82 may includes one or more SPM blocks or units such as blocks SPM1 - SPM4 which may collect process variable or other data within the transmitter and perform one or more statistical calculations on the collected data to determine, for example, a mean, a median, a standard deviation, etc. of the collected data. While the blocks 80 and 82 are illustrated as including four SPM blocks each, the blocks 80 and 82 could have any other number of SPM blocks therein for collecting and determining statistical data.

[0064] Further details regarding the implementation and configuration of abnormal situation prevention systems and components thereof can be found in U.S. Pat. Publ. No. 2005/0197803, now U.S. Patent No. 7,079,984 ("Abnormal situation prevention in a process plant"), U.S. Pat. Publ. No. 2005/0197806 ("Configuration system and method for abnormal situation prevention in a process plant"), and U.S. Pat. Publ. No. 2005/0197805 ("Data

presentation system for abnormal situation prevention in the process plant"), each of which is hereby incorporated by reference for all purposes.

[0065] In the abnormal situation prevention systems and techniques described above and in the referenced documents, the SPM (or abnormal situation prevention) blocks 80, 82 may be associated with, or considered components of, one or more abnormal situation prevention modules. While abnormal situation prevention blocks may reside in a field device, where the faster-sampled data is available, abnormal situation prevention modules may reside in a host system or controller. The abnormal situation prevention modules may take data from one or more abnormal situation prevention blocks, and use the data to make a decision about the larger system. More generally, an abnormal situation prevention module may be developed and configured to receive data from one or more function blocks (e.g., abnormal situation prevention blocks) to support diagnostics for each type of field device, instrumentation or other equipment (e.g., valve, pump, etc.). Nonetheless, the function blocks associated with an abnormal situation prevention module may reside and be implemented by devices other than the specific equipment for which it was developed. In such cases, the abnormal situation prevention module has a distributed nature. Other abnormal situation prevention modules may be implemented entirely within one device, such as the process controller 60, despite being directed to diagnostics for a specific field device. In any event, a diagnostics routine or technique may be developed for each equipment type for detecting, predicting and preventing abnormal situations or operation of the equipment (or process). For ease in description only, the term "abnormal situation prevention module" will be used herein to refer to such routines or techniques. An abnormal situation prevention module is therefore responsive to a set of measurements needed to perform the diagnostics, and further includes (i) a set of abnormal conditions to be detected by the module, and (ii) a set of rules, which link a change in the measurements to a corresponding abnormal condition. Furthermore, references to abnormal situation prevention modules in the description of the disclosed techniques to follow are set forth with the understanding that the techniques may be utilized in conjunction with abnormal situation prevention blocks as well.

[0066] In some cases, the configuration application 38 or other component of the abnormal situation prevention system 35 may support the development or generation of a template for each abnormal situation prevention module. For example, the configuration and development platform provided by the DeltaV™ control system may be used to create

specific instances, or instantiations, of abnormal situation prevention modules from corresponding composite template blocks.

[0067] Although shown and described in connection with Fig. 2 as abnormal situation prevention functionality, the modules and blocks described above may be more generally directed to implementing multivariate statistical techniques configured for process monitoring and diagnostics and fault detection as described below. In some cases, the techniques described below may include or be integrated with abnormal situation prevention modules or blocks. In any case, references below to systems and techniques (and any modules, function blocks, applications, software or other components or aspects thereof) may be utilized, included, be integrated with, or otherwise be associated with the workstation tools 17, 19, operator interfaces 12A, 14A, applications 23, abnormal situation prevention system 25 and interfaces 72, 74 described above.

[0068] The monitoring and diagnostics techniques described herein may be used to detect abnormal operations, also referred to as abnormal situations or abnormal conditions, that have occurred or are occurring. In addition, the monitoring and diagnostics techniques may be used to predict the occurrence of abnormal operations before these abnormal operations actually arise, with the purpose of taking steps to prevent the predicted abnormal operation before any significant loss within the process or the process plant takes place, for example, by operating in conjunction with the abnormal situation prevention system 35.

[0069] Turning to Fig. 3, a number of multivariate monitoring and diagnostic techniques described herein may build on and incorporate visualization and other aspects of multivariate and univariate diagnostics tools. For example, the process variable (PV) is shown with its corresponding process requirements in Fig. 3 in a chart often referred to as a Shewhart chart. Each process variable may be described in terms of the Shewhart chart, which plots the variable relative to an upper control limit (UCL) 100, a lower control limit (LCL) 102, and a Target value 104. The UCL 100 and LCL 102 are real physical limits and not statistical limits. During operation, each process variable is preferably constrained between its UCL and LCL, with the ideal value for the process variable being the Target value. Within these limits, the process variable is said to be in-control, otherwise it is said to be out of control. As indicated, the UCL and LCL represent physical limits, which may be provided by the process operating requirements. Although this manner of visualization is based on a multidimensional view of process variable data, this description of the process

variables may be useful in combination with further visualization techniques based on process variable normalization, as described below.

[0070] However, a brief summary the algorithm underlying PCA is provided for reference to the various aspects that will be described further below. Although PCA is disclosed herein as a multivariate statistical analysis that may be used with various aspects of this disclosure, it should be understood that other multivariate statistical analyses may be used in place of PCA, including, but not limited to, Principal Component Regression (PCR), Partial Least Squares (PLS), Fisher Discriminant Analysis (FDA) and Canonical Variate Analysis (CVA).

[0071] For a given process, process unit, process device, etc., there are numerous measured process variables. Each of these can be used as an input variable to the PCA algorithm. Data from each of the process variables is collected either for a certain length of time, or a certain number of points, and are referred to herein as observations of the process variable. Generally, all of the process variables used in PCA are sampled simultaneously.

[0072] If m is the number of input variables, and n is the number of observations of each input variable, then the matrix X is an n x m matrix which contains all of the observations for all input variables. In a typical process, some variables have magnitudes significantly larger than others. In order to ensure that all process variables have an equal effect on the PCA model, the X data may be autoscaled (for each input variable, subtract the mean and divide by the standard deviation). After the input variables are autoscaled, the sample covariance matrix is calculated by: $S = X^T \bullet X / (n-1)$, where T is a transpose operation of the matrix X.

[0073] An Eigenvalue-Eigenvector decomposition is performed on the sample covariance matrix: $S = V \bullet D \bullet V^T$, where D is a diagonal matrix containing the m Eigenvalues, sorted from largest to smallest. The columns of V are the corresponding Eigenvectors, and T is a transpose operation of the matrix V. The largest Eigenvalue, and its corresponding Eigenvector, correspond to the direction in the original input spaces containing the largest amount of variation in the original process data. This is considered the first principal component. The second largest Eigenvalue, and its corresponding Eigenvector, correspond to the direction, orthogonal to the first principal component, containing the next largest amount of variation. This continues until a new orthogonal vector space, explaining all of the original process data, is created.

[0074] After the Eigenvalues and Eigenvectors are computed, PCA determines which ones of the Eigenvalues and Eigenvectors are the significant principal components, usually corresponding to the important variation in a process, and which are insignificant, usually corresponding to noise. There are many different methods that have been proposed for selecting the number of significant principal components, including Kaiser's Rule (i.e., selecting principal components with Eigenvalues greater than one), percent variance test (i.e., selecting the number of principal components that explain a certain percent (e.g., 80% or 90%) of the total variation in the data), parallel analysis, and cross-validation.

[0075] Whichever method is utilized, the final result determines the number of significant principal components, a, that will be retained for the PCA model. Then, the loading matrix $P \in \Re^{m \times a}$ is created by taking the first a columns from V. The projection of the original observations onto the subspace defined by the loading matrix is called the score matrix, and denoted by $T = X \bullet P$. The columns of T are called the scores of the PCA model, referred to as $t_1, t_2, \ldots t_a$. The loading matrix P, along with the means and standard deviations used in the autoscaling, are considered together to be the PCA model. This PCA model may then be applied to any future data set.

[0076] After the PCA model is created, statistically-based limits, based on the original process data, may be created for defining the normal operation of the process defined by the original data. One method that is used is Hotelling's T^2 test, though it should be understood that other methods for deriving statistically-based limits may be utilized. When the scores exceed the limit, it can be an indication that something is abnormal in the process. Also, 2-σ and 3-σ limits, which accordingly limit the amount of variation, may alternatively be applied.

[0077] One can make several plots, based upon score data from the PCA model. The simplest plot is a plot of one of the scores versus time. In this plot, upper and lower limits are created based on statistics, and an alarm could be triggered if either threshold is exceeded. A second common plot is a scatter plot of two of the scores. Most often the first two scores, t_1 and t_2, are plotted, because these describe the largest amount of variation in the process. A number of exemplary two-score scatter plots are shown in the user interfaces described below. In these and other cases, the statistical-based limit may include or involve a circle or ellipse around the normal process data. Again, an alarm could be triggered if scores exceed these limits.

[0078] Turning to Figs. 4 and 5, and in accordance with an aspect of the disclosure generally directed to supporting on-line monitoring and analysis of multivariate real-time process data, a process visualization technique utilizes the control limits within a multidimensional (or multivariate) set of process variable data, such as the control limits associated with the Shewhart chart above. Despite the multivariate data set, the limits may be represented in univariate form using a transformation, such as the one described below. Generally speaking, the transformation may be based on a univariate technique for monitoring multivariate data. Using the technique, process data and control limits of any number of process variables can be normalized and brought to the same standard by using a single projection parameter.

[0079] By contrast, there exist different statistical tools for off-line applications, and some of them can be used for on-line monitoring. For instance, as discussed above, Principal Component Analysis (PCA) is a technique often used to reduce dimensionality of multivariable data space to a few dimensions (often two to three). However, to implement PCA, the PCA model is developed using training data, or data from a "normal" process operation, and then the model is applied for further monitoring of the process. As discussed above, the model derived is in terms of the principal components, which are essentially dominant eigenvectors of the covariance matrix determined by the training data. As such PCA results, also referred to as scores, are not directly related to the true process variables, and hence the physical process limits of variables monitored are lost in analysis. For this reason, when using PCA tool, some statistical interpretation is often used to quantify "goodness of data", (i.e., 95% or 99% confidence intervals are computed using Hotelling's T^2 statistics).

[0080] Using a univariate analysis, a deterministic method is proposed for monitoring multivariate process data with a single monitoring variable. The method can be applied easily in both off- and on-line data monitoring and analysis applications. The method and model development are described as follows.

[0081] At the outset, $X_1, X_2, \ldots, X_n$ are disclosed as representing the process variables. Each variable X_i, i=1,…, n, is a vector containing time dependent data $x_{i,j}$, where j designates a j-th sample or component of X_i. For example, $X_i=[x_{i,1}, x_{i,2}, \ldots, x_{i,m}]^T$ where T implies a transpose operation on X_i, and *m* is the vector dimension that is determined by the data sampling rate and the total time of data collection. With such nomenclature each process

variable is described according to the transformation, $p_i = [(y - \alpha_i)^2 + k_i x_{i,j}^2]$, where i=1,..., *n*, and j=1,..., *m*, and where y is a variable designated to bundle together multivariable process variables, α_i is a unique position assigned for variable X_i, k_i is a scaling parameter and $x_{i,j}$ is the j-th component of X_i. Since the transformation p_i can uniquely be defined for each variable X_i, i=1,..., n, then the multivariable transformation can be defined as $P(y) = p_1(y)*p_2(y).....*p_n(y)$.

[0082] Next P is used to derive the projection model $M_v(y)=K_p/P(y)$, where K_p is a scaling factor and $M_v(y)$ designates a multivariate projection. Using these assignments, multivariate data is projected by the single variable y.

[0083] Figs. 4 and 5 illustrate the resulting visualization of multiple process variables (PV1, PV2, etc.) on a single plot in relation to the respective physical operating limits for each variable. Fig. 4 depicts a plot 106 for a data point in which each of the process variables remain within the operating limits. Fig. 5 depicts a plot 108 for another data point, one in which one of the process variables may have exceed its control limit.

[0084] To further illustrate this method (including how the visualization indicates that a control limit has been exceeded), an example is provided using six process variables, X_1, X_2, X_3, X_4, X_5 and X_6. Under normal operating conditions, each of these variables has an operating limit defined by $|X_{iL}|$ (with the assumption that data points are centered). With this information each transformation p_i is tuned to the same level with coefficients k_i so the visualizations in Figs. 4 and 5 are obtained. This effectively defines a group of multivariate monitor settings. As can be seen, each of the six process variables has a unique position, given by α_i's=-10, -5, 0, 5, 10, 15, with respect to the variable y (the horizontal axis). Furthermore, all process variables are tuned by the parameters k_is to achieve the equal maximum response, or limit, for the maximum operating ranges given by X_{iL}s.

[0085] Now if during the monitoring stage any of the six peaks pierce through the process variable indicators, then that particular variable will have encountered an out of range event. This is shown in Fig. 5. As can be seen, process variable #2 is marginal with respect to the allowable limit, while process variable #4 clearly exceeds the limit. Nonetheless, the situation depicted in Fig. 5 clearly illustrates the normal operating condition for the six process variables.

[0086] During the monitoring stage, the events of Figs. 4 and 5 are not isolated, or stand alone, events. They are in reality visual snippets, or frames, of continuously changing

variables. That is, as new process data enters the model (at a given sampling rate) the response changes instantly. The monitoring image is therefore refreshed in accordance with the sampling rate, which may be anywhere from 0.1 to 1 second, or longer.

[0087] In this monitoring approach, when the limits for all process variables are known, the disclosed method does not require training data. For example, to create the multivariate monitor, one applies the design procedure described above and the univariate model is completed, which is not possible with other approaches. In other cases, if the process variable limits are not all known, then training data for the process variables without predefined limits may be used to determine process variable limits under normal process operating conditions, and the model can then be computed. In the case of either known or unknown limits, each of the process variables may be normalized using the univariate analysis method to define a common process variable limit (see e.g., Fig. 4), and each of the process variables may be monitored relative to the common process variable limit. Further examples of normalization, using both known and unknown process variable limits, as discussed further below and may be implemented with the univariate analysis method.

[0088] The disclosed univariate analysis method is not restricted to any particular set or type of process variables. It can be applied to any mix of process variables, (e.g., process inputs, outputs and external variables). In addition the univariate method offers a unique way for creating and analyzing process data patterns. These patterns can further be used for process diagnostics, including abnormal situation prevention applications. For instance, if most of the time two or more peaks have a tendency to move synchronously up or down, then the corresponding process variables are correlated. Otherwise the variables are uncorrelated. Similarly, the univariate technique can be used to evaluate whether a process variable is malfunctioning. In short, the disclosed univariate data modeling and visualization method offers a technique to implement multivariate monitoring, for either on-line and off-line contexts using a single variable. The method does not require training data if the limits of all process variables are known, and as such it provides monitoring results that can easily be interpreted by the process operators.

[0089] Further information is now provided regarding three alternative methods that may be used for normalizing process variables for display in the form illustrated in Figs. 4 and 5. To describe them ,the typical process variable description illustrated in Fig. 3 (the Shewhart chart) may be considered. In this figure, the process variable is constrained

between the UCL 100 and the LCL 102, and the ideal value for the process variable is the Target value. Within these limits, the process variable is said to be in-control, otherwise it is said to be out of control. As discussed above, UCL and LCL represent physical limits which may be provided by the process operating requirements. Based on this, the following data normalization methods may be utilized.

[0090] The first method may be used when process variable control limits are known for each process variable. In particular, the process variable, PV, may be normalized as a function of the control limits and the Target value:

$$PV^* = 100\% \times n \times \sqrt{\frac{(PV - T)^2}{CL^2}}, \text{ with } 0<n<1$$

where Target = T, CL = |UCL-T| = |LCL-T|.

[0091] A second method may be used when process variable control limits are not known for all of the process variables. In particular, the process variable, PV, may be normalized as a function of the training data set collected for the process variable when the process is operating normally, which includes the observations of the process variable within the training data set:

$$PV^* = 100\% \times n \times \sqrt{\frac{(PV - M_{PV})^2}{CL^2}}, \text{ with } 0<n<1$$

where the normal process operation is represented by the training data set T{PV}, M_{PV} is the mean of T{PV}, and CL = $|PV - M_{PV}|_{max}$ in T{PV}.

[0092] A third method uses a statistical autoscaling approach, which may also be used when process variable control limits are not known for all process variables. In particular, the process variable, PV, may be normalized as a function of the training data set collected for the process variable when the process is operating normally, which includes the observations of the process variable within the training data set:

$$PV^* = 100\% \times n \times \sqrt{\frac{(PV - M_{PV})^2}{k^2\sigma^2}}, \text{ with } 0<n<1$$

where the normal process operation is represented by the training data set T{PV}, σ^2 is the variance of T{PV}, M_{PV} is the mean of T{PV}, and $k\sigma$, k=1,2,..., n determines a statistical control limit (e.g., 3σ).

[0093] Variations of these methods and other user defined control limits (and methods of determining them) may be utilized in connection with the disclosed techniques as desired.

[0094] The above-described data monitoring and analysis technique is ideal for a rapid on-line visualization and understanding of process data behavior. It can be used in process diagnostics and abnormal situation prevention applications. For instance data from multiple intelligent devices spread over a Fieldbus can rapidly be analyzed and monitored. The same is true for data from sensor arrays and bio-microarrays. The monitoring results can also be used for process optimization and quality control. Furthermore, the method can be used with statistical data modeling methods to enhance data analysis. For instance, it can be used in parallel with the PCA method in order to speed-up and enhance interpretation of scored data.

[0095] In accordance with certain aspects of the disclosure, the above-described visualization techniques may be integrated in an operator interface that displays the process data in other ways, such as after processing via other multivariate statistical techniques, as described below. For example, the visualization of process variables with respect to their physical limits can be used in conjunction with a PCA score plot to enable more accurate decisions about the current state of the process.

[0096] The above-described univariate technique may be incorporated into a deterministic multivariate data analysis tool (examples of which are described below) that offers a method for monitoring multivariable real-time process data. In this way, the single variable monitoring model may be used for visualization of the behavior of multivariate data.

[0097] More generally, an advantage of this approach is that all process variables are bundled together within a single monitoring variable. The disclosed univariate method is well suited for both on-line and off-line applications. As mentioned above, the disclosed method requires no training data when the operational limits of all process variables are known. The operational limits associated with process variables are used to develop the single variable monitoring model. In some cases, training data may be used for process variables for which the operational limits are not known. The model may be implemented on a number of software platforms, including, for instance, DeltaV™.

[0098] With reference now to Figs. 6 and 7, another aspect of the disclosure is directed to the functionality provided by principal component analysis (PCA) and other

multivariate statistical techniques. The PCA and other multivariate statistical techniques described below may be integrated components of a process control system, such as the distributed control systems DeltaV™ and Ovation®. Such integration supports the plant personnel utilization of PCA techniques for on-line monitoring applications. As described below, the visualization and alarms supported by the PCA and other techniques may be fully integrated into the control system along with other visualization schemes, such as the ones described above in connection with Figs. 4 and 5. For these and other reasons, plant personnel may utilize a streamlined or single user interface without the need to monitor, maintain or support separate software applications. Implementing PCA as part of a process control system also makes it more practical to implement monitoring and diagnostics using on-line process data, as described below.

[0099] Figs. 6 and 7 depict exemplary PCA function blocks 110, 112 for implementation of PCA-based techniques, which were discussed above, within a process control system. Generally speaking, each PCA function block 110, 112 can collect data during a training period, and at the end of the training period, develop a PCA model, and then apply the PCA model to all future data. On the left-hand side of the function block 110, 112 are the inputs to the PCA, which are the raw process variables. The outputs of the function block are the scores, corresponding to the most significant Eigenvalues.

[0100] For instance, the function blocks 110, 112 may be custom function blocks made available via the Control Studio of DeltaV™, also sold by Emerson Process Management of Austin, Texas, an interface 114 of which may be used to depict the function block input/output connections and other details. In this example, up to 20 process variables can be provided as inputs to the PCA. Of course, in general any number of inputs can be used for PCA. The inputs to the PCA block 110, 112 are process variables normally available in the DCS. There is also an input for a dependent process variable (Y_{IN}) which can be used to do Principal Component Regression (PCR), a statistical technique that may utilized in one or more of the disclosed methods described herein. Although the disclosed example shows a single variable, in general it should be understood that there can be multiple dependent variables.

[0101] The PCA function block 110, 112 may include a number of operational modes for both configuration and application of the PCA model. In the exemplary embodiments shown, the PCA function blocks 110, 112 run in two different modes: Learning

and Monitoring (as determined by the True/False on the LEARN function block input). During the Learning Mode, the function block collects data from each of the input variables. For example, when a user gives the Learn command, the PCA block 110, 112 begins collecting process data for all of the input variables. The process data is stored in the computer memory, and data collection is continued indefinitely until the user gives the Monitor command.

[0102] After a sufficient amount of data has been collected, the user gives the Monitor command to the PCA block. After the Monitor command is given, the PCA block 110, 112 performs the PCA algorithm to develop a PCA model based on the data collected. Model development may include the following steps: autoscaling the input data (for each input variable, subtract its mean, and divide by its standard deviation), calculating the covariance matrix from the scaled data, calculating the Eigenvalues and Eigenvectors, determining the number of significant Eigenvectors to retain for the PCA loading matrix, and calculating statistical limits using σ-based limits, or T^2 Hotelling's statistics (e.g. 95% or 99%) for the PCA scores.

[0103] When this calculation is complete, the PCA block 110, 112 proceeds into monitoring mode, where it calculates the scores, based on new process data input to the block. In particular, the new process data may be on-line process data generated in real-time. If any of the scores exceed the limit, this could indicate an abnormal situation and an alarm parameter on the block 110, 112 is set. This alarm parameter may be tied to any other part of the process control system or network.

[0104] Each visualization plot of the PCA results may be provided or generated as part of the control system operator interface. For instance, the interface may be used to generate a PCA score plot in a DeltaV™ Process History View that plots the score over time. Alternatively or additionally, data for two PCA scores may be plotted in a two-dimensional graph, as shown and addressed above and below. The statistical limit computed results either in circular or elliptical shape that may also be provided via the operator interface. An alarm may be triggered if the score exceeds the limit.

[0105] In some cases, the multivariate monitoring and diagnostics techniques described herein may be implemented in a module (or other element or system component) of a system directed to abnormal situation prevention. Like other abnormal situation prevention algorithms, the techniques may therefore be directed to detecting abnormal process situations

before they occur. These algorithms generally exhibit "alertness" and accuracy continuously for a number of abnormal situations. Accordingly, the abnormal situation prevention module may be a continuous process, rather than offering a one-time, or non-real time, solution.

[0106] The multivariate monitoring and diagnostics techniques described herein may also be utilized in batch processing, whereby the function blocks 110, 112 may collect different sets of process data for the input variables, where each set corresponds to a different on-line state of a process. For example, a user may give the Learn command when the process is operating at various capacities, such that the PCA function blocks collect process data for the input variables for a particular capacity level (e.g., a particular on-line state) when the process is operating normally. Thereafter, the user may give the Monitor command for the PCA block 110, 112 to perform the PCA algorithm to develop a PCA model based on the data collected for the particular on-line state (e.g., level of capacity). As such, a variety of PCA models may be developed, each corresponding to a different on-line state for the process when the process is operating normally. Thereafter, when the process is being executed in a particular on-line state (e.g., 50% capacity), the the PCA block 110, 112 proceeds into monitoring mode using the corresponding PCA model, where it calculates the scores, based on new process data input to the block. Accordingly, a user may select from a plurality of multivariate statistical models to select the one most closely associated with the on-line state of the process to analyze the on-line, real-time operation of the process to monitor the process, detect abnormal situations, etc. In one example, the user may select a particular training session and elect to build new models customized to a particular on-line state of a process.

[0107] This aspect of the disclosure proposes a general (open source) data-driven approach for on-line (i.e., real time) multivariate monitoring and diagnostics applications. Moreover, the approach generally supports creating and manipulating data for developing on-line models and comparing model parameters for use in diagnostics, fault detection, etc. The disclosed approach includes the definition of an integration platform for process monitoring and diagnostics, which ultimately may be implemented by using a plant-wide or networked architecture, such as the PlantWeb® architecture sold by Emerson Process Management of Austin, Texas. The proposed integration platform is based on a multivariate approach and can accommodate different computational algorithms, including those utilized in the detection techniques described below. With reference to the open source nature of the data-driven platform, it is understood that the disclosed methods and systems can receive data

from different input domains (e.g., FOUNDATION™ Fieldbus, HART®, Modbus, OPC, wireless, etc.). The data received may be on-line process data collected from any number of different transmitters, sensors and actuators and is used to define a multivariate process domain. The same data is used by the process control system and can be stored in a data historian.

[0108] The flow of data through the disclosed system and module is shown in Fig. 8. Once data from the process enters the disclosed module, the data may be used for on-line training, on-line monitoring, or both during the same time period. Two switches SW1 and SW2 are implemented in way to support the different data paths, as desired. If training is desired, data may be accumulated until user/operator stops the accumulation process. This procedure may be accomplished by placing SW1 into an on-line training state. Any accumulated data may be stored in a file 120, and then be used in a model builder or generator 122. Each generated model may be stored as shown, and made available to the components involved in the monitoring mode via, for instance, the switch SW2. When the model building operation is completed, the model together with a training data is stored, typically as a .txt file in a model folder which resides in a database or memory (drive C:/).

[0109] Models that may be built by this data include, but are not limited to, Principal Component Analysis (PCA), Artificial Neural Networks (ANN), Fuzzy Logic and Bayesian Decision Trees. The tools for each of the algorithms may be provided in a drop-down-window fashion by, for instance, the abnormal situation prevention module. Each of these models may be used to develop predictions, as described below. The same model may also be used for on-line monitoring. When the model is in an on-line monitoring mode, the abnormal situation prevention module uses the model output (O/P) to generate three types of outputs: statistical, normalized process variables (PVs), and process variable (PV) ratings. As shown in Fig. 8, each of the outputs (PV ratings, Normalized PVs & Physical Limits, and Scores & Statistical Limits) may be further used to generate alarm status (alarm diagnostics), visualizations (Operator Graphics Interface) and/or verification of the models.

[0110] The statistical output is composed of scores which are low dimensional representations of multivariate data points. The low dimensional space is obtained based on directions with largest data variation in the multivariate domain. The PCA modeling is a technique that may be used for developing scores, as has been discussed above, although

other modeling techniques produce scores as well. The statistical output can also contain 95%(99%) confidence limits, or σ-based limits.

[0111] The concept of normalized process variables was described above in relation to both known and unknown process variable limits. Based on this concept, a bar chart (or other two-dimensional plot or graph) may be used for presenting normalized data. These charts may also contain physical or process limits that are useful for alarm identification. As described below in connection with further aspects of the disclosure, the combination of the statistical limits and the physical limits in a single user interface provides a useful way for accurately identifying abnormal situation prevention alarm status, and as such offers a technique supportive of the abnormal situation prevention initiative.

[0112] Process variable ratings may be useful in connection with abnormal situation prevention because the process variable rating is basically a rating that is given to a process variable for its variability. The more variation that the process variable exhibits the higher is its rating. This is a useful parameter in that it can easily be cross-validated with ratings obtained for monitoring the same process situation at different times. For example, if all process variables exhibit the same ratings for two or more monitoring models, the confidence in monitoring models increases. To obtain the process variable rating the following calculation may be used:

$$PV_{j\,rating} = \frac{1}{trS}\sqrt{\sum_{i=1}^{k}\sigma_i^2 p_{i,j}^2}$$

where $PV_{jrating}$ is the rating of the j^{th} process variable, S is the n x n auto-scaled covariance matrix, n is the dimension of multivariate data space, trS is the trace of S, k<n is the principal component space, σ_i and p_i for i=1,..., k are, respectively, the principal Eigenvalues and Eigenvectors of S, and $p_{i,j}$ is the j^{th} component of the eigenvector p_i. $p_{i,j}$s are also referred to as the loadings. The ratings may be computed for each developed PCA model and can be stored in model files with other model parameters.

[0113] The above equation is composed of model parameters, and the process variable rating obtained in this fashion is intimately related to the model. Thus, if two models correctly describe the same process event, then they rate process variables in the same order. As such, the models may be verified based upon their respective ratings. This also is generally true when models are different. For example, one model may be PCA-based and the other ANN based, and their process variable ratings are generally in the same order.

Exceptions may include when the process data is nonlinear (i.e., deviates significantly from the normal distribution).

[0114] It is further noted that other model parameters may be used for model validation. PCA parameters such as the eigenvector loadings $p_{i,j}$ and Eigenvalues σ_i, or ANN parameters such as the number of hidden layers and the number of outputs, are also useful parameters. Every model exhibits certain set of parameters that can be used for the validation task.

[0115] In accordance with another aspect of the disclosure, the model output data may be used to generate alarm status and, more generally, a composite, multivariate graphical interface for operators. Figs. 9-13 illustrate exemplary graphical interfaces that display normalized process variables, process variable ratings (see Fig. 13), scores and alarm status. The graphical interface also contains tabs from which other model information and graphs can be accessed. As shown in Fig. 13, the process variable ratings may be given in parentheses, [e.g., (2)], next to the variable description in the upper left part of Fig. 13. In this particular instance there are 10 process variables, and ratings range from 1 to 10. Process variable S21531 has the rating 1, implying it is the variable with largest variation.

[0116] With reference now to Fig. 14, accessing a BiPlot tab 130 provides the operator with an opportunity to view two score presentations as depicted in a score plot panel 132. The bi-plot is used to represent a score simultaneously with process variables, which are in the graph defined by their loadings $p_{i,j}$. The lines indicate how much of the principal component is present in process variables. For instance it can be seen that the process variable, S21531 rated 1, has largest first principal component t1 (horizontal axis). In contrast the process variable, S21008 rated 8, has the largest third principal component t3 (vertical axis). This makes perfect sense because t1 defines the direction with the largest variation in the data space, while t3 is significantly inferior. One can explore process variable relation with other principal components by simply changing the presentation in the window in the upper right part of the graph, (e.g., scores for t1-t2 can be obtained).

[0117] The bi-plot offers an easy way to relate an on-line score (black dot) to the process variables. The line closest to the score determines the dominant process variable for the score. Now since the lines are always in the same locations and the score is the only dynamic component on the graph, the operator can quickly understand which process variable, or a group of process variables, are most influential for a score projected at any

given instant of time. As such, the operator can become quickly familiar with all process variables through a single display.

[0118] With regard to implementation options, the disclosed user interface displays (and the system or module responsible for generating them) may be implemented in DeltaV™, Ovation® or any other control system platform. For instance, the PCA-based function blocks may be utilized to this end. Alternatively, Visual Basic (VB) with .NET technology may also be utilized as an implementation platform. In one case, an exemplary implementation strategy may incorporate some combination of VB.NET and DeltaV™ or Ovation®. Notwithstanding the foregoing, implementation may also be accomplished on different software platforms, (i.e., Java, Delphi, C++). Further details regarding the operator display interfaces of Figs. 9-13 are provided below.

[0119] It is noted that despite the usefulness of PCA, the PCA technique does not readily convey information about process dynamics. This is generally due to the use of scatter plots to display the results of the PCA analysis (scores). The scatter plots, although useful, are generally static representations of process data. As discussed above, multidimensional training or monitoring data sets are projected (scored) onto the lower dimensional principal component space. Typically the first two principal components capture the largest variations in data, and as a result they produce a scatter plot with most significant information about data behavior. Hotelling's T^2 distribution may also be used to compute 95% or 99% confidence region, or apply σ-based limits. A knowledge of this region may be important for detection of abnormal situations because if the training data represents a normal process performance, then for any new score point inside the region one can be 95% (99%) certain that the data point is normal. In contrast, for score points outside the limit region one can be 95% (99%) certain that the data point is not normal (abnormal).

[0120] As is apparent from the above, recordation and interpretation of scores through scatter plots results in a degree of uncertainty associated with the data being interpreted incorrectly. Unfortunately, as discussed above, experience has shown that this approach often leads to false alarms. The main reason for this is that the statistically calculated limits do not take into account the actual physical limits of the process variable. Each score is a linear combination of the mean-centered input variables. For example, inside and outside the confidence limit region, there is an ambiguity of 5% (1%), such that there is not a 100% assurance that any given score is normal or abnormal, and traditional PCA is not

always reliable in detecting abnormal situations. For example, a PCA score may trigger an abnormal situation when one does not exist (false alarm), and conversely a PCA score can indicate a normal behavior when the process is actually behaving abnormally (missed diagnostics). It is apparent this is not generally acceptable when determining abnormal situations and issuing abnormal alerts. The question is what happens when one takes the same linear combination of the physical limits. Because process variables have different scales, it is not always clear how one would transform these limits onto a PCA score plot.

[0121] The interface displays of Figs. 9-13 show an exemplary manner in which an operator interface may be used to visualize the PCA model applied to the current process data, while addressing the foregoing concerns. In short, both a PCA score plot 146, which may be generated from a PCA or other multivariate analysis, and a physical limit plot 148, which may be generated from univariate analysis, are included. The combination of these two plots gives an indication of the current state of the process. In the example of Fig. 9, because the PCA scores are within the statistical limits and all of the process variables are within their physical limits, this indicates the process is in the normal operating condition, and an indicator block 150 is highlighted to that effect.

[0122] Fig. 10 illustrates an example of how, according to the PCA score plot 146, everything appears to be normal. However, one of the process variables has exceeded its physical operating limits, as shown in the physical limit plot 148. Therefore, an indicator block 152 highlights a Missed Alarm.

[0123] Conversely, it is also possible for the PCA Score plot to indicate that there is a problem, while all of the process variables are still within their physical operating limits. Fig. 11 illustrates this case, which we call a False Alarm as depicted by an indicator block 154. Therefore, a true alarm condition is detected when both the statistical limits on the PCA plot and the physical operating limits are exceeded. This is illustrated in Fig. 12 with an indicator block 156.

[0124] Referring to Fig. 13, it is shown that the graphical representations are accompanied by a user-selectable graphic for selecting previous sets of collected data, shown as a slide bar, and a user-selectable graphic of principal components, shown as a drop down menu. The slide bar, or other graphical representation, enables the user to select previously collected data based on a previous operation of the process, and generate PCA scores based on the previously collected data. The graphical representations may then be updated with the

PCA score, such that a user may readily access and analyze any previous process operations for abnormal behavior. Still further, it is noted that the process variables may be expressed as a function of their process limits, which may include a common process limit as discussed above. Likewise, selection of different principal components enables a user to view the operation of different process variables having different degrees of significance on the process, and the graphical representations may be updated accordingly.

[0125] Figs. 15-19 illustrate different states of an exemplary user interface that presents dynamic process data visualization together with the PCA scoring of control limits in a manner similar to that shown in connection with the embodiments of Figs. 9-14. As described above, the control limits of seven process variables have been normalized to a unique value, and if the PCA score for this case is outside the 95% confidence region, a warning may be issued. These two graphical illustrations run simultaneously on-line for any incoming data. In addition, the PCA score plot contains one point, which is dynamic, while the peaks in the process variable monitor change according to the data inputs. When all peaks are below a control limit line 160, the process is in control and operation is normal. The corresponding statistical score should also be inside a limit circle 162. That situation is depicted in Fig. 16. Similarly when one or more peaks cross the limit line 160, the out of control or abnormal condition occurs and the PCA score should be outside the limit circle 162, as shown in Fig. 17.

[0126] False call diagnostics may also be experienced. In Fig. 18, the score is outside the limit circle 162 (abnormal), while all process variables are below the limit line 160 (normal). This case is clearly a false alert to be ignored. However, the opposite can also occur. The case illustrated in Fig. 19 shows the situation in which the score is inside the limit circle 162 (normal), while the process variable indicator indicates that process variable #2 has violated limit requirements (abnormal). This is the case of missed detection, or missed alert.

[0127] The two on-line graphical representations can accurately diagnose whether a data point corresponds to a normal or abnormal process situation. Moreover if abnormal situation occurs, process variables that have violated normal status can instantly be identified. The significance of these variables for process operations, however, may not be the same. The importance of process variables for a particular monitoring requirement are determined from the PCA loading values in the loading matrix P. If the process variable signaling an alert is associated with a small PCA loading value it may have a very little significance to

process operations, and the alert may be classified simply as warning. In contrast, if the loading value is high a variable is likely to be significant for process operations and high alert should be advised.

[0128] The foregoing visualization techniques and accompanying operator interfaces may be realized within any process control system architecture or platform and, in any case, may be supported by the functionality implemented by the PCA block described above.

[0129] Figs. 20 and 21 depict further alternative operator interfaces generated in accordance with the disclosed techniques, both of which support on-line monitoring, diagnostics, and false alarm detection functionality, as described above. In particular, Fig. 20 depicts the process variables in a bar graph as a percentage of their process variable limits, which may include a common process variable limit, as discussed above.

[0130] **Detection of Steady-State Operation with Multivariate Statistical Techniques**

[0131] Figs. 22 and 23 are directed to another aspect of the disclosure relating to recognizing and detecting steady-state operation using a PCA-based statistical signature. Many abnormal situation prevention algorithms rely on making a detection by learning the initial state of one or more of the process variables, and then triggering an alarm when the process variables change in a certain manner. However, many of these abnormal situation prevention algorithms also require that the process be in a steady-state, both before training for the initial conditions, and before making a detection. While it is generally easy for an operator to look at a plot of one or more process variables, and tell whether or not they are at a steady-state, it is generally more difficult to create an algorithm which can reliably make the same determination. A technique directed to detecting a steady-state in a process is now described. The technique is capable of making the determination based on monitoring multiple process variables.

[0132] Given a process variable x, the mean and standard deviation can be calculated over non-overlapping sampling windows of a given length (e.g. five minutes). In one example, the mean is calculated by:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

and the standard deviation is calculated by:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

where n is the number of samples, and $x_1, x_2, \ldots x_n$ are the samples of the process variable x taken over the sampling window.

[0133] In this example, $\bar{x}_1$ and s_1 are the mean and standard deviation, respectively, calculated in one sampling window, and $\bar{x}_2$ and s_2 are the mean and standard deviation, respectively, calculated over the next sampling window. Then, if $|\bar{x}_1 - \bar{x}_2| \leq 3 \cdot s$, the process may be described as being in steady-state. Conversely, if $|\bar{x}_1 - \bar{x}_2| > 3 \cdot s$, then the process is not in a steady-state.

[0134] Next, the parameter s used in the calculation is determined. If the process is in steady-state, and the sampling window is chosen correctly, then $s_1 \approx s_2$. In that case, it should not matter whether $s = s_1$ or $s = s_2$. However a more robust algorithm, which would result in smaller limits for declaring steady-state (which may provide greater certainty that the process is actually in a steady-state when the abnormal situation prevention module starts running) utilizes the minimum of s_1 and s_2, or $s = \min(s_1, s_2)$. In such a case, the criteria for declaring steady-state is $|\bar{x}_1 - \bar{x}_2| \leq 3 \cdot \min(s_1, s_2)$. The multiple of "3" may be a user-configurable parameter, though it is understood this may complicate the configuration of an abnormal situation prevention system, particularly when there are many process variables. The functionality of calculating mean and standard deviation of a process variable and determining steady-state, as well as other functionality, may be encapsulated into a Statistical Process Monitoring (SPM) block, as described above.

[0135] In accordance with another aspect of the disclosure, a steady-state detection technique utilizes scores generated by a PCA model (rather than the process variable data directly) to determine whether a process with multiple process variables is in a steady-state. Fig. 22 illustrates an SPM block 170 coupled to a PCA block 172, which may be used to determine whether or not a system with multiple process variables (IN_1, IN_2, IN_3, ...) is currently at steady-state. The PCA block 172 may correspond to the function block described above.

[0136] The example shown in Fig. 22 implements steady-state detection using the first score of a PCA model. As discussed above with respect to PCA models, the first score,

which corresponds to the largest Eigenvalue value and is calculated from the first loading vector, represents the largest amount of variation in the process. For example, in some processes, this may correspond to a load change that propagates through to many of the other process variables. In such a case, if only the first principal component meets the steady-state criteria, then it may be determined that the whole system is in steady-state.

[0137] While the first component may be the most statistically significant, alternative examples may use multiple scores of the PCA model, as shown in Fig. 23. In this case, the PCA function block 172 is coupled to multiple SPM blocks 170, with each SPM block 170 monitoring one of the scores of the PCA model. In this case, there would be one SPM block 170 for each principal component that was determined to be significant according to the chosen method (e.g., Kaiser's rule, percent variance, parallel analysis, etc.). A logic block 174 may be used to make a steady-state determination for the entire system. If all of the SPM blocks 170 indicate a steady-state, then steady-state may be determined for the entire system. If any of the blocks 170 show an unsteady process, then it may be determined that the entire system is not at steady-state.

[0138] A method implementing this technique may include collecting process data from all process variables, creating a PCA model of the process, coupling the first PCA score to a Statistical Process Monitoring (SPM) block, and declaring the system to be at a steady-state if the SPM block declares the first PCA score to be at a steady-state. The SPM block calculates means ($\bar{x}_1$ and $\bar{x}_2$) and standard deviations (s1 and s2) of an input variable over consecutive non-overlapping sampling windows, and declares that the input variable is at a steady-state if $|\bar{x}_1 - \bar{x}_2| \le n \cdot \min(s_1, s_2)$, where n is any real number. In some cases, the steady-state may be determined by $|\bar{x}_1 - \bar{x}_2| \le n \cdot s_1$. Alternatively, the steady-state may be determined by $|\bar{x}_1 - \bar{x}_2| \le n \cdot s_2$. In an exemplary embodiment, n=3.

[0139] The techniques and methods described above may be implemented with one or more additional SPM blocks coupled to the second and greater PCA scores, and a logic module which takes as inputs the steady-state detection of each of the SPM blocks and generates an output signal that indicates that the entire system is at a steady-state if all of the SPM blocks indicate a steady-state. The logic module may additionally determine the entire system to be an unsteady process if any of the SPM blocks indicates an unsteady process.

Alternatively or additionally, other multivariate techniques may be used with the SPM block for detecting steady-state. Such techniques include PLS, PCR, FDA, CVA, etc.

[0140] Although shown in Figs. 22 and 23 as involving multiple, separate units, the disclosed techniques and methods may be implemented in a single, integrated function block (e.g., a Fieldbus function block), a field device interface module, a control system, a standalone software application, or the like.

[0141] **Detection of Transient Dynamics**

[0142] Figs. 24-37 are directed to another aspect of the disclosure relating to the detection of transient dynamics using multivariate statistical techniques. Dynamic behavior is commonly encountered in process operations. For instance, a process exhibits transient behavior due to changes in input or load variables, such as flow rates. This is quite common in fired heaters because they usually run under various flowrates of crude oil.

[0143] When implementing fault or abnormal situation detection, it is important to distinguish between a normal operating condition and a transient operating condition. In particular, a process does not behave in a normal manner in a transient region, and a false diagnostic (e.g., abnormal operation) may easily be construed from the transient operation. For example, in fired heaters the outlet temperature of crude oil overshoots and takes time to drop to the normal level when the flowrate of crude oil increases. This is because the change in flowrate is faster than the change in the manipulated variable (fuel flowrate). In other words, in first order dynamic systems, the final temperature will need time equal to the time constant plus any lag time to reach its new steady-state point. Hence, it is useful to distinguish between dynamic transient region and steady-state, or set point, regions of operation when detecting a fault or abnormal situation.

[0144] In accordance with another aspect of the disclosure, a transient operation is detected using a multivariate statistical technique based on, for instance, principal component analysis (PCA). The disclosed technique is described in connection with dynamic transients in fired heaters, although the disclosed technique is well suited for implementation in connection with other process control contexts. The detection method may also integrate PCA-based techniques with principal component regression (PCR) to differentiate between abnormal situations and dynamic transients in processes, such as fired heaters. The example of detecting coking (or fouling) in fired heater tube-passes will be used to illustrate the robustness of the proposed method. It is important to identify transient behavior of the

process to avoid false alarms in the detection algorithm. Detection during transient operations may lead to false results and false alarms. It is also noted that the disclosed transient detection techniques may be utilized to address transient modes in other processes or unit operations, such as distillation columns and heat exchangers, for example.

[0145] Turning to Fig. 24, fired heaters are often used in refineries and petrochemical plants to raise the temperature of crude oil or heavy hydrocarbons to some elevated temperatures. A fired heater indicated generally at 180 includes one or more tube-passes 182. In this example, the feed is split evenly among heater passes 182. Each tube-pass 182 acts as a heat exchanger where the feed flowing inside the tube-pass 182 is heated by the burning fuel. The flow of fuel is often manipulated to achieve desired target final temperature of the crude oil (feed stream).

[0146] A common abnormal condition in coker heaters, also referred to as fired heaters, is coking. In particular, due to elevated temperatures in the fired heaters, crude oil cracks and forms a residue carbon called coke. Coke gets deposited over time on the inside surface of tubes, also referred to as passes, causing lower performance of the tubes or the fired heater in general. This process is called fouling. As time progresses, the performance of coker heaters gets poorer and sudden shut down may take place to clean the coker heater. In some instances, some tubes may get severe blockage. As such, coke formation is monitored inside coker heater tube passes to schedule unit cleaning.

[0147] Coker heaters present a multivariate data structure that is well suited for analysis using the multivariate monitoring and diagnostics techniques and tools described herein, referred to herein at times as MMDM (i.e., multivariate monitoring and diagnostics module) tools. For example, the process variables (PVs) set forth in Table 1 may monitored and processed by the MMDM tools in accordance with the techniques described herein. These process variables are also illustrated in Fig. 24. However, it is noted that process variable data may be supplied in any desired fashion via the process control system or network, as described above, and in accordance with a variety of different architectures and platforms, such as the PlantWeb® digital plant architecture and by using Emerson FOUNDATION™ Fieldbus devices (e.g., Micro Motion coriolis flow meters, Rosemount temperature devices, etc.) for example.

Monitored Process Variables	Abbreviation
Inlet temperature of crude oil	T_{in}

Flow rate to each pass	F_i
Total flow of oil	F
Exit temperature of the targeted pass	T_{out}
Final temperature of crude oil (controlled variable)	TF
Fuel flow rate	FIF
Final temperature of fuel	FIT

TABLE 1: Monitored Variables for Fired Heater

[0148] Fig. 25 presents training data scores developed from PCA in a space defined by principal components t_1 and t_2. In this example, the underlying training data was gathered for various flowrates with a single set point of the controlled variable. Approximately 30,000 real-time process data points were collected. The training data was used to develop models for the MMDM tool, and one of the scatter plots produced by the MMDM tool may correspond with the plot shown in Fig. 25. The eight clusters in the scatter plot correspond with the eight different flowrate regions in the training data.

[0149] The region inside the dashed line 190 shown in the plot of Fig. 25 corresponds with the ellipsoid shown in Figs. 26 and 27, which, in turn, represents the 99% confidence ellipsoid. That is, the area inside the ellipsoid gives 99% probability that any score within the boundaries of the ellipsoid belongs to a normal operating condition of the fired heater unit.

[0150] Turning to Figs. 26 and 27, the outlet temperature of a tube pass (T_{out}) was used as the dependent variable in this aspect of the disclosure and the other six variables were treated as independent variables. The independent variables are selected for monitoring, and used to create a normalized process variable data model, as well as a PCA model with score plots. The process variable data provides training data for the model development. The model is then used to define the operator user interface for on-line monitoring, as depicted in Fig. 26. As described above, the interface displays the independent process variables in both normalized and scoreplot fashions, as well as the process alarm status indicators. Generally speaking, the interface may be generated as described above in connection with Fig. 8 and may incorporate any one or more aspects of the exemplary interfaces described in connection with Figs. 9-21.

[0151] As for the dependent process variable (T_{out}), the training data is used to create a regression type PCR model. This is accomplished by the MMDM tool during the model building session. That is, components from the data space created by the PCA model

are used as independent components in creating the PCR model that models the dependent process variable as a function of the independent process variables. In the context of the interface, the PCR model may be accessed by pushing the "PCR" tab in the operator interface window, and the display window depicted in Fig. 27 appears. This display window shows regression coefficients computed for the score components T_1, T_2, ..., T_n.

[0152] To validate a detection of transient behavior by using the PCA model developed, a validation data set (4300 data points) was created for various crude oil flowrates. Fig. 28 shows seven different regions for the crude oil flow rate for the validation data. In this example, there are six different transient (dynamic) regions in the validation data, and the PCR model was used to predict these regions from the validation data. Fig. 29 shows the predicted T_{out} (Y) vs. actual T_{out}. The overlapping plots shown in Fig. 29 exhibit the success of the PCR model in predicting the dependent variable (Y) in both steady-state and transient regions. The prediction in the transient regions is slightly lagging, which is in agreement with expectations. Nevertheless, the six transient regions present in the validation data set are clearly identifiable in Fig. 29.

[0153] Figs. 30-34 depict the use of the Hotelling (T^2) computation, and the corresponding portion of the operator interface, for transient region detection. Specifically, Fig. 30 shows a T^2 plot at the moment the transient region starts. In particular, T^2 spikes up as flowrate of crude oil starts to change. In addition, Fig. 30 shows that the point of operation is outside the ellipse of scoreplot t_2-t_3. Fig. 31 shows that dynamic (transient) region was detected by PCA even though none of the independent variables are outside the limit. As dynamic transient progresses T^2 stays outside the acceptable limit and the scoreplots are outside the ellipsoid (Fig. 32). Once the process settles to the new steady-state region (Fig. 33), the T^2 response is below the limit and the scoreplot is inside the ellipsoid region. As can be seen, the PCA scoreplots and T^2 representation may be used to detect transients in process variables, as well as to estimate a transient time constant. Fig. 34 illustrates all six transient regions detected by T^2 for the whole validation data. The width of each transient region represents the transient duration.

[0154] Figs. 35 and 36 depict the results of an approach that does not utilize the controlled variable in the PCR analysis. In so doing, the usefulness of incorporating the controlled variable, and how it affects the detection robustness of the proposed method, is described. In the coker heater example used in this work, the final temperature (TF) of crude

oil is the controlled variable. In this example, it is desired to heat crude oil to a desired set point, final temperature. As before, a PCA model was developed using 5 of the 6 independent variables for the same training data: final temperature (TF) of crude oil was removed from the list. The new model was verified by using the same validation data as was used in the 6 variable analysis. Fig. 35 shows that the new model is not able to detect the dynamic behavior of the process. T^2 was also obtained for the validation data using the new model. Fig. 36 shows that spikes obtained in the 6 variable model no longer exist. Based on the foregoing, the inclusion of controlled variables in the model is useful in detecting transient or dynamic behavior, which may be expected given that controlled variables are often directly affected by changing load or input variables.

[0155] Fig. 37 is directed to a procedure for coking detection. From the discussion above, it is apparent that during the transition period the process behaves dynamically and the predicted Y is different than the actual Y. As such, abnormal condition detection may be halted during a transient condition, which may be accomplished using the behavior of T^2. For example, abnormal condition detection may be disabled if T^2 is outside the limit. If T^2 goes back within the acceptable limit, then abnormal condition detection will resume/start. For example, referring to Fig. 37, on-line process control data from a process is collected as training data at a block 202, from which the independent and dependent variables are determined at a block 204. At a block 206, PCA, or another multivariate analysis technique, is utilized on the independent variables to reduce the dimensions of the multivariate data space. At a block 208, the Hotelling (T^2) limits are determined from the PCA executed at block 206. A PCR model is developed at a block 210 using the components from the PCA model as the independent variable(s) and using the dependent variable(s) determined from block 204 to develop a model of the dependent variable(s) as a function of the independent variable(s). At a block 212, both PCA and PCR may be used to monitor the process based on monitored process control variables and detect abnormal conditions. For example, if an abnormal condition is detected using PCA, PCR may be utilized to determine whether the abnormal condition is a transient condition or an actual (or predicted) abnormal condition. In particular, PCR may be used to develop a prediction as to when transient conditions occur and may be compared against the occurrence of an abnormal condition. At a block 214, the T^2 values are generated based on the monitored data, and if the T^2 values are within acceptable limits, abnormal condition detection continues (block 218) and the occurrence may be treated as an abnormal condition. Otherwise, the procedure determines the

occurrence as a transient condition. When T^2 returns within acceptable limits, abnormal condition detection may resume. The disclosed detection method and technique may be applied to other processes and unit operations such as distillation columns, heat exchangers, turbines, etc, and is well suited for different process applications in chemical, refining, power, and life science industries.

[0156] In accordance with the foregoing aspect of the disclosure, a PCA-based monitoring technique has been implemented for on-line detection of transient behavior in process operations. A PCR regression technique may also be for on-line detection of transient behavior in process operations. The combination of both PCA and PCR techniques supports the characterization of process alerts, as the disclosed techniques can distinguish between an alert of an abnormal situation prevention type and one caused by process transients. Alternative designs are possible by using techniques such as multivariable regression, artificial neural-networks and Bayesian decision trees.

[0157] **Exemplary Application: Coke Detection in Coker Heaters with PCA and PCR**

[0158] Turning to Figs. 38-41, another aspect of this disclosure is directed to using PCA and PCR techniques to detect coking (or fouling) in coker heater tube-passes. In the embodiments described below, PCA and PCR techniques are used to collect and analyze off-line data and then monitor the process on-line to detect coking. The disclosed methods and techniques may be integrated with any of the controllers, field devices and other process control system elements described herein.

[0159] As described below, the disclosed coking detection method generally includes three steps: gathering training data, developing a model using PCA and PCR techniques, and using actual and predicted values of a targeted process variable to detect the coking fault. The training data gathered is generally real-time data, which is free of faults and representative of the expected operational range of the load variables. For example, training data may include minimum and maximum crude flow rates that are expected to be encountered in running the fired heater. The model to be developed may also benefit from having several flowrates between the minimum and maximum flowrates. In some cases, training data is gathered to cover each range of load variables, and may also include transient (i.e., dynamic) data for changing load variables. Generally, the sampling time and set point for the controlled variable may remain constant throughout the training data set. Given data

having the foregoing characteristics, the training data may be collected in a variety of ways. For instance, data may be collected on-line or be extracted from the process data historian. In any case, practice of the disclosed method is not limited to the manner in which the training data is gathered.

[0160] Multiple process variables are used to support robust coke detection in the tube-passes. In this exemplary case, the disclosed method utilizes the process variables set forth in Table 1 and depicted in Fig. 24. Of these, tube-pass outlet temperature (T_{out}) is the monitored dependent process variable, as it should decrease with coke formation inside the tubes. For coker heaters, as coke (fault) starts forming inside the tubes, the overall heat transfer across the tube from hot side to cold side decreases. Compared to normal operation conditions (fault free), the tube-pass outlet temperature (T_{out}) should also decrease. The other six variables in Table 1 are considered to be independent variables.

[0161] The coker heater provides a multivariate data structure well suited for explaining the analysis using the above-described multivariate monitoring and diagnostics module (MMDM). Once the independent variables are selected for monitoring, the selected variables are used to create a normalized process variable data model, as well as a PCA model with score plots. The gathered process variable data provides the training data for model development. The developed model is then used to generate the operator interface for on-line monitoring, as described above. As shown in, for instance, Fig. 26, the six independent process variables are displayed in both normalized and scoreplot fashions in conjunction with the four process alarm status indicators. The details of the model and operator interface are set forth above.

[0162] As for the dependent process variable (T_{out}), the training data is also used to create a regression-type PCR model. As described above, the generation of the PCR model is accomplished automatically by the MMDM tool during implementation of the model building procedure. In the context of the interface, the PCR model may be accessed by selecting the "PCR" tab in the operator interface, which may generate the exemplary display window shown in Fig. 27. As described above, this display window shows regression coefficients computed for the score components T_1, T_2, ..., T_n.

[0163] Embodiments of the disclosed fired heater monitoring system and method suitable for coke detection are now described. Coke formation may be indicated if the actual value of T_{out} (Y) is consistently lower than the predicted value $\hat{T}_{out}$ ($\hat{Y}$). For example this

may mean that the actual pass outlet temperature is continuously lower than the predicted temperature. In this case, no false alarms or alerts are generated. Absolute values or mean values of Y can be used for this purpose. As illustrated further below, the deviation of the actual Y from the predicted $\hat{Y}$ can give an accurate indication of the severity of coking. This tool may be utilized by plant engineers and operators to schedule unit cleaning or shut down based on the actual unit performance. In one example, the ratio $Y/\hat{Y}$ may be plotted as a function of time, such that plant engineers or operators may continuously monitor the performance of the unit and detect unit degradation or coking. Alternatively, the difference between the two values can be monitored as function of time. Based on the ratio of Y to $\hat{Y}$, plant personnel can decide what action to take. Since coking is a slow process, plant personnel may not select to enforce alarms or alerts. However, alerts or alarms may be activated based on threshold limits. For example, alerts can be issued if the ratio gets down to a certain limit (λ). For instance,

$$if \quad \left(\frac{Y}{\hat{Y}}\right) \leq \lambda ,$$

then an alert may be issued. Various alerts may be issued to indicate the severity of coking. For example, λ_1 indicates low coking, λ_2 indicates medium coking, etc.

[0164] More generally, the disclosed coke detection method includes the following steps: obtaining sufficient training data, developing a statistical model, and using the model for monitoring unit performance. In one example simulation, the training and validation data for the coker heater was generated by, and obtained from, a hi-fidelity simulator, SSPRO, commercially available from GSE Systems. The sampling rate was one second. The flowrate of fuel was manipulated to achieve the desired target final temperature of the crude oil. The simulation had flexibility to introduce coking of various severity levels. The simulation was used to obtain training data and simulation data. The seven process variables were recorded and used in the multivariate analysis provided by the MMDM tool.

[0165] The training was run with various flowrates. The flowrate and inlet temperature of the crude oil were allowed to have uniform noise, as to imitate real plant operation. Approximately 30,000 real-time process data points were collected. The training data was used to develop MMDM models, and the scatter plot produced by the MMDM tool is illustrated in Fig. 25, which depicts eight clusters which correspond to the eight flow regimes in the training data. Fig. 25 also displays the 99% confidence ellipsoid. The area

inside the ellipsoid gives a 99% probability that any score within the boundaries of the ellipsoid belongs to a normal operating condition of the fired heater.

[0166] Several real-time simulation data files were collected with various coking levels to test the robustness of the proposed abnormal situation detection. The simulation data points contain both good and faulty data. In order to introduce coking into the tube-pass, the overall heat transfer coefficient (U) was intentionally lowered. Several data sets were gathered as follows. In Set 1, the coefficient U was lowered from 1000 to 995, then to 990, to 980, and to 950. In Set 2, the coefficient U was reduced from 1000 to 500.

[0167] In this exemplary application, the PCR model developed for this specific case is $Y = B_0 + B_1*T_1 + B_2*T_2 + B_3*T_3$, where $B_0 = 316.9227895$, $B_1 = -0.215154485$, $B_2 = 0.274120980$, and $B_3 = -0.276083039$. T_1, T_2, T_3 are the score components determined on-line (validation data) based on variable contributions.

[0168] Before using this model for monitoring new data for abnormal condition (e.g., coke) detection, the model was tested by using a new set of validation data in order to ensure robustness of the model. Fig. 38 shows how accurate the PCR model was in predicting Y (T_{out}). The model has accuracy higher than 99.9%, excellent accuracy given that the data was scattered among various operation conditions (Fig. 25). Accuracy would be even higher if the range of operation conditions was narrower.

[0169] The model was also used to predict Y(T_{out}) for flow rates higher than what was used in the training data by 12%. In other words, the model was used to predict behavior of the process for flow rates outside the range of flow rates used in the training data. Fig. 39 shows that the prediction operated again with very high accuracy. It is noted that, the model may be used to data that remains within the training range (e.g., the range for which the model was trained).

[0170] The above-described model was used to detect coking in the validation data. The model was able to detect coking for all simulated validation data. Fig. 40 shows how the actual Y is lower than the predicted Y ($\hat{Y}$). The ratio of actual Y to $\hat{Y}$ gets lower as the coking increases (overall heat transfer coefficient gets lower). The predicted Y is the value of Y at normal conditions (fault/coking free). From the behavior of actual Y with respect to predicted Y, plant personnel can determine the severity of coking inside tube-passes.

[0171] Another set of data was used to verify the proposed detection methodology. The set contains data or a normal process where a sudden large abnormal condition (coking)

of high value is introduced. Fig. 41 shows how the ratio of actual Y to predicted Y changes. The first part of data in Fig. 41 demonstrated how good the model is in predicting the normal behavior.

[0172] In summary, the above-described technique (and underlying model) is directed to coke, fouling and other abnormal condition detection in coker heater tube passes using PCA and PCR. To that end, the inlet temperature of the crude oil, the tube-pass crude flowrate, the total crude flowrate, the tube-pass outlet temperature, the final temperature, the fuel flowrate, and the fuel outlet temperature may be monitored, and the tube-pass outlet temperature may be used as a performance indicator of the coker heater. In some cases, the performance indicator is modeled using PCA and predicted using PCR for operating conditions within the range of training data for the model. Nonetheless, PCR may be used for the prediction of the key performance indicator of the coker heater for operating conditions outside the range of the training data.

[0173] Turning to the aspects of the disclosure addressed in Figs. 42 and 43, a PCA parameter-based technique for on-line multivariate monitoring and diagnostics of coking in fired heaters is now described. The approach is based on developing multiple Principal Component Analysis (PCA) models, and comparing the model parameters for different process operating conditions. Apart from coking detection, the disclosed technique method may also be effective for determining or detecting other slow varying process changes involving a variety of different process equipment (e.g., soot buildup in furnaces, fouling and other degradation of reactors, compressors, heat exchangers, distillation columns, and membranes, etc.).

[0174] As described above, the PCA-based techniques may be integrated with a distributed process control systems (DCS) for implementation in connection with on-line monitoring and diagnostics. The disclosed techniques also enabled the creation and manipulation of data for developing on-line models and comparing model parameters. PCA-based techniques were also applied to the specific details presented by the challenge of determining coking levels in coker heaters using Principal Component Regression (PCR) analysis.

[0175] This aspect of the disclosure provides a method for detecting slow varying process changes, such as coking, without using a regression-based approach. The disclosed method is instead based on continuously, repeatedly or otherwise constantly developing PCA

models during process operations, and comparing the developed models to a baseline model developed under normal process operating conditions. The disclosed technique may provide a more robust solution in certain applications (e.g., abnormal situation prevention applications) relative to others that use regression for the reasons set forth below. For example, a regression-based technique generates a prediction every time data points are read from a field device communication link (e.g., Fieldbus, Modbus, wireless, etc.). As a result, regression-based techniques provide a current prediction of a process state, which may then be processed or evaluated via logic to determine a diagnostic status. This process often creates false alerts/alarms due to dynamic process changes such as setpoint and steady-state changes, as well as outlier noise in data strings. These types of alerts often become troublesome for operators and, accordingly, should be minimized, if not eliminated all together.

[0176] In contrast, the disclosed approach creates data files of predetermined dimension in the background of process operation either continually or on-demand. These data files are then used to generate PCA model updates used for evaluating process performance. As a result, the disclosed method may be more robust, as it is not entirely dependent on the current process data values. Instead, the disclosed method may generate an alert report based on process performance diagnostic based on a long term data behavior, which, for instance, may be determined by the size of data files. Further details of the disclosed method are described below.

[0177] At the outset, a training data file may be established to develop a PCA. The training data may represent data collected during normal process operating conditions. As described above, the data file may thus contain rows of data (observations) and columns, where each column corresponds to an assigned process variable (PV). The process variables may be either raw data, or statistical signatures obtained from the raw data, (e.g., abnormal situation prevention block data). Consequently, the data file contains **n** columns and **m** rows with a general assumption that **m**>>**n**. With such an assignment, the model addresses **n** process variables with **m** observations and contains **mxn** data matrix **X,** which is autoscaled as described above. The actual model may then be constructed by evaluating spectral components of the **nxn** covariance matrix **S,** where $\mathbf{S} = \mathbf{X}^{T} \bullet \mathbf{X} / (m-1)$, which are used to

rewrite **S** as $\mathbf{S} = \mathbf{V} \bullet \mathbf{D} \bullet \mathbf{V}^T$, where $\boldsymbol{D} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$ and where σ_i^2, i=1,..,n, are Eigenvalues (variances), sorted from largest to smallest, and the columns of V are the corresponding Eigenvectors. The largest Eigenvalue, and its corresponding eigenvector, correspond to the direction in the original input spaces containing the largest amount of variation in the original process data. This is considered the first principal component. The second largest Eigenvalue, and its corresponding eigenvector, correspond to the direction, orthogonal to the first principal component, containing the next largest amount of variation. This continues until a new orthogonal vector space, explaining all of the original process data, is created.

[0178] After the Eigenvalues and Eigenvectors are computed, it is necessary to determine which ones are the significant principal components, usually corresponding to the important variation in a process, and which are insignificant, usually corresponding to the noise. It should be understood that a number of methods may be used to determine the significant and insignificant principal components, examples of which have been provided above.

[0179] Whichever method is chosen, the final result is determining the number of significant principal components, **a**, that will be retained for the PCA model. Then, the loading matrix $P \in \Re^{n \times a}$ may be created by taking the first **a** columns (Eigenvectors) from **V**. The projection of the original observations onto the subspace defined by the loading matrix is called the score matrix, and denoted by $\mathbf{T} = \mathbf{X} \bullet \mathbf{P}$. The columns of **T** are called the scores of the PCA model, referred to as t_1, t_2, ... t_a, and the **nxa** matrix **P** is the PCA model.

[0180] Training data file **X**. For the purpose of this aspect of the disclosure, the training data X may be selected by selecting independent process variables that affect the process behavior to be resolved and letting those represent the first $\mathbf{k} \leq \mathbf{n}$ columns of **X**, as well as selecting dependent process variables that describe the process behavior and letting them be represented by the last $\mathbf{n}-\mathbf{k}>0$ columns of **X**. If a regression technique were to be implemented, then the first k process variables in **X** are used to for prediction of the last n-k process variables. However, when developing the PCA model, **X** has k independent process variables only and **n=k**. They are used to create the loading **nxa** matrix **P**, which contains **a**

principal component vectors (loading vectors) used to create scores t_1, t_2, ..., t_a that are then employed to predict dependent variables. In the disclosed approach, the independent and dependent variables are combined to create the training data file **X**.

[0181] Once the PCA model is specified, the loading matrix $P=[p_1,p_2,\ldots,p_a]$, where $p_i=[p_{i,1},..,p_{i,k},p_{i,k+1},..,p_{i,n}]^T$ is a loading vector for i=1,..,a, becomes defined. The first $p_{i,1},..,p_{i,k}$ components of p_i are loading values corresponding to the first **k** independent variables in **X**. Similarly, $p_{i,k+1},..,p_{i,n}$ components are loading values corresponding to the last **n-k** dependent variables in **X**. As a result, **P** defines the space of interrelated loading values that are directly related to the original process variables. Now each dependent variable in **X**, **n-k** of them, has a loading value metric in this space given by

$$z_j = \sqrt{\sum_{i=1}^{a} \frac{p_{i,j}^2}{\sigma_i^2}} ,$$

where j=k+1,...,n, and σ_i is i-th Eigenvalue in **D**.

[0182] The loading value metric z_j is the model outcome of a j^{th} dependent process variable with the following properties. Where **X** and **Y** are data files from the same process operation, and collected at different times, and z_j and v_j are the model outcomes respectively corresponding to **X** and **Y** of the same dependent process variable, then if $z_j \approx v_j$ the process behavior described by the j^{th} process variable is similar for **X** and **Y**. In contrast, if $z_j \neq v_j$, then the process behavior described by the j^{th} process variable is different for **X** and **Y**.

[0183] This property is useful for process diagnostics, and in particular for diagnostics of slowly changing process behavior. The reason is that the loading value metric z_j reflects process behavior over a period of time specified by the operator, and, as such, provides a more complete diagnostics than the most current update given by the regression approach.

[0184] Coker Heater Application. The foregoing technique may used to determine the level of coking in coker heaters. As described above, the data file X was composed of the process variables defined in Table 1 and Fig. 24. In that example, T_{out} was the only dependent variable, while all other variables were independent. In the technique in accordance with this aspect of the disclosure, however, all variables are used to create the PCA model, and PCR is not utilized.

[0185] With such an assignment, the baseline model was created under normal operating conditions (e.g., 0% coking), and the loading value metric Z was computed for T_{out}. Subsequently, the same was done for different percent of coking in fired heaters, and Fig. 42 illustrates the results obtained. As can be seen in Fig. 42, there is a monotonic decrease in Z as percent of coking increases, which implies that there is a structural change in the loading space due to changes in process variable contributions as the percent of coking is varied. This change is collaborative and may not be easily determined by monitoring one variable at the time.

[0186] Based on the foregoing, by placing different limits on Z, this technique may be used in a diagnostics/fault detection method or system. Alerts and alarms issued using this technique are more reliable because each model represents the behavior of Z based on the length of time used to gather the model data. Moreover, while running, the detection method may be entirely transparent to operators. This is because the detection method may be running in the background of the monitoring console, and may be configured to alert operators of abnormal situations only when the set limits are exceeded or violated. The flow chart in Fig. 43 illustrates how the method or system may be implemented in accordance with one embodiment. With this approach, false alarms and alerts may be avoided and performance of process parameters may nonetheless be rigorously monitored with time.

[0187] Referring to Fig. 43, at a block 302 the procedure specifies the size of the data file X and creates the data file X based on training data generated from process variables of the process being monitored. As discussed above, the training data may correspond to the normal on-line operation of the process. At a block 304, the multivariate model, such as a PCA model, is developed from the data file X and evaluated to determine the loading matrix P and the diagonal matrix D. The Z variables are evaluated at a block 306 for the dependent variables of the data file X, as discussed above. At a block 308, the limits L are set to the Z values evaluated at block 308 for the data file X.

[0188] Having generated the limits L, the procedure from blocks 302-306 may be repeated with a new data file Y in order to detect abnormal operations. In particular, data generated from process variables of the system may be received, where the data corresponds to data generated while the process is operating on-line which may correspond to a monitoring state where the process is being monitored for abnormal operations. As discussed above, the data of the data file Y is generally collected at a different time from the same

process. At a block 310, the size of the new data file Y is specified, and the data file Y is created to model the on-line operation of the process. A multivariate model, such as a PCA model, is developed and evaluated at a block 312 to determine the loading matrix P and the diagonal matrix D for the data file Y. The Z values are evaluated at a block 314 for the dependent variables of the data file Y, and the Z values for the data file Y are compared to the limits L at a block 316. If the Z values for the dependent variables of the data file Y exceed the limits, an alert may be generated at a block 318 to indicate an abnormal operation. Otherwise, the process may continue with a new data file generated from the same process, and generally at yet another different time than the data files X and Y. Thus, the procedure may be repeated for various data files corresponding to different time of operation of the process to detect abnormal conditions based on long term data behavior, where the abnormal conditions may relates to slow varying changes in the process. Although the block 316 indicates that an alert may be generated is the Z values are greater than the limits L, it should be recognized that an alert may also be generated if the Z values are less than the limits L.

[0189] With this aspect of the disclosure, a PCA model parameter may be defined and applied for on-line diagnostics, which may be useful in connection with coking in coker heaters and a variety of process equipment faults or abnormal situations. The model parameter may be derived using PCA loadings and variance information, and a loading value metric may be defined for the dependent or independent process variables. In some cases, the disclosed method may be used for observing long term coking, rather than instantaneous or most current changes. For instance, the disclosed method may be used for on-line long term collaborative diagnostics. Alternatively or additionally, the disclosed method may provide an alternative approach to regression analysis. The disclosed method may be implemented in connection with a number of control system platforms, including, for instance, DeltaV™ and Ovation®, and with a variety of process equipment and devices, such as the Rosemount 3420 FF Interface Module. Alternatively, the disclosed method and system may be implemented as a stand alone abnormal situation prevention application. In either case, the disclosed method and system may be configured to generate alerts and otherwise support the regulation of coking levels in fired heaters.

[0190] One of ordinary skill in the art will recognize that the exemplary systems and methods described above may be modified in various ways. For example, blocks may be omitted or reordered, additional blocks may be added, etc. For example, with regard to Fig. 7, the block 146 could be implemented at a different point in the flow. Similarly, the block

148 could be implemented as part of a separate routine, and thus it could actually occur at various points with the flow of Fig. 7 depending upon when a suitable command is received to initiate implementation of the separate routine.

[0191] The above-described examples involving abnormal situation prevention modules and abnormal situation prevention blocks are disclosed with the understanding that practice of the disclosed systems, methods, and techniques is not limited to such contexts. Rather, the disclosed systems, methods, and techniques are well suited for use with any diagnostics system, application, routine, technique or procedure, including those having a different organizational structure, component arrangement, or other collection of discrete parts, units, components, or items, capable of selection for monitoring, data collection, etc. Other diagnostics systems, applications, etc., that specify the process parameters being utilized in the diagnostics may also be developed or otherwise benefit from the systems, methods, and techniques described herein. Such individual specification of the parameters may then be utilized to locate, monitor, and store the process data associated therewith. Furthermore, the disclosed systems, methods, and techniques need not be utilized solely in connection with diagnostic aspects of a process control system, particularly when such aspects have yet to be developed or are in the early stages of development. Rather, the disclosed systems, methods, and techniques are well suited for use with any elements or aspects of a process control system, process plant, or process control network, etc.

[0192] The methods, processes, procedures and techniques described herein may be implemented using any combination of hardware, firmware, and software. Thus, systems and techniques described herein may be implemented in a standard multi-purpose processor or using specifically designed hardware or firmware as desired. When implemented in software, the software may be stored in any computer readable memory such as on a magnetic disk, a laser disk, or other storage medium, in a RAM or ROM or flash memory of a computer, processor, I/O device, field device, interface device, etc. Likewise, the software may be delivered to a user or a process control system via any known or desired delivery method including, for example, on a computer readable disk or other transportable computer storage mechanism or via communication media. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and

not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency, infrared and other wireless media. Thus, the software may be delivered to a user or a process control system via a communication channel such as a telephone line, the Internet, etc. (which are viewed as being the same as or interchangeable with providing such software via a transportable storage medium).

[0193] Thus, while the present invention has been described with reference to specific examples, which are intended to be illustrative only and not to be limiting of the invention, it will be apparent to those of ordinary skill in the art that changes, additions or deletions may be made to the disclosed embodiments without departing from the spirit and scope of the invention.

Claims

1. A system for facilitating detection of a steady-state operation of a process in a process plant, the system comprising:

a data collection tool adapted to collect process data from a process within the process plant, wherein the collected process data is representative of an operation of the process and wherein the collected process data is generated from a plurality of process variables;

an analysis tool comprising a multivariate statistical analysis engine adapted to generate a representation of the operation of the process based on a set of the collected process data representative of the operation of the process, wherein the analysis tool is adapted to execute the representation based on the set of the collected process data to generate a first outcome corresponding to the largest amount of variation in the process; and

a statistical calculation tool adapted to perform a statistical calculation on the first outcome corresponding to the largest amount of variation in the process to generate a statistical measure of the first outcome, and adapted to determine a steady-state operation status related to the process based on the statistical measure.

2. The system of claim 1, wherein the analysis tool is adapted to execute the representation based on the set of the collected process data representative of the operation of the process to generate a second outcome corresponding to a significant amount of variation in the process different from the largest amount of variation in the process.

3. The system of claim 2, wherein the statistical calculation tool comprises a first and second statistical calculation tool, wherein the first statistical calculation tool is adapted to perform a statistical calculation on the first outcome to generate a statistical measure of the first outcome and the second statistical calculation tool is adapted to perform a statistical calculation on the second outcome generate a statistical measure of the second outcome, and wherein the first and second statistical calculation tools are each adapted to determine the steady-state operation status related to the process based on the corresponding statistical measures.

4. The system of claim 3, wherein the first and second statistical calculation tools are each adapted to generate an indication pertaining to the corresponding determination of a steady-state operation status related to the process, the system further comprising a

monitoring tool adapted to analyze each of the steady-state indications of the first and second calculation tools and adapted to determine the steady-state operation status of the process based on any one of the steady-state indications generated by the first and second calculation tools.

5. The system of claim 1, wherein the statistical calculation tool is adapted to perform a statistical calculation on the first outcome to generate a first mean and a first standard deviation of the first outcome corresponding to a first sampling window from the set of collected process data and adapted to generate a second mean and a second standard deviation of the first outcome corresponding to a second sampling window from the set of collected process data.

6. The system of claim 5, wherein the statistical calculation tool is adapted to determine the steady-state operation status related to the process based on a difference between the first and second mean as compared to at least one of the first or second standard deviations.

7. The system of claim 6, wherein the statistical calculation tool is adapted to determine the steady-state operation status related to the process based on the difference between the first and second mean as compared to the minimum of the first and second standard deviations.

8. The system of claim 1, wherein the multivariate statistical analysis engine is adapted to perform a principal component analysis to generate the representation of the operation of the process based on the set of the collected process data representative of the operation of the process, wherein the first outcome corresponds to the most significant component generated from the principal component analysis.

9. A method of facilitating detection of a steady-state operation of a process in a process plant, the method comprising:

collecting process data from a process within the process plant, wherein the collected process data is representative of an operation of the process and wherein the collected process data is generated from a plurality of process variables;

generating a multivariate statistical representation of the operation of the process based on a set of the collected process data representative of the operation of the process;

generating a first outcome corresponding to the largest amount of variation in the process from the multivariate statistical representation;

generating a statistical measure of the first outcome; and

determining the presence of a steady-state operation status related to the process based on the first statistical measure.

10. The method of claim 9, further comprising generating a second outcome corresponding to a significant amount of variation in the process from the multivariate statistical representation, wherein the significant amount of variation in the process is different from the largest amount of variation in the process.

11. The method of claim 10, further comprising generating a statistical measure of the second outcome, wherein determining the presence of a steady-state operation related to the process comprises determining the presence of a steady-state operation status related to the process based on the statistical measures of the first and second outcomes.

12. The method of claim 11, further comprising:

generating a first indication pertaining to the determination of a steady-state operation status based on the statistical measure of the first outcome; and

generating a second indication pertaining to the determination of a steady-state operation status based on the statistical measure of the second outcome;

wherein determining the presence of a steady-state operation status related to the process comprises determining the presence of a steady-state operation status related to the process based on any one of the first and second steady-state indications.

13. The method of claim 9, wherein generating a statistical measure of the first outcome comprises:

generating a first mean and a first standard deviation of the first outcome corresponding to a first sampling window from the set of collected process data; and

generating a second mean and a second standard deviation of the first outcome corresponding to a second sampling window from the set of collected process data.

14. The method of claim 13, wherein determining the presence of a steady-state operation status related to the process comprises determining the presence of a steady-state operation status related to the process based on a difference between the first and second mean as compared to at least one of the first or second standard deviations.

15. The method of claim 14, wherein determining the presence of a steady-state operation status related to the process comprises determining the presence of a steady-state operation status related to the process based on the difference between the first and second mean as compared to the minimum of the first and second standard deviations.

16. The method of claim 9, wherein generating a multivariate statistical representation of the operation of the process comprises performing a principal component analysis to generate a representation of the operation of the process based on the set of the collected process data representative of the operation of the process, and wherein generating a first outcome comprises generating a first outcome corresponding to the most significant component generated from the principal component analysis.

17. A method of facilitating detection of a steady-state operation of a process in a process plant, the method comprising:

collecting process data from a process within the process plant, wherein the collected process data is representative of an operation of the process and wherein the collected process data is generated from a plurality of process variables of the process comprising a first data space having a plurality of dimensions;

generating a model of the operation of the process using a set of the collected process data of the process, wherein the model comprises a measure of the operation of the process when the process is operating at different times within a second data space having fewer dimensions than the first data space;

generating a plurality of outputs from the model of the operation of the process, each output corresponding to a different significant variation in the process;

generating a statistical measure for each of the plurality of outputs; and

determining the presence of a steady-state operation status of the process based on any one of the plurality of statistical measures.

18. The method of claim 17, wherein generating a model comprises performing a principal component analysis on the set of the collected process data to represent the operation of the process in a loading matrix defining a subspace of the first data space, wherein each output corresponds to a different significant component generated from the principal component analysis.

19. The method of claim 17, wherein generating a statistical measure for each of the plurality of outputs comprises generating a first and second mean and a first and second standard deviation for each of the plurality of outputs, wherein the first mean and first standard deviation correspond to a first sampling window from the set of collected process data and the second mean and second standard deviation correspond to a second sampling window from the set of collected process data, and wherein determining the presence of a steady-state operation status of the process comprises determining the presence of a steady-state operation status of the process based on a difference between the first and second mean as compared to at least one of the first or second standard deviations for any one of the plurality of outputs.

20. The method of claim 19, wherein determining the presence of a steady-state operation status of the process comprises determining the presence of a steady-state operation status of the process based on a difference between the first and second mean as compared to the minimum of the first and second standard deviations for any one of the plurality of outputs.

21. A system for facilitating detection of a steady-state operation of a process in a process plant, the system comprising:

a data collection tool adapted to collect process data from a process within the process plant, wherein the collected process data is representative of an operation of the process wherein the collected process data is generated from a plurality of process variables of the process comprising a first data space having a plurality of dimensions;

an analysis tool adapted to generate a model of the operation of the process based on a set of the collected process data of the process, wherein the model comprises a measure of the operation of the process when the process is operating at different times within a second data space having fewer dimensions than the first data space, and wherein the analysis tool is adapted to execute the model of the operation of the process to generate a plurality of outputs each corresponding to a different significant variation in the process; and

a statistical calculation tool adapted to perform a statistical calculation on each of the plurality of outputs corresponding to a different significant variation in the process to generate a statistical measure for each of the plurality of outputs, and adapted to determine a steady-state operation status related to the process based on any one of the plurality of statistical measures.

22. The system of claim 21, wherein the analysis tool comprises a multivariate statistical engine adapted to perform a principal component analysis on the set of the collected process data to represent the operation of the process in a loading matrix defining a subspace of the first data space, wherein each output corresponds to a different significant component generated from the principal component analysis.

23. The system of claim 21, wherein the statistical calculation tool is adapted to perform a statistical calculation on each of the plurality of outputs to generate a first and second mean and a first and second standard deviation for each of the plurality of outputs, wherein the first mean and first standard deviation correspond to a first sampling window from the set of collected process data and the second mean and second standard deviation correspond to a second sampling window from the set of collected process data, and wherein the statistical calculation tool is adapted to determine the presence of a steady-state operation status of the process based on a difference between the first and second mean as compared to at least one of the first or second standard deviations for any one of the plurality of outputs.

24. The system of claim 23, wherein the statistical calculation tool is adapted to determine the steady-state operation status of the process based on a difference between the first and second mean as compared to the minimum of the first and second standard deviations for any one of the plurality of outputs.

FIG. 1

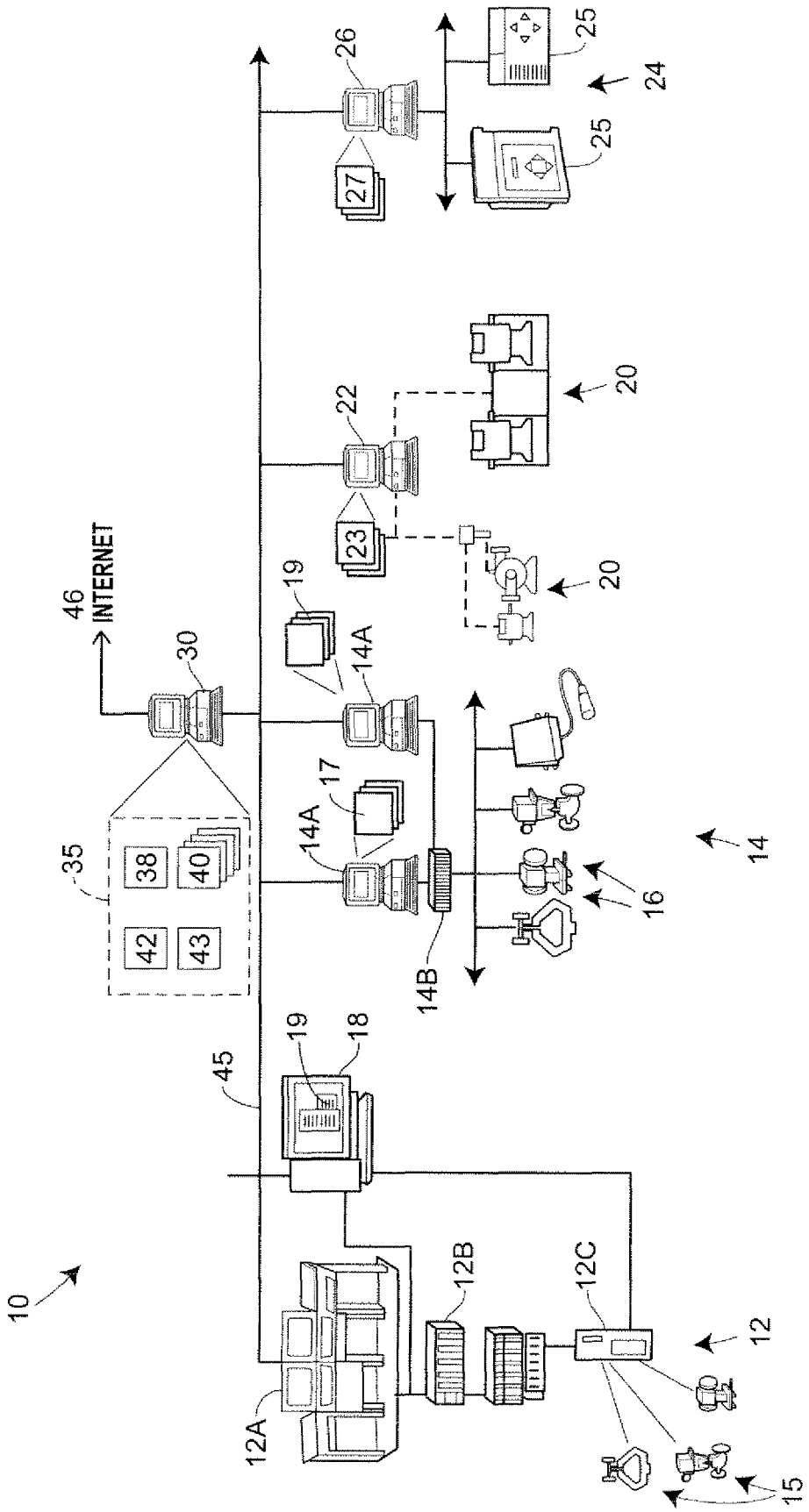

10
12
12A
12B
12C
14
14A
14B
15
16
17
18
19
20
22
23
24
25
26
27
30
35
38
40
42
43
45
46
INTERNET

FIG. 2

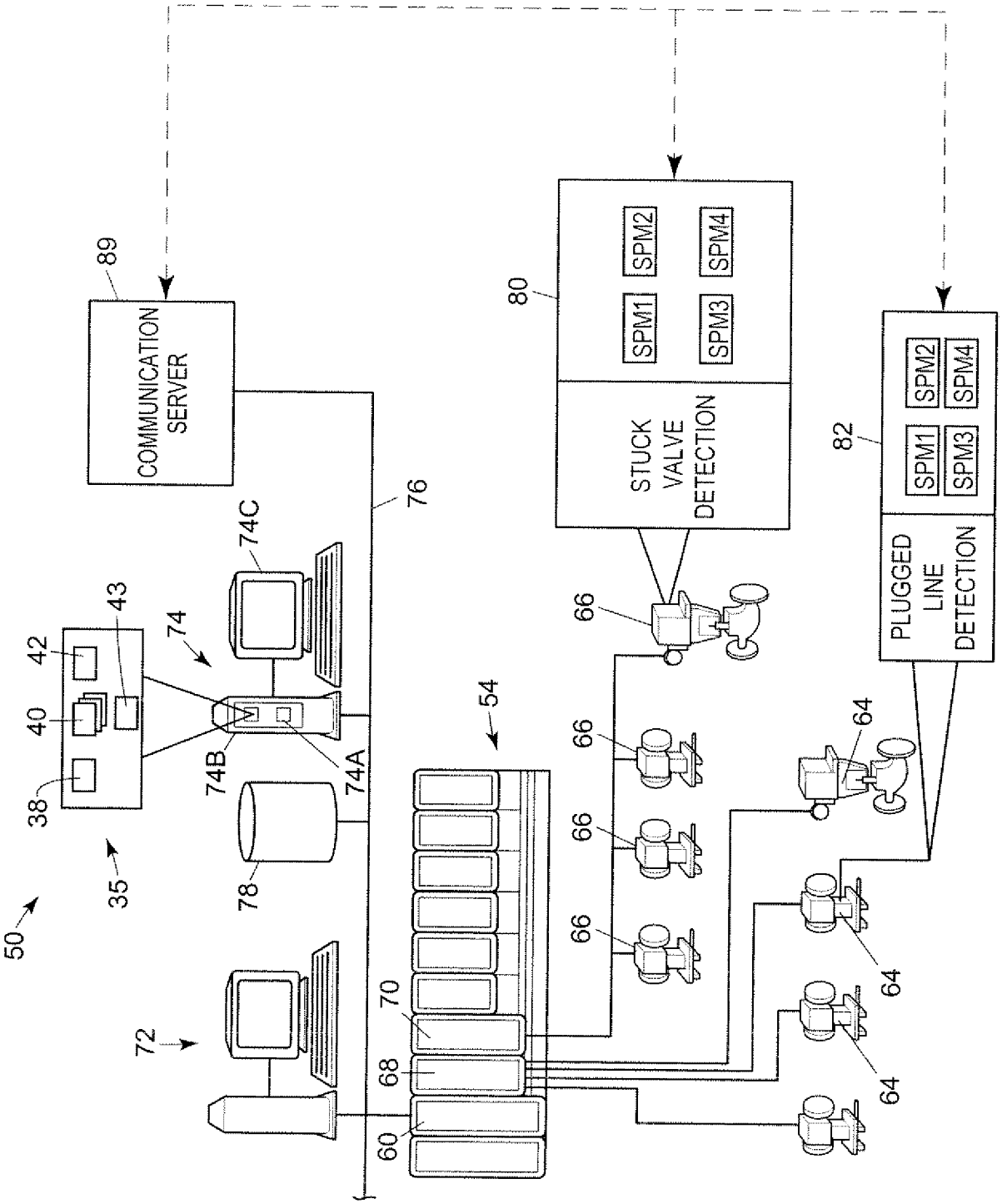

89
COMMUNICATION SERVER
80
SPM1
SPM2
SPM3
SPM4
STUCK VALVE DETECTION
82
PLUGGED LINE DETECTION
76
74C
74
74B
74A
78
42
43
40
38
35
50
54
66
64
72
70
68
60

FIG. 3

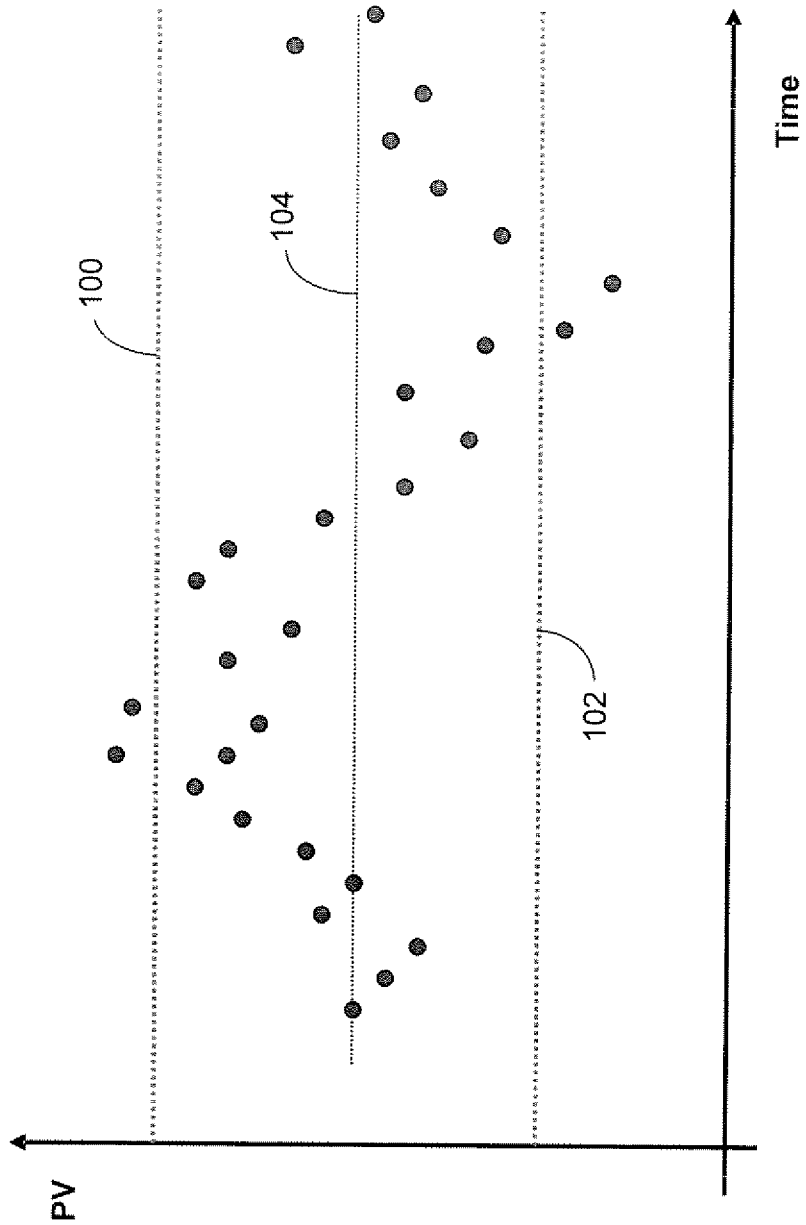


FIG. 4

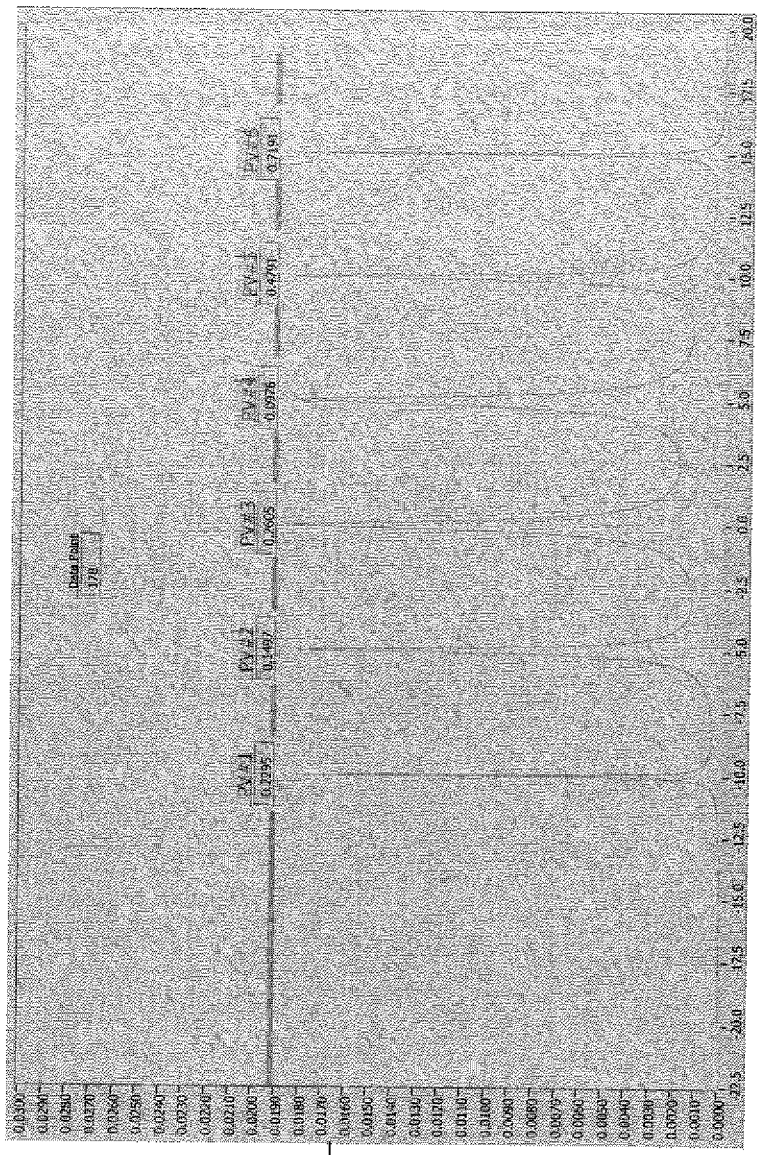
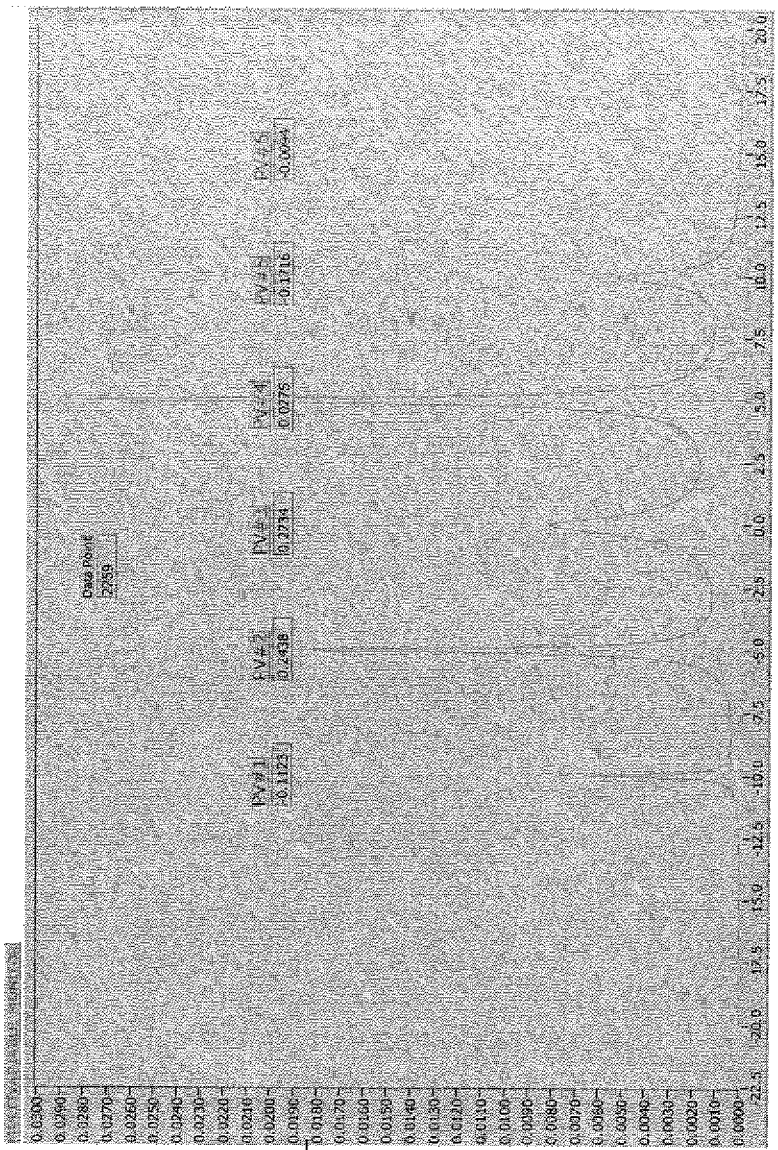


FIG. 5



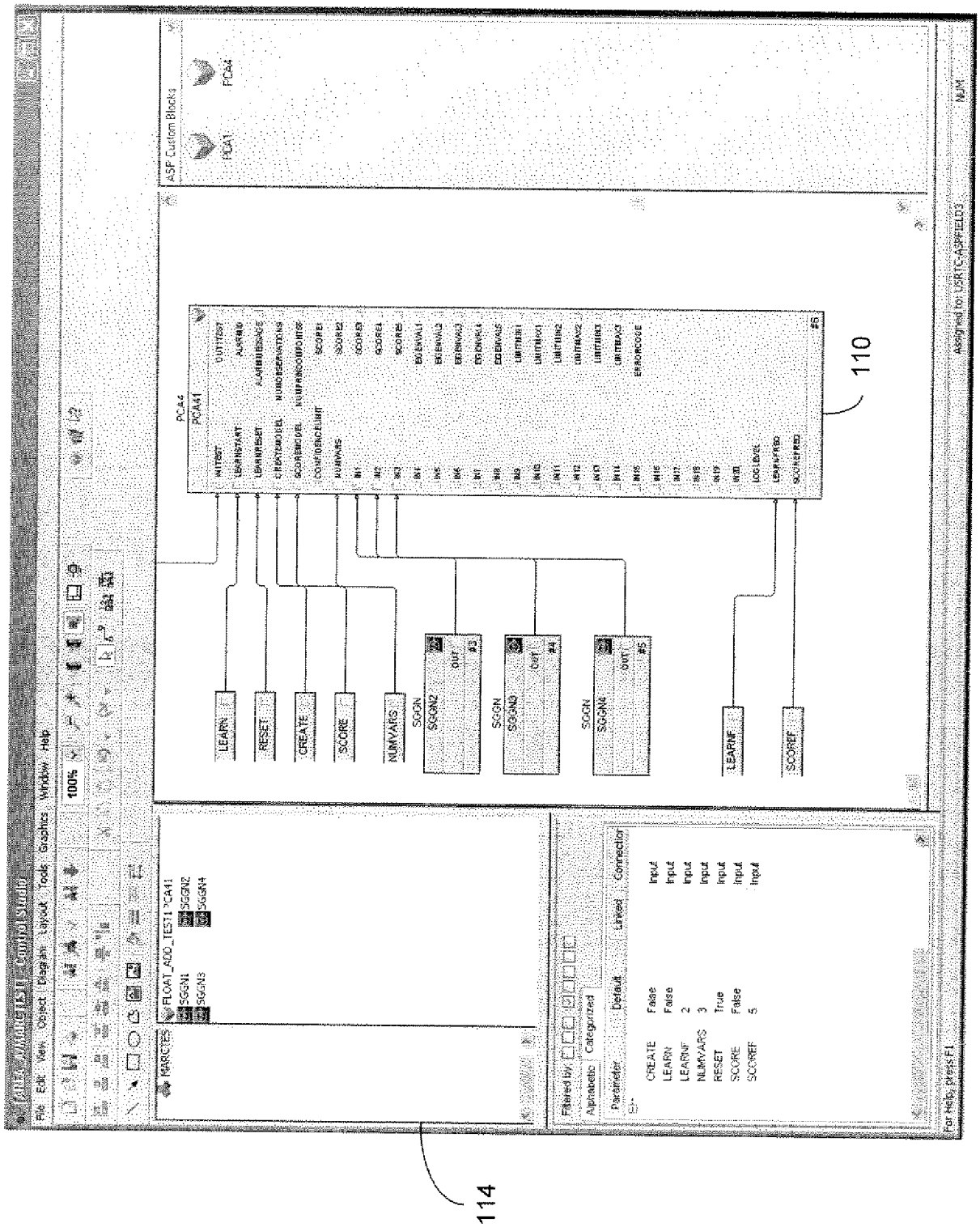


FIG. 6

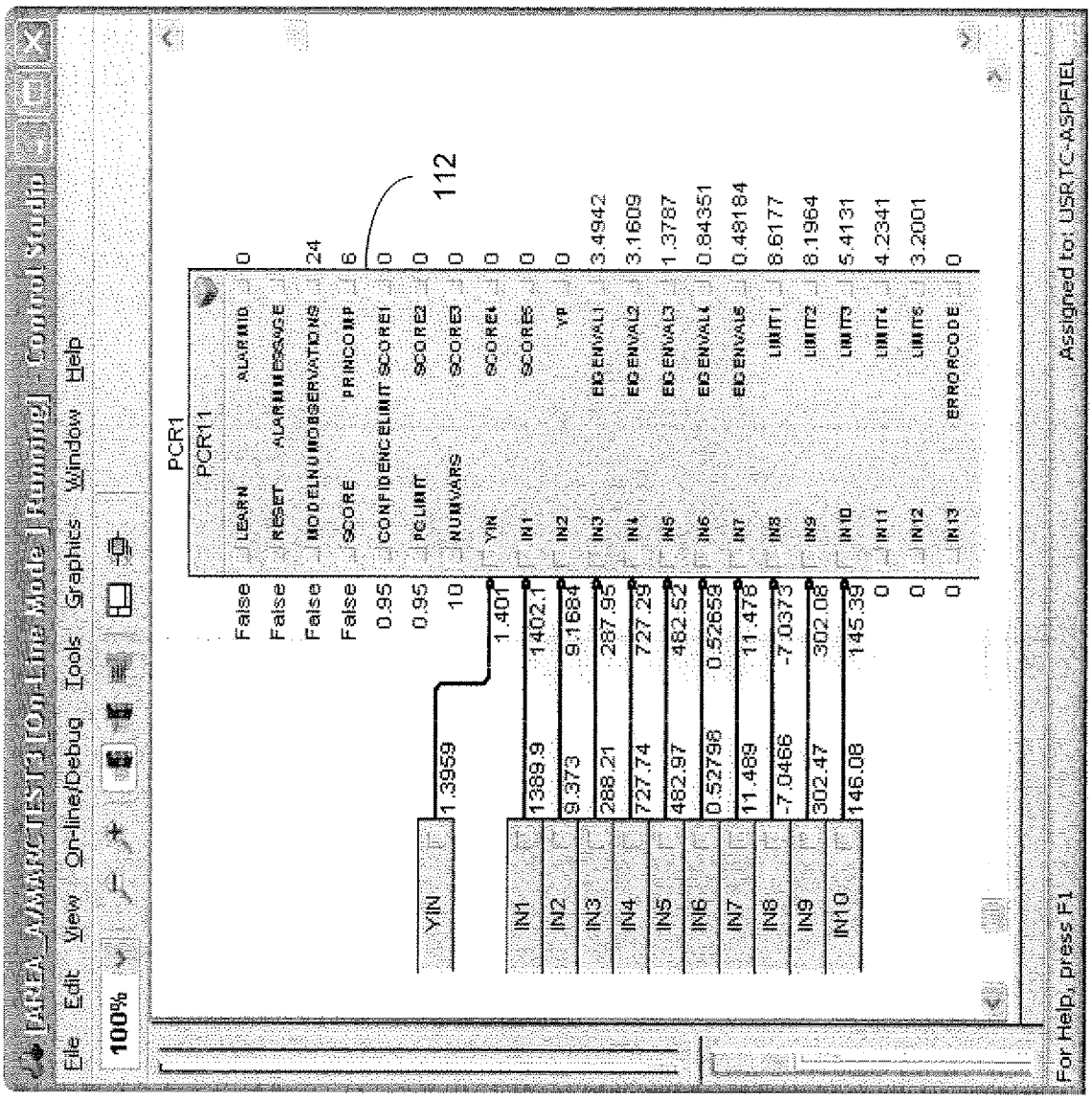
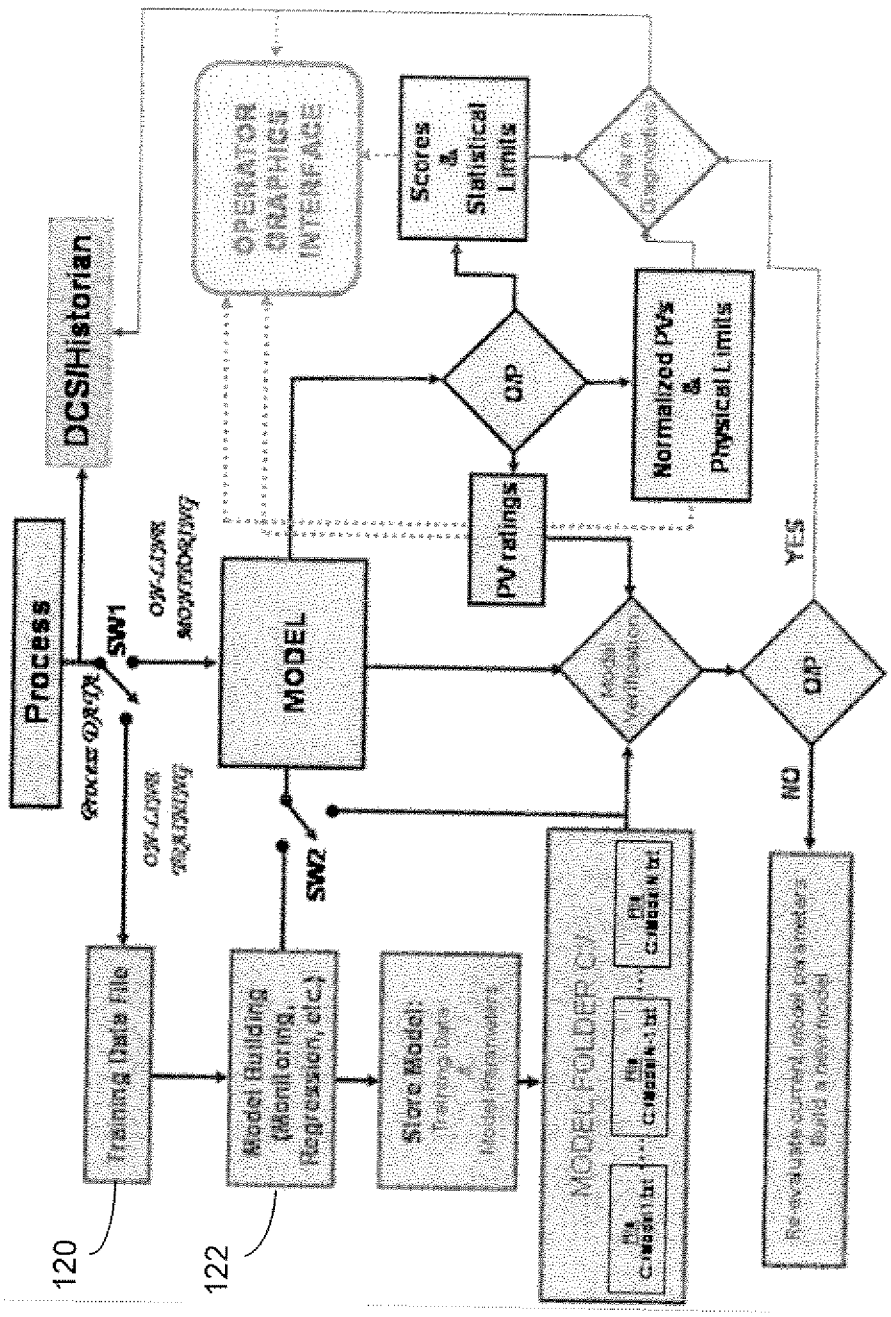

PCR1
PCR11
LEARN
RESET
MODELNUMOBSERVATIONS
SCORE
CONFIDENCELIMIT
PCLIMIT
NUMVARS
YIN
IN1
IN2
IN3
IN4
IN5
IN6
IN7
IN8
IN9
IN10
IN11
IN12
IN13
ALARMID
ALARMMESSAGE
PRINCOMP
SCORE1
SCORE2
SCORE3
SCORE4
SCORE5
YP
EIGENVAL1
EIGENVAL2
EIGENVAL3
EIGENVAL4
EIGENVAL5
LIMIT1
LIMIT2
LIMIT3
LIMIT4
LIMIT5
ERRORCODE
112
For Help, press F1
Assigned to: USRTC-ASPFIEL

FIG. 7

FIG. 8


DCS/Historian
OPERATOR GRAPHICS INTERFACE
Scores & Statistical Limits
Out of Operation
OP
Normalized PVs & Physical Limits
PV ratings
Process
SW1
MODEL
Model Verification
OP
YES
NO
SW2
Training Data File
Model Building (Monitoring, Regression, etc.)
Store Model
MODEL FOLDER
120
122

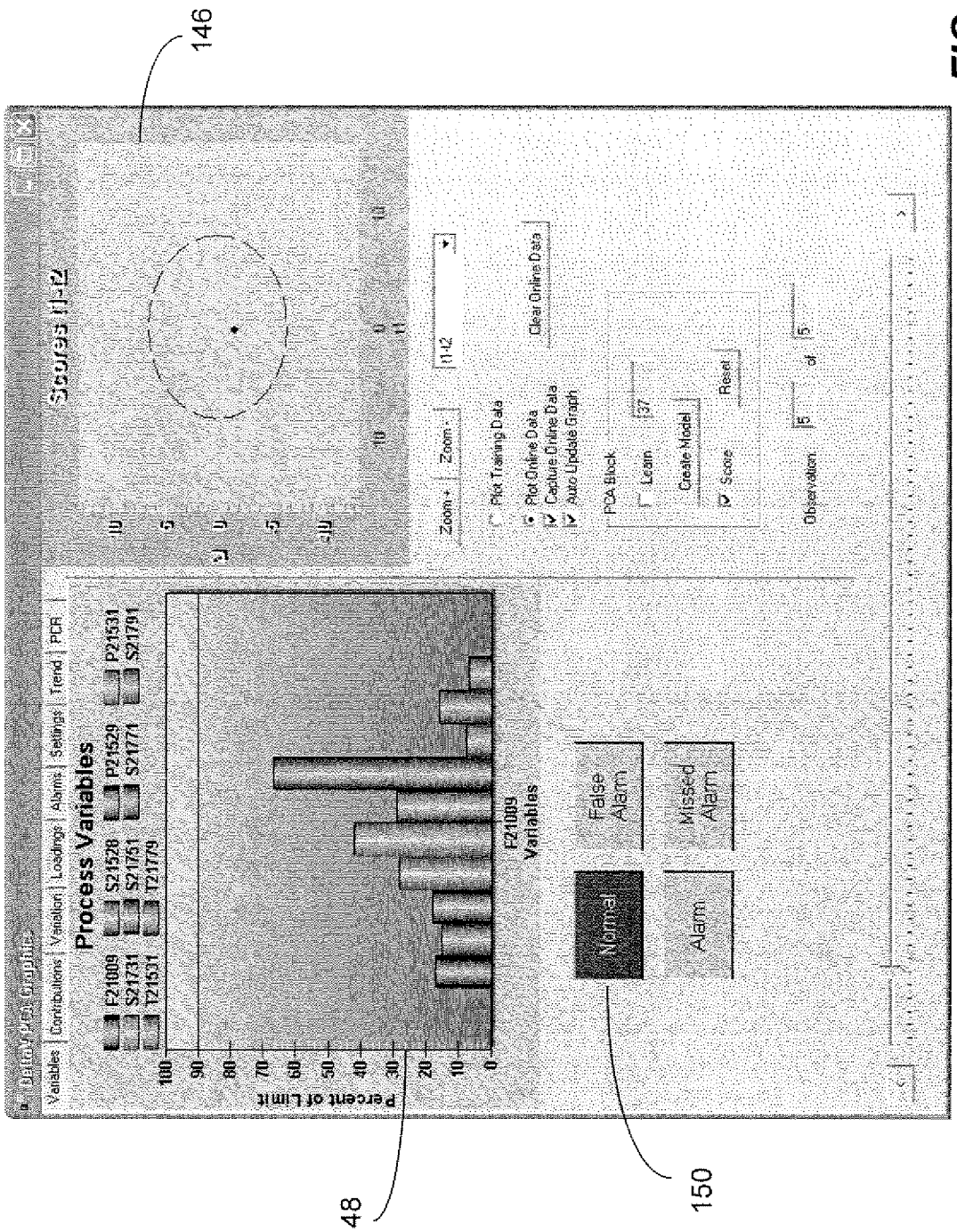

146
148
150
Process Variables
Percent of Limit
F21009 Variables
Normal
Alarm
False Alarm
Missed Alarm
Scores Plot
Zoom
Reset
Create Model
Clear Online Data
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
PCA Block
Learn
Score
Observation

FIG. 9

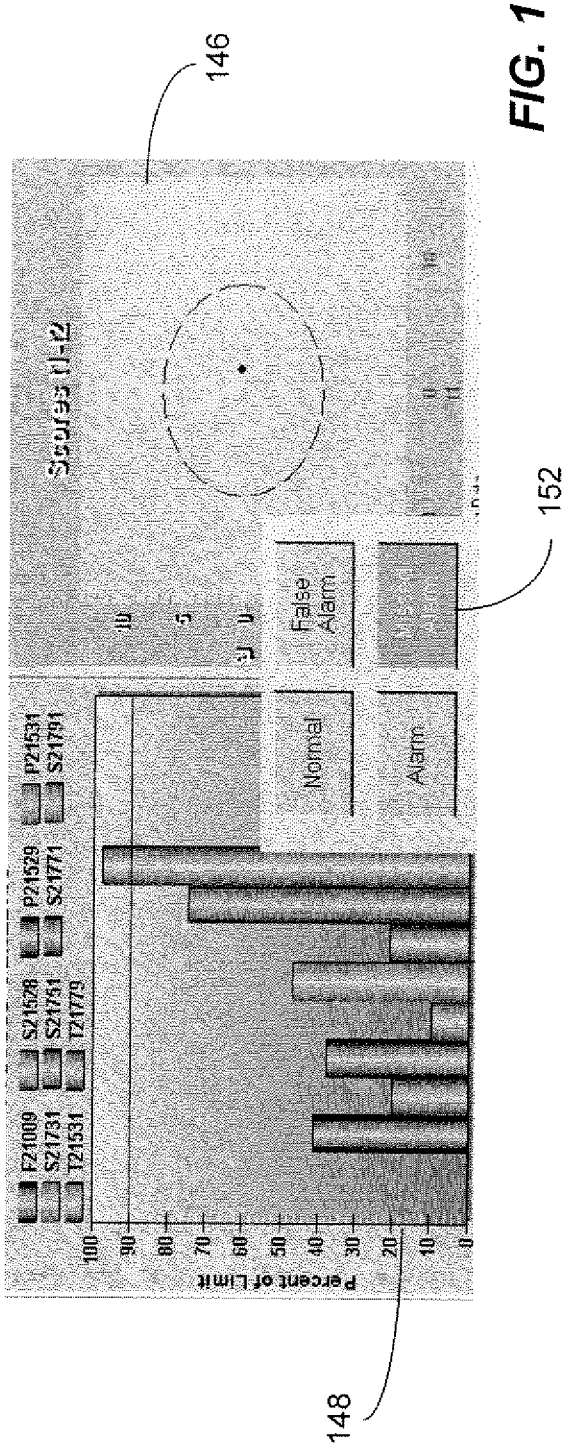

146
152
148
Percent of Limit
100
90
80
70
60
50
40
30
20
10
0
F21009
S21731
T21531
S21528
S21751
T21779
P21529
S21771
P21531
S21791
Normal
Alarm
False
Alarm

FIG. 10

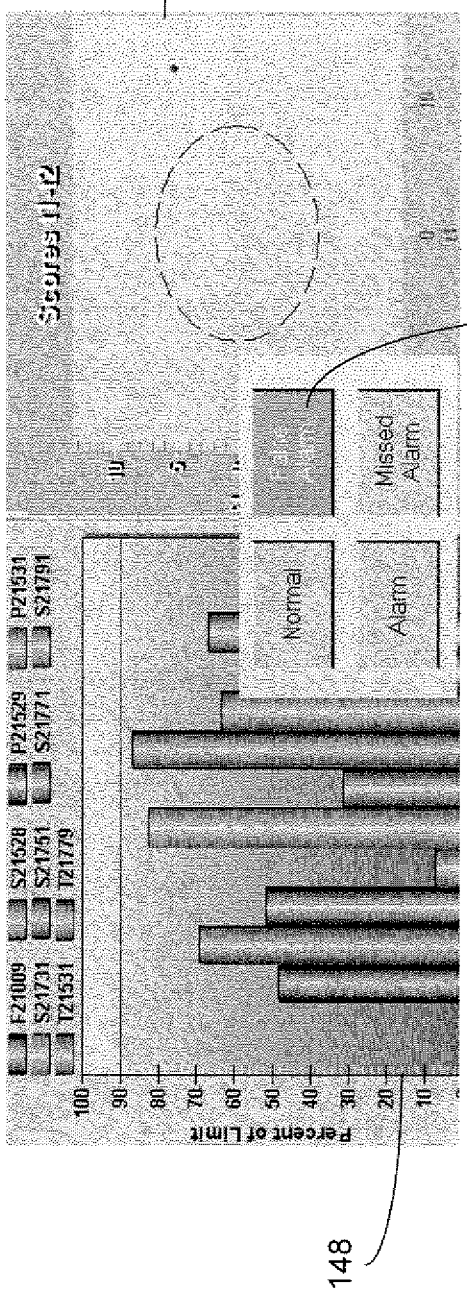

146
154
148
Normal
Alarm
Missed Alarm
Percent of Limit
100
90
80
70
60
50
40
30
20
10
F21009
S21731
T21531
S21528
S21751
T21779
P21529
S21771
P21531
S21791

FIG. 11

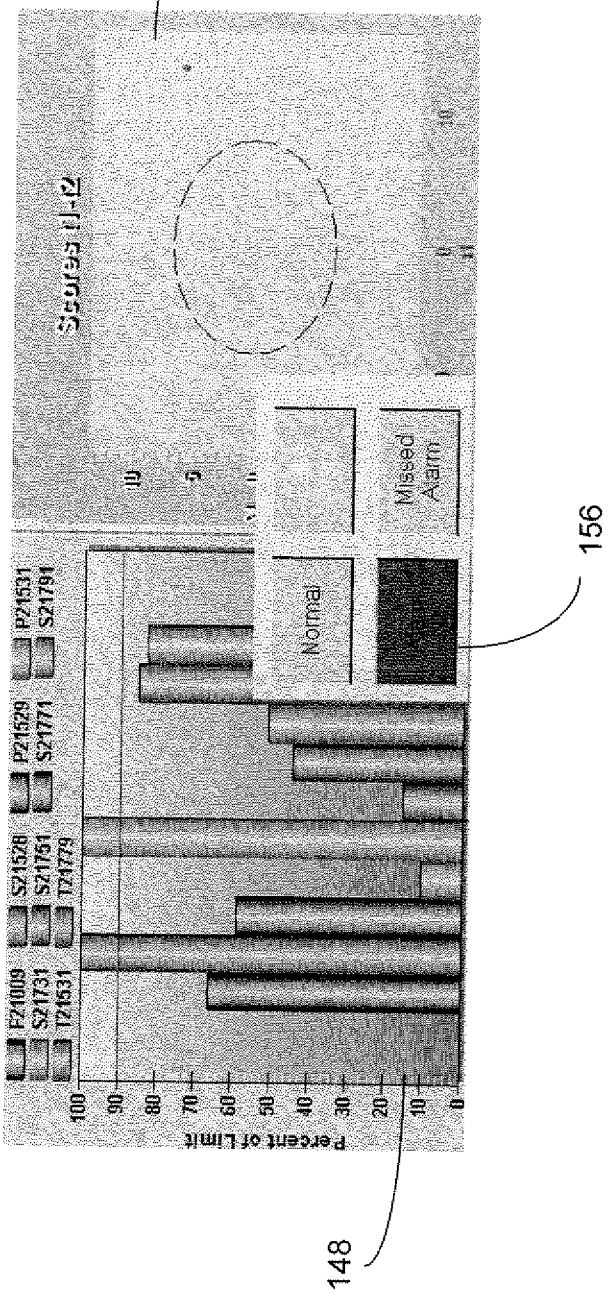

146
156
148
Normal
Missed Alarm
Percent of Limit
P21009
S21731
T21531
S21528
S21751
T21779
P21529
S21771
P21531
S21791

FIG. 12

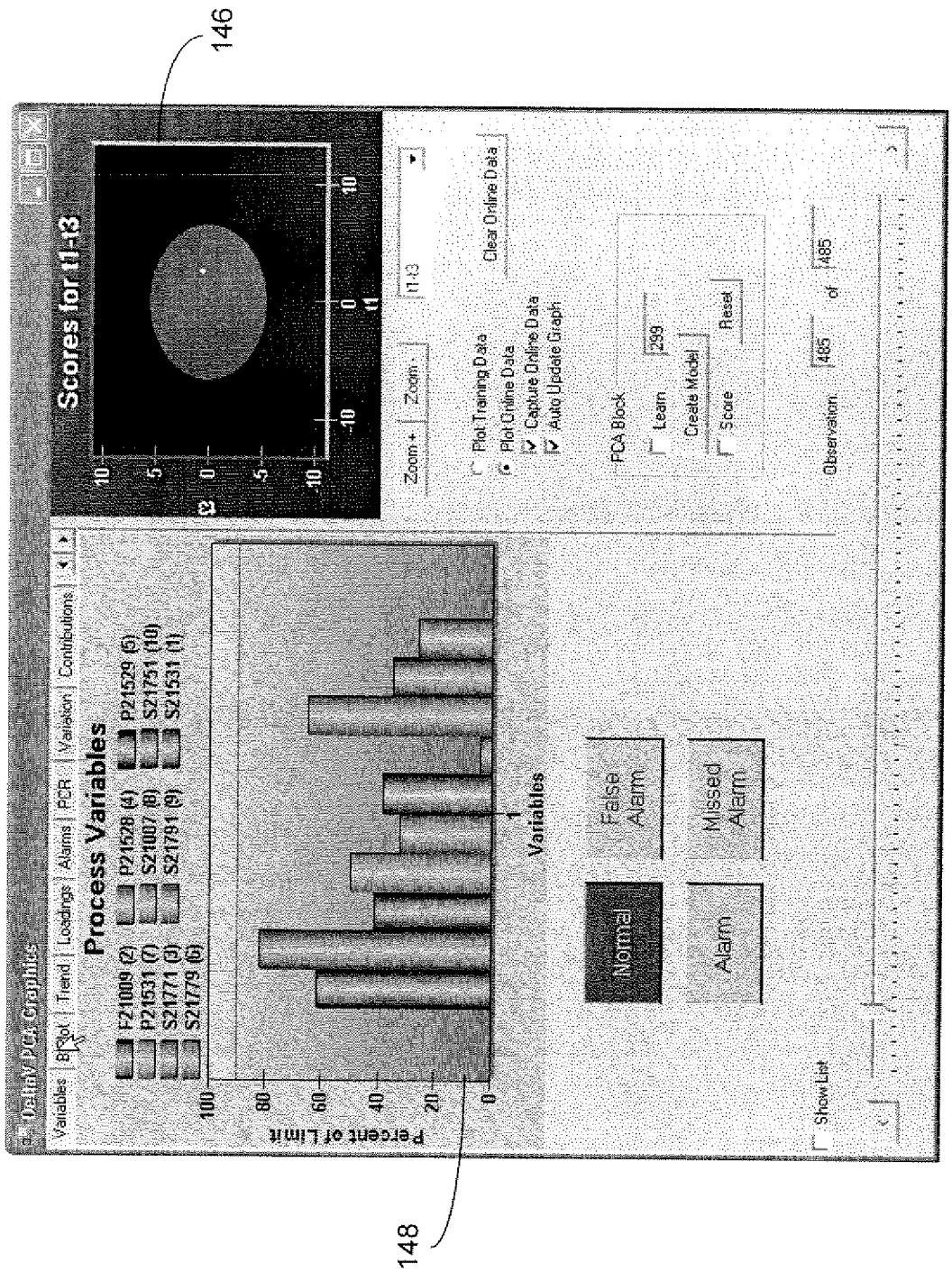

146
148
Scores for t[-3
Process Variables
Percent of Limit
Variables
Normal
Alarm
False Alarm
Missed Alarm
Zoom +
Zoom -
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
PCA Block
Learn
Create Model
Score
Reset
Observation
Show List

FIG. 13

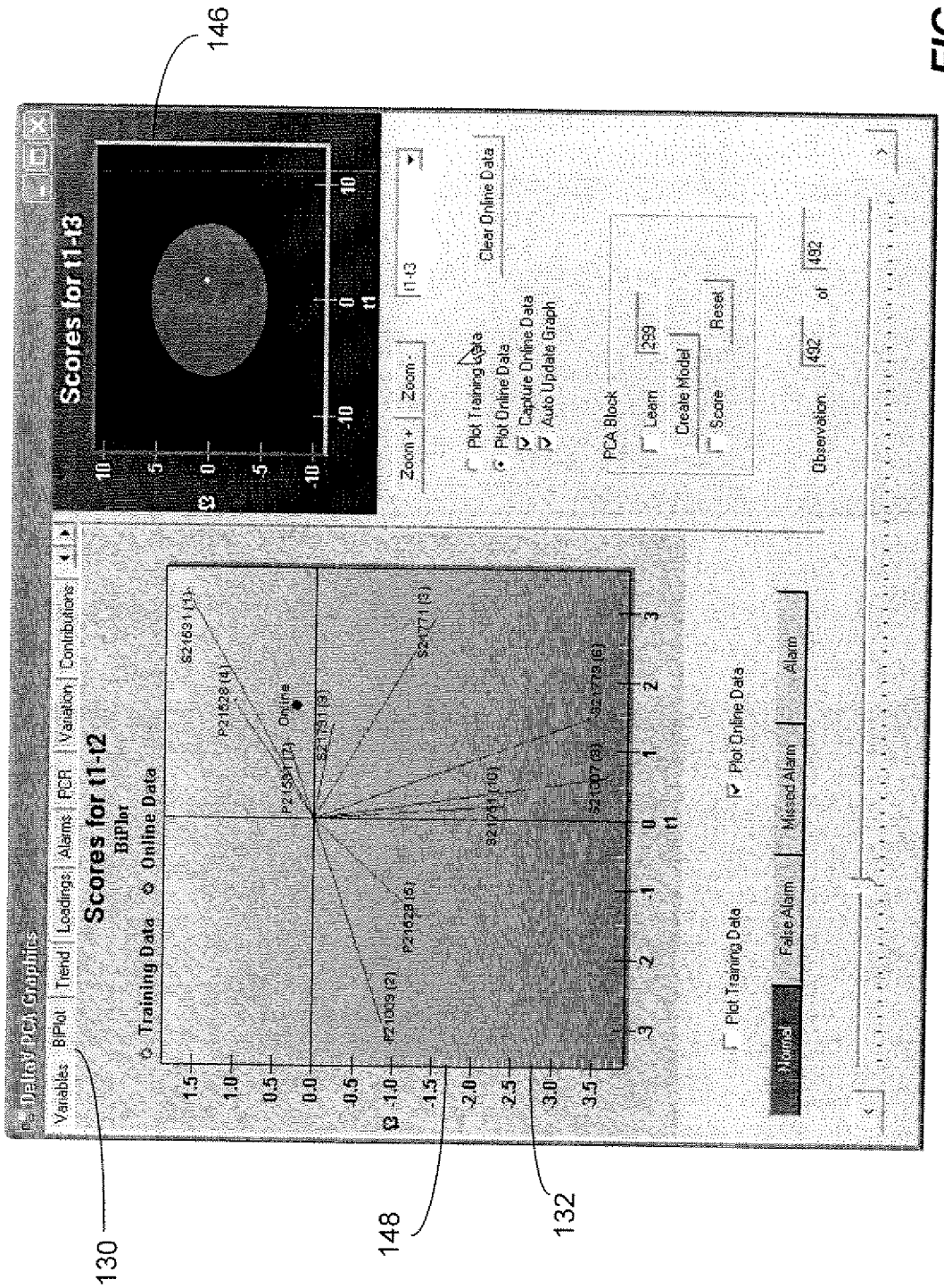

146
Scores for t1-3
Scores for t1-2
BiPlot
Training Data
Online Data
Variables
BiPlot
Trend
Loadings
Alarms
RCR
Variation
Contributions
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Zoom +
Zoom -
PCA Block
Learn
Create Model
Score
Reset
Observation
Plot Training Data
Plot Online Data
Normal
False Alarm
Missed Alarm
Alarm
130
148
132

FIG. 14

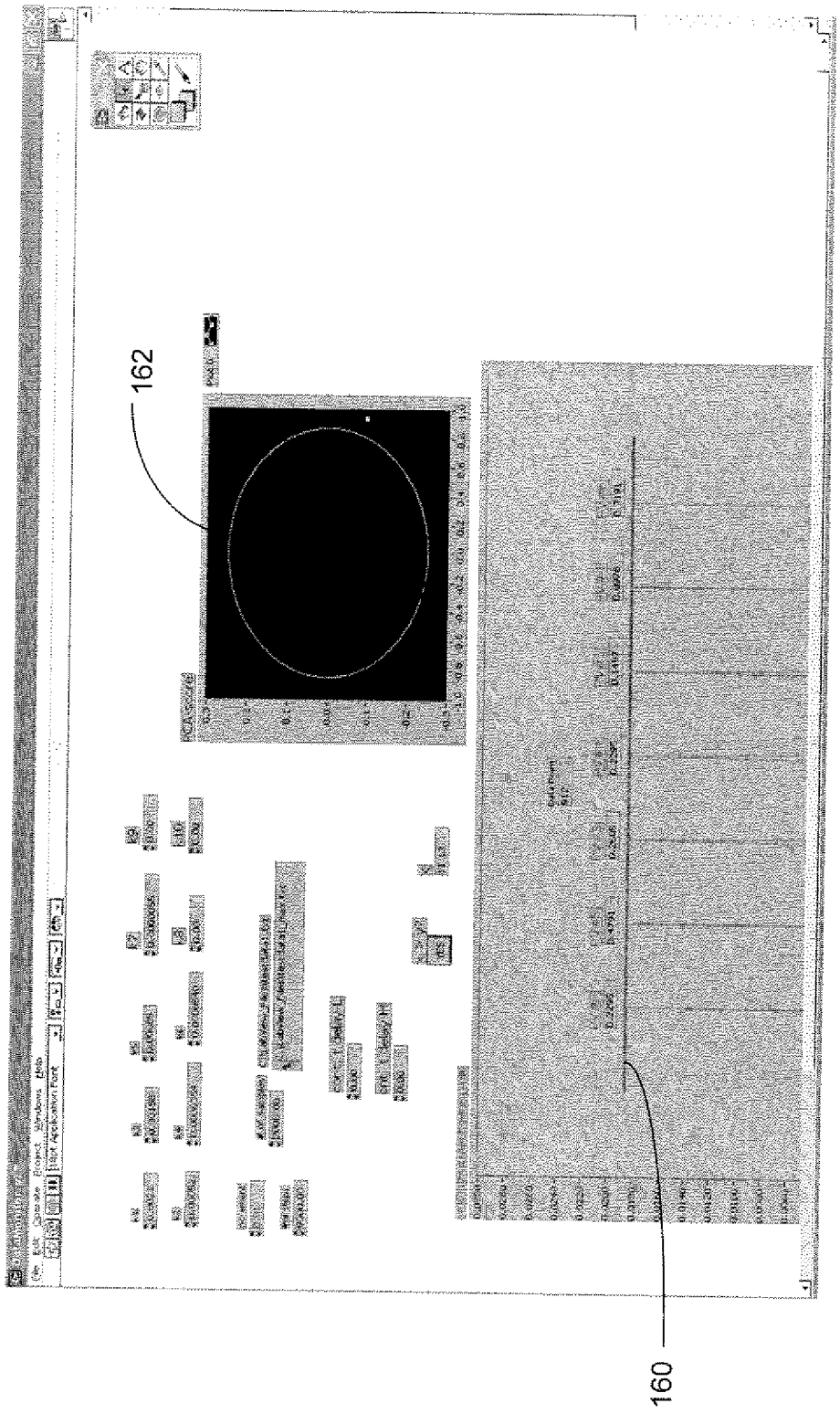


FIG. 15

FIG. 16

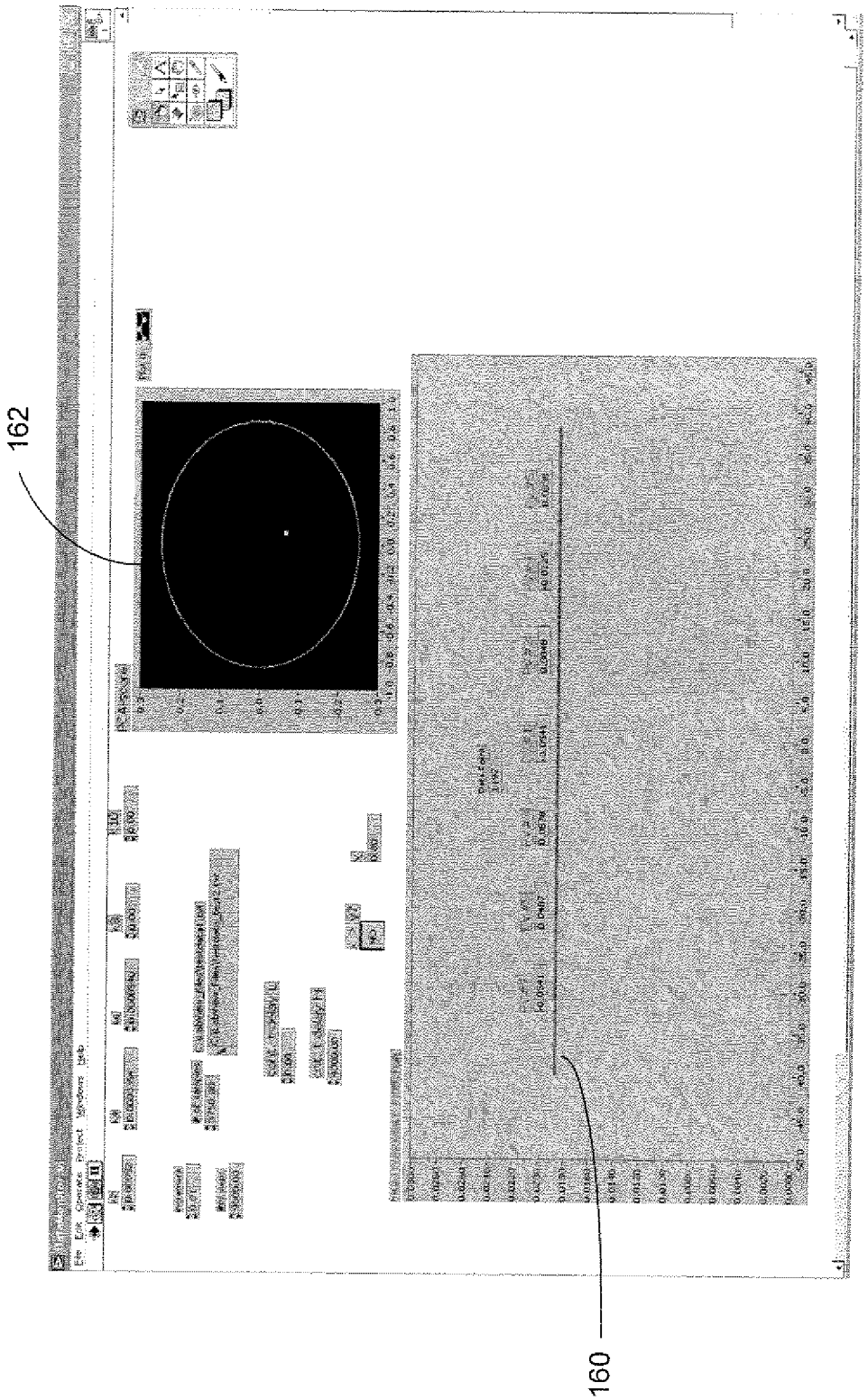


FIG. 17

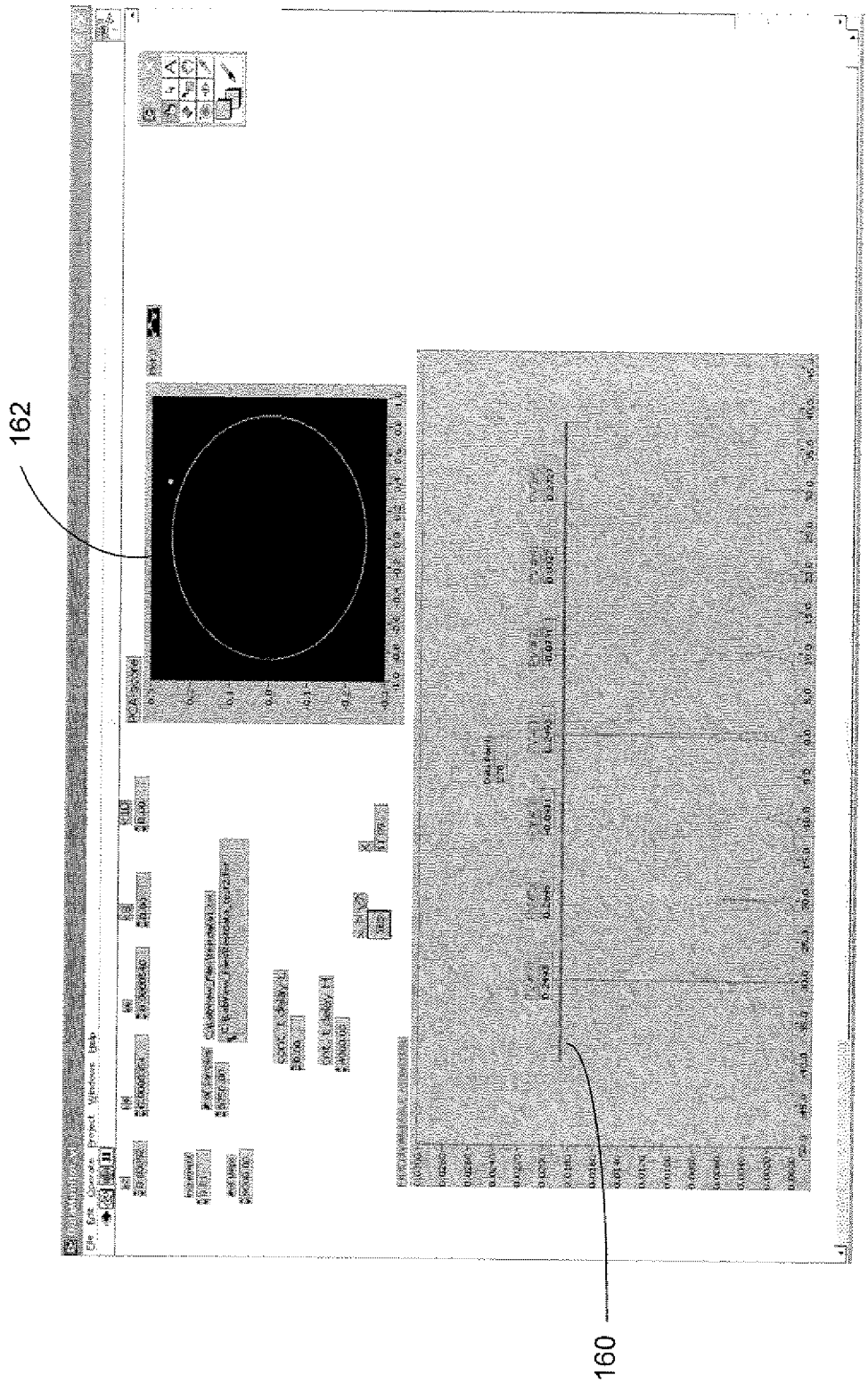


FIG. 18

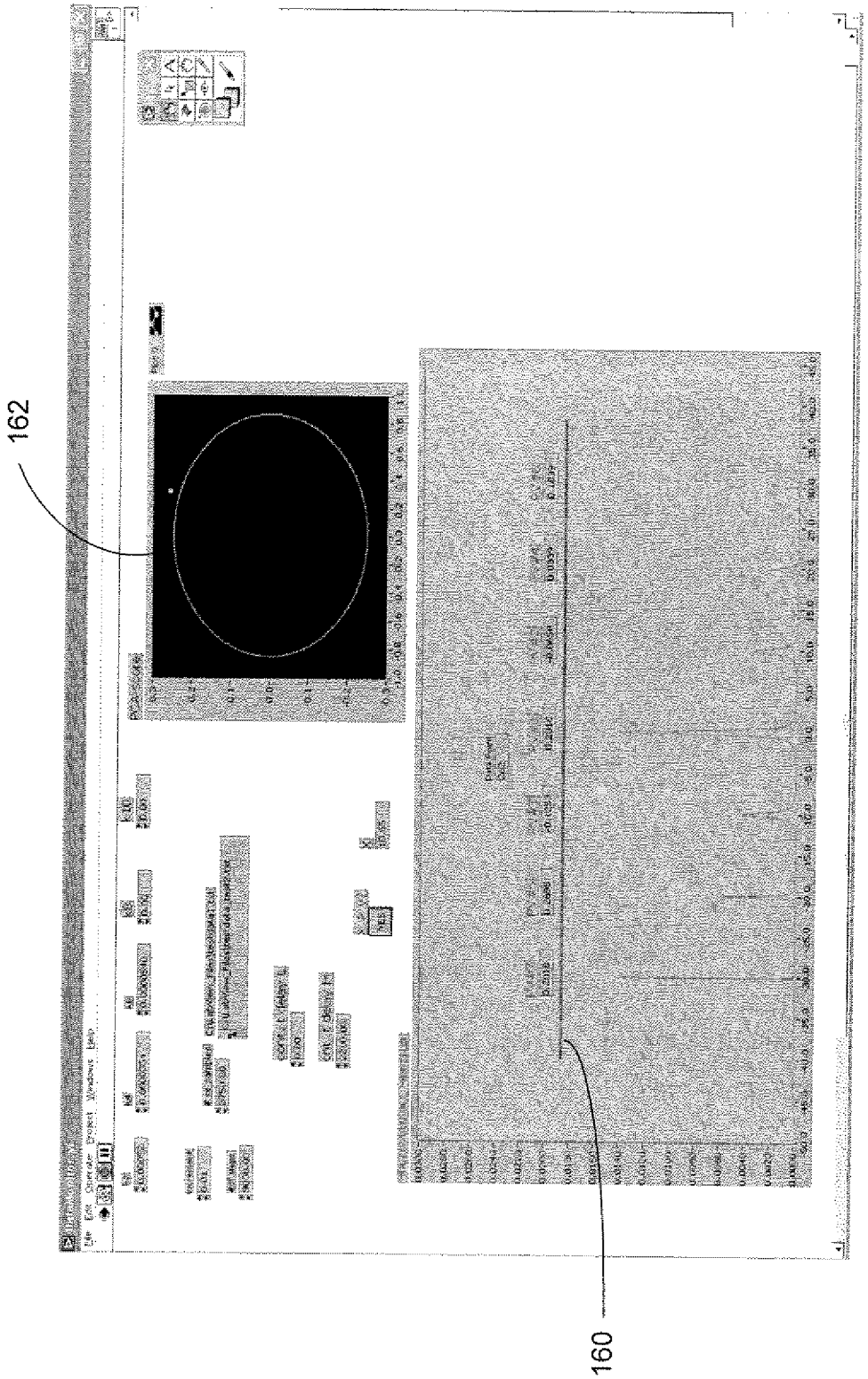
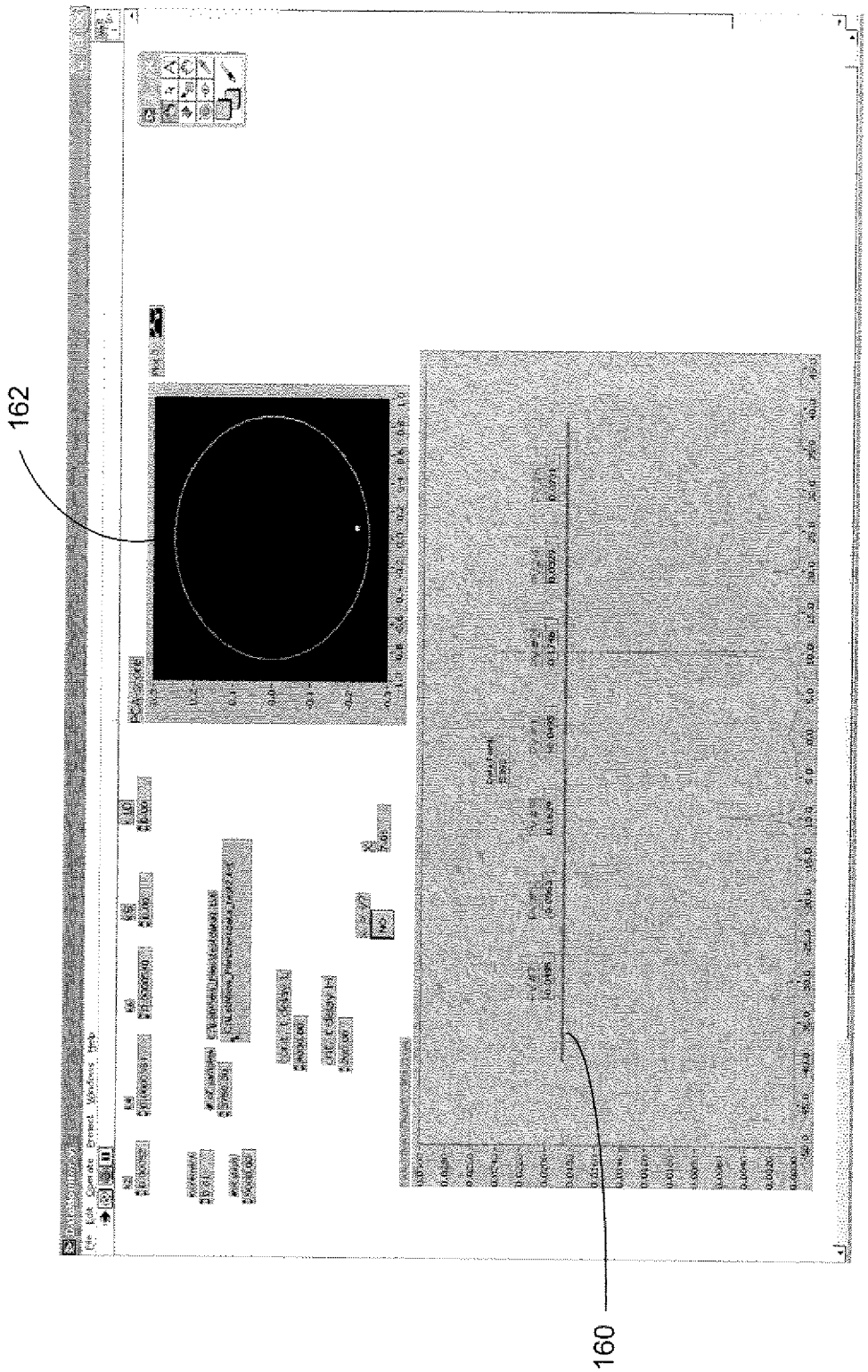


FIG. 19



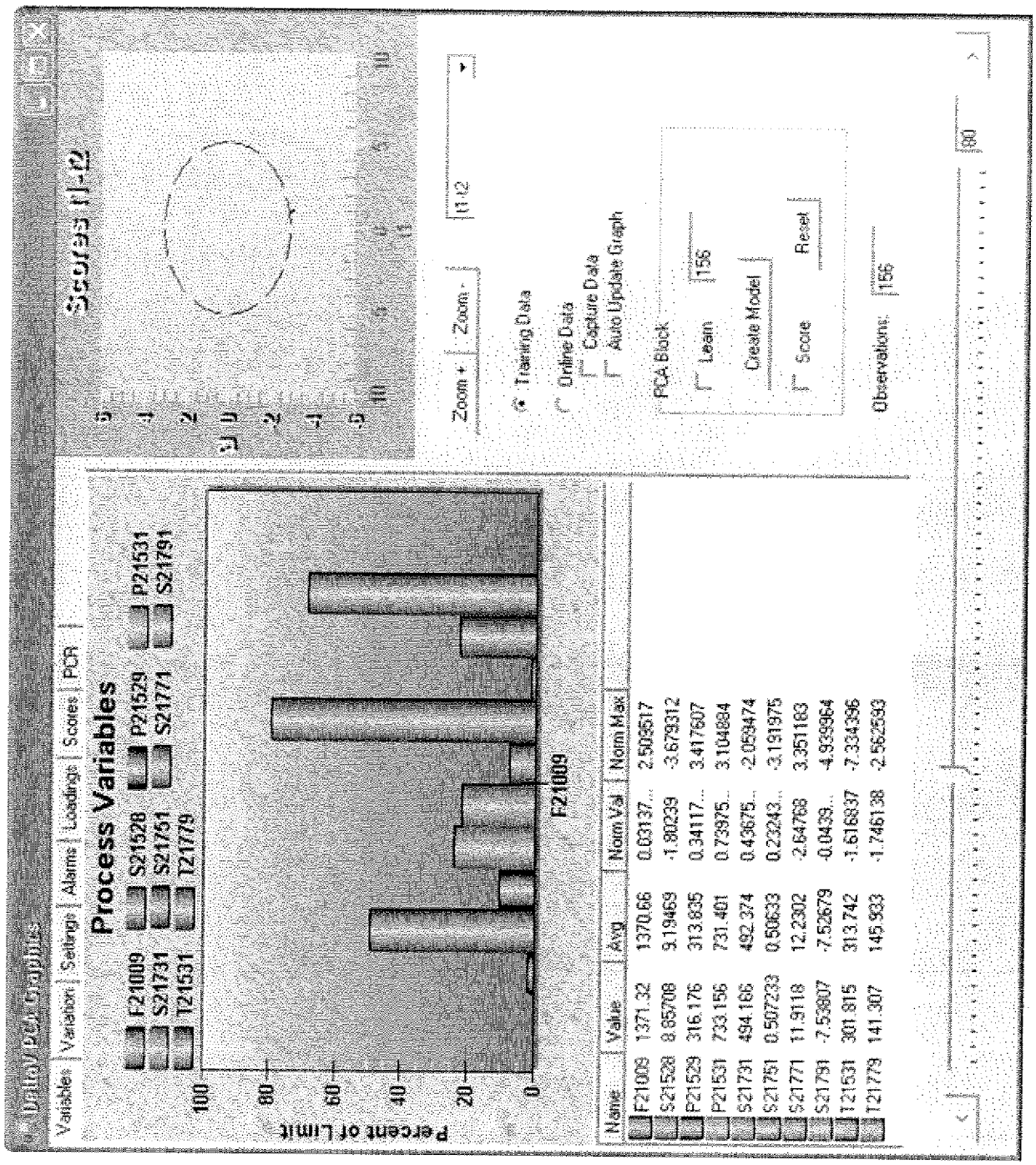

Variables
Variation
Settings
Alarms
Loadings
Scores
PCR
Process Variables
F21009
S21528
P21529
P21531
S21731
S21751
S21771
S21791
T21531
T21779
Percent of Limit
F21009
Scores [1-2]
Zoom +
Zoom -
1:1.2
Training Data
Online Data
Capture Data
Auto Update Graph
PCA Block
Learn
156
Create Model
Score
Reset
Observations:
156
Name
Value
Avg
Norm Val
Norm Max
F21009 1371.32 1370.66 0.03137... 2.509517
S21528 8.65708 9.19469 -1.80239 -3.679312
P21529 316.176 313.835 0.34117... 3.417607
P21531 733.156 731.401 0.73975... 3.104884
S21731 494.186 492.374 0.43675... -2.059474
S21751 0.507233 0.50633 0.23243... -3.191975
S21771 11.9118 12.2302 -2.64768 3.351163
S21791 -7.53807 -7.52679 -0.0439... -4.939964
T21531 301.815 313.742 -1.61683 -7.334396
T21779 141.307 145.933 -1.74613 -2.562593

FIG. 20

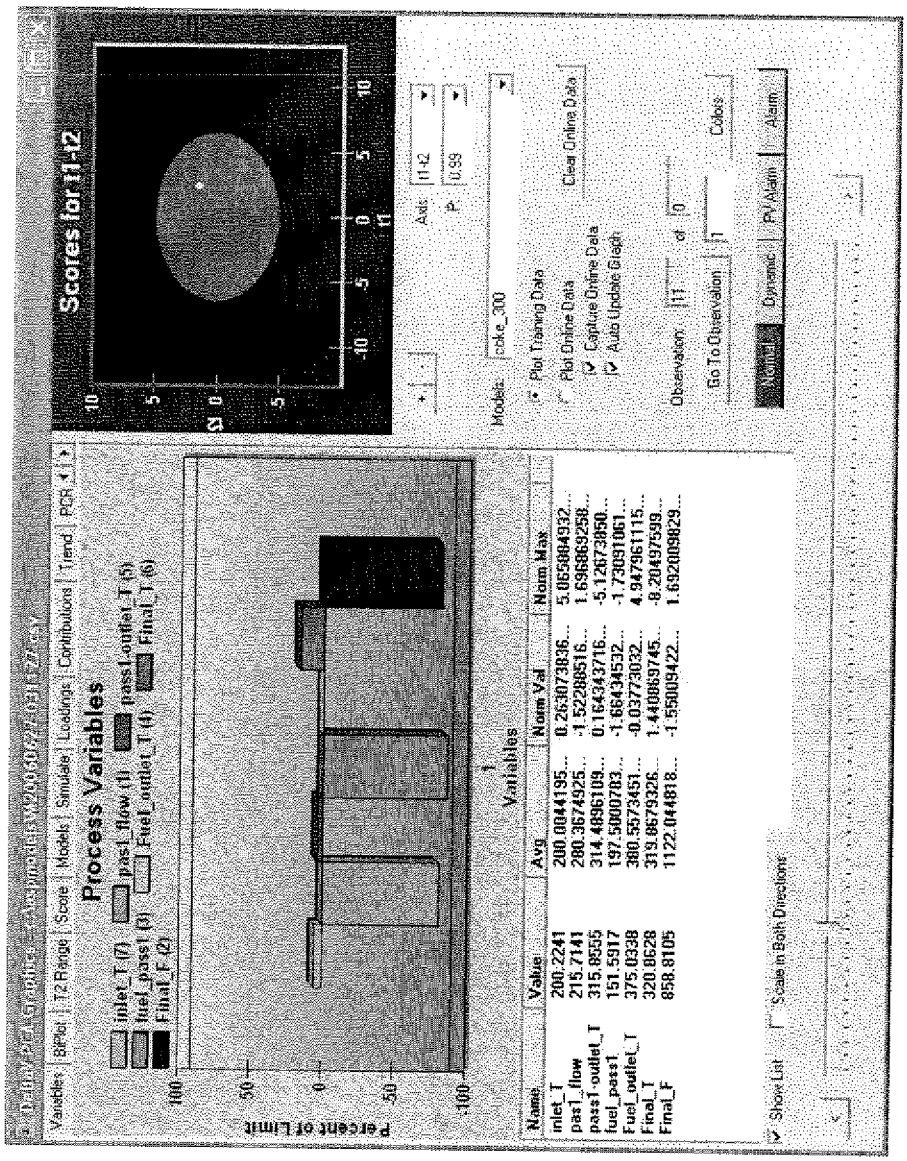

Process Variables
inlet_T (7)
past_flow (1)
pass1-outlet_T (5)
fuel_pass1 (3)
Fuel_outlet_T (4)
Final_T (6)
Final_F (2)
Percent of Limit
Variables
Scores for t1 t2
Name
Value
Avg
Norm Val
Norm Max
inlet_T 200.2241 200.0044195... 0.2630738361... 5.065884932...
past_flow 216.7141 280.3674925... -1.522886516... 1.696869258...
pass1-outlet_T 315.8555 314.4896109... 0.1643437161... -5.126738501...
fuel_pass1 151.5917 197.5000783... -1.664345321... -1.730910611...
Fuel_outlet_T 375.0338 380.5573451... -0.03773032... 4.947961115...
Final_T 320.8628 319.8679326... 1.440869745... -8.204975991...
Final_F 858.8105 1122.044818... -1.550009422... 1.692009829...
Axis:
P:
Models:
cake_300
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation:
of
Go To Observation
Colors
Normal
Dynamic
PI Alarm
Alarm
Show List
Scale in Both Directions

FIG. 21

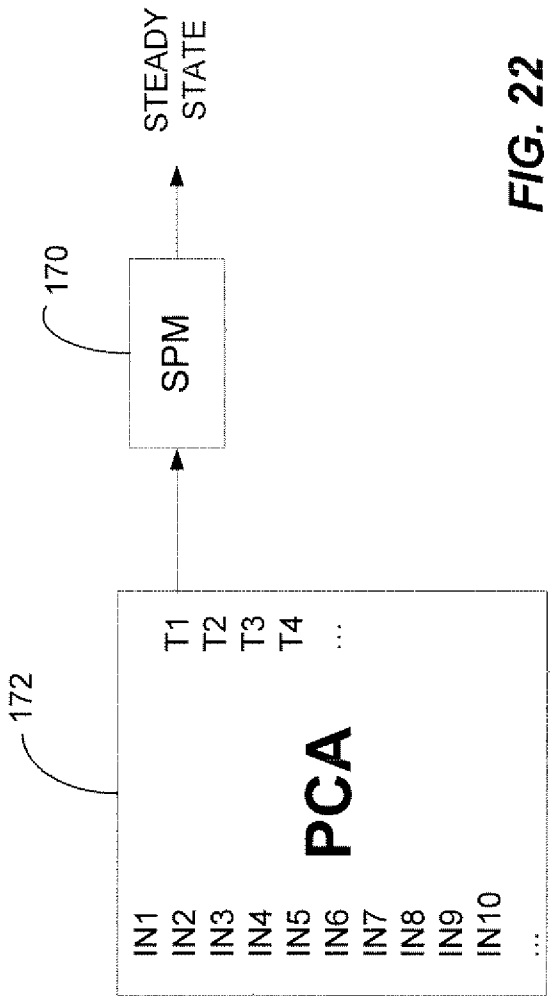

STEADY STATE
170
SPM
172
PCA
T1
T2
T3
T4
...
IN1
IN2
IN3
IN4
IN5
IN6
IN7
IN8
IN9
IN10
...

FIG. 22

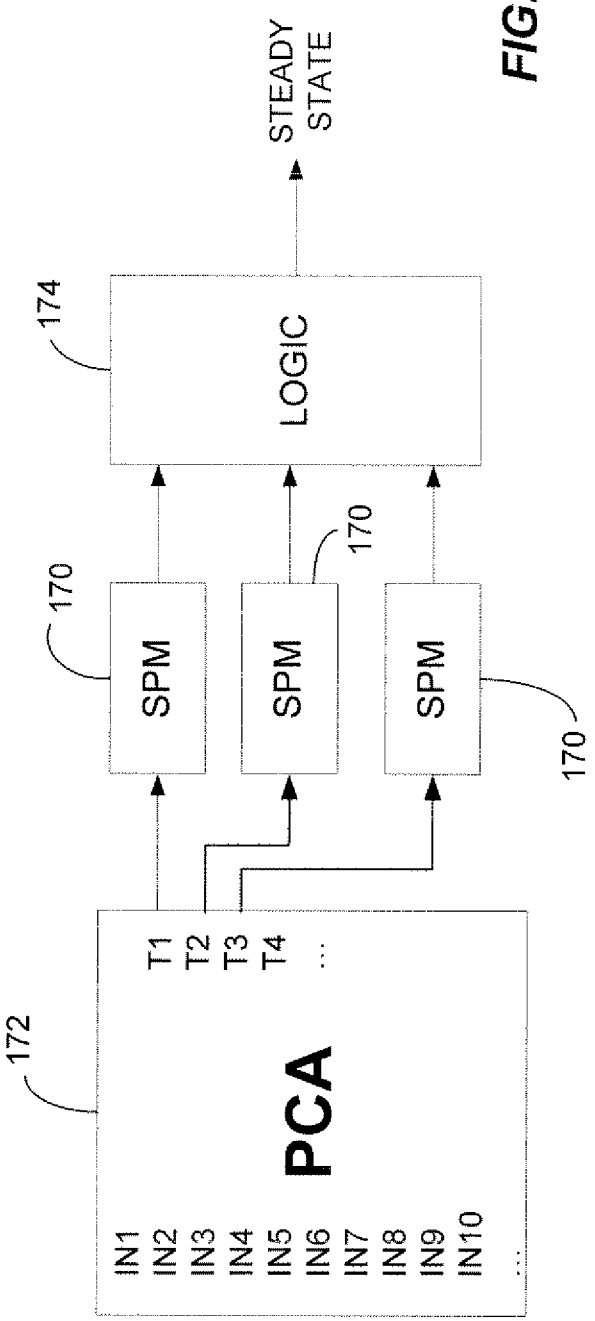

STEADY STATE
174
LOGIC
170
SPM
SPM
SPM
170
170
172
PCA
T1
T2
T3
T4
...
IN1
IN2
IN3
IN4
IN5
IN6
IN7
IN8
IN9
IN10
...

FIG. 23

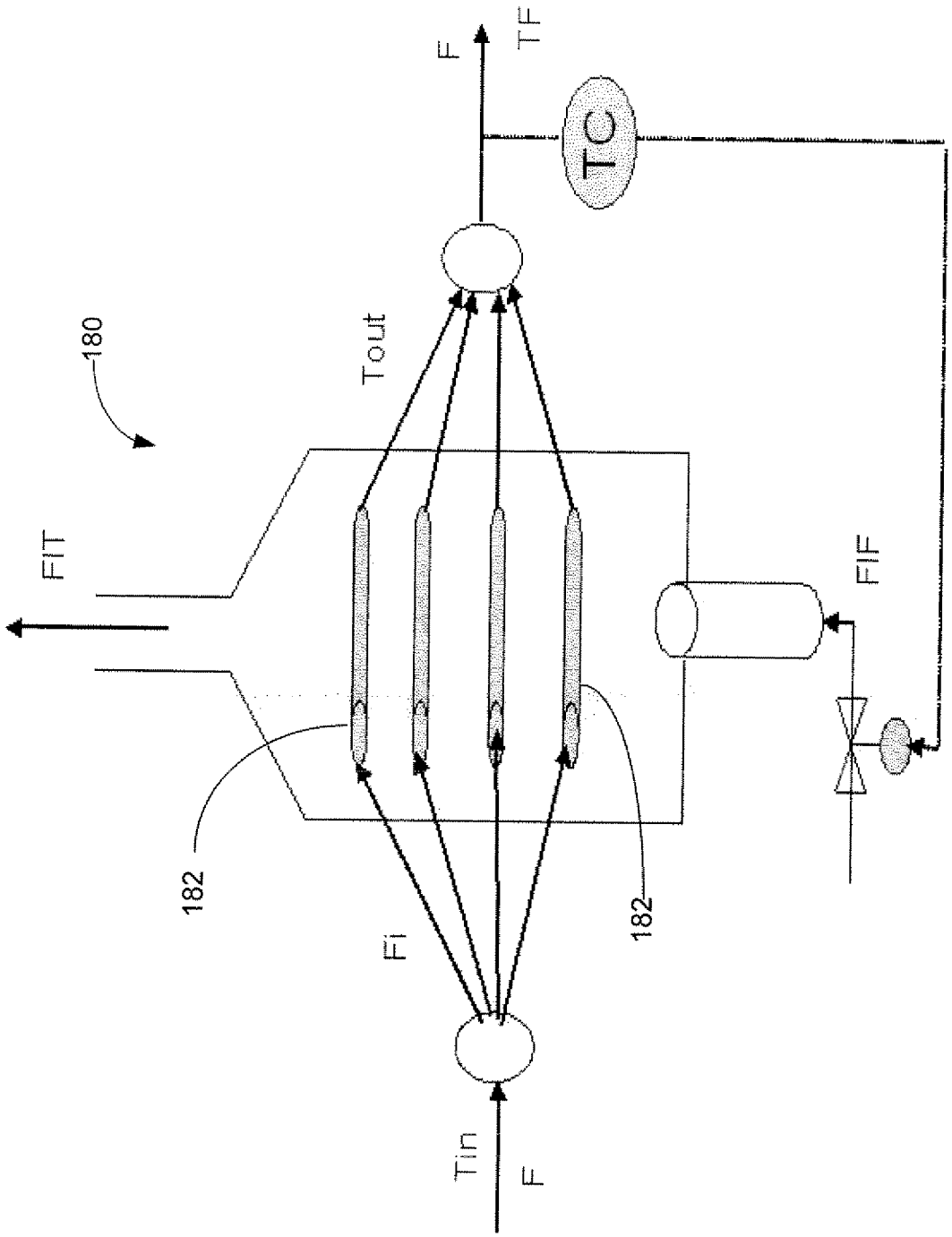


FIG. 24

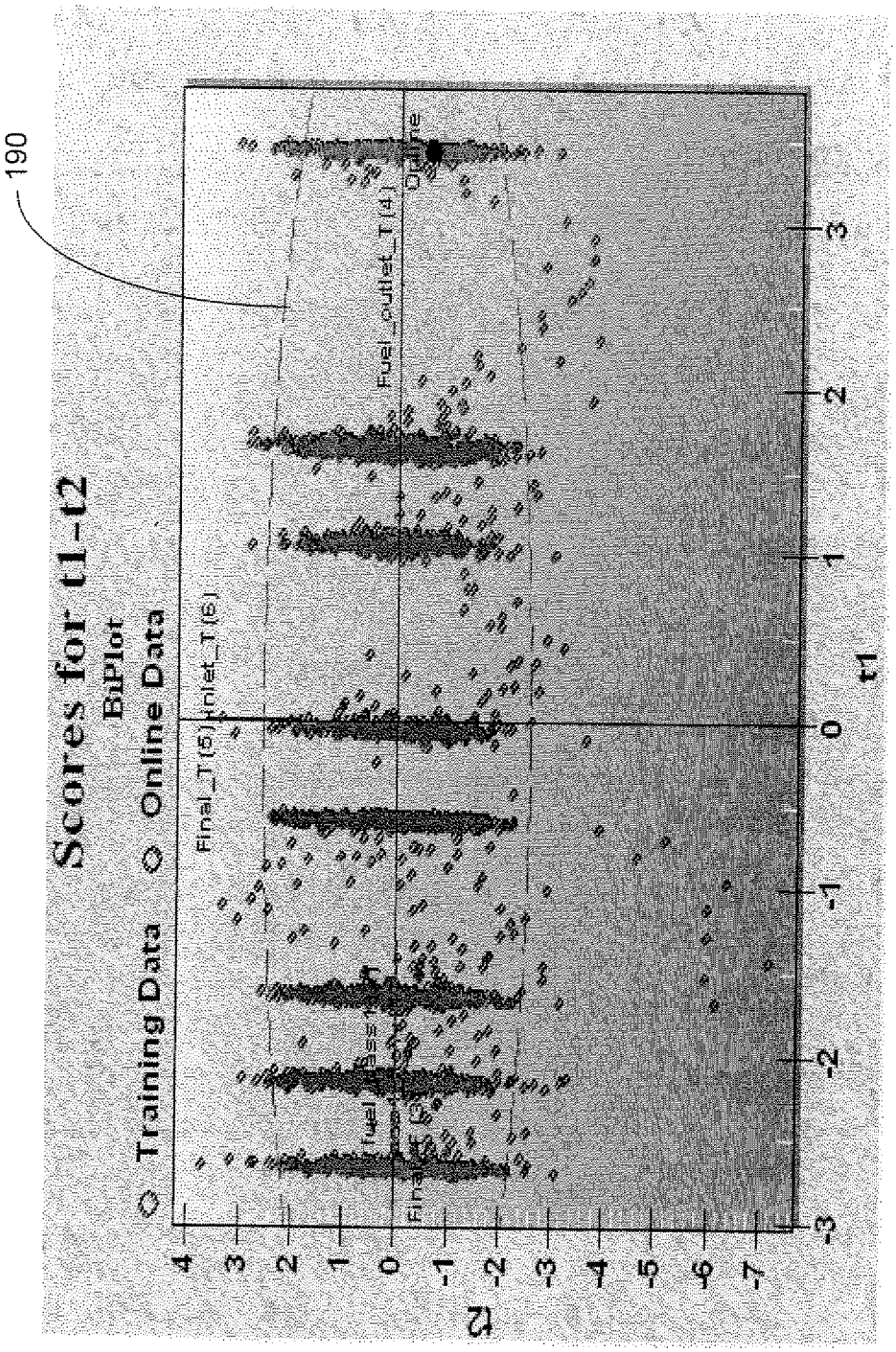

190
Scores for t1-t2
BiPlot
Training Data
Online Data
Final_T (5) Inlet_T (6)
Fuel_outlet_T (4)
Outlet
t2
t1

FIG. 25

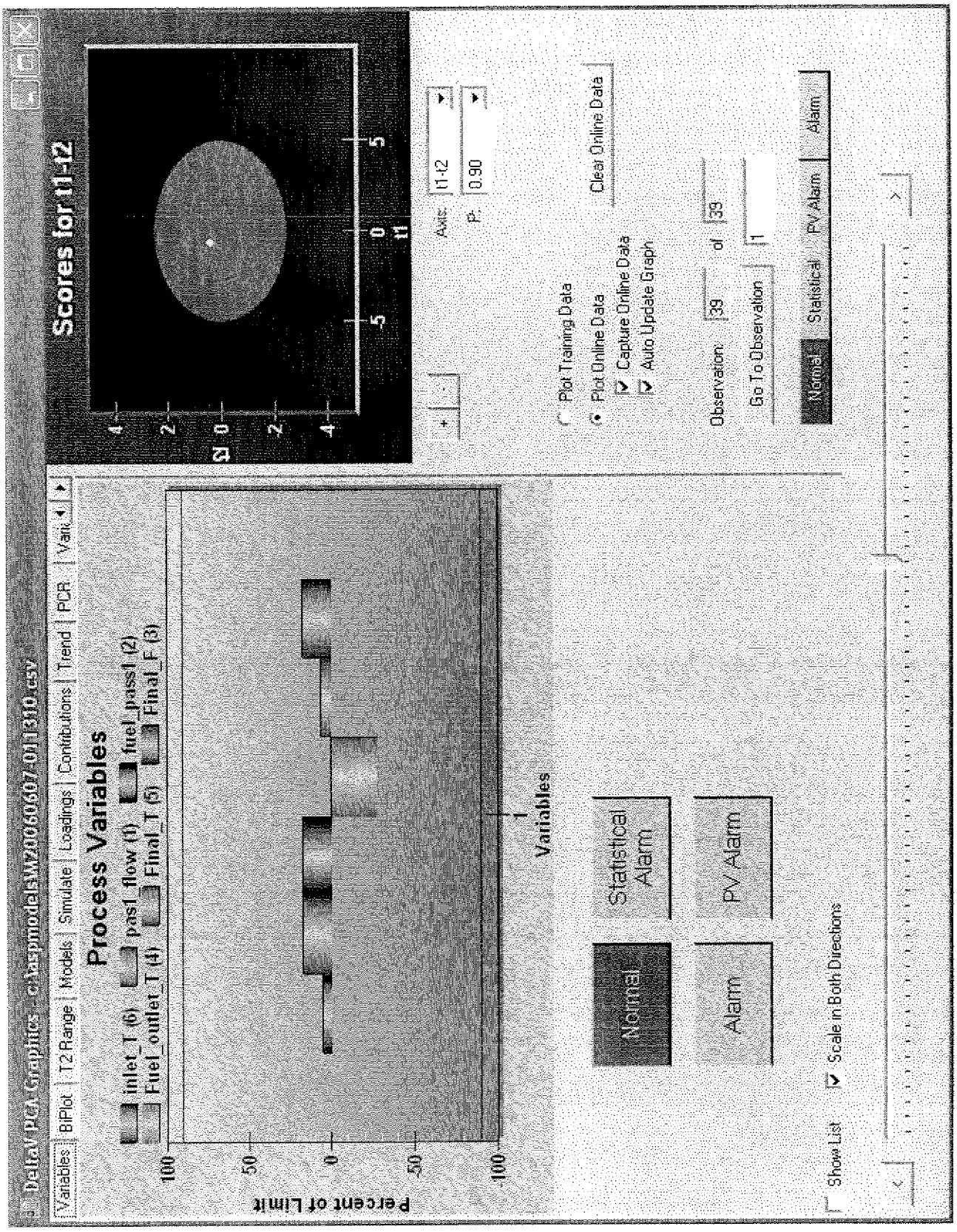

Variables
BiPlot
T2 Range
Models
Simulate
Loadings
Contributions
Trend
PCR
Process Variables
inlet_T (6)
pas1_flow (1)
fuel_pas1 (2)
Fuel_outlet_T (4)
Final_T (5)
Final_F (3)
Percent of Limit
Variables
Scores for t1-t2
t1-t2
0.90
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation: 39 of 39
Go To Observation 1
Normal
Statistical
PV Alarm
Alarm
Show List
Scale in Both Directions

FIG.

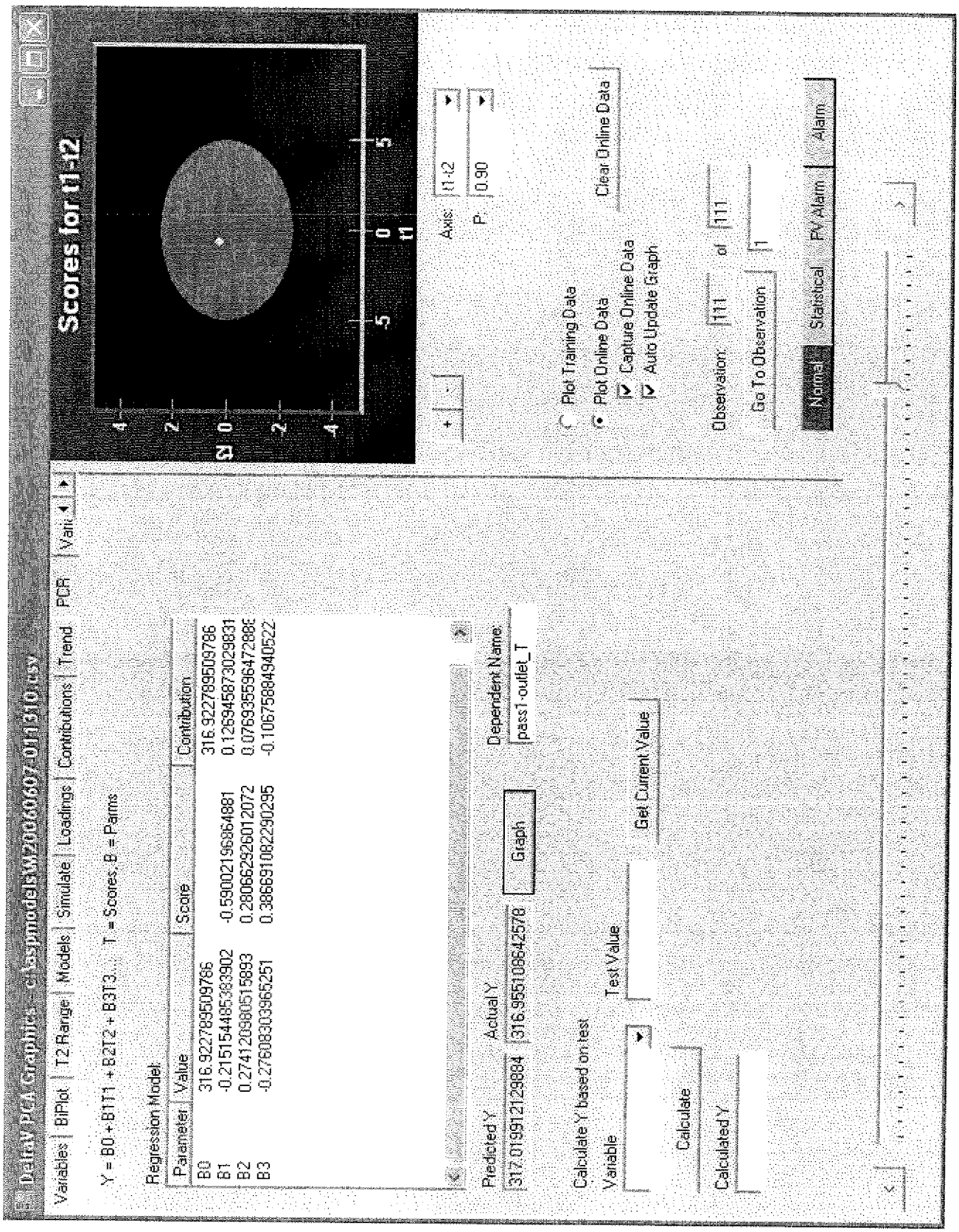

Scores for t1-t2
Variables
BiPlot
T2 Range
Models
Simulate
Loadings
Contributions
Trend
PCR
Regression Model
Parameter
Value
Score
Contribution
B0 316.92278950978 316.92278950978
B1 -0.21515448538390 -0.59002196364881 0.12694587302983
B2 0.27412098051589 0.28066329260120 0.07693559647288
B3 -0.27608303965251 0.38669108229029 -0.10675884940522
Predicted Y
Actual Y
Dependent Name
316.01991212988 316.95510986425 Graph pass1-outlet_T
Calculate Y based on test
Variable
Test Value
Get Current Value
Calculate
Calculated Y
Axis: t1-t2
P: 0.90
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation: 111 of 111
Go To Observation 1
Normal
Statistical
FV Alarm
Alarm

FIG.

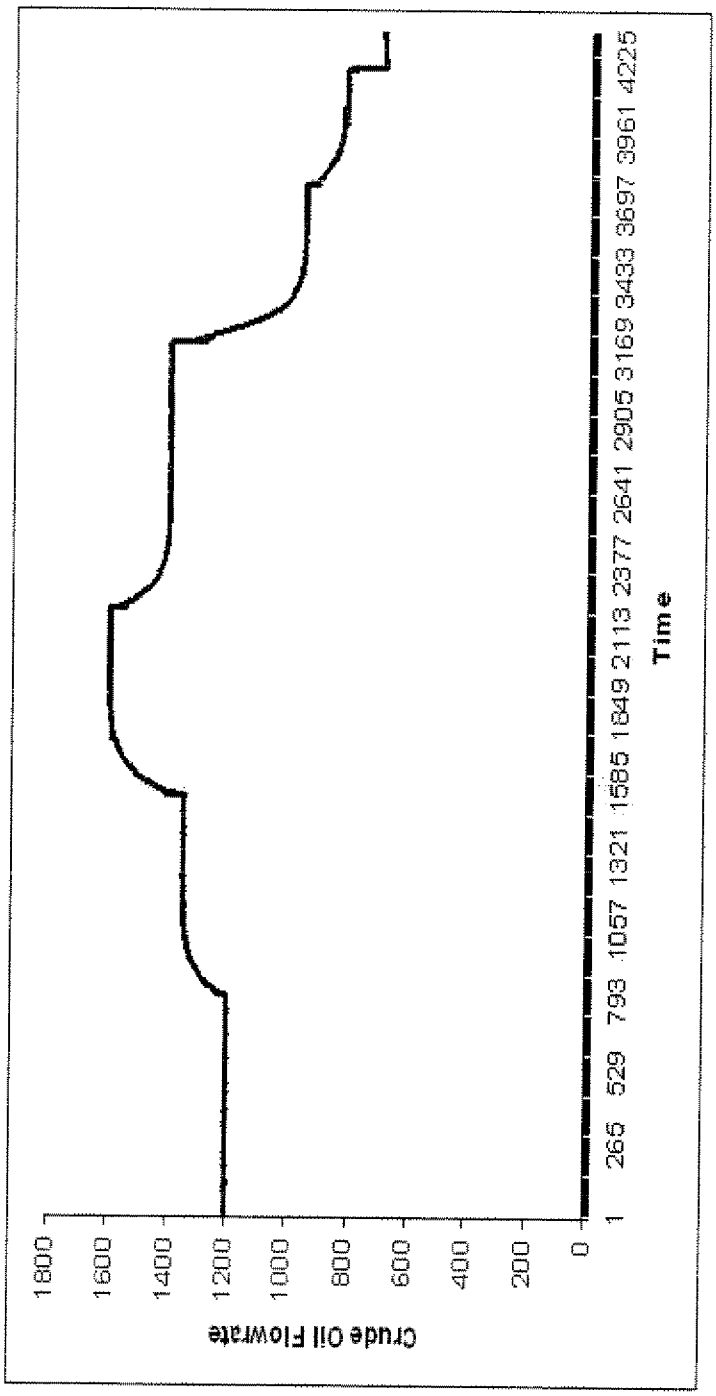

Crude Oil Flowrate
1800
1600
1400
1200
1000
800
600
400
200
0
1
265
529
793
1057
1321
1585
1849
2113
2377
2641
2905
3169
3433
3697
3961
4225
Time

FIG. 28

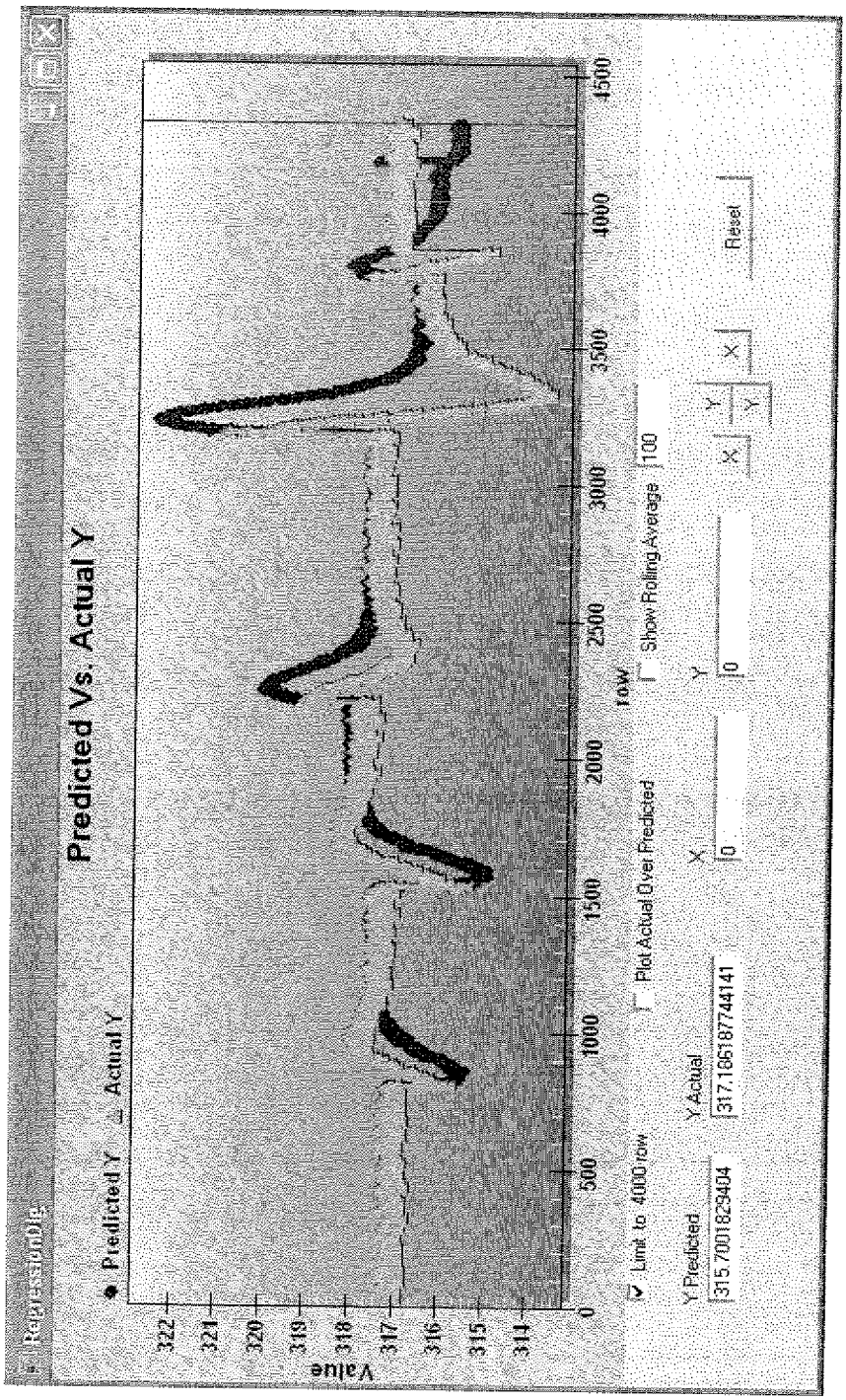

RegressionDlg
Predicted Vs. Actual Y
Predicted Y
Actual Y
Value
322
321
320
319
318
317
316
315
314
0
500
1000
1500
2000
2500
3000
3500
4000
4500
row
Limit to 4000 row
Plot Actual Over Predicted
Show Rolling Average
100
Y Predicted
315.70018629404
Y Actual
317.16618744141
X
0
Y
0
X
Y
X
Y
Reset

FIG. 29

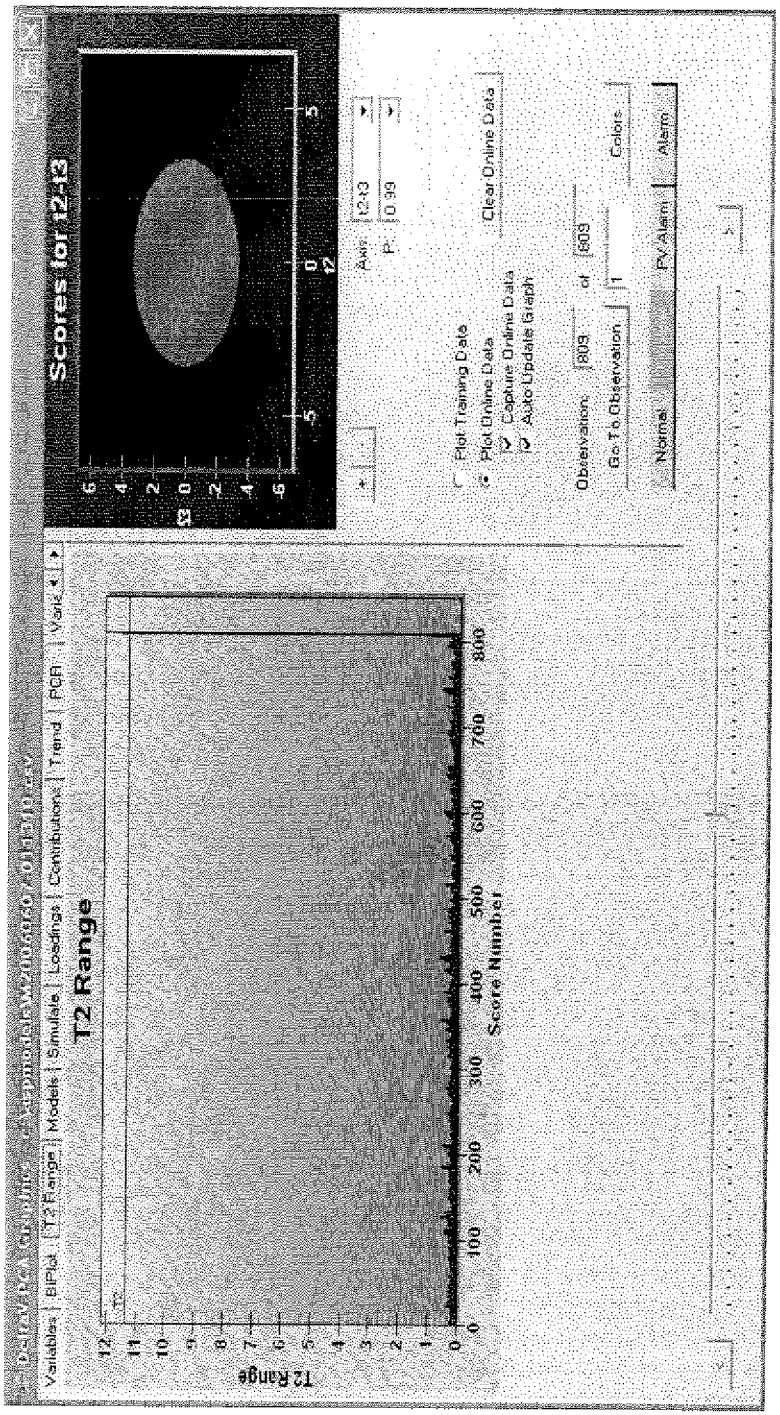

Variables | BiPlot | T2 Range | Models | Simulate | Loadings | Contributions | Trend | PCR | Var4
T2 Range
T2 Range
Score Number
Scores for 1243
Axes 1243
P: 0.99
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation 809 of 809
Go To Observation 1
Colors
Normal
PV Alarm
Alarm

FIG. 30

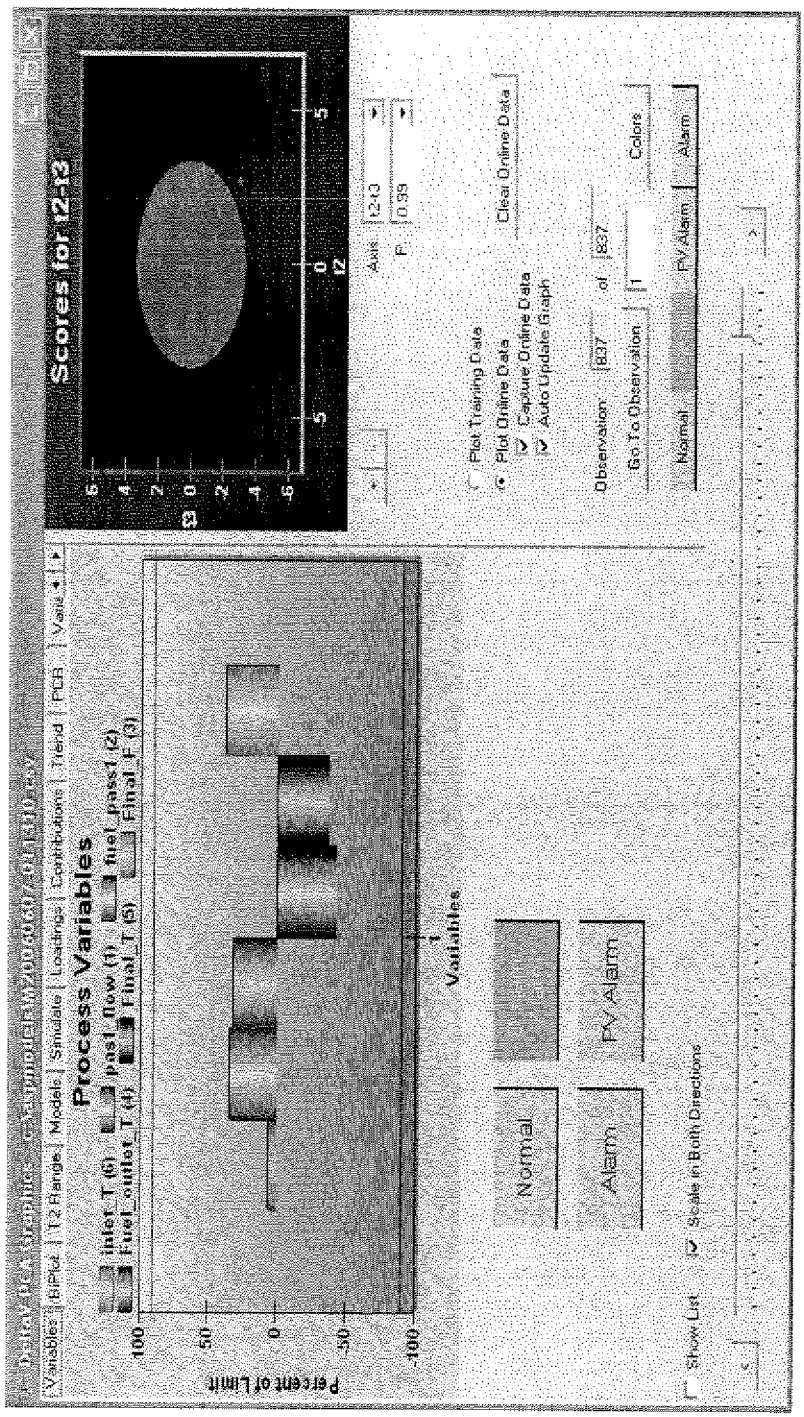

Process Variables
Scores for t2-t3
Normal
Alarm
PV Alarm
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation
Go To Observation
Colors
Show List
Scale in Both Directions
Variables
Percent of Limit

FIG. 31

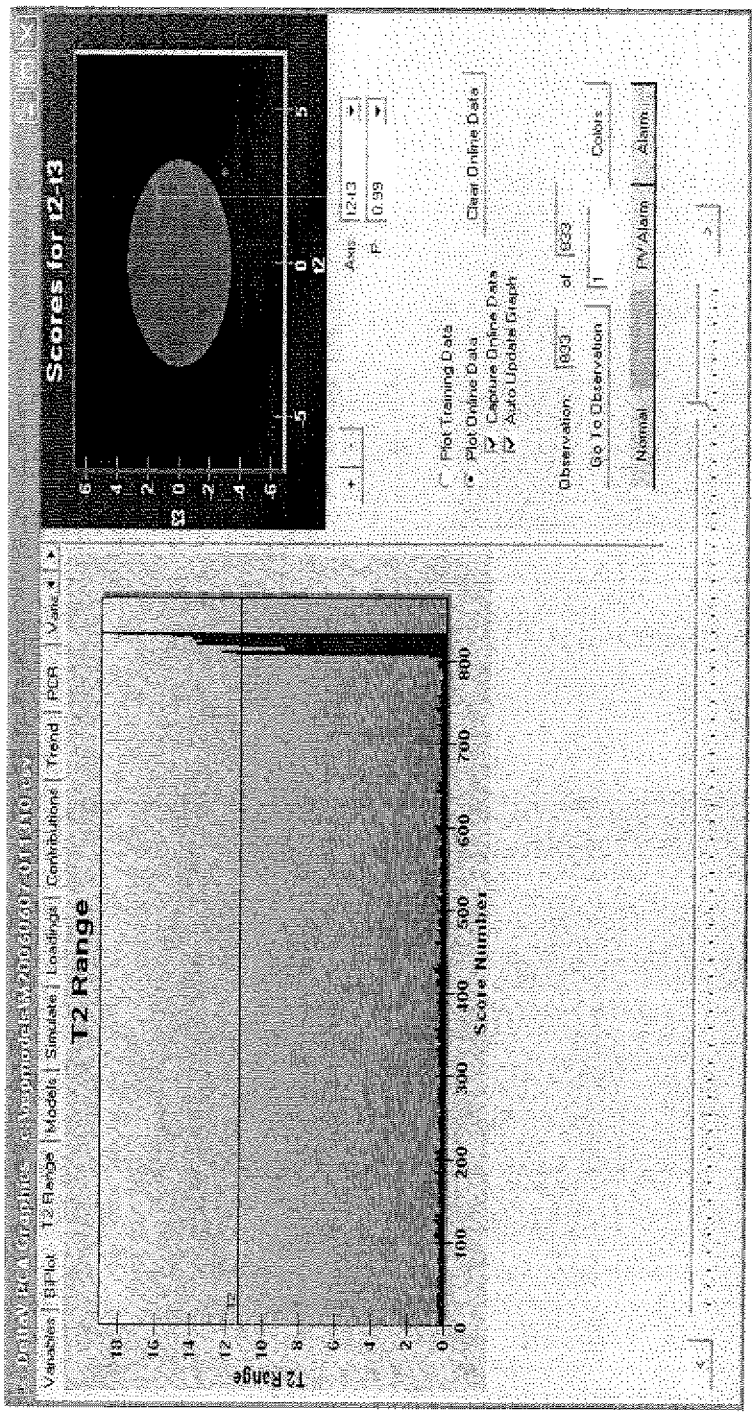

Scores for 2-3
T2 Range
T2 Range
Score Number
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation:
of
Go To Observation
Colors
Normal
FV Alarm
Alarm

FIG. 32

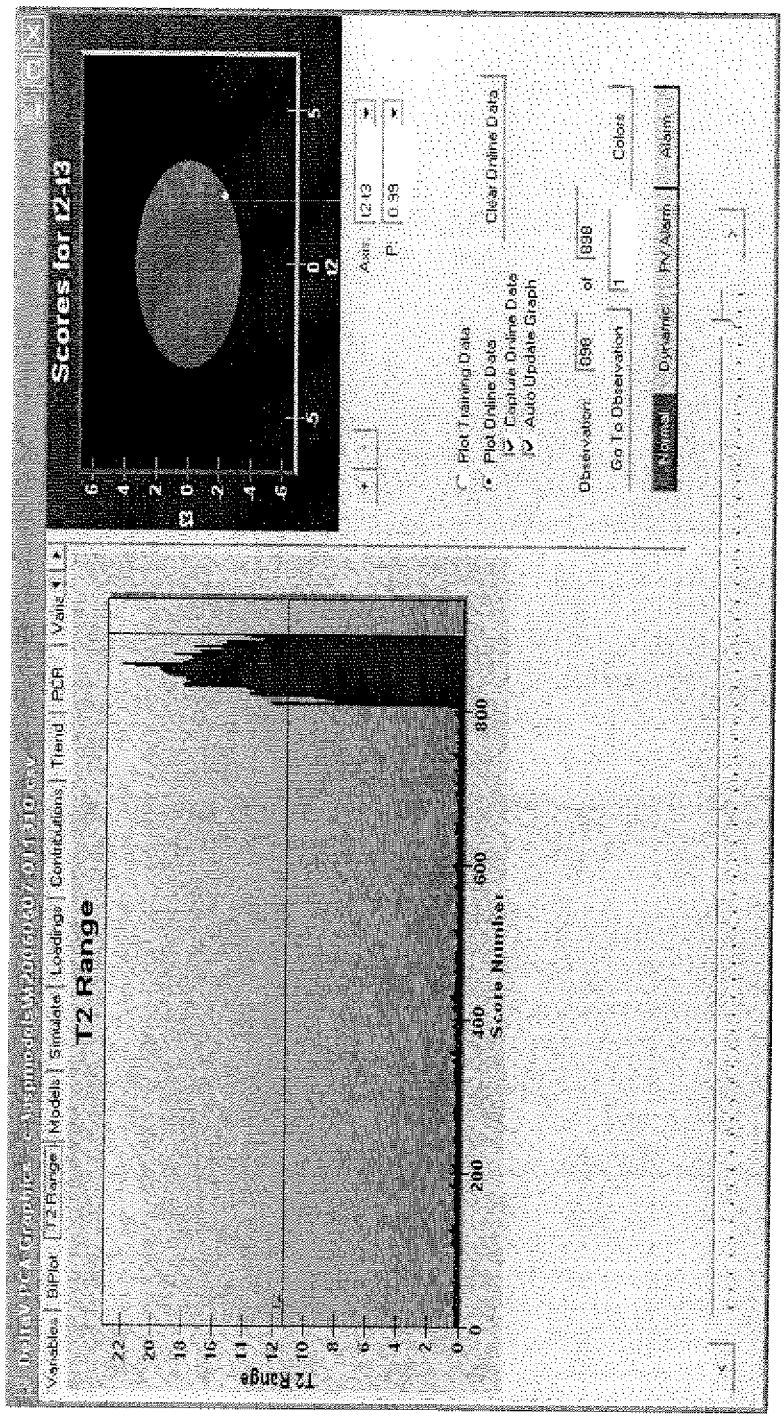

Variables
BiPlot
T2Range
Models
Simulate
Loadings
Contributions
Trend
RCR
Valid
T2 Range
T2 Range
Score Number
Scores for t2-t3
Axes
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation
of
Go To Observation
Colors
Normal
Dynamic
PV Alarm
Alarm

FIG. 33

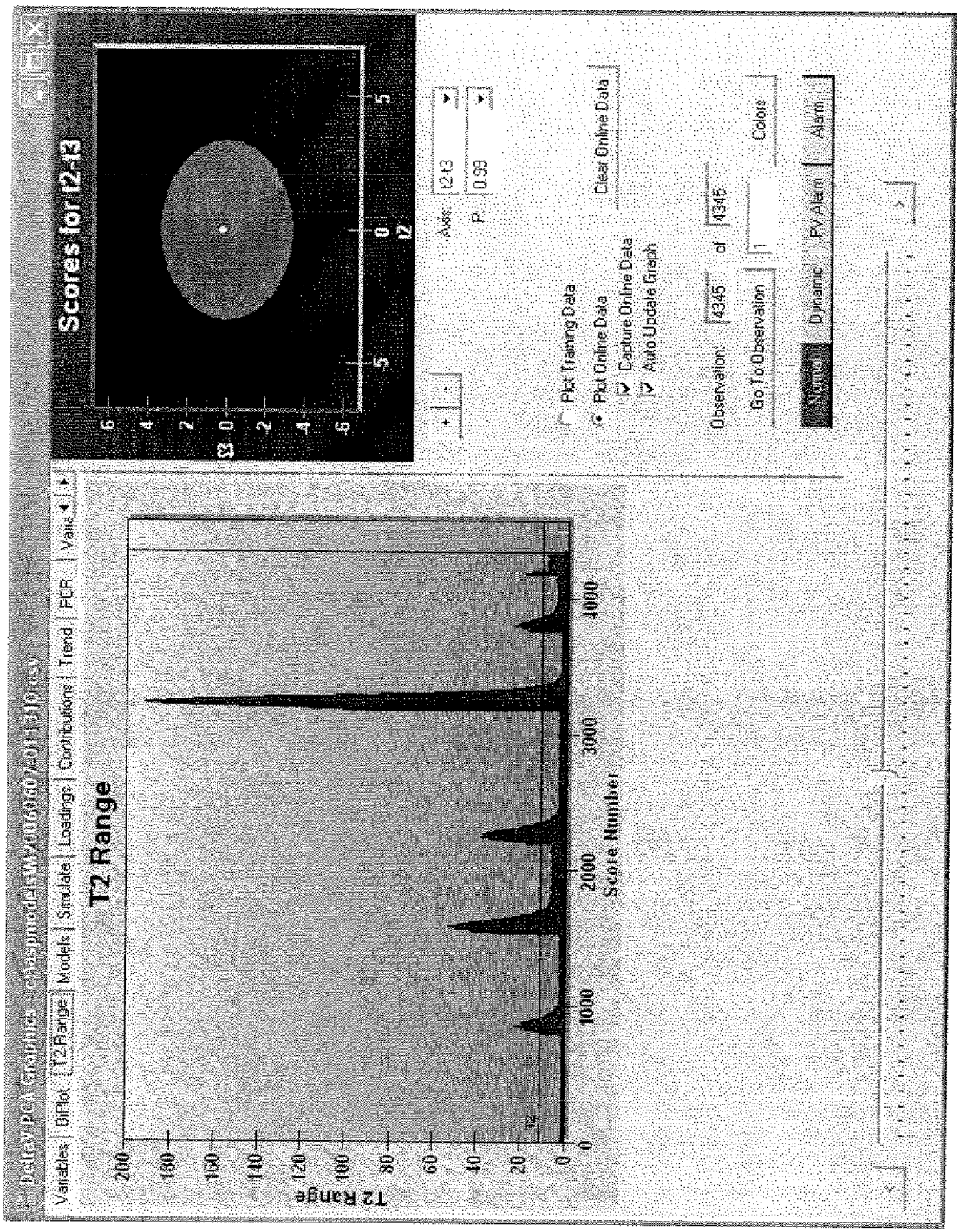

Variables
BiPlot
T2 Range
Models
Simulate
Loadings
Contributions
Trend
PCR
T2 Range
Score Number
Scores for [2-3]
Axes
2-3
P
0.99
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation
4345
of
4345
Go To Observation
1
Colors
Name
Dynamic
PV Alarm
Alarm

FIG. 34

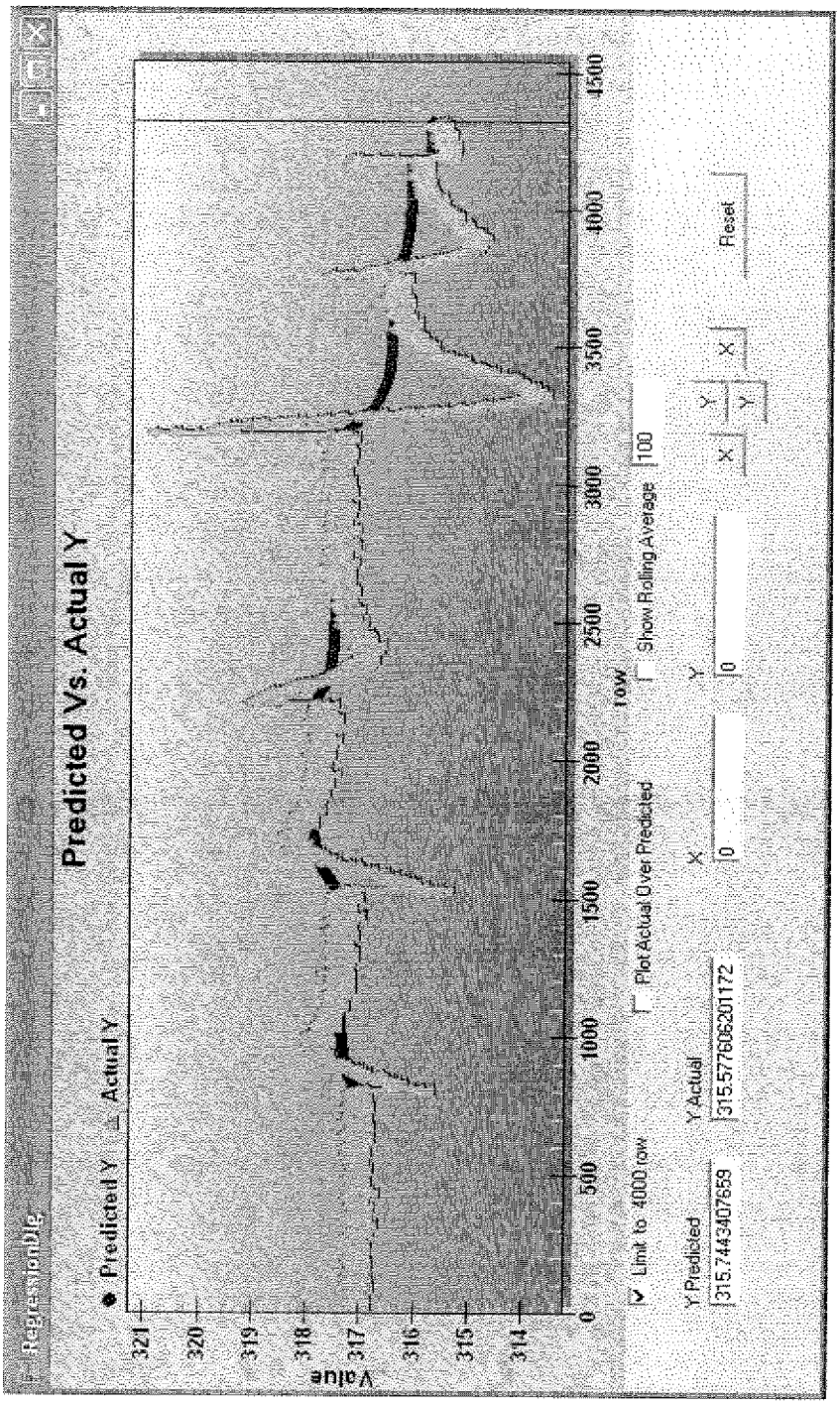

Predicted Vs. Actual Y
Predicted Y
Actual Y
Value
321
320
319
318
317
316
315
314
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Row
Limit to 4000 row
Plot Actual Over Predicted
Show Rolling Average
100
Y Predicted
315.7443407659
Y Actual
315.57766201172
X
0
Y
0
X
Y
X
Y
Reset

FIG. 35

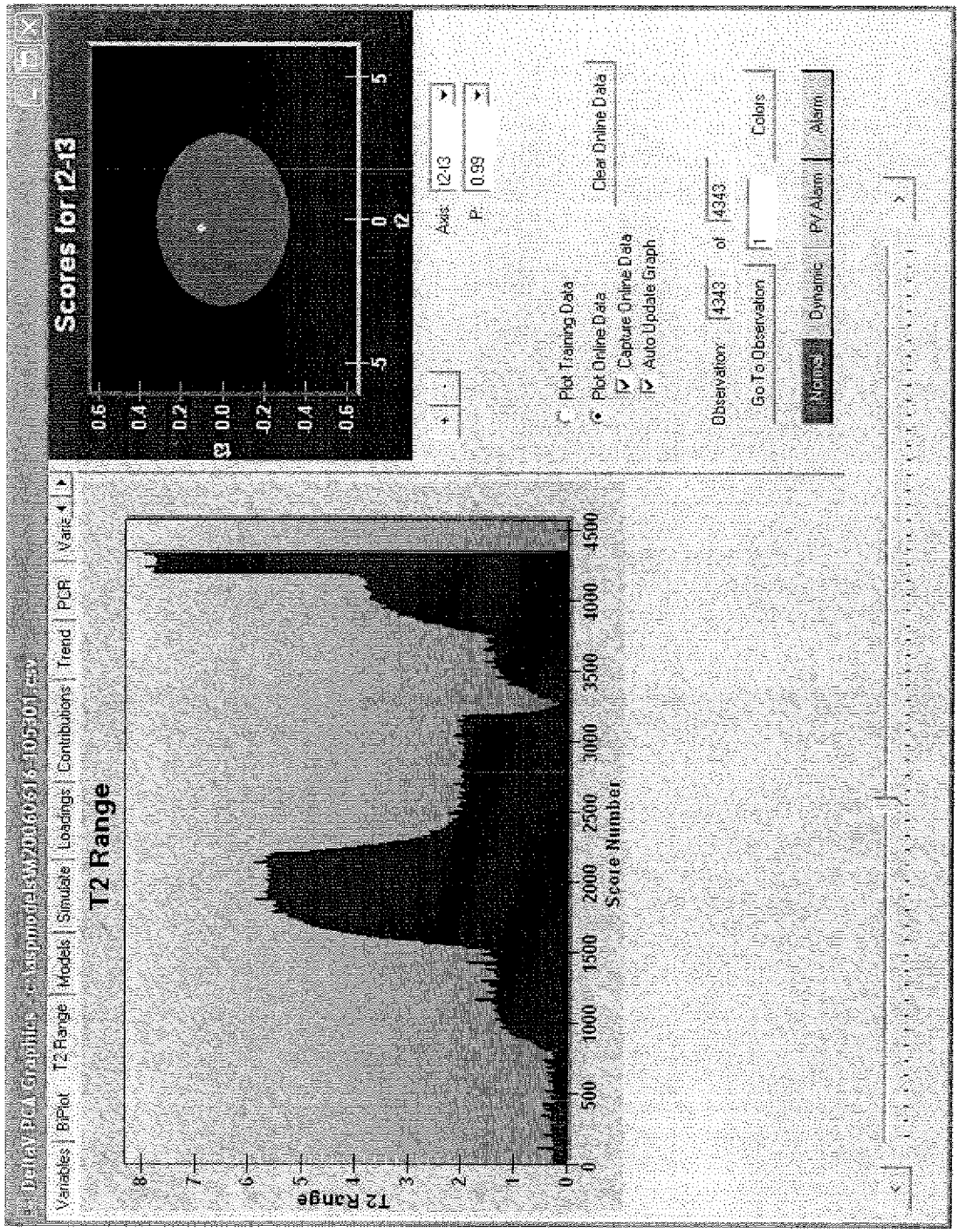

Variables
BPlot
T2 Range
Models
Simulate
Loadings
Contributions
Trend
PCR
Vars
T2 Range
T2 Range
Score Number
Scores for t2-t3
t3
t2
Axis:
t2-t3
P:
0.99
Plot Training Data
Plot Online Data
Capture Online Data
Auto Update Graph
Clear Online Data
Observation:
4343
of
4343
Go To Observation
1
Colors
Alarm
Dynamic
P/ Alarm

FIG. 36

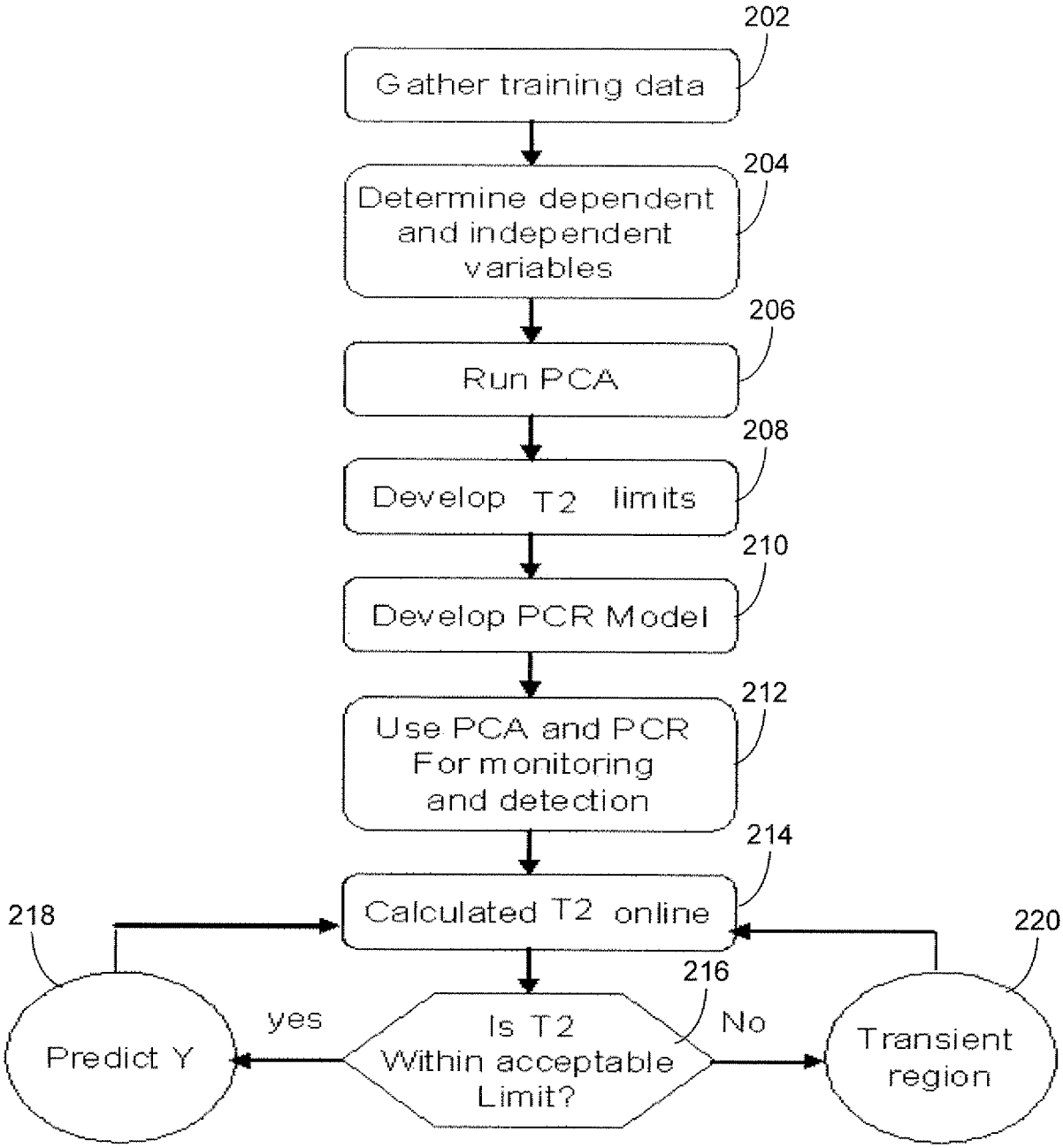

202
Gather training data
204
Determine dependent and independent variables
206
Run PCA
208
Develop T2 limits
210
Develop PCR Model
212
Use PCA and PCR For monitoring and detection
214
Calculated T2 online
218
220
216
Predict Y
yes
Is T2 Within acceptable Limit?
No
Transient region

FIG. 37

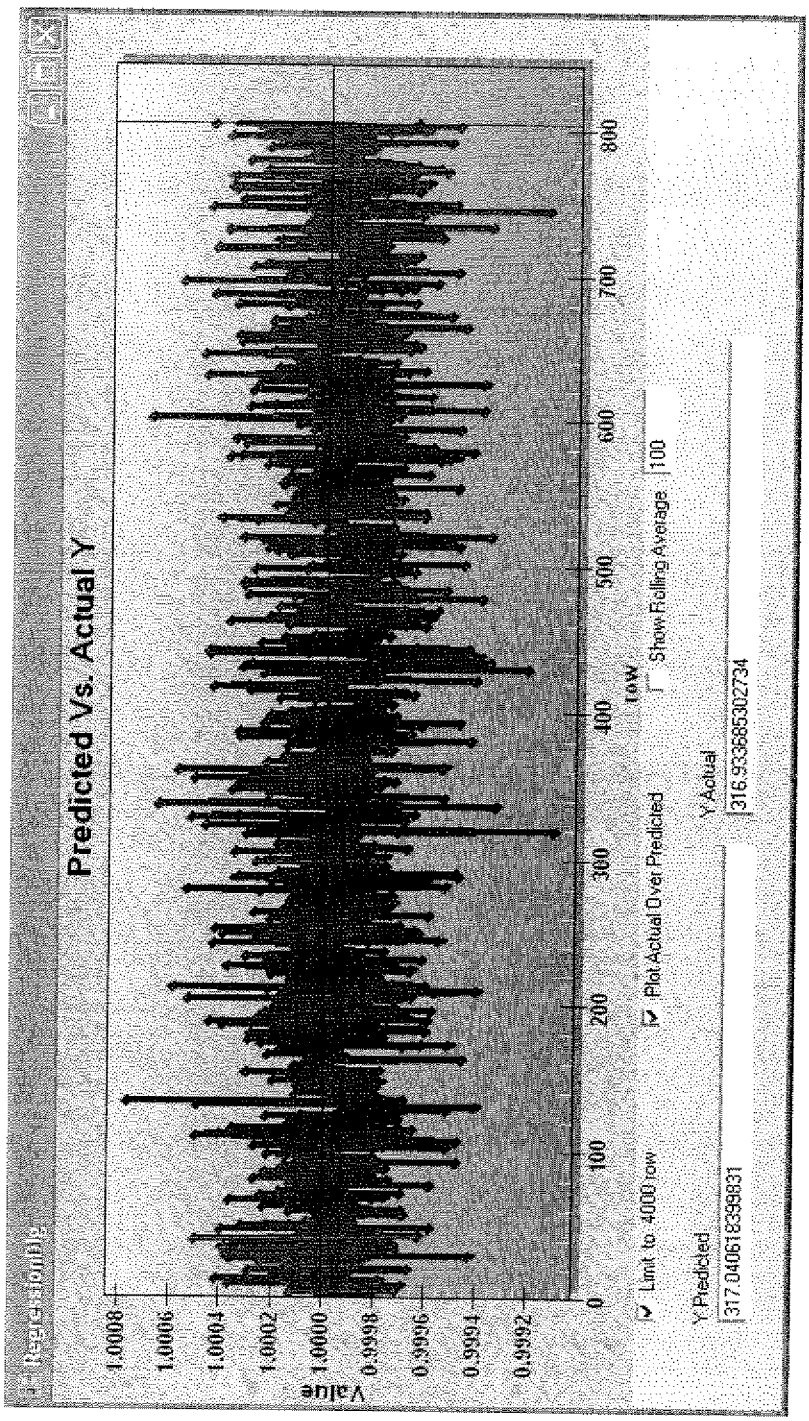

Predicted Vs. Actual Y
Value
1.0008
1.0006
1.0004
1.0002
1.0000
0.9998
0.9996
0.9994
0.9992
0
100
200
300
400
500
600
700
800
row
Limit to 4000 row
Plot Actual Over Predicted
Show Rolling Average
100
Y Predicted
317.040618399831
Y Actual
316.933685302734

FIG. 38

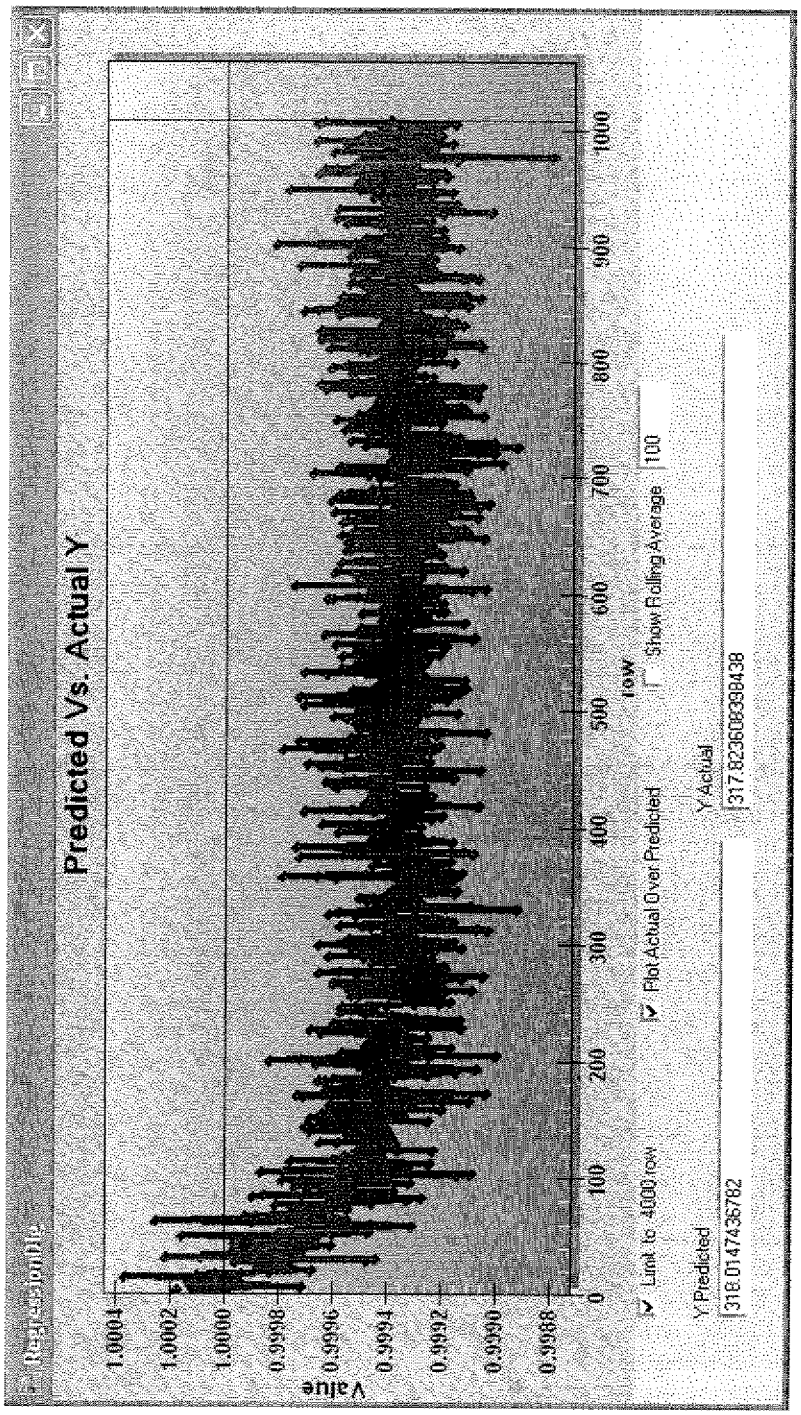

Predicted Vs. Actual Y
Value
row
Link to 4000 row
Plot Actual Over Predicted
Show Rolling Average
100
Y Predicted
318.0147436762
Y Actual
317.823608398438

FIG. 39

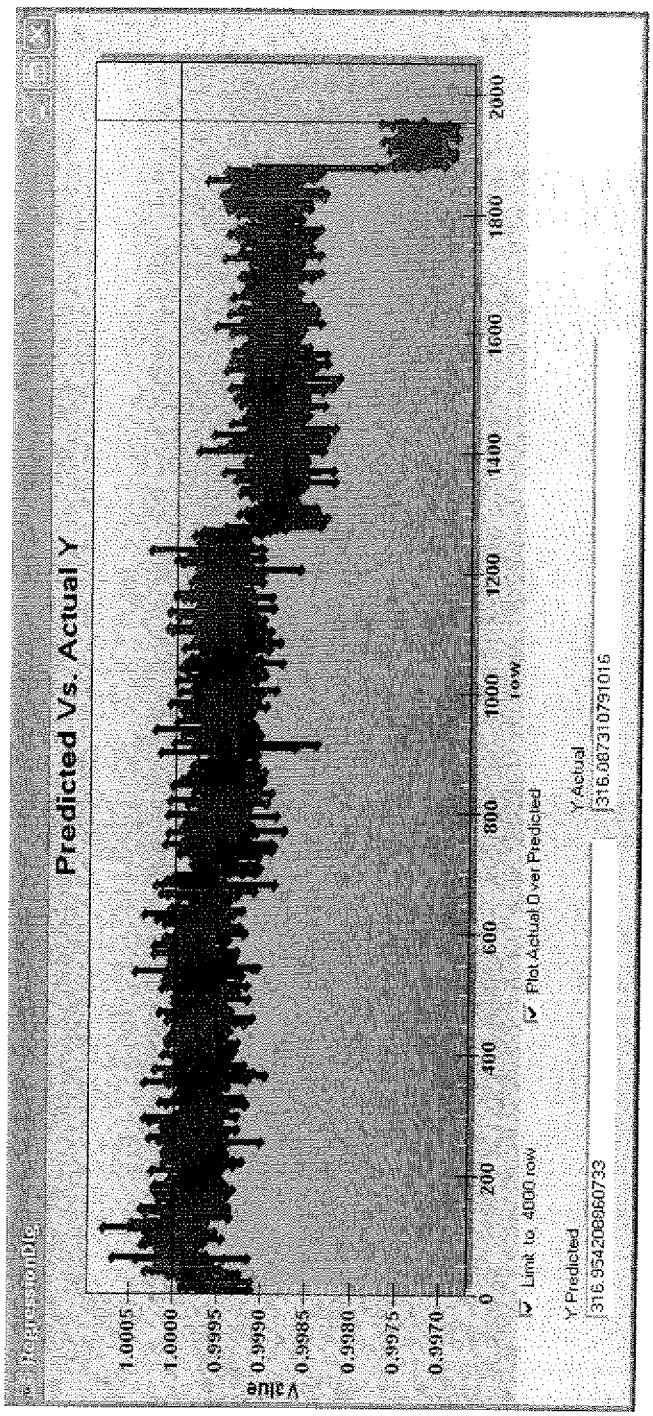

Predicted Vs. Actual Y
Value
1.0005
1.0000
0.9995
0.9990
0.9985
0.9980
0.9975
0.9970
0
200
400
600
800
1000
1200
1400
1600
1800
2000
Row
Limit to 4000 row
Plot Actual Over Predicted
Y Predicted
316.95420860733
Y Actual
316.98731079101 6

FIG. 40

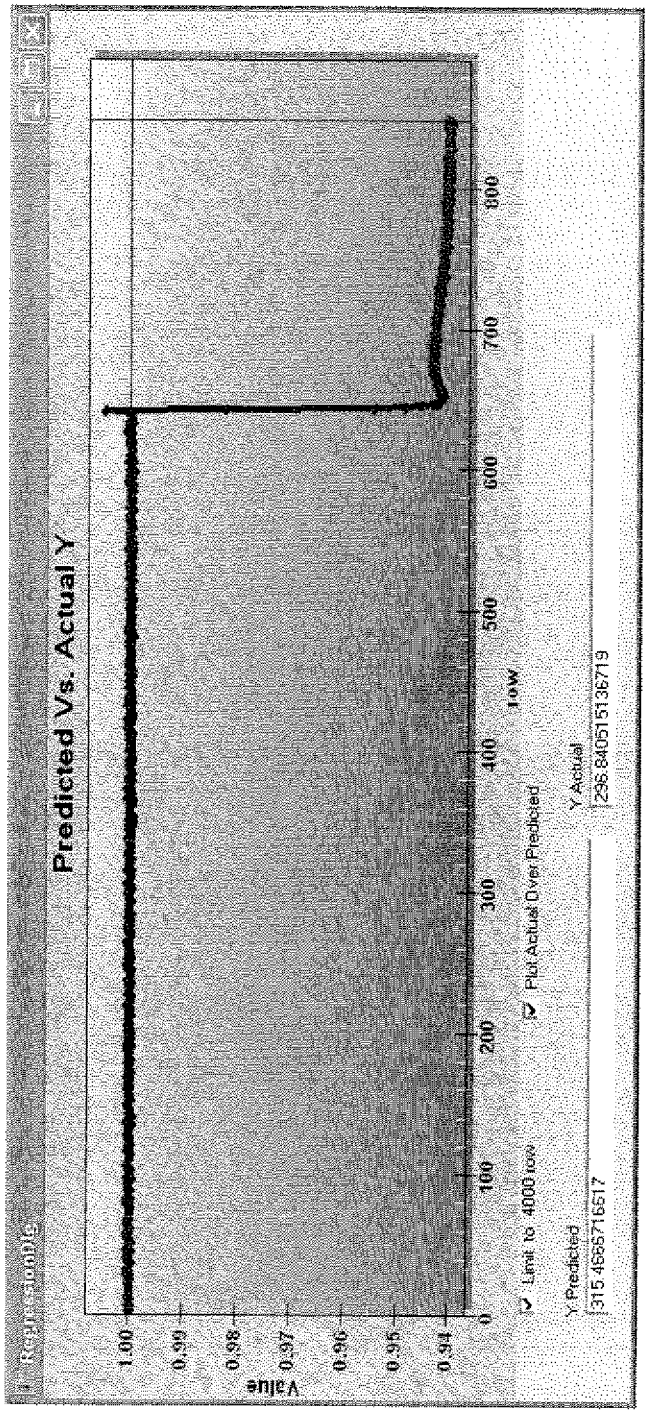

Predicted Vs. Actual Y
Value
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0
100
200
300
400
500
600
700
800
Row
Limit to 4000 rows
Plot Actual Over Predicted
Y Predicted
315.466571667
Y Actual
296.840515136719

FIG. 41

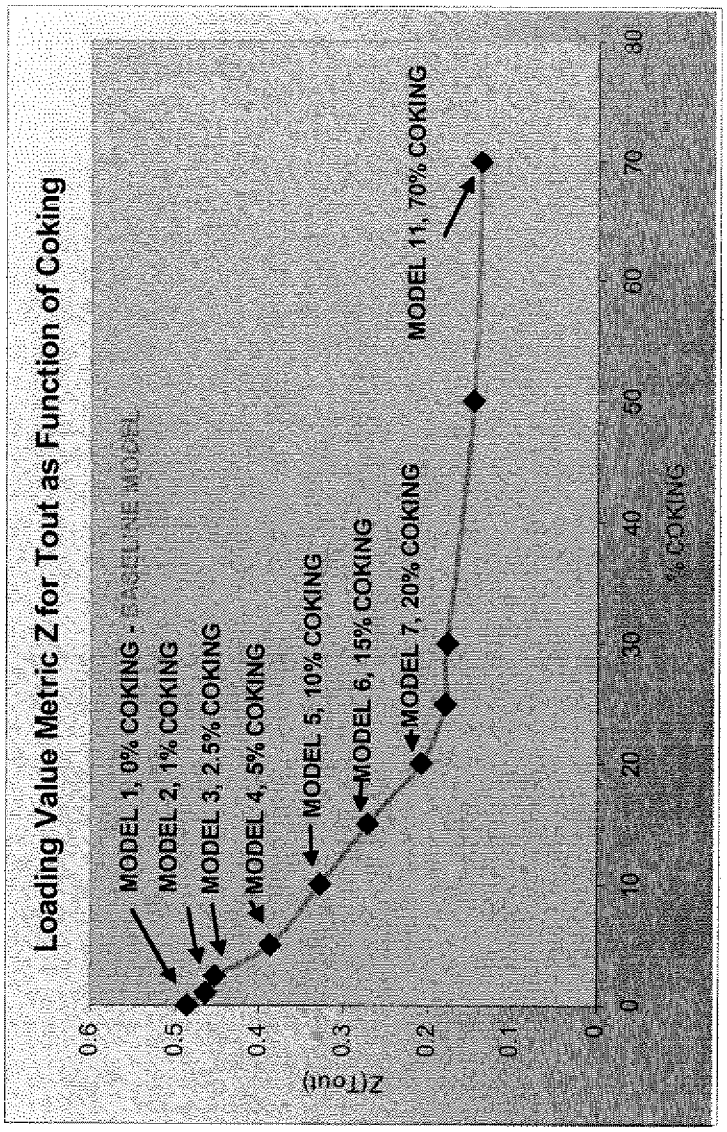

Loading Value Metric Z for Tout as Function of Coking
MODEL 1, 0% COKING - BASELINE MODEL
MODEL 2, 1% COKING
MODEL 3, 2.5% COKING
MODEL 4, 5% COKING
MODEL 5, 10% COKING
MODEL 6, 15% COKING
MODEL 7, 20% COKING
MODEL 11, 70% COKING
Z(Tout)
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
60
70
80
% COKING

FIG. 42

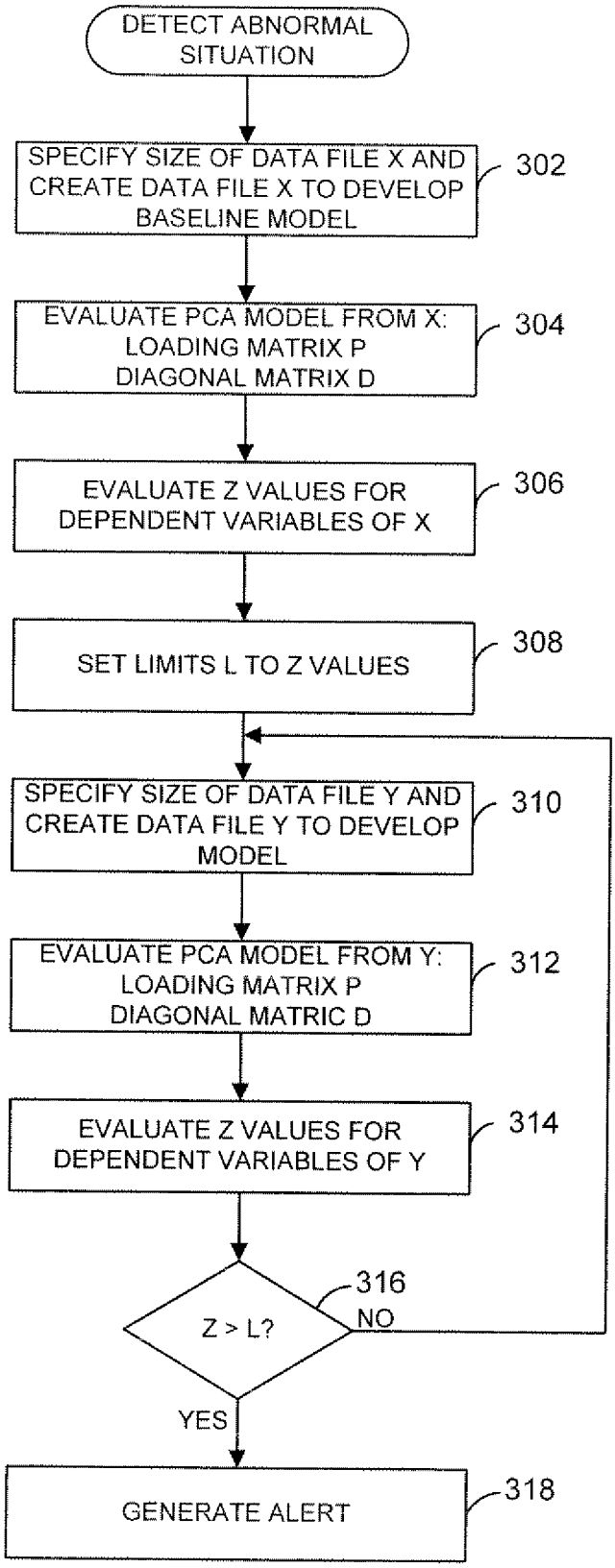

DETECT ABNORMAL SITUATION
SPECIFY SIZE OF DATA FILE X AND CREATE DATA FILE X TO DEVELOP BASELINE MODEL
302
EVALUATE PCA MODEL FROM X: LOADING MATRIX P DIAGONAL MATRIX D
304
EVALUATE Z VALUES FOR DEPENDENT VARIABLES OF X
306
SET LIMITS L TO Z VALUES
308
SPECIFY SIZE OF DATA FILE Y AND CREATE DATA FILE Y TO DEVELOP MODEL
310
EVALUATE PCA MODEL FROM Y: LOADING MATRIX P DIAGONAL MATRIC D
312
EVALUATE Z VALUES FOR DEPENDENT VARIABLES OF Y
314
Z > L?
316
NO
YES
GENERATE ALERT
318

FIG. 43