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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 January 2013 (24.01.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/012534 A1

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

(21) International Application Number:
PCT/US2012/044614

(22) International Filing Date:
28 June 2012 (28.06.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/186,128 19 July 2011 (19.07.2011) US

(71) Applicant *(for all designated States except US)*: **SMARTSIGNAL** [US/US]; 901 Warrenville Road, Suite 300, Lisle, IL 60532 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **HERZOG, James, P.** [US/US]; 901 Warrenville Road, Lisle, IL 60532 (US).

(74) Agents: **CONKLIN, Mark, A.** et al.; General Electric Company, Global Patent Operation, 2 Corporate Drive, Suite 648, Shelton, CT 06484 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: MONITORING SYSTEM USING KERNEL REGRESSION MODELING WITH PATTERN SEQUENCES

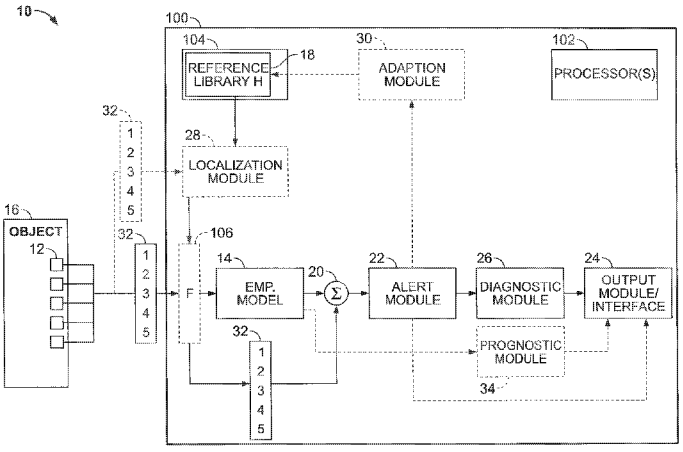

FIG. 1

(57) Abstract: A monitoring system for determining the condition of an object includes an empirical model with reference data that indicates the normal operational state of the object, and input pattern arrays. Each input pattern array has a plurality of time-ordered input vectors, while each input vector has input values representing a plurality of parameters indicating the current condition of the object. The empirical model generates estimate values based on a calculation that uses an input pattern array and the reference data to determine a similarity measure between the input values and reference data. A differencing module compares the estimates values to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

WO 2013/012534 A1

MONITORING SYSTEM USING KERNEL REGRESSION MODELING WITH PATTERN SEQUENCES

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] The subject matter disclosed herein relates generally to the field of kernel regression modeling used for predictive condition monitoring and prognostics of an object such as machine, system, or process, and more particularly to the use of multivariate models for analysis of measurements of parameters to provide an assessment of the object being monitored.

Brief Description of the Related Art

[0002] Kernel regression is a form of modeling used to determine a non-linear function or relationship between values in a dataset and is used to monitor machines or systems to determine the condition of the machine or system. One known form of kernel regression modeling is similarity-based modeling (SBM) disclosed by U.S. Patent Nos. 5,764,509 and 6,181,975. For SBM, multiple sensor signals measure physically correlated parameters of a machine, system, or other object being monitored to provide sensor data. The parameter data may include the actual or current values from the signals or other calculated data whether or not based on the sensor signals. The parameter data is then processed by an empirical model to provide estimates of those values. The estimates are then compared to the actual or current values to determine if a fault exists in the system being monitored.

[0003] More specifically, the model generates the estimates using a reference library of selected historic patterns of sensor values representative of known operational states. These patterns are also referred to as vectors, snapshots, or observations, and include values from multiple sensors or other input data that indicate the condition of the machine being monitored at an instant in time. In the case of the reference vectors from the reference library, the vectors usually indicate normal operation of the machine being monitored. The model compares the vector from the current time to a number of selected learned

vectors from known states of the reference library to estimate the current state of the system. Generally speaking, the current vector is compared to a matrix made of selected vectors from the reference library to form a weight vector. In a further step, the weight vector is multiplied by the matrix to calculate a vector of estimate values. The estimate vector is then compared to the current vector. If the estimate and actual values in the vectors are not sufficiently similar, this may indicate a fault exists in the object being monitored.

[0004] However, this kernel regression technique does not explicitly use the time domain information in the sensor signals, and instead treat the data in distinct and disconnected time-contemporaneous patterns when calculating the estimates. For instance, since each current vector is compared to the reference library vectors individually, it makes no difference what order the current vectors are compared to the vectors of the reference library – each current vector will receive its own corresponding estimate vector.

[0005] Some known models do capture time domain information within a kernel regression modeling construct. For example, complex signal decomposition techniques convert time varying signals into frequency components as disclosed by U.S. Patent Nos. 6,957,172 and 7,409,320, or spectral features as disclosed by U.S. Patent No. 7,085,675. These components or features are provided as individual inputs to the empirical modeling engine so that the single complex signal is represented by a pattern or vector of frequency values that occur at the same time. The empirical modeling engine compares the extracted component inputs (current or actual vector) against expected values to derive more information about the actual signal or about the state of the system generating the time varying signals. These methods are designed to work with a single periodic signal such as an acoustic or vibration signal. But even with the system for complex signals, the time domain information is not important when calculating the estimates for the current vector since each current vector is compared to a matrix of vectors with reference or expected vectors regardless of the time period that the input vectors represent.

BRIEF DESCRIPTION OF THE INVENTION

[0006] In one aspect, a method for monitoring the condition of an object includes obtaining reference data that indicates the normal operational state of the object, and obtaining input pattern arrays. Each input pattern array has a plurality of time-ordered input vectors, while each input vector has input values representing a plurality of parameters indicating the current condition of the object. Then, at least one processor generates estimate values based on a calculation that uses an input pattern array and the reference data to determine a similarity measure between the input values and reference data. The estimate values are compared to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

[0007] In another aspect, a method for monitoring the condition of an object includes obtaining reference data that indicates the normal operational state of the object and in the form of a plurality of learned sequential pattern matrices. In this case, each learned sequential pattern matrix has a plurality of reference vectors, while each reference vector has reference values representing a plurality of parameters. The method also includes obtaining input data representing a plurality of parameters indicating the current condition of the object, and then using at least one processor to generate estimate values based on a calculation that uses the input data and the learned sequential pattern matrices to determine a similarity measure between the input data and reference values in the plurality of reference matrices. After that, the estimate values are compared to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

[0008] In yet another aspect, a method for monitoring the condition of an object includes obtaining reference data that indicates the normal operational state of the object, and obtaining input data representing a plurality of parameters indicating the current condition of the object. The method then includes generating, by at least one processor, estimate values based on a calculation that uses both the input data and the reference data to determine similarity measures between the input data and the reference data. The estimate values are generated in the form of an estimate matrix having a plurality of time-

ordered estimate vectors, where each estimate vector has estimate values representing multiple parameters. The method then includes comparing at least one estimate vector for each time period represented by the estimate matrices to the input data so that the resulting values from the comparison can be used to determine the condition of the object.

[0009] In another form, a monitoring system for determining the condition of an object has an empirical model with reference data that indicates the normal operational state of the object, and input pattern arrays where each input pattern array has a plurality of time-ordered input vectors. Each input vector has input values that represent a plurality of parameters indicating the current condition of the object. The empirical model is configured to generate estimate values based on a calculation that uses an input pattern array and the reference data to determine a similarity measure between the input values and the reference data. A differencing module then compares the estimate values to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

[0010] In a further form, a monitoring system for determining the condition of an object has an empirical model operated by at least one processor. The model comprises reference data that indicates the normal operational state of the object and in the form of a plurality of learned sequential pattern matrices. Each learned sequential pattern matrix has a plurality of reference vectors, while each reference vector has reference values representing a plurality of parameters. The model also has input data representing a plurality of parameters that indicate the current condition of the object. The empirical model is configured to generate estimate values based on a calculation that uses the input data and the learned sequential pattern matrices to determine a similarity measure between the input data and reference values in the plurality of reference matrices. The system also has a differencing module comparing the estimate values to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

[0011] In yet a further form, a monitoring system for determining the condition of an object has an empirical model with reference data that indicates the normal operational state of the object. The model also has input data representing a plurality of parameters that indicate the current condition of the object. Otherwise, the empirical model is configured to generate estimate values based on a calculation that uses both the input data and the reference data to determine similarity measures between the input data and the reference data. The estimate values are generated in the form of an estimate matrix having a plurality of time-ordered estimate vectors. Each estimate vector has estimate values representing multiple parameters. The system also has a differencing module comparing at least one estimate vector for each time period represented by the estimate matrices to the input data so that the resulting values from the comparison can be used to determine the condition of the object.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] FIG. 1 shows a block diagram of an example arrangement of a monitoring system;

[0013] FIG. 2 is flow chart showing the basic process for the monitoring system;

[0014] FIG. 3 is a schematic diagram of the autoassociative similarity based modeling equation;

[0015] FIG. 4 is a schematic diagram of one form of the inferential similarity based modeling equation;

[0016] FIG. 5 is a schematic diagram of another form of the inferential similarity based modeling equation;

[0017] FIG. 6 is a schematic diagram of the autoassociative sequential similarity based modeling equation;

[0018] FIG. 7 is a schematic diagram of one form of the inferential sequential similarity based modeling equation that extrapolates in the modeled sensor dimension;

[0019] FIG. 8 is a schematic diagram of another form of the inferential sequential similarity based modeling equation that extrapolates in the modeled sensor dimension;

[0020] FIG. 9 is a schematic diagram of an inferential sequential similarity based modeling equation that extrapolates in the time dimension;

[0021] FIG. 10 is a schematic diagram of an inferential sequential similarity based modeling equation that extrapolates in the time dimension; and

[0022] FIG. 11 is a schematic diagram of an inferential sequential similarity based modeling equation that extrapolates in the time dimension and the sensor dimension.

DETAILED DESCRIPTION OF THE INVENTION

[0023] It has been determined that the accuracy of the estimates in a kernel regression model, and specifically a similarity based model, can be substantially improved by incorporating time domain information into the model. Thus, one technical effect of the present monitoring system and method is to generate estimate data by capturing time domain information from the large numbers of periodic and non-periodic sensor signals that monitor industrial processes, systems, machines, or other objects. The technical effect of the present system also is to operate an empirical model that extends the fundamental non-linear mathematics at the core of kernel regression modeling from a vector-to-vector operation to a matrix-to-matrix (or array-to-array) operation as explained in detail below. Another alternative technical effect of the monitoring system and method is to generate virtual or inferred estimate values for future time points to determine a future condition of the object being monitored whether the reference data used to generate the estimates is data representing normal operation of the object being monitored or failure mode data to better match data from the object that indicates a fault.

[0024] Referring to FIG. 1, a monitoring system 10 incorporating time domain information can be embodied in a computer program in the form of one or more modules and executed on one or more computers 100 and by one or more processors 102. The computer 100 may have one or more memory storage devices 104, whether internal or external, to hold sensor data and/or the computer programs whether permanently or temporarily. In one form, a standalone computer runs a program dedicated to receiving sensor data from sensors on an instrumented machine, process or other object including a living being,

measuring parameters (temperature, pressure, and so forth). The object being monitored, while not particularly limited, may be one or more machines in an industrial plant, one or more vehicles, or particular machines on the vehicles such as jet engines to name a few examples. The sensor data may be transmitted through wires or wirelessly over a computer network or the internet, for example, to the computer or database performing the data collection. One computer with one or more processors may perform all of the monitoring tasks for all of the modules, or each task or module may have its own computer or processor performing the module. Thus, it will be understood that processing may take place at a single location or the processing may take place at many different locations all connected by a wired or wireless network.

[0025] Referring to FIG. 2, in the process (300) performed by the monitoring system 10, the system receives data or signals from sensors 12 on an object 16 being monitored as described above. This data is arranged into input vectors 32 for use by the model 14. Herein, the terms input, actual, and current are used interchangeably, and the terms vector, snapshot, and observation are used interchangeably. The input vector (or actual snapshot for example) represents the operational state of the machine being monitored at a single moment in time.

[0026] Additionally, or alternatively, the input vector 32 may include calculated data that may or may not have been calculated based on the sensor data (or raw data). This may include, for example, an average pressure or a drop in pressure. The input vector 32 may also have values representing other variables not represented by the sensors on the object 16. This may be, for example, the average ambient temperature for the day of the year the sensor data is received, and so forth.

[0027] The model 14 obtains (302) the data in the form of the vectors 32 and arranges (304) the input vectors into an input array or matrix. It will be understood, however, that the model 14 itself may form the vectors 32 from the input data, or receive the vectors from a collection or input computer or processor that organizes the data into the vectors and arrays. Thus, the input data may be arranged into vector 32 by computer 100, another computer near location of computer 100, or at another location such as near the object 16.

[0028] The model 14 also obtains (306) reference data in the form of reference vectors or matrices from reference library 18 and sometimes referred to as a matrix **H**. The library 18 may include all of the historical reference vectors in the system. The model 14 then uses the reference data and input arrays to generate estimates (310) in the form of a resulting estimate matrix or array. The estimate matrix is provided to a differencing module 20 that determines (312) the difference (or residual) between the estimate values in the estimate matrix and corresponding input values in the input array. The residuals are then used by an alert or analysis management module (or just alert module) 22 to determine (314) if a fault exists.

[0029] As shown in dashed line, the monitoring system 10 also may have a Localization Module 28 that changes which data from the reference library is used to form (308) a subset or matrix **D(t)** (referred to as a three-dimensional collection of learned sequential pattern matrices below (FIG. 6)) to compare to the vectors in each input array. Otherwise, the matrix **D(t)** of reference data may remain the same for all of the input matrices as explained in detail below. Also, the monitoring system may have an adaption module 30 that continuously places the input vectors into the reference library to update the data in the library or when a certain event occurs, such as when the model receives data that indicates a new normal condition of the machine not experienced before, for example. This is also described in detail below.

[0030] The alert module 22 may provide alerts as well as the residuals directly to an interface or output module 24 for a user to perform their own diagnostic analysis, or a diagnostic module 26 may be provided to analyze the exact nature of the cause of the fault to report diagnostic conclusions and severity ratings to the user through the output module 24.

[0031] The output module 24, which may include mechanisms for displaying these results (for example, computer screens, PDA screens, print outs, or web server), mechanisms for storing the results (for example, a database with query capability, flat file, XML file), and/or mechanisms for communicating the results to a remote location or to other computer programs (for example, software interface, XML datagram, email data packet,

asynchronous message, synchronous message, FTP file, service, piped command and the like).

[0032] A more detailed explanation of the empirical model 14 requires certain knowledge of kernel regression. In pattern recognition techniques such as kernel regression, a pattern consists of input data (as described above) grouped together as a vector. The data for each vector is collected from a piece of equipment at a common point in time. Here, however, and as explained in greater detail below, the pattern (vector) of contemporaneous sensor values associated with existing kernel regression methods is augmented with temporally-related information such as sequential patterns from successive moments in time or the output from time-dependent functions (for example, filters, time-derivatives and so forth) applied to the patterns from successive moments in time. Therefore, the individual patterns (vectors) processed by traditional kernel regression methods are replaced by temporally-related sequences of patterns that form an array (or simply pattern arrays or pattern matrices).

[0033] All kernel-based modeling techniques, including kernel regression, radial basis functions, and similarity-based modeling can be described by the equation:

$$\mathbf{x}_{est} = \sum_{i=1}^{L} \mathbf{c}_i K(\mathbf{x}_{new}, \mathbf{x}_i) \tag{1}$$

where a vector $\mathbf{x}_{est}$ of sensor signal or sensor value estimates is generated as a weighted sum of results of a kernel function **K**, which compares the input vector $\mathbf{x}_{new}$ of sensor measurements to **L** learned patterns of sensor data, $\mathbf{x}_i$. $\mathbf{x}_i$ is formed of reference or learned data in the form of vectors (also referred to as observations, patterns, snapshots, or exemplars). The kernel function results are combined according to weights $\mathbf{c}_i$, which may be in the form of vectors and can be determined in a number of ways. The above form is an "autoassociative" form, in which all estimated output signals are also represented by input signals. In other words, for each input value, an estimate sensor value is calculated. This contrasts with the "inferential" form in which certain estimate

output values do not represent an existing input value, but are instead inferred from the inputs:

$$y_{est} = \sum_{i=1}^{L} \mathbf{c}_i K(\mathbf{x}_{new}, \mathbf{x}_i) \tag{2}$$

where in this case, $\mathbf{y}_{est}$ is an inferred sensor estimate obtained from the kernel-based comparison of the input vectors $\mathbf{x}_{new}$ of other parameters to the L learned exemplars $\mathbf{x}_i$ of those parameters. Each learned exemplar $\mathbf{x}_i$ is associated with another exemplar vector $\mathbf{y}_i$ of the parameters to be estimated, which are combined in a weighted fashion according to the kernel K and vectors $\mathbf{c}_i$ (which are functions at least in part of the $\mathbf{y}_i$) to predict output $\mathbf{y}_{est}$. In a similar fashion, more than one sensor can be simultaneously inferred.

[0034] What is common to the kernel-based estimators is the kernel function, and the generation of a result from a linear combination of exemplars (for example, a matrix of the exemplars or vectors), based on the kernel results and the vectors $\mathbf{c}_i$ that embodies the exemplars. Kernel function **K** is a generalized inner product, but in one form has the further characteristic that its absolute value is maximum when $\mathbf{x}_{new}$ and $\mathbf{x}_i$ are identical.

[0035] According to one embodiment of the invention, a kernel-based estimator that can be used to provide the model is Kernel Regression, exemplified by the Nadaraya-Watson kernel regression form:

$$\mathbf{y}_{est} = \frac{\sum_{i=1}^{L} \mathbf{y}_i^{out} K(\mathbf{x}_{new}, \mathbf{x}_i^{in})}{\sum_{i=1}^{L} K(\mathbf{x}_{new}, \mathbf{x}_i^{in})} \quad \text{(Inferential form)} \tag{3}$$

$$\mathbf{x}_{est} = \frac{\sum_{i=1}^{L} \mathbf{x}_i K(\mathbf{x}_{new}, \mathbf{x}_i)}{\sum_{i=1}^{L} K(\mathbf{x}_{new}, \mathbf{x}_i)} \quad \text{(Autoassociative form)} \tag{4}$$

In the inferential form, a multivariate estimate of inferred parameters $\mathbf{y}_{est}$ is generated from the results of the kernel **K** operator on the input vector of parameter measurements $\mathbf{x}_{new}$ and the **L** learned exemplars $\mathbf{x}_i$, linearly combined according to respective learned vectors $\mathbf{y}_i$, which are each associated with each $\mathbf{x}_i$, and normalized by the sum of kernel results. The $\mathbf{y}_i$ represent the **L** sets of learned measurements for the parameters in **Y**, which were associated with (such as, measured contemporaneously with) the learned measurements of parameters in **X**. By way of example, **X** may comprise a plurality of pressure readings, while **Y** may represent a corresponding plurality of temperature readings from a common system. In other words, the pressure readings may be used to calculate weights which are then used in a calculation with $\mathbf{y}_i$ (the reference vector with previous values of the missing parameter) to calculate estimated temperature readings or sensor values for $\mathbf{y}_{est}$.

[0036] In the autoassociative form of the kernel regression, a multivariate estimate of parameters $\mathbf{x}_{est}$ is generated by a normalized linear combination of the learned measurements of those parameters $\mathbf{x}_i$ (for example, in the form of a matrix **D** of exemplars described below), multiplied by the kernel operation results for the input vector $\mathbf{x}_{new}$ vis-à-vis the learned observations $\mathbf{x}_i$.

[0037] In kernel regression for the present example, the $\mathbf{c}_i$ from equations 1 and 2 above are composed of the learned exemplars normalized by the sum of the kernel comparison values. The estimate vectors, $\mathbf{y}_{est}$ or $\mathbf{x}_{est}$, comprise a set of estimated parameters that are, according to one example, differenced with actual measured values ($\mathbf{x}_{new}$, or $\mathbf{y}_{new}$, which is not input to the model in the inferential case) to provide residuals.

[0038] In a specific example of Kernel regression, a similarity-based model (SBM) can be used as the model according to the present invention. Whereas the Nadaraya-Watson kernel regression provides estimates that are smoothed estimates given a set of (possibly noisy) learned exemplars, SBM provides interpolated estimates that fit the learned exemplars when they also happen to be the input as well, such as if the input vector is identical to one of the learned exemplars. This can be advantageous in detecting deviations in

parameters, since noise in these signals will be overfit to a certain extent (if noise was similarly present on the exemplars from which the model was made), thus removing the noise somewhat from the residuals as compared to the Nadaraya-Watson kernel regression approach. SBM can be understood as a form of kernel-based estimator by rewriting the kernel function **K** as the operator ⊗, and equating the set of learned exemplars $\mathbf{x_i}$ as a matrix **D** with the elements of $\mathbf{x_i}$ forming the rows, and the $\mathbf{x_i}$ observations forming its columns. Then:

$$K_{i=1}^{L}(\mathbf{x}_i, \mathbf{x}_{new}) = (\mathbf{D}^T \otimes \mathbf{x}_{new}) \quad (5)$$

where **D** has been transposed, which results in a column vector of kernel values, one for each observation $\mathbf{x_i}$ in **D**. Similarly, the comparison of all exemplars with each other can be represented as:

$$K_{i,j=1}^{L}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{D}^T \otimes \mathbf{D}) \quad (6)$$

[0039] Then, the autoassociative form of SBM generates an estimate vector according to:

$$\mathbf{x}_{est} = \mathbf{D} \cdot (\mathbf{D}^T \otimes \mathbf{D})^{-1} \cdot (\mathbf{D}^T \otimes \mathbf{x}_{new}) \quad (7)$$

where $\mathbf{x_{est}}$ is the estimate vector, $\mathbf{x_{new}}$ is the input observation, and **D** is a learned vector matrix comprising the set (or subset) of the learned exemplary observations of the parameters. The similarity operator or kernel is signified by the symbol ⊗, and has the general property of rendering a similarity score for the comparison of any two vectors from each of the operands. Thus, the first term ($\mathbf{D^T} \otimes \mathbf{D}$) would yield a square matrix of values of size equal to the number of observations in **D** as shown in equation (6) above. The term ($\mathbf{D^T} \otimes \mathbf{x_{new}}$) would yield a vector of similarity values, one similarity value for each vector in **D** as shown in equation 5. This similarity operator is discussed in greater detail below. The equation is shown schematically on FIG. 3 and shows how each component of the equation is formed by vectors as represented by the rectangular boxes.

In this example, each vector contains sensor values for parameters 1-5 (although this could also include other non-sensor values as described above). It will be understood that the numbers 1-5 indicate which parameter is being represented and not the exact sensor value. Thus, the sensor value itself will be different for the different parts of the equation (for example, the value for parameter 1 may be different in $\mathbf{x}_{new}$ versus that in $\mathbf{D}$ versus that in $\mathbf{x}_{est}$).

[0040] It will also be understood that for equation (7), time domain information among a group of input vectors is ignored to generate estimates. In other words, since equation (7) generates an estimate vector by using a single input vector $\mathbf{x}_{new}$, the order in which the vectors in a group of input vectors are analyzed to generate estimate vectors is largely unimportant. If a certain order related to time (such as sequential) is needed later in the process to determine if a fault exists or to diagnose the particular type of fault for example, then the vectors can be ordered as desired after generating the estimates.

[0041] The estimate can further be improved by making it independent of the origin of the data, according to the following equation, where the estimate is normalized by dividing by the sum of the "weights" created from the similarity operator:

$$\mathbf{x}_{est} = \frac{\mathbf{D} \cdot (\mathbf{D}^T \otimes \mathbf{D})^{-1} \cdot (\mathbf{D}^T \otimes \mathbf{x}_{new})}{\sum \left((\mathbf{D}^T \otimes \mathbf{D})^{-1} \cdot (\mathbf{D}^T \otimes \mathbf{x}_{new}) \right)} \qquad (8)$$

[0042] In the inferential form of similarity-based modeling, the inferred parameters vector $\mathbf{y}_{est}$ is estimated from the learned observations and the input according to:

$$\mathbf{y}_{est} = \mathbf{D}_{out} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{D}_{in} \right)^{-1} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{x}_{in} \right) \qquad (9)$$

where $\mathbf{D}_{in}$ has the same number of rows as actual sensor values (or parameters) in $\mathbf{x}_{in}$, and $\mathbf{D}_{out}$ has the same number of rows as the total number of parameters including the inferred parameters or sensors. Equation (9) is shown schematically on FIG. 4 to show the location of the vectors, the input values (1 to 5), and the resulting inferred values (6-7).

[0043] In one form, the matrix of learned exemplars $\mathbf{D_a}$ can be understood as an aggregate matrix containing both the rows that map to the sensor values in the input vector $\mathbf{x_{in}}$ and rows that map to the inferred sensors:

$$\mathbf{D}_a = \left[\frac{\mathbf{D}_{in}}{\mathbf{D}_{out}}\right] \tag{10}$$

Normalizing as before using the sum of the weights:

$$\mathbf{y}_{est} = \frac{\mathbf{D}_{out} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{D}_{in}\right)^{-1} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{x}_{in}\right)}{\sum\left(\left(\mathbf{D}_{in}^T \otimes \mathbf{D}_{in}\right)^{-1} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{x}_{in}\right)\right)} \tag{11}$$

It should be noted that by replacing $\mathbf{D_{out}}$ with the full matrix of leaned exemplars $\mathbf{D_a}$, similarity-based modeling can simultaneously calculate estimates for the input sensors (autoassociative form) and the inferred sensors (inferential form):

$$\left[\frac{x_{est}}{y_{est}}\right] = \frac{\mathbf{D}_a \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{D}_{in}\right)^{-1} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{x}_{in}\right)}{\sum\left(\left(\mathbf{D}_{in}^T \otimes \mathbf{D}_{in}\right)^{-1} \cdot \left(\mathbf{D}_{in}^T \otimes \mathbf{x}_{in}\right)\right)} \tag{12}$$

[0044] Referring to FIG. 5, Equation (12) uses the matrix $\mathbf{D_a}$ with reference values for both the input and inferred values. This results in an estimate vector with both representative input values and inferred values.

[0045] Yet another kernel-based modeling technique similar to the above is the technique of radial basis functions. Based on neurological structures, radial basis functions make use of receptive fields, in a special form of a neural network, where each basis function forms a receptive field in the *n*-dimensional space of the input vectors, and is represented by a hidden layer node in a neural network. The receptive field has the form of the kernels described above, where the "center" of the receptive field is the exemplar that particular hidden unit represents. There are as many hidden unit receptive fields as there are

exemplars. The multivariate input observation enters the input layer, which is fully connected with the hidden layer. Thus, each hidden unit receives the full multivariate input observation, and produces a result that is maximum when the input matches the "center" of the receptive field, and diminishes as they become increasingly different (akin to SBM described above). The output of the hidden layer of receptive field nodes is combined according to weights $\mathbf{c_i}$ (as above in equation 1).

[0046] As mentioned above, the kernel can be chosen from a variety of possible kernels, and in one form is selected such that it returns a value (or similarity score) for the comparison of two identical vectors that has a maximum absolute value of all values returned by that kernel. While several examples are provided herein, they are not meant to limit the scope of the invention. Following are examples of kernels/similarity operators that may be used according to the invention for the comparison of any two vectors $\mathbf{x_a}$ and $\mathbf{x_b}$.

$$K_h(\mathbf{x}_a, \mathbf{x}_b) = e^{-\frac{\|\mathbf{x}_a - \mathbf{x}_b\|^2}{h}} \tag{13}$$

$$K_h(\mathbf{x}_a, \mathbf{x}_b) = \left(1 + \frac{\|\mathbf{x}_a - \mathbf{x}_b\|^\lambda}{h}\right)^{-1} \tag{14}$$

$$K_h(\mathbf{x}_a, \mathbf{x}_b) = 1 - \frac{\|\mathbf{x}_a - \mathbf{x}_b\|^\lambda}{h} \tag{15}$$

[0047] In equations 13-15, the vector difference, or "norm", of the two vectors is used; generally this is the 2-norm, but could also be the 1-norm or p-norm. The parameter **h** is generally a constant that is often called the "bandwidth" of the kernel, and affects the size of the "field" over which each exemplar returns a significant result. The power λ may also be

used, but can be set equal to one. It is possible to employ a different **h** and λ for each exemplar $\mathbf{x}_i$. By one approach, when using kernels employing the vector difference or norm, the measured data should first be normalized to a range of 0 to 1 (or other selected range), for example, by adding to or subtracting from all sensor values the value of the minimum reading of that sensor data set, and then dividing all results by the range for that sensor. Alternatively, the data can be normalized by converting it to zero-centered mean data with a standard deviation set to one (or some other constant). Furthermore, a kernel/similarity operator according to the invention can also be defined in terms of the elements of the observations, that is, a similarity is determined in each dimension of the vectors, and those individual elemental similarities are combined in some fashion to provide an overall vector similarity. Typically, this may be as simple as averaging the elemental similarities for the kernel comparison of any two vectors **x** and **y**:

$$K(\mathbf{x},\mathbf{y}) = \frac{1}{L}\sum_{m=1}^{L} K(x_m, y_m) \tag{16}$$

Then, elemental similarity operators that may be used according to the invention include, without limitation:

$$K_h(x_m, y_m) = e^{\frac{-|x_m - y_m|^2}{h}} \tag{17}$$

$$K_h(x_m, y_m) = \left(1 + \frac{|x_m - y_m|^{\lambda}}{h}\right)^{-1} \tag{18}$$

$$K_h(x_m, y_m) = 1 - \frac{|x_m - y_m|^{\lambda}}{h} \tag{19}$$

[0048] The bandwidth **h** may be selected in the case of elemental kernels such as those shown above, to be some kind of measure of the expected range of the *mth* parameter of the observation vectors. This could be determined, for example, by finding the difference between the maximum value and minimum value of a parameter across all exemplars. Alternatively, it can be set using domain knowledge irrespective of the data present in the exemplars or reference vectors. Furthermore, it should be noted with respect to both the vector and elemental kernels that use a difference function, if the difference divided by the bandwidth is greater than 1, it can be set equal to one, resulting in a kernel value of zero for equations 14, 15, 18 and 19, for example. Also, it can readily be seen that the kernel or similarity operator can be modified by the addition or multiplication of different constants, in place of one, **h**, λ, and so on. Trigonometric functions may also be used, for example:

$$K_h(x_m, y_m) = \left(1 + \sin\left(\frac{\pi}{2h}|x_m - y_m|\right)\right)^{-1} \qquad (20)$$

[0049] In one form, the similarity operator or kernel generally provides a similarity score for the comparison of two identically-dimensioned vectors, which similarity score:

1. Lies in a scalar range, the range being bounded at each end;
2. Has a value of one (or other selected value) at one of the bounded ends, if the two vectors are identical;
3. Changes monotonically over the scalar range; and
4. Has an absolute value that increases as the two vectors approach being identical.

[0050] All of the above methods for modeling use the aforementioned kernel-based approach and use a reference library of the exemplars. The exemplars (also called reference observations or reference vectors) represent "normal" behavior of the modeled system. Optionally, the available reference data can be down-selected to provide a characteristic subset to serve as the library of exemplars, in which case a number of techniques for

"training" the kernel-based model can be employed. In this case, the down-selected library itself may form the matrix **D** used in the equations above. According to one training method, at least those observations are included in the library that have a highest or lowest value for a given parameter across all available reference observations. This can be supplemented with a random selection of additional observations, or a selection chosen to faithfully represent the scatter or clustering of the data. Alternatively, the reference data may be clustered, and representative "centroids" of the clusters formed as new, artificially generated exemplars, which then form the library. A wide variety of techniques are known in the art for selecting the observations to comprise the library of exemplars. Thus, at least in general terms for this case, the matrix **D** remains the same in equation (7) for all of the input vectors $\mathbf{x}_{in}$ unless the library is changed (i.e. such as when the library is updated).

[0051] In an alternative arrangement for both the inferential and autoassociative forms of the empirical kernel-based model, matrix **D** can be reconfigured for each input vector $\mathbf{x}_{in}$ so that the model can be generated "on-the-fly" based on qualities of the input observation, and drawing from a large set of learned observations, i.e., a reference set. One example of this is described in U.S. Patent No. 7,403,869. This process is called localization. Accordingly, the inferential and autoassociative forms of kernel-based modeling can be carried out using a set of learned observations $\mathbf{x}_i$ (matrix **D**) that are selected from a larger set of reference observations, based on the input observation. Kernel-based models are exceptionally well suited for this kind of localization because they are trained in one pass and can be updated rapidly. Advantageously, by drawing on a large set of candidate exemplars, but selecting a subset with each new input observation for purposes of generating the estimate, the speed of the modeling calculation can be reduced and the robustness of the model improved, while still well characterizing the dynamics of the system being modeled.

[0052] For the monitoring system 10, the localization module 28 can use a variety of criteria to constitute the localized matrix membership for collection **D(t),** including the application of the similarity operator itself. In general, however, the input observation 32, comprising the set of parameters or derived features that are to be estimated by the model

as part of the monitoring process, are provided to the localization module 28, which accesses a large store of exemplar observations in the form of reference library 18, in order to select a subset of those exemplar observations to build the model. Localization module 28 selects exemplars from library 18 that are relevant to the input observation 32, which can be a much smaller set than the size of the library. By way of example, the reference library 18 might comprise 100,000 exemplar observations that characterize the normal dynamics of the system represented by the parameters being modeled, but the localization module 28 might select only a few dozen observations to build a localized model in response to receiving the input observation 32. The selected exemplar observations are then provided to the now localized model 14. In the vector-based system, these observations then comprise the set of learned exemplars $\mathbf{x}_i$ for purposes of the kernel-based estimator (also shown as **D** in connection with SBM above). The estimate observation $\mathbf{x}_{est}$ is then generated accordingly as described above. For the monitoring system 10, the selected learned exemplars each may represent a vector at time point $\mathbf{t}_p$, such that a sequential pattern matrix is built for each vector at $\mathbf{t}_p$ to form the collection **D(t)** described below. As the next input observation 32 is presented to the monitoring system 10, the process is repeated, with selection of a new and possibly different subset of exemplars from library 18, based on the new input observation.

[0053] According to one approach, the input observation 32 can be compared to the reference library 18 of learned observations, on the basis of a clustering technique. Accordingly, the exemplar observations in library 18 are clustered using any of a number of techniques known in the art for clustering vectors, and the localization module 28 identifies which cluster the input observation 32 is closest to, and selects the member exemplars of that cluster to be the localized observations provided to the localized model 14. Suitable clustering methods include k-means and fuzzy *c*-means clustering, or a self-organizing map neural network.

[0054] According to another approach, a kernel can be used to compare the input observation 32 to each exemplar in the library 18 to yield a similarity value that provides a ranking of the reference observations vis-à-vis the input observation. Then, a certain top fraction of them can be included in the localized collection **D(t)**. As a further refinement of this

localization aspect, observations in the ranked list of all reference observations are included in localized collection **D(t)** to the extent one of their component elements provides a value that "brackets" the corresponding value in the input vector. For example, a search down the ranked list is performed until values in the input vector are bracketed on both the low and high side by a value in one of the reference observations. These "bracketing" observations are then included in localized collection **D(t)** even if other observations in library 18 have higher similarity to the input. The search continues until all input values in the input vector are bracketed, until a user-selectable maximum limit of vectors for building sequential pattern matrices to include in collection **D(t)** is reached, or until there are no further reference observations that have sufficiently high similarity to the input to surpass a similarity threshold for inclusion.

[0055] Other modifications in determining the membership of localized collection **D(t)** are contemplated. By way of example, in both the clustering selection method and the similarity selection method described above, the set of elements, i.e., parameters used to comprise the vectors that are clustered or compared with the kernel for similarity, may not be identical to those used to generate the model and the estimate, but may instead be a subset, or be a partially overlapping set of parameters. As mentioned above, an additional step for the system 10 and model 14 is then performed to generate the collection **D(t)**. Specifically, once the vectors (referred to as primary vectors $\mathbf{t_p}$) are selected for inclusion in collection **D(t)**, other temporally related vectors (whether looking forward or looking back in time) are selected for each primary vector to form a learned sequential pattern matrix for each primary vector and included in the collection **D(t)**. The process for choosing the temporally related vectors is explained below. It will be understood that the localization by the module 28 can be applied to any of the three-dimensional collections of learned sequential pattern matrices described in detail below.

[0056] Turning now to the incorporation of the time domain information into the model 14, by one approach for the monitoring system 10 described herein, the above kernel function, which operates to compare the similarity of two vectors, is replaced by an extended kernel function **K** that operates on two identically-dimensioned arrays:

$$\vec{\vec{K}}(\vec{\vec{X}}_{\text{new}}, \vec{\vec{X}}_{\text{i}}) \qquad (20)$$

where $\mathbf{X_{new}}$ is an input pattern array and $\mathbf{X_i}$ is a learned pattern array. A pattern array or pattern matrix is composed of a sequence of temporally-related vectors, where each of its constituent vectors contains sensor measurements from a distinct moment in time. One of the vectors in a pattern array is designated the primary vector, and the time at which its data is derived is designated the current primary time point $\mathbf{t_p}$. The other vectors are associated with time points that relate to the primary time point in a systematic manner.

[0057] In one form, the primary time point is the most recent of the time points that compose a sequence of the time-ordered points (or time-ordered vectors that represent those time points) in the pattern array. By one approach, the other time points are equally-spaced and precede the primary time point by integer multiples of a time step $\mathbf{\Delta t}$ providing uniform time intervals between the time points. For a given number of samples $\mathbf{n_{lb}}$, the time points form an ordered sequence: $(\mathbf{t_p - n_{lb}\Delta t, t_p - (n_{lb}\text{-}1)\Delta t, \ldots, t_p - 2\Delta t, t_p - \Delta t, t_p})$. The sequence of time points defines a look-back pattern array,

$$\vec{\vec{X}}(t_p) = \left[x(t_p - n_{1b}\Delta t), x(t_p - (n_{1b} - 1)\Delta t), \cdots x(t_p - 2\Delta t), x(t_p - \Delta t), x(t_p)\right] \qquad (21)$$

[0058] As shown in FIG. 6, the primary vector $\mathbf{t_p}$ is positioned as the right-most column of each pattern array, and the other ($\mathbf{n_{lb}}$) data vectors are column vectors that are located to the left of the primary vector $\mathbf{t_p}$. The rows of the pattern arrays correspond to short segments of the time-varying signals from the modeled sensors.

[0059] By using look-back pattern arrays, the extended kernel function in equation (20) can be applied to real-time system monitoring. The primary vector $\mathbf{t_p}$ (which means the vector at time point $\mathbf{t_p}$) in the input pattern array $\mathbf{X_{new}}$ contains system data from the current point in time, and the remainder of the array consists of data vectors from recent time points in the past. Thus, not only does the input pattern array contain the current, albeit static, vector used by traditional kernel methods, but it also contains a sequence of vectors that express the developing, dynamic behavior of the monitored system. As system time progresses, new input pattern arrays are formed which contain much of the same data as

preceding arrays except that new primary vectors appear in the right-most position of the arrays, and the oldest vectors are dropped from the left-most position. Thus, a single input vector representing a single instant in time will be used in multiple input pattern arrays $\mathbf{X_{new}}$, and assuming the vectors are used in sequence, the vectors will be used the same number of times as there are vectors in the array. In this manner, the input pattern array describes a moving window of patterns through time. Here, moving window means a set or group of a fixed number of vectors in chronological order that changes which vectors are included in the set as the window moves along the timeline or along a sequence of time-ordered sensor value vectors.

[0060] The pattern array defined in equation (21) above contains $\mathbf{n_{lb}}$ data vectors that span a window in time equal to $\mathbf{n_{lb}*\Delta t}$. The data vectors are equally-spaced in time for this example. Another way to say this is that each input pattern array or matrix is defined only by uniform time intervals between time points represented by the input vectors within the input pattern array $\mathbf{X_{new}}$.

[0061] Alternatively, a kernel can be used to compare pattern arrays that span differing lengths of time. If a pattern array contains data from time points that are spaced by one time step $\mathbf{\Delta t_1}$ (say one second apart for example), and if the time points of another pattern array differ by a second time step $\mathbf{\Delta t_2}$ (say ten seconds apart for example), then the pattern arrays will span two differing time windows: $\mathbf{n_{lb}*\Delta t_1}$ and $\mathbf{n_{lb}*\Delta t_2}$ so that there are two pattern arrays that represent different durations. In one form, as long as the pattern arrays contain the same number of vectors even though one pattern array may have different time intervals between the vectors (or time points) than in another pattern array, a kernel function that matches vectors from the same positions in the two pattern arrays (such as right-most with right-most, second from right with second from right, and onto left-most with left-most) will be capable of operating across varying time scales. Thus, in one example, the matrices may extend across differently spaced time points so that the time interval spacing could correspond to the harmonics (**1/f**) of the peaks in a spectral time signal. It also will be understood that this difference in time period or duration covered by the pattern arrays may be used between the learned pattern arrays and input pattern arrays, from input pattern array to input pattern array, from learned pattern array to

learned pattern array, or any combination of these as long as each vector in the input pattern array has a corresponding learned exemplar in the learned pattern arrays (or in other words, both learned and input matrices have the same number of vectors).

[0062] According to another example, a kernel can be used to compare pattern arrays whose pattern vectors are not equally-spaced in time. Instead of spacing pattern vectors by a constant time interval or step, the time step can vary by position within the pattern array. By using small time steps for most recent vectors (positioned near the right side of the array) and larger time steps for the older vectors (positioned near the left side of the array), the kernel function will focus attention on the most recent changes while still retaining some effect from changes in the more distant past.

[0063] Referring again to FIG. 1, an additional filtering step may be performed on the pattern arrays by a filter module 106 prior to analysis by the kernel function (equation (21)). When the filtering is used, it is performed on both the reference vectors and the input vectors to avoid any substantial, unintentional mismatch between the two resulting signal values to be used for generating estimates. In the filtering step, each of the time-varying sensor segments (rows of a pattern array) are processed by a filtering algorithm to either smooth the data in the segment or to calculate statistical features from the data. Smoothing algorithms, such as moving window averaging, cubic spline filtering, or Savitsky-Golay filtering, capture important trends in the original signal, but reduce the noise in the signal. Since smoothing algorithms produce smoothed values for each of the elements in the input signal, they produce a pattern array that has the same dimensions as the original pattern array of sensor data. Alternately, the filtering step can consist of the application of one or more feature extraction algorithms to calculate statistical features of the data in each signal. These features may include the mean, variance, or time derivatives of the signal data. As long as the same number of feature extraction algorithms is applied to the data in the pattern arrays, the number of data vectors in the original pattern array can vary.

[0064] As described above, there are numerous methods in which pattern arrays are used to represent temporal information from the system being modeled. These methods include,

but are not limited to, sequences of data vectors from equally-spaced time points, sequences of data vectors that span differing time periods such that the pattern arrays have varying durations, and sequences whose data vectors are not equally-spaced in time. The input pattern array may have different intervals than the reference pattern arrays, or they may be the same. In addition, the pattern sequences can be filtered by smoothing or feature extraction algorithms. The only limitation on the form of the pattern arrays or the arrays produced by filtering algorithms are that the two arrays processed by the extended kernel function (equation 20) be identically-dimensioned (i.e., having the same number of rows and columns).

[0065] Similar to the vector-based kernel function described above, the extended kernel function returns a scalar value or similarity measure, although here, the scalar value represents the similarity between two arrays rather than two vectors. The extended kernel function produces a similarity score that displays the same properties as the vector-based kernel function enumerated above. Namely, the similarity score is a scalar whose range is bounded; has a value of one (or other selected value) for one of the bounds when the two arrays are identical; varies monotonically over the range; and whose absolute value increases as the two arrays approach being identical. In addition, the extended kernel function operates on the matching temporal components of the two arrays. This means, for the example of two look-back pattern arrays, that the extended kernel function finds the similarity between the two primary vectors $\mathbf{t_p}$ from the reference and input pattern arrays respectively, then on the two data vectors to the left of the primary vectors **-1**, and so forth across the preceding vectors in the arrays.

[0066] One example of an extended kernel function is based on the similarity operator described in U.S. Patent No. 6,952,662. Letting $\mathbf{X_{new}}$ and $\mathbf{X_i}$ be two identically-dimensioned pattern arrays, containing data from $\mathbf{n_{sens}}$ sensors (or parameters) and spanning $\mathbf{n_{lb}}$ sequential time points, the extended kernel function is written as follows:

$$S(\vec{\vec{X}}_{new}, \vec{\vec{X}}_i) = \frac{1}{1 + \frac{1}{\rho}\left(\frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \theta_j(t)\right)^{\lambda}} \qquad (22)$$

where ρ and λ are constants. The time-dependent function **θ(t)** in equation 22 operates on the temporal elements of the pattern arrays, matching data from the same time point vectors in the two arrays. One means of accomplishing this temporal data matching is to use a weighted average of the temporal data for a given sensor **j**:

$$\theta_j(t) = \left[\sum_{k=1}^{n_{lb}} \left(W_k s_{j,k} \right) \right] \Big/ \sum_{k=1}^{n_{lb}} W_k \tag{23}$$

The similarity ($\mathbf{s_{j,k}}$) between data elements for a given sensor **j** is defined as the absolute difference of the data elements normalized by the range of normal operating data for a sensor $\mathbf{range_j}$. Thus, the time-dependent similarity function **θ(t)** for a given sensor's data is:

$$\theta_j(t) = \left[\sum_{k=1}^{n_{lb}} \left(\frac{W_k \left| \vec{X}_{new;j,k} - \vec{X}_{i;j,k} \right|}{range_j} \right) \right] \Big/ \sum_{k=1}^{n_{lb}} W_k \tag{24}$$

Combining equations 22 and 24, produces an extended kernel function for two pattern arrays:

$$S(\vec{X}_{new}, \vec{X}_i) = \frac{1}{1 + \frac{1}{\rho} \left[\frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \left(\frac{\sum_{k=1}^{n_{lb}} \left(\frac{W_k \left| \vec{X}_{new;j,k} - \vec{X}_{i;j,k} \right|}{range_j} \right)}{\sum_{k=1}^{n_{lb}} W_k} \right) \right]^{\lambda}} \tag{25}$$

[0067] Another example of an extended kernel function is based on the similarity operator described in U.S. Patent No. 7,373,283. Again letting $\mathbf{X_{new}}$ and $\mathbf{X_i}$ be two identically-

dimensioned pattern arrays, containing data from $\mathbf{n_{sens}}$ sensors and spanning $\mathbf{n_{lb}}$ sequential time points, this second extended kernel function is written as follows:

$$S(\vec{\vec{X}}_{new}, \vec{\vec{X}}_i) = \frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \left[\frac{1}{1 + \frac{1}{\rho}\left(\theta_j(t)\right)^{\lambda}} \right] \tag{26}$$

This extended kernel function utilizes the same time-dependent function **θ(t)** as defined by equations 23 and 24 to compare the temporal data of a given sensor in the two pattern matrices:

$$S(\vec{\vec{X}}_{new}, \vec{\vec{X}}_i) = \frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \left[\frac{1}{1 + \frac{1}{\rho}\left(\frac{\sum_{k=1}^{n_{lb}} \left(\frac{W_k \left| \vec{\vec{X}}_{new;j,k} - \vec{\vec{X}}_{i;j,k} \right|}{range_j} \right)}{\sum_{k=1}^{n_{lb}} W_k} \right)^{\lambda}} \right] \tag{27}$$

[0068] While referring to FIG. 6, the two extended kernel functions (equations 25 and 27) differ only in how they aggregate information from the modeled sensors, with the first equation representing the elemental form of a kernel function, and the second equation representing the vector difference form (such as 1-norm) of a kernel function. Both equations utilize weighted averaging to account for differences between the segments of time-varying signals in the two arrays $\mathbf{X_{new}}$ and $\mathbf{X_i}$. Specifically, for both example equations 25 and 27, and for each sequential learned pattern matrix **a** to **g**, the absolute difference is calculated for each corresponding pair of learned and input values. The

values correspond when they represent (1) the same sensor (or parameter) and (2) either the same time point within the pattern array (such as both values being from the primary time $\mathbf{t_p}$) or the same position relative to the other vectors in the array (such as when both values are on vectors that are second from the right within the pattern array). The absolute differences from the pairs of learned and input values are combined via weighted averaging to obtain a resulting single average value for the particular sensor. This is repeated for each sensor or parameter (1 to 5) represented by the pattern matrices **a** to **g** and pattern arrays $\mathbf{X_{new}}$ so that there is one resulting average scalar for each sensor/parameter in the weighted averaging step.

[0069] Then, in the first extended kernel function (equation 25), the results from the weighted averaging step are in turn averaged across all sensors to produce a scalar value for the array-to-array comparison. Finally, this scalar value is transformed into a value that adheres to the properties of a similarity score as described above so that it falls within a range of zero to one for example, with one meaning identical. This process is then repeated for each learned sequential pattern matrix **a** to **g** in the three-dimensional collection **D(t)**. In the second extended kernel function (equation 27), the results from the weighted averaging step are converted into similarity scores right away, one for each sensor. Then this vector of similarity scores is averaged so that a single similarity score is returned by the function for each learned sequential pattern matrix **a** to **g** in the three-dimensional collection **D(t)**.

[0070] When used within context of similarity-based modeling, the extended kernel functions described above can also be termed extended similarity operators without loss of generality. The notation used in the above equations ($\mathbf{S(X_{new},X_i)}$) can also be written using the traditional similarity operator symbol ($\mathbf{X_{new} \otimes X_i}$).

[0071] Extended versions of other vector-based kernel functions defined above (for example, equations 13 through 20) can be constructed by using weighted averaging to match temporal data from the same time points in two sequential pattern arrays. For instance, letting $\mathbf{X_{new}}$ and $\mathbf{X_i}$ be two identically-dimensioned pattern arrays, containing data from

$\mathbf{n_{sens}}$ sensors and spanning $\mathbf{n_{lb}}$ sequential time points, an extended version of the kernel function defined in equation 16, using the elemental similarity operator of equation 17, is:

$$K_h(\vec{\vec{X}}_{new}, \vec{\vec{X}}_i) = \frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \left[\exp\left(-\frac{1}{h} \left[\frac{\sum_{k=1}^{n_{lb}} W_k \left| \vec{\vec{X}}_{new;j,k} - \vec{\vec{X}}_{i;j,k} \right|}{\sum_{k=1}^{n_{lb}} W_k} \right]^2 \right) \right] \quad (28)$$

[0072] Weighted averaging (equation 22) is used to account for differences between segments of the time-varying signals in pattern arrays since the weights can be selected such that more recent data are more heavily weighted than outdated data. Thus, data from the primary time point $\mathbf{t_p}$ are typically given the highest weight, with data from preceding time points (equation 21) given ever-decreasing weights. Numerous schemes can be used to define the weights, such as having them decline linearly or exponentially with time relative to the primary time point.

[0073] It will be understood that various other time-dependent functions **θ(t)** can be used to match data from sequential time points in two segments of time-varying signals. Such methods include, but are not limited to, other weighted norms (2-norm and p-norm) and maximum, minimum, or median difference. All that is required of the function is that it returns a scalar value that is minimized (a value of 0) if the two sequences are identical and increases in value as the sequences become more different.

[0074] In order to combine the concept of sequential pattern arrays with an extended similarity operator (for example, equation 25 or 27) in the autoassociative form of SBM (equation 7), the concept of the vector-based learned vector matrix **D** is extended. In the standard form of SBM described above, the learned vector matrix consists of a set of learned exemplars (vectors) selected from various points in time during periods of normal operation. Letting the time points from which these vectors are selected represent primary time points, each learned vector can be expanded into a learned sequential pattern matrix by collecting data from a sequence of time points that precede each primary time point. In this manner, the learned vector matrix **D** is expanded into a

collection of learned sequential pattern matrices **D(t)**. This collection of learned pattern matrices forms a three-dimensional matrix, wherein the dimensions represent the modeled sensors or parameters in a first dimension, the learned exemplars (vectors) from various primary time points in a second dimension, and time relative to the primary time points in a third dimension.

[0075] The training methods described above that are used for constructing the learned vector matrix used in vector-based forms of SBM can be utilized to create the three-dimensional collection of learned sequential pattern matrices **D(t)** required by the sequential pattern forms of SBM. This is accomplished by augmenting each reference vector selected by a training algorithm with reference vectors from preceding time points to construct a sequential pattern matrix. The collection of learned pattern matrices, one for each reference vector selected by a training algorithm, is drawn from reference library 18 of exemplars which represents "normal" behavior of the modeled system. If the time-inferential form of sequential SBM (described below) is used, then additional vectors from succeeding time points are added to each sequential pattern matrix.

[0076] The training methods that are used for the vector-based forms of SBM select exemplars (vectors) from various points in time during periods of normal operation, without regard to the time domain information inherent in the reference data. In the sequential pattern array forms of SBM, that time domain information is supplied by augmenting each of the selected exemplars with data vectors from a sequence of time points that immediately precede and (possibly) succeed the primary time points. In an alternative process for building and localizing the collection **D(t)** of sequential learned pattern matrices while factoring in the time domain information, each input pattern array may be compared to every sequence of reference vectors that is equal in number (namely, $\mathbf{n_{lb}+1}$) to that in the input pattern array. The comparison is accomplished by using an extended form of the similarity operator (for example, equation 25 or 27) to identify those sequences of reference vectors that are most similar to the input pattern array. Each of the identified sequences of reference vectors forms one of the sequential learned pattern matrices in the collection **D(t)**. Whatever the selection process, it is possible for a training method to select exemplars from primary time points that are quite near to one another. When two

exemplars are selected from nearby primary time points, the corresponding sequential pattern matrices may contain data vectors in common.

[0077] Referring to FIG. 6, equation 7 is shown with an input pattern array $\mathbf{X}_{new}$ and a three-dimensional collection of learned sequential pattern matrices **D(t)**. The input pattern array $\mathbf{X}_{new}$ may also be referred to as the current or actual pattern array or matrix since it includes the vector $\mathbf{t}_p$ representing a current instant in time, and in contrast to the learned pattern matrices in **D(t)**. In the illustrated example, the input pattern array $\mathbf{X}_{new}$ includes four vectors where vector $\mathbf{t}_p$ is the last (right-most) vector in the array. The other vectors are numbered as -3 to -1 referring to the number of time intervals before $\mathbf{t}_p$ for simplicity. Thus, it will be understood that vector -3 on FIG. 6 represents the same thing as $(\mathbf{t}_p - \mathbf{n}_{lb}\Delta\mathbf{t})$ where $\mathbf{n}_{lb} = 3$. As shown in FIG. 6, the three dimensions of the collection of learned sequential pattern matrices (modeled sensors, primary time points, and pattern sequences) are depicted as follows: the numbers 1 through 5 represent data from five modeled sensors, the four columns (or vectors) of numbers represent four sequential time points, and the seven layered rectangles each represent a sequential pattern matrix **a** to **g** each with a primary time point $\mathbf{t}_p$ selected from various periods of normal operation. The three-dimensional collection of learned sequential pattern matrices **D(t)** contains the seven sequential pattern matrices **a** to **g**. Thus, each sequential pattern matrix **a** to **g** comprises data from five sensors and four sequential points in time, and has the same dimensions as the input pattern matrix $\mathbf{X}_{new}$. For comparison, another way to visualize the difference between the prior vector-based equation with a two-dimensional matrix **D** (FIG. 3) and the three-dimensional collection of learned sequential pattern matrices **D(t)** (FIG. 6) is that the prior two-dimensional array would merely have been formed by a single matrix cutting across the seven sequential pattern arrays **a** to **g** to include only the $\mathbf{t}_p$ vectors from the three-dimensional collection **D(t)**.

[0078] In the right-most bracket in FIG. 6, the extended similarity operator (⊗) calculates the similarity between the input pattern array $\mathbf{X}_{new}$ and the seven learned sequential pattern matrices **a** to **g** as explained above. In the example of FIG. 6, and using the weighted averaging step from equations 25 or 27, the model compares the time-varying signal for sensor 1 in sequential pattern matrix **a** to the time-varying signal for sensor 1 in the input

pattern array $\mathbf{X_{new}}$ to obtain a single average value for sensor 1. This is repeated for sensors 2-5 until one average value is provided for each sensor. Then, these scalar values (or similarity scores for equation 27) are averaged to determine a single similarity measure for sequential pattern matrix **a**. This is then repeated for each sequential pattern matrix **b** to **g**, returning a similarity vector containing seven similarity scores, one similarity score for each learned sequential pattern matrix **a** to **g**.

[0079] The operation in the middle bracket produces a seven-by-seven square similarity matrix of similarity values, one for each combination of a pair of learned sequential pattern matrices **a** to **g** in collection **D(t)**. Multiplication of the inverse of the resulting similarity matrix with the similarity vector produces a weight vector containing seven elements. In a final step, the weight vector is multiplied by the collection **D(t)** to create an estimate matrix $\mathbf{X_{est}}$. In one form, the estimate matrix $\mathbf{X_{est}}$ is the same size as the input pattern array $\mathbf{X_{new}}$ so that it has an estimate vector that corresponds to each of the time periods represented by the input vectors in the input pattern arrays. In the present example of FIG. 6, the estimate matrix $\mathbf{X_{est}}$ has an estimate vector for the current moment in time $\mathbf{t_p}$ and for each of the three preceding time points **-1** to **-3** as if formed in a look-back window. The use of the estimate matrix $\mathbf{X_{est}}$ is described in further detail below. It also should be noted that the preceding vectors grouped together with or without the current or primary vector may be called a look-back window anywhere herein, and the succeeding vectors grouped together with or without the current or primary vector may be called a look-ahead window explained below and anywhere herein.

[0080] Extensions to the inferential form of SBM (equation 9) that utilize sequential pattern matrices with an extended similarity operator are readily apparent. Analogous to the vector-based form of inferential modeling, the three-dimensional collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ can be understood as an aggregate matrix containing learned sequential pattern matrices **a** to **g** that map to the sensor values in the input pattern array $\mathbf{X_{in}}$ and sequential pattern matrices **a** to **g** that map to the inferred sensors $\mathbf{D_{out}(t)}$. Referring to FIG. 7, equation 9 is shown with an input pattern array $\mathbf{X_{in}}$ and a three-dimensional collection of learned sequential pattern matrices $\mathbf{D_{in}(t)}$ with seven learned sequential pattern matrices **a** to **g** for the five input sensors 1 to 5. It is

understood that the aggregate matrix $\mathbf{D_a(t)}$ is a three-dimensional extension of the two-dimensional aggregate matrix defined in equation 10. Comparing the illustration in FIG. 7 to that in FIG. 6, the matrices within the brackets of both figures are identical except for how they are denoted. Therefore, the calculation of the weight vector for an inferential model proceeds in the same manner as that described above for an autoassociative model. Then, as in FIG. 4, the weight vector is multiplied by the learned sequential pattern array for the inferred sensors in FIG. 7 except that here matrix $\mathbf{D_{out}(t)}$ is now a three-dimensional collection of learned sequential pattern matrices, and this step forms an estimate matrix $\mathbf{Y_{est}}$ representing only the inferred sensors. As described above for the vector-based form of inferential modeling, the weight vector can also be multiplied by the full three-dimensional collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ that includes both $\mathbf{D_{in}(t)}$ and $\mathbf{D_{out}(t)}$ to generate estimate matrices for both input and inferred sensors (depicted in FIG. 8).

[0081] Inferential modeling enables calculation of estimates for sensors whose data are not included in the input data stream because reference data for these sensors are included in the three-dimensional collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ or $\mathbf{D_{out}(t)}$. Conceptually, an inferential model extrapolates along the dimension of the modeled sensors. It is also possible to create an inferential model that extrapolates in the time dimension. This can be understood by revisiting the concept of the primary time point and the look-back window of equation 21. The time points in the look-back window precede the primary time point, meaning that they lie in the past relative to the primary time. One can also define a look-ahead window, constructed of time points that succeed the primary time. The time points in a look-ahead window are in the future relative to the primary time. Consider an ordered sequence of time points composed of a given number ($\mathbf{n_{lb}}$) of time points that precede the primary time point and a given number ($\mathbf{n_{la}}$) of time points that succeed the primary time point: $\mathbf{(t_p - n_{lb}\Delta t,\ t_p - (n_{lb}\text{-}1)\Delta t,\ \ldots,\ t_p - 2\Delta t,\ t_p - \Delta t,\ t_p,\ t_p + \Delta t,\ t_p + 2\Delta t,\ldots,\ t_p + (n_{la}\text{-}1)\Delta t,\ t_p + n_{la}\Delta t)}$. The sequence of time points defines a pattern array that contains both look-back and look-ahead data,

$$\vec{X}(t_p) = \begin{bmatrix} x(t_p - n_{1b}\Delta t), x(t_p - (n_{1b} - 1)\Delta t), \cdots x(t_p - 2\Delta t), x(t_p - \Delta t), x(t_p), \\ x(t_p + \Delta t), x(t_p + 2\Delta t), \cdots x(t_p + (n_{la} - 1)\Delta t), x(t_p + n_{la}\Delta t) \end{bmatrix} \quad (29)$$

[0082] Referring to FIG. 9, an extension to the inferential form of SBM (equation 9) that supports extrapolation into the time dimension is produced if the three-dimensional collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ is created with sequential pattern matrices **a** to **g** that contain both look-back and look-ahead data. Since the input pattern array $\mathbf{X_{in}}$ contains data only from the current time point and preceding time points (data from future time points do not exist yet), the collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ is an aggregate matrix composed of two sub-matrices separated along the time dimension. The first of these sub-matrices $\mathbf{D_{lb}(t)}$ contains the data from the various primary time points and from the look-back time points. The second sub-matrix $\mathbf{D_{la}(t)}$ contains the data from the look-ahead time points. Equation 9 is shown with an input pattern array $\mathbf{X_{in}}$ of five input sensors and a look-back window of three time intervals between the time points $\mathbf{t_p}$ to -3. The look-back portion or sub-matrix $\mathbf{D_{lb}(t)}$ is a three-dimensional collection of learned sequential pattern matrices that contains data from five input sensors (1-5), seven primary time points each on its own sequential pattern matrix **a** to **g**, and four look-back time points or reference vectors $\mathbf{t_p}$ to -3 on each sequential pattern matrix **a** to **g**. The look-ahead portion or sub-matrix $\mathbf{D_{la}(t)}$ is a three-dimensional collection of learned sequential pattern matrices that contains data from five input sensors (1-5), seven learned sequential pattern matrices **a** to **g** each with its own primary time point, and two future or succeeding time points or vectors **+1** and **+2**. The resulting weight vector, generated by the operations within the two sets of brackets, is multiplied by the look-ahead collection of learned sequential pattern matrices $\mathbf{D_{la}(t)}$ to create an estimate matrix $\mathbf{Y_{la}}$ that extrapolates in time. In this example, two extrapolated estimate vectors **+1** and **+2** are calculated for estimate matrix $\mathbf{Y_{la}}$, representing the time points that are one and two time steps $\mathbf{\Delta t}$ into the future. As described above with the vector-based equation (FIG. 5), the weight vector can also be multiplied by the full collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ that includes both $\mathbf{D_{la}(t)}$ and $\mathbf{D_{lb}(t)}$ to generate estimate matrices $\mathbf{X_{1b}}$ and $\mathbf{Y_{la}}$ within an estimate matrix $\mathbf{XY_{e1}}$ that contains estimate data for past, current, and future time points (depicted in FIG. 10).

[0083] Comparing the illustrations in FIGS. 9 and 10 to those in FIGS. 7 and 8, the matrix calculations within the brackets of all four figures are identical. This means that the

calculation of the weight vector for an inferential model that extrapolates in the time dimension is identical to that for an inferential model that extrapolates along the dimension of the modeled sensors. The two forms of inferential modeling differ only by the data that are included in the full collection of learned sequential pattern matrices. A model that includes data for time points that are in the future relative to the primary time points extrapolates into the future. A model that includes data for sensors that are not in the input data stream extrapolates into these sensors. Referring to FIG. 11, an inferential model that extrapolates into both the time and modeled sensor dimensions is shown. Its three-dimensional collection of learned sequential pattern matrices $\mathbf{D_a(t)}$ is an aggregate matrix composed of four sub-matrices separated along the modeled sensor and time dimensions. Its sub-matrices contain data for the look-back window of the input sensors $\mathbf{D_{lb}(t)}$, data for the look-ahead window of the input sensors $\mathbf{D_{la}(t)}$, data for the look-back window of the output (inferred) sensors $\mathbf{D_{lbout}(t)}$, and data for the look-ahead window of the output (inferred) sensors $\mathbf{D_{laout}(t)}$. The calculations generate estimate matrices $\mathbf{X_{1b}}$ and $\mathbf{Y_{la}}$ within an estimate matrix $\mathbf{XY_{e2}}$ that contains estimate data for past, current, and future time points (depicted in FIG. 10) for both input and output (inferred) sensors.

[0084] Each of the various forms of kernel regression modeling with sequential pattern arrays described above produces an estimate matrix of model estimate data. In one example, estimate matrix $\mathbf{X_{est}}$ is formed for each input pattern array $\mathbf{X_{new}}$ (FIG. 6). As understood from the examples described above, in addition to the estimate vector corresponding to the current time point, the estimate matrix contains vectors for each of the time points in the look-back and/or look-ahead windows. The number of sequential vectors in the estimate matrix depends on the form of the modeling equation (autoassociative or inferential) and the number of time points $\mathbf{n_{lb}}$ in the look-back window and the number of time points $\mathbf{n_{la}}$ in the look-ahead window. As system time progresses, each fixed time point along the timeline accumulates multiple estimate vectors as the input pattern array reaches, moves through, and past the time point. The total number of estimate vectors that will be calculated for a fixed moment in time equals the total number of sequential patterns (vectors) in the sequential pattern matrix and analyzed by the model. For an autoassociative model or an inferential model that extrapolates along the sensor

dimension, this total number is given by $\mathbf{n_{lb}} + 1$, corresponding to an estimate vector for each pattern in the look-back window and an estimate vector for the primary (current) time point. For an inferential model that extrapolates along the time dimension, this total number is given by $\mathbf{n_{lb}} + 1 + \mathbf{n_{la}}$, corresponding to an estimate vector for each pattern in the look-back and look-ahead windows and an estimate vector for the primary (current) time point.

[0085] Because multiple estimate vectors are calculated for a fixed point in time, utilizing sequential kernel regression models to feed algorithms for condition monitoring or diagnostics is complicated by the fact that many of these algorithms expect that only a single estimate vector exists for a time point. The simplest means of dealing with the multiple estimate vectors is to simply designate less than all of the multiple vectors in the estimate matrix as the source of the model estimates and to ignore any others. In one form, only one of the estimate vectors from each estimate matrix is selected for further diagnostic analysis. Typically, this means that the estimate vector in the estimate matrix selected for a fixed, arbitrary point in time $\mathbf{t_i}$ while looking across multiple estimate matrices is the one generated when that time point becomes the current time point ($\mathbf{t_i} = \mathbf{t_{cur}}$) or in other words, the most recent time point ($\mathbf{t_p}$ in the example estimate matrices of FIGS. 6 to 8). As the input pattern window moves past $\mathbf{t_i}$, and $\mathbf{t_i}$ becomes part of the look-back window to the new current time point, new estimate data calculated for $\mathbf{t_i}$ are ignored. In other words, the older or preceding vectors relative to the current vector $\mathbf{t_p}$ in the estimate matrix are ignored.

[0086] Other, more complex methods can be used to produce or select a single estimate vector for each fixed time point across multiple estimate matrices, while taking advantage of the information in the multiple vectors. Such methods include, but are not limited to, an average; weighted average; other weighted norms (2-norm and p-norm); maximum, minimum or median value, and so forth. The estimate vector chosen for diagnostic analysis could also be the vector with the greatest similarity to its corresponding input vector, and may use a similar similarity equation as that used to determine the weight vector. It will also be understood these methods can be applied to provide a single estimate vector for each estimate matrix to represent multiple sequential time points

within the estimate matrix rather than a single fixed time point across multiple estimate matrices.

[0087] For an inferential model that extrapolates in the time dimension, a prognostic module 34 (FIG. 1) can use the future estimate matrix $\mathbf{X_{la}}$ to feed prognostics algorithms, such as calculations of the remaining useful life of an asset (or to state it another way, to determine the future condition or operational state of the object being monitored). This is based on the fact that the sequence of extrapolated estimates of a modeled sensor is a trend-line that predicts the future behavior of the modeled sensor. As system time progresses and new input pattern arrays are formed containing new primary vectors, new future estimate matrices are calculated. Like the other kernel regression models described above, the new estimate matrices substantially overlap previous matrices, meaning that multiple estimate values are produced for each sensor at each time point.

[0088] Also similar to the other kernel regression models, the inferential time extrapolating model can use various methods devised to reduce the multiple estimate values that are calculated at a fixed time point to a single value suitable for trending of the sensor. The simplest method is to select the most-recently calculated estimate matrix to supply the estimate data at each of the time points in the look-ahead window. Specifically, for a fixed time point $\mathbf{t_i}$ well into the future, an estimate vector will be generated for it when the look-ahead pattern window first reaches it: $\mathbf{t_i = t_{cur} + n_{la}*\Delta t}$. At each succeeding time step as the look-ahead window passes through the fixed point, a new estimate vector is calculated for it, which replaces the last vector. Thus, all of the estimate vectors are used to build a trend line, and the results for each time point (or fixed point) represented by estimate vectors are constantly being updated by the more recent estimate values to correspond to vectors as they past through the look-ahead window used to build the estimate matrices.

[0089] Besides being simple, this approach produces sensor trends that react quickly to dynamic changes since only the most-recently calculated estimate matrix is used. Since estimate data in the trend-lines are replaced for each succeeding time step, the trends are susceptible to random fluctuations. This means that the trend value at a fixed time point

can vary dramatically between successive time steps. Other more complex methods, such as average, weighted average, or other weighted norms, utilize two or more, or all, of the estimate values calculated at a fixed time point across multiple estimate matrices to produce a single estimate value for it. Trend lines produced by these methods are smoother, but less responsive to rapid dynamic changes. In addition to the above methods, which are designed to produce trend-lines representative of expected system behavior, other trend-lines can be produced that indicate the range of possible behaviors. For instance, a trend-line that connects the maximum estimate values at each future time point coupled with a trend-line connecting the minimum estimate values, bound the results produced by the model.

[0090] Returning again to FIG. 1, the full estimate matrix $\mathbf{X}_{est}$ or a single representative estimate vector, as described above, is passed to differencing engine 20. The differencing engine subtracts the estimate matrix from the input pattern array ($\mathbf{X}_{in}$ or $\mathbf{X}_{new}$) or it subtracts the representative estimate vector from the current time point's input vector. Specifically, each selected estimate value from the estimate matrix is subtracted from a corresponding input value from the input pattern array. This array of residual vectors or a single representative residual vector is then provided to the alert module 22. Alert module 22 applies statistical tests to the residual data to determine whether the estimate and input data are statistically different. The alert module 22 performs any of a variety of tests to make the fault determination. This may include a rules engine for assessing rules logic using one or more residual values. The rules can be of any of a variety of commonly used rules, from simple univariate threshold measures, to multivariate and/or time series logic. Furthermore, the output of some rules may be the input to other rules, as for example when a simple threshold rule feeds into a windowed alert counting rule (e.g., x threshold alerts in y observations). Furthermore, statistical techniques may be used on the residual data to derive other measures and signals, which themselves can be input to the rules. Applicable statistical analyses can be selected from a wide variety of techniques known in the art, including but not limited to moving window statistics (means, medians, standard deviations, maximum, minimum, skewness, kurtosis, etc.),

statistical hypothesis tests (for example, Sequential Probability Ratio Test (SPRT)), trending, and statistical process control (for example, CUSUM, S-chart).

[0091] The alert module 22 may determine that any differences between the estimate and input data is due to the normal operating conditions that were not encountered during training. In this case, sensor data indicative of the new operating conditions are provided to the optional adaptation module 30, which incorporates that data into the learning of model 14 via library 18, for example. In addition, adaptation module 30 may optionally perform its own automated tests on the data and/or residual analysis results to determine which input vectors or input arrays should be used to update the model 14.

[0092] The process of adapting a model comprises adding sensor data indicative of the new operating conditions to the set of reference data in the library **H** from which the original kernel-based model was "trained". In the simplest embodiment, all reference data are used as the model exemplars, and therefore adapting a model means adding the new sensor data to the exemplar set of the model. Since sequential kernel regression models operate on sequences of observation vectors by design, new operating data added to the reference data must consist of a sequence of observation vectors. The minimum number of vectors added during any adaptation event equals the total number of sequential patterns (vectors) analyzed by the model. As described above, this total number is given either by $\mathbf{n_{Ib}} + 1$ for an autoassociative model or an inferential model that extrapolates along the sensor dimension, or by $\mathbf{n_{Ib}} + 1 + \mathbf{n_{Ia}}$ for an inferential model that extrapolates along the time dimension. If a training method has been used to down-select the reference observations to a subset stored as "representative" of system dynamics as described above for forming the three-dimensional collection of learned sequential pattern matrices **D(t)**, then the new sequence of observation vectors (or in other words the entire input pattern array) is added to the original reference dataset, and the down-selection technique is applied to derive a new representative exemplar set, which should then include representation of the new observations. It is also possible to merely add the new sequence to a down-selected set of learned pattern arrays, without rerunning the down-selection technique. Furthermore, in that case, it may be useful to remove some learned pattern arrays from the model so that they are effectively replaced by the new

data, and the model is kept at a manageable size. The criteria for which old learned pattern arrays are removed can include clustering and similarity determinations using equations described above which compare the observations at the new primary time points to the observations at old primary time points and replace those sequential pattern arrays most like the new sequential pattern arrays.

[0093] To this point, the invention describes sequential kernel regression models that are trained with representative data from periods of normal operation. It has been shown that such models can be used to detect and diagnosis system faults. In addition, the time-inferential form of the invention produces models that can extrapolate system behavior into the future. But since the models are trained only with normal operating data, their utility as a fault progresses is limited as the system behavior departs further and further from normality.

[0094] To improve diagnostics and prognostics during developing faults, separate sequential kernel regression models that are trained with data collected during fault conditions (or failure mode reference data) can be utilized. These fault models are activated only after there is an indication that a fault is developing in the system. The fault indication can be provided by sequential models trained with normal system data, or by numerous other means; including, but not limited to, vector-based kernel regression models (for example, SBM), neural networks, k-means clustering models, and rule-based fault detection models. The fault models are trained with full transient histories of known fault events for the asset being monitored. These fault events need not have occurred on the actual asset from an earlier period in time, they can come from fault events that have occurred on other machinery that are substantially equivalent to the asset being monitored. The fault histories consist of all system data collected from the time at which the fault was first indicated to the final end state of the event, such as system failure or system shutdown.

[0095] It will be appreciated by those skilled in the art that modifications to the foregoing embodiments may be made in various aspects. Other variations clearly would also work, and are within the scope and spirit of the invention. The present invention is set forth

with particularity in the appended claims. It is deemed that the spirit and scope of that invention encompasses such modifications and alterations to the embodiments herein as would be apparent to one of ordinary skill in the art and familiar with the teachings of the present application.

What is claimed is:

1. A monitoring system for determining the condition of an object, comprising:

an empirical model having reference data that indicates the normal operational state of the object, and input pattern arrays, each input pattern array having a plurality of time-ordered input vectors, each input vector having input values representing a plurality of parameters indicating the current condition of the object,

the empirical model being configured to generate estimate values based on a calculation that uses an input pattern array and the reference data to determine a similarity measure between the input values and the reference data; and

a differencing module comparing the estimate values to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

2. The system of claim 1, wherein at least one of the input vectors is in multiple input pattern arrays.

3. The system of claim 1, wherein each input pattern array represents a time period including a plurality of time points along a sequence of the input vectors representing the operational state of the object.

4. The system of claim 3, wherein each input pattern array represents a time period that is different than, but overlaps, the time period of at least one of the other input pattern arrays.

5. The system of claim 3, wherein each input pattern array has a current primary vector from the sequence of input vectors and representing the most recent time point within the input pattern array.

6. The system of claim 5 wherein the input vectors are selected for each input pattern array by moving a look-back window along the sequence, selecting the current primary

vector, and selecting one or more older vectors that follow the primary vector along the sequence and fit within the window.

7. The system according to claim 6 wherein the older vectors are consecutive along the sequence.

8. The system according to claim 6 wherein the older vectors are not consecutive along the sequence.

9. The system according to claim 1 wherein time intervals between the time points represented by the input vectors within an input pattern array are uniform.

10. The system according to claim 1 wherein time intervals between the time points represent by the input vectors within an input pattern array are unequal.

11. The system according to claim 1 wherein time intervals between time points represented by the input vectors within an input pattern array are shorter between input vectors that are more recent than the time intervals between later input vectors within the same input pattern array.

12. The system according to claim 1 wherein the total duration represented by the input pattern arrays varies.

13. The system according to claim 1 wherein the calculation is based on an equation that comprises:

$$S(\vec{\vec{X}}_{new}, \vec{\vec{X}}_i) = \frac{1}{1 + \frac{1}{\rho}\left[\frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \left(\frac{\sum_{k=1}^{n_{lb}} \left(\frac{W_k \left| \vec{\vec{X}}_{new;j,k} - \vec{\vec{X}}_{i;j,k} \right|}{range_j} \right)}{\sum_{k=1}^{n_{lb}} W_k} \right)\right]^{\lambda}}$$

14. The system according to claim 1 wherein the calculation is based on an equation that comprises:

$$S(\vec{\vec{X}}_{new}, \vec{\vec{X}}_i) = \frac{1}{n_{sens}} \sum_{j=1}^{n_{sens}} \left[\frac{1}{1 + \frac{1}{\rho}\left(\frac{\sum_{k=1}^{n_{lb}} \left(\frac{W_k \left| \vec{\vec{X}}_{new;j,k} - \vec{\vec{X}}_{i;j,k} \right|}{range_j} \right)}{\sum_{k=1}^{n_{lb}} W_k} \right)^{\lambda}} \right]$$

15. The system of claim 1 wherein the similarity measure is a value on a bounded range that indicates an input value and reference value are identical at one end of the range and have no similarity at another end of the range.

16. The system of claim 1, wherein the reference data comprises at least one learned sequential pattern matrix, each learned sequential pattern matrix having a plurality of reference vectors, each reference vector having reference values representing a plurality of parameters.

17. The system of claim 16 wherein the reference data forms a three-dimensional collection of the learned sequential pattern matrices.

18. The system of claim 17 wherein the three-dimensional collection is defined by three dimensions representing the parameters along a first dimension, the learned sequential pattern matrices along a second dimension, and time represented by the reference vectors in a third dimension.

19. The system of claim 17 wherein each input pattern array is compared to each learned sequential pattern matrix in the three-dimensional collection to determine a similarity measure for each learned sequential pattern matrix.

20. The system of claim 17 herein each learned sequential pattern matrix represents a time period including time points represented by the reference vectors, wherein the time period is different than, but overlaps, the time period of at least some of the other learned sequential pattern matrices in the three-dimensional collection.

21. The system of claim 17 wherein the learned sequential pattern matrices have the same number of vectors as the input pattern arrays.

22. The system of claim 17 wherein the learned sequential pattern matrices represent the same time points as the input pattern arrays.

23. The system of claim 17 wherein the similarity between one of the learned sequential pattern matrices and one of the input pattern arrays is based on the similarity between corresponding reference and input values that represent the same vector position relative to the position of the other vectors represented within their respective array.

24. The system of claim 17 wherein the similarity between one of the learned sequential pattern matrices and one of the input pattern arrays comprises calculating an average scalar similarity value for each parameter represented by the arrays.

25. The system of claim 17 wherein at least some of the input pattern arrays represent a time period that is different than the time period represented by at least one of the learned sequential pattern matrices.

26. The system according to claim 17 wherein the time period represented by the learned sequential pattern matrices varies.

27. The system according to claim 1 wherein the empirical model is configured to generate the estimate values in the form of an estimate matrix having a plurality of estimate vectors.

28. The system of claim 27 wherein the estimate vectors in each estimate matrix corresponds to the time points represented by the input vectors in the input pattern arrays.

29. The system of claim 27 wherein less than all of the estimate values within the estimate matrices are used to determine the condition of the object.

30. The system of claim 27 wherein the differencing module only uses the estimate values that represent the most recent time point within each estimate matrix to determine the condition of the object.

31. The system of claim 27 comprising providing a single estimate vector for each time point represented across multiple estimate matrices to determine the condition of the object.

32. The system of claim 31 wherein the single estimate vector is provided by using at least one of:

selecting the estimate vector with one or more estimate values that has a maximum, minimum, or median value relative to other estimate vectors at the time point across multiple estimate matrices,

calculating an average, a weighted average, or weighted norm of the estimate values at the time point across multiple estate matrices, and

selecting an estimate vector with the greatest similarity to a corresponding input vector relative to other estimate vectors at the time point across multiple estimate matrices.

33. The system of claim 27 wherein the empirical model provides an estimate vector to determine the condition of the object by using at least one of:

an estimate vector within each estimate matrix that has a maximum, minimum, or median estimate value,

a weighted average or weighted norm of the estimate values in the estimate matrix to form a single estimate for each matrix, and

the estimate vector within the estimate matrix with the greatest similarity to a respective corresponding input vector.

34. The system of claim 1 wherein the empirical model is configured to use the similarity measures as weights in a calculation with a three-dimensional collection of learned sequential pattern matrices that represents parameters that are not represented in the input pattern arrays.

35. The system of claim 34 wherein the reference data used to generate the similarity measures do not represent all of the parameters represented by the three-dimensional collection.

36. The system of claim 34 wherein the three-dimensional collection represents only parameters that are not represented by the input pattern arrays.

37. The system of claim 35 wherein the three-dimensional collection represents both parameters that are not represented by the input pattern arrays and parameters that are represented by the input pattern arrays.

38. The system of claim 1 comprising a localization module to redetermine which reference data is to be used to generate the estimate values for each input pattern array.

39. The system of claim 1 comprising an adaption module configured to add the input pattern arrays to the reference data when the estimate values indicate the corresponding input pattern array represents normal operation.

40. A monitoring system for determining the condition of an object, comprising:

an empirical model operated by at least one processor and comprising reference data that indicates the normal operational state of the object and in the form of a plurality of learned sequential pattern matrices, each learned sequential pattern matrix having a plurality of reference vectors, each reference vector having reference values representing a plurality of parameters, and

input data representing a plurality of parameters indicating the current condition of the object,

wherein the empirical model is configured to generate estimate values based on a calculation that uses the input data and the learned sequential pattern matrices to determine a similarity measure between the input data and reference values in the plurality of reference matrices; and

a differencing module comparing the estimate values to the corresponding input values so that resulting values from the comparison can be used to determine the condition of the object.

41. A monitoring system for determining the condition of an object, comprising:

an empirical model comprising reference data that indicates the normal operational state of the object, and

input data representing a plurality of parameters indicating the current condition of the object,

wherein the empirical model is configured to generate estimate values based on a calculation that uses both the input data and the reference data to determine similarity measures between the input data and the reference data, wherein the estimate values are generated in the form of an estimate matrix having a plurality of time-ordered estimate vectors, each estimate vector having estimate values representing multiple parameters; and

a differencing module comparing at least one estimate vector for each time period represented by the estimate matrices to the input data so that the resulting values from the comparison can be used to determine the condition of the object.

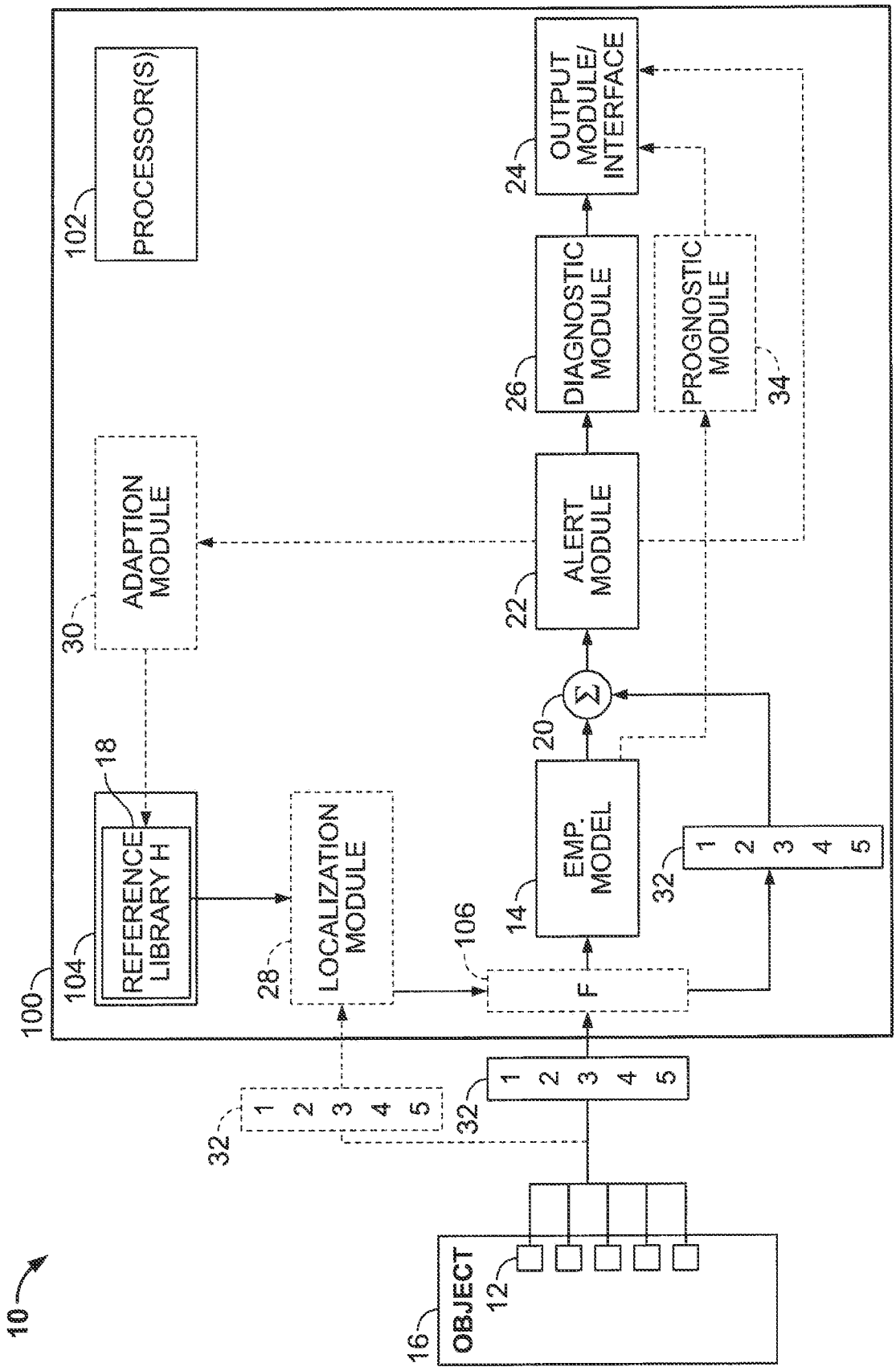
10
100
102
PROCESSOR(S)
104
REFERENCE LIBRARY H
18
30
ADAPTION MODULE
28
LOCALIZATION MODULE
106
F
14
EMP. MODEL
20
Σ
22
ALERT MODULE
26
DIAGNOSTIC MODULE
34
PROGNOSTIC MODULE
24
OUTPUT MODULE/ INTERFACE
32
1 2 3 4 5
16
OBJECT
12

FIG. 1

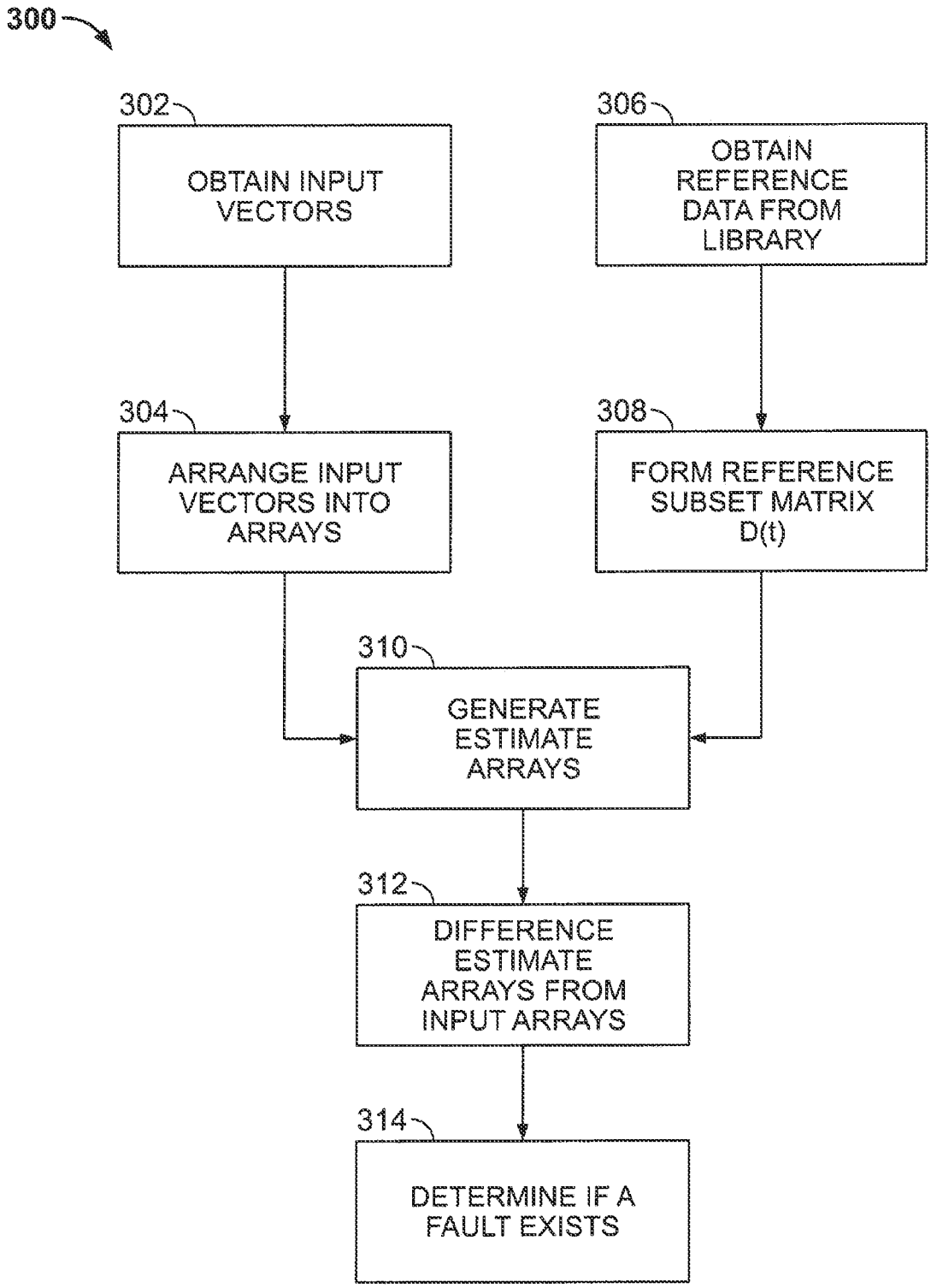
300
302
OBTAIN INPUT VECTORS
306
OBTAIN REFERENCE DATA FROM LIBRARY
304
ARRANGE INPUT VECTORS INTO ARRAYS
308
FORM REFERENCE SUBSET MATRIX D(t)
310
GENERATE ESTIMATE ARRAYS
312
DIFFERENCE ESTIMATE ARRAYS FROM INPUT ARRAYS
314
DETERMINE IF A FAULT EXISTS

FIG. 2

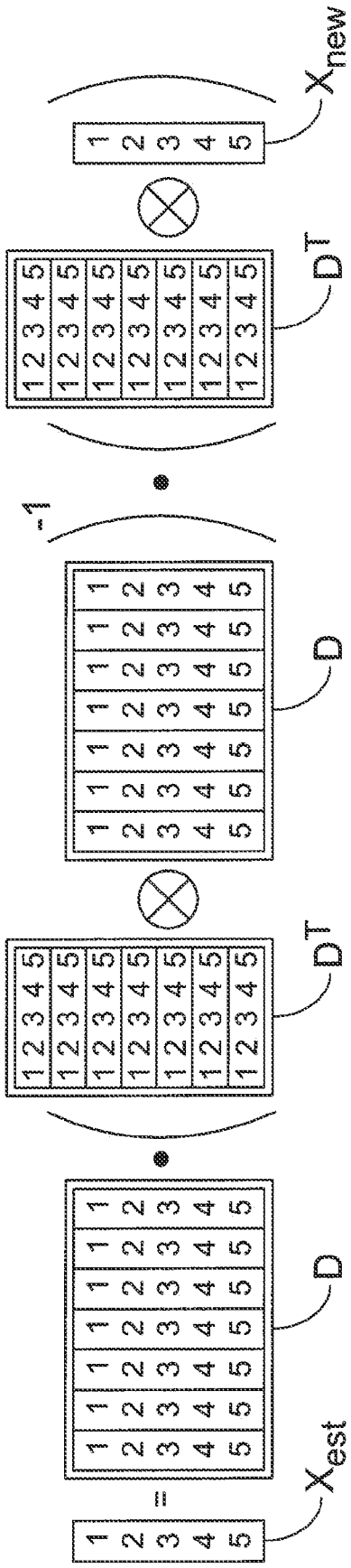

FIG. 3

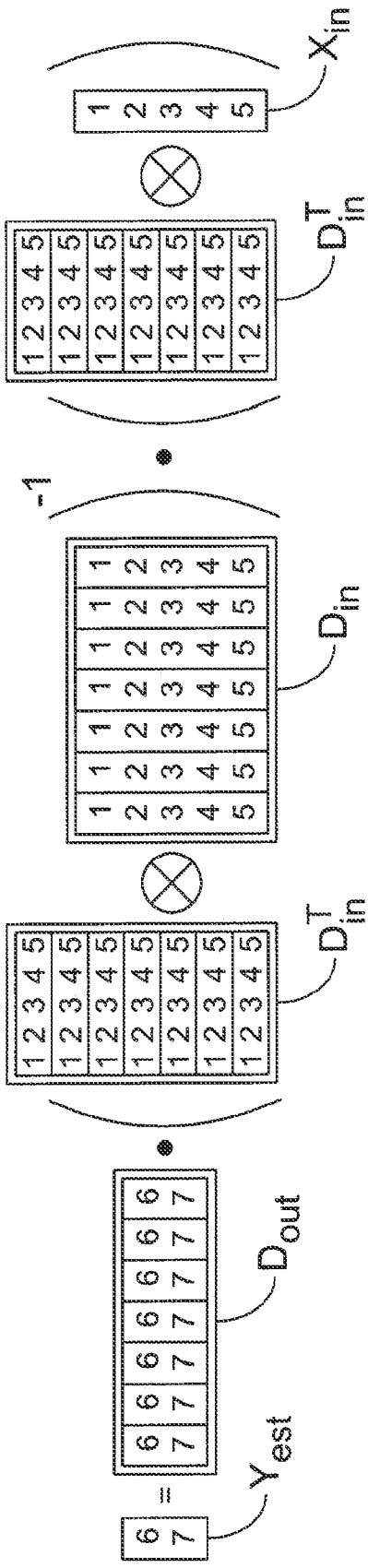

FIG. 4

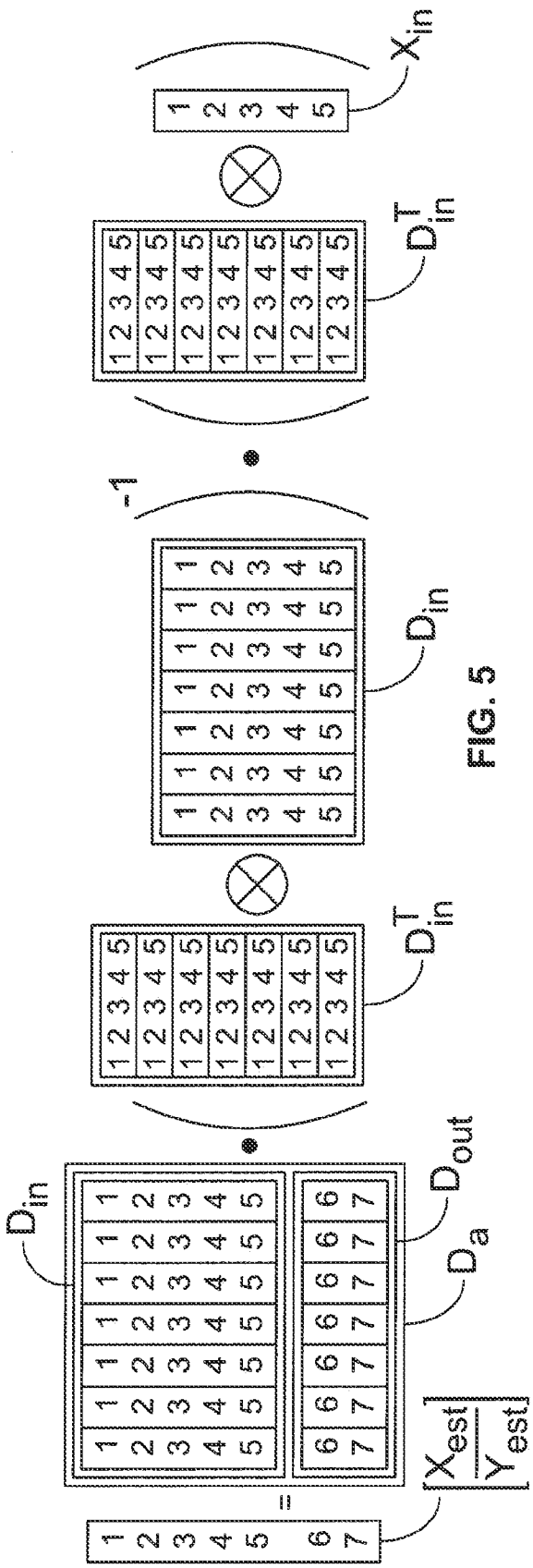
X_{in}
D^T_{in}
-1
D_{in}
D^T_{in}
D_{in}
D_{out}
D_a
X_{est}
Y_{est}

FIG. 5

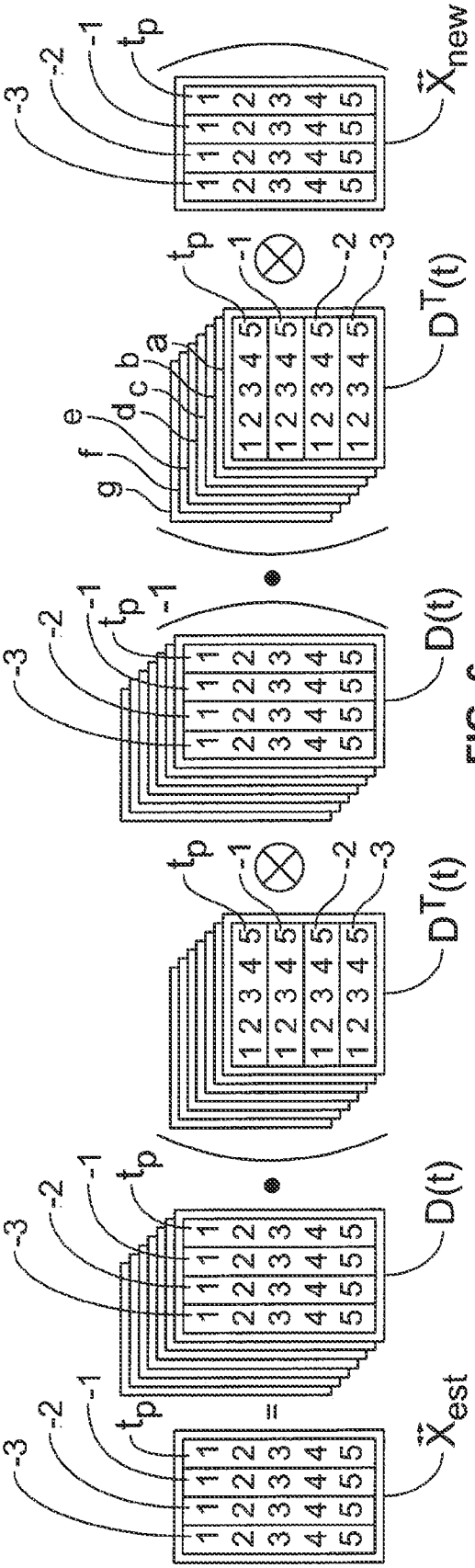
$\tilde{X}_{new}$
$D^T(t)$
$D(t)$
$D^T(t)$
$D(t)$
$\tilde{X}_{est}$

FIG. 6

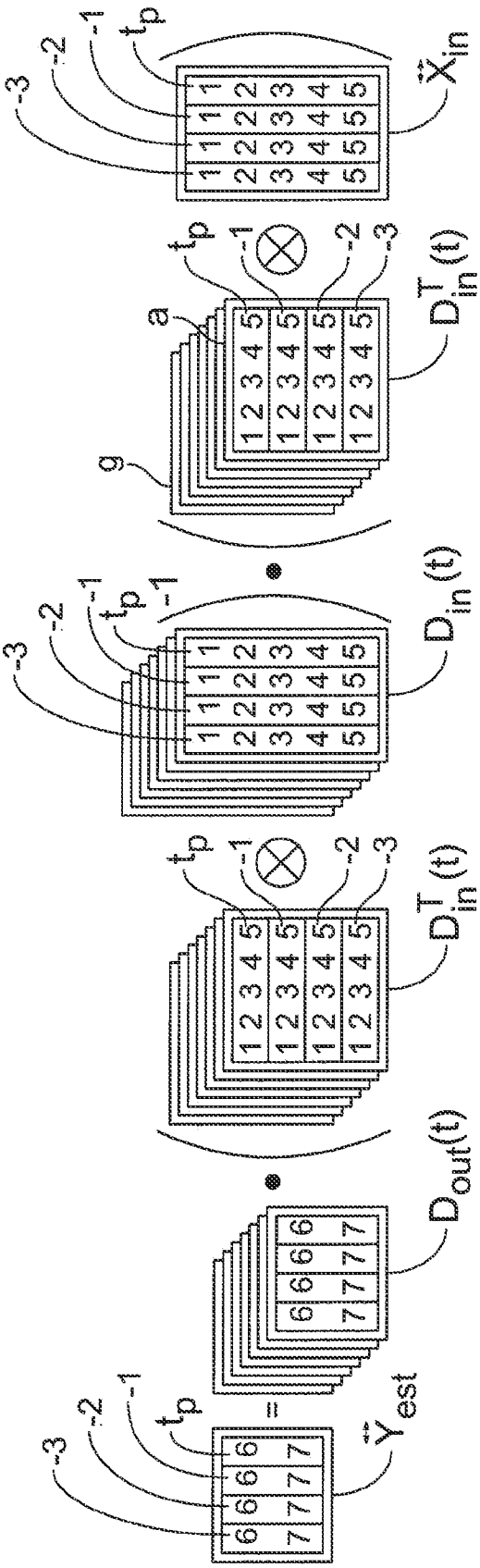
-3
-2
-1
t_p
$\vec{Y}_{est}$
=
$D_{out}(t)$
$D_{in}(t)$
$D_{in}^T(t)$
a
g
$\vec{X}_{in}$

FIG. 7

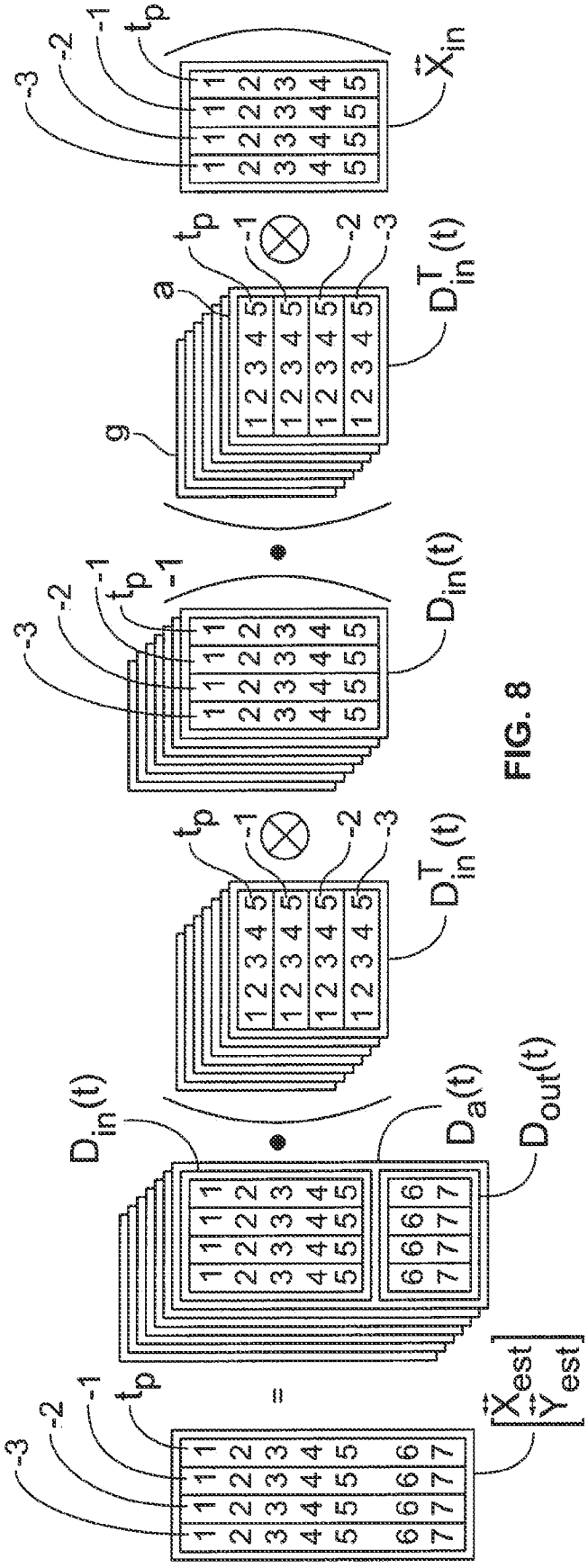
-3
-2
-1
t_p
$\vec{X}_{est}$
$\vec{Y}_{est}$
=
$D_{in}(t)$
$D_a(t)$
$D_{out}(t)$
$D_{in}^T(t)$
a
g
$\vec{X}_{in}$

FIG. 8

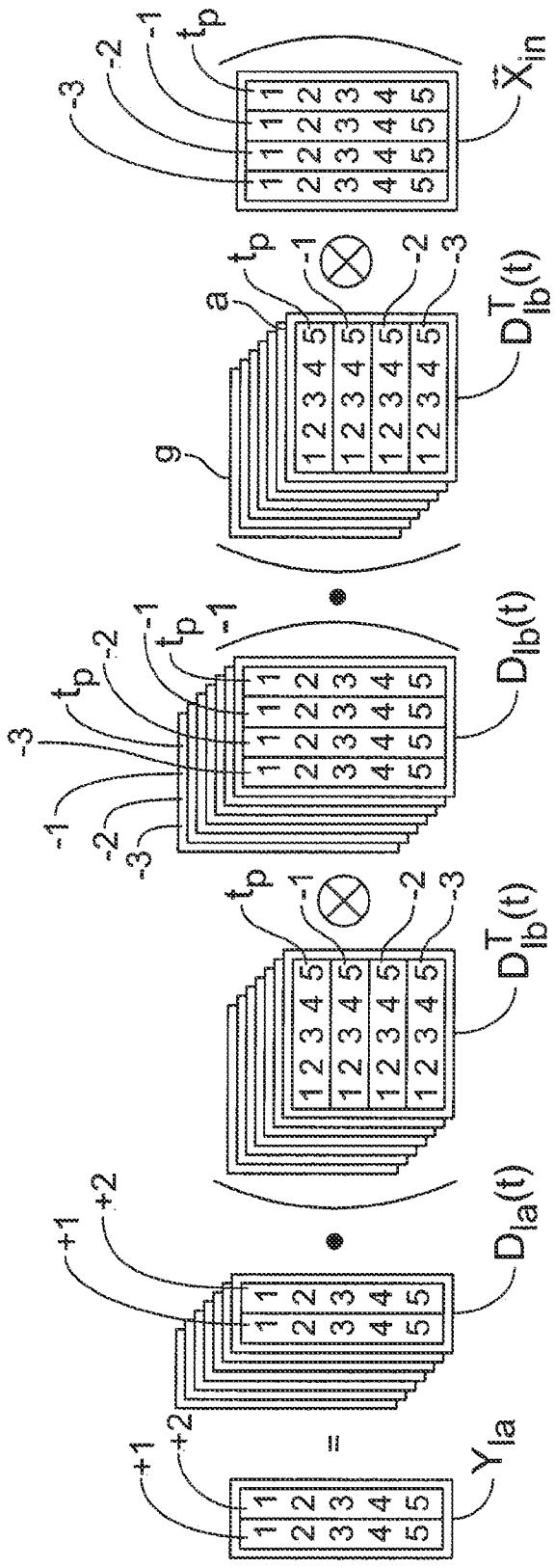

FIG. 9

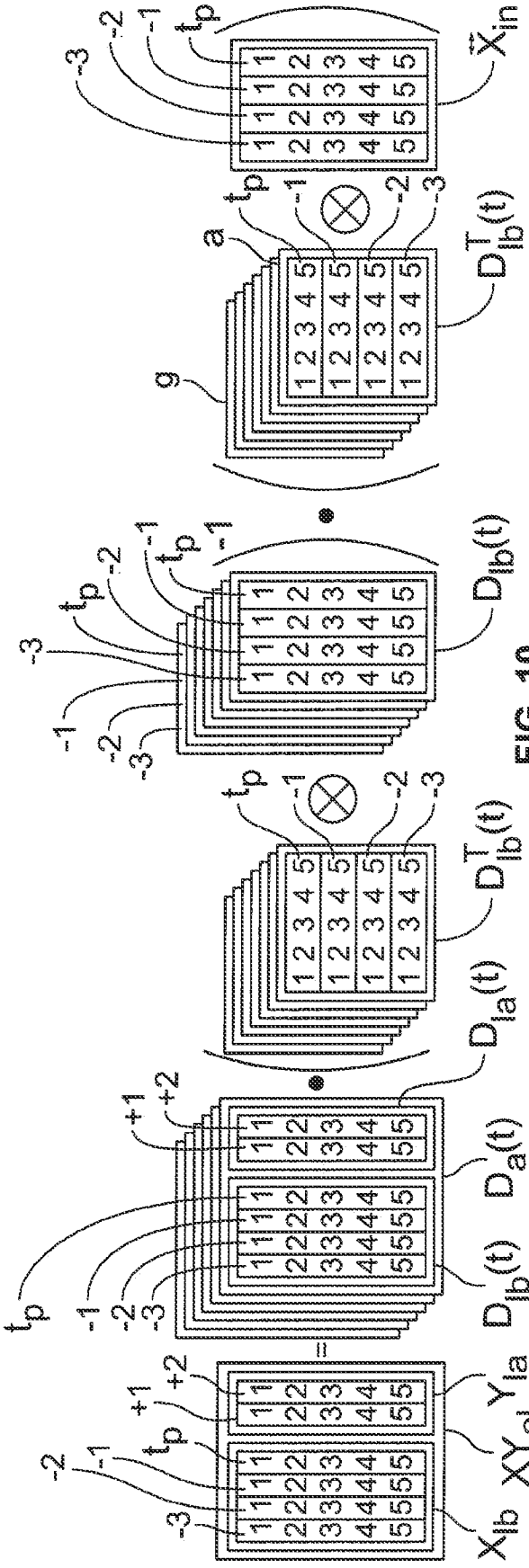

FIG. 10

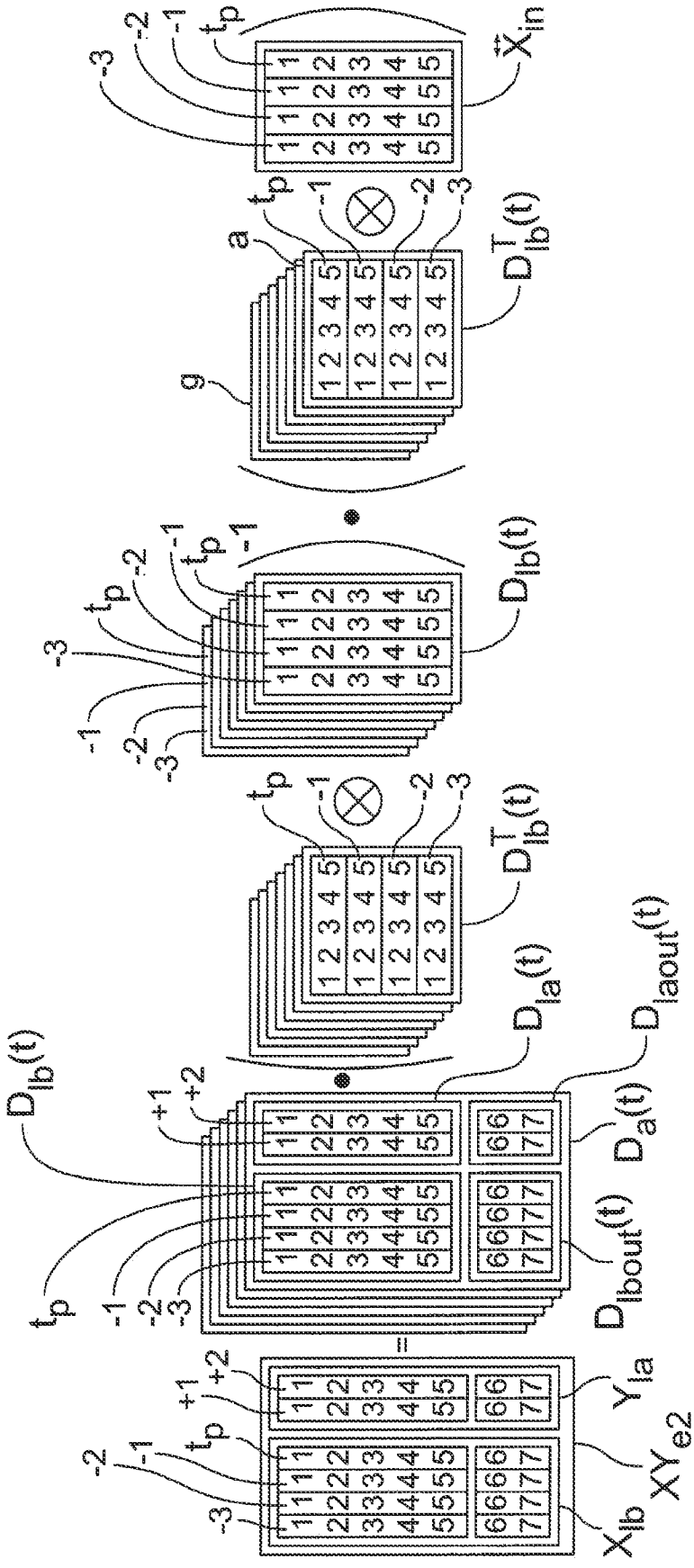
$\vec{X}_{in}$
$D^T_{lb}(t)$
$D_{lb}(t)$
$D_{la}(t)$
$D_{laout}(t)$
$D_a(t)$
$D_{lbout}(t)$
Y_{la}
XY_{e2}
X_{lb}
a
g
t_p

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/044614

A. CLASSIFICATION OF SUBJECT MATTER
INV. G05B23/02
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G05B

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 539 597 B2 (WEGERICH STEPHAN W [US] ET AL) 26 May 2009 (2009-05-26) column 1, line 16 - column 3, line 61 column 5, line 6 - column 14, line 38 column 17, line 11 - line 60 column 20, line 45 - column 22, line 15 column 23, line 60 - column 24, line 14 abstract; figures 2,3,4,5,9,18	1-41
X	US 6 952 662 B2 (WEGERICH STEPHAN W [US] ET AL) 4 October 2005 (2005-10-04) cited in the application column 1, line 13 - column 10, line 10 figures 4-7	1,13,40, 41

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 September 2012	19/09/2012

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Postemer, Patricia

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2012/044614

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 7 373 283 B2 (HERZOG JAMES P [US] ET AL) 13 May 2008 (2008-05-13) cited in the application the whole document	1,14,40, 41
X	US 2002/128731 A1 (WEGERICH STEPHAN W [US] ET AL) 12 September 2002 (2002-09-12)	40
A	paragraphs [0010] - [0038] abstract; figures 1-3, 9b	1,41
A	EP 2 333 629 A1 (HITACHI LTD [JP]) 15 June 2011 (2011-06-15) paragraphs [0001], [0004], [0012], [0017] - [0045], [0065], [0074] figures 1,2,31	1,41

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2012/044614

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7539597 B2	26-05-2009	NONE	
US 6952662 B2	04-10-2005	NONE	
US 7373283 B2	13-05-2008	NONE	
US 2002128731 A1	12-09-2002	NONE	
EP 2333629 A1	15-06-2011	CN 102112933 A	29-06-2011
		EP 2333629 A1	15-06-2011
		JP 2010092355 A	22-04-2010
		KR 20110026515 A	15-03-2011
		US 2011191076 A1	04-08-2011
		WO 2010041355 A1	15-04-2010