US 20200327264A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2020/0327264 A1**

Wang et al. (43) **Pub. Date: Oct. 15, 2020**

(54) **SYSTEMS AND METHODS FOR ENHANCED POWER SYSTEM MODEL CALIBRATION**

(71) Applicant: **General Electric Company**, Schenectady, NY (US)

(72) Inventors: **Honggang Wang**, Clifton Park, NY (US); **Kaveri Mahapatra**, State College, PA (US)

(21) Appl. No.: **16/698,058**

(22) Filed: **Nov. 27, 2019**

Related U.S. Application Data

(60) Provisional application No. 62/833,492, filed on Apr. 12, 2019.

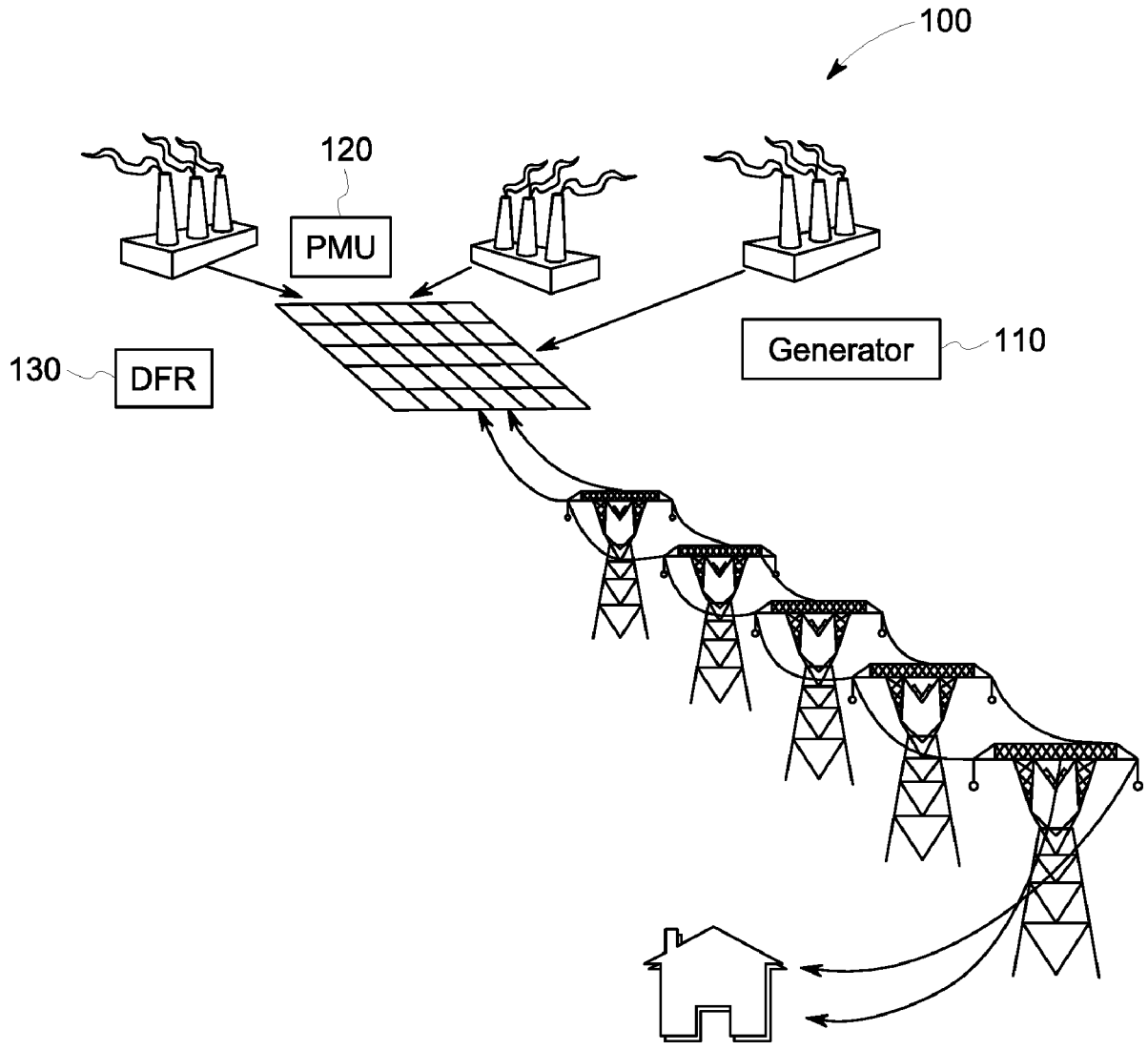
Publication Classification

(51) **Int. Cl.**
G06F 30/20 (2006.01)

(52) **U.S. Cl.**
CPC ***G06F 30/20*** (2020.01); *G06F 2111/10* (2020.01)

(57) **ABSTRACT**

A system for enhanced power system model calibration is provided. The system is programmed to store a model of a device. The model includes a plurality of parameters. The system is also programmed to receive a plurality of events associated with the device, receive a first set of input calibration values for the plurality of parameters, sequentially analyze the plurality of events in a first sequence to determine a set of calibrated parameter values for the model, validate the set of calibrated parameter values for the model to determine fit, and perform Bayesian optimization on the determined fit, the set of calibrated parameter values for the model, and the plurality of events.



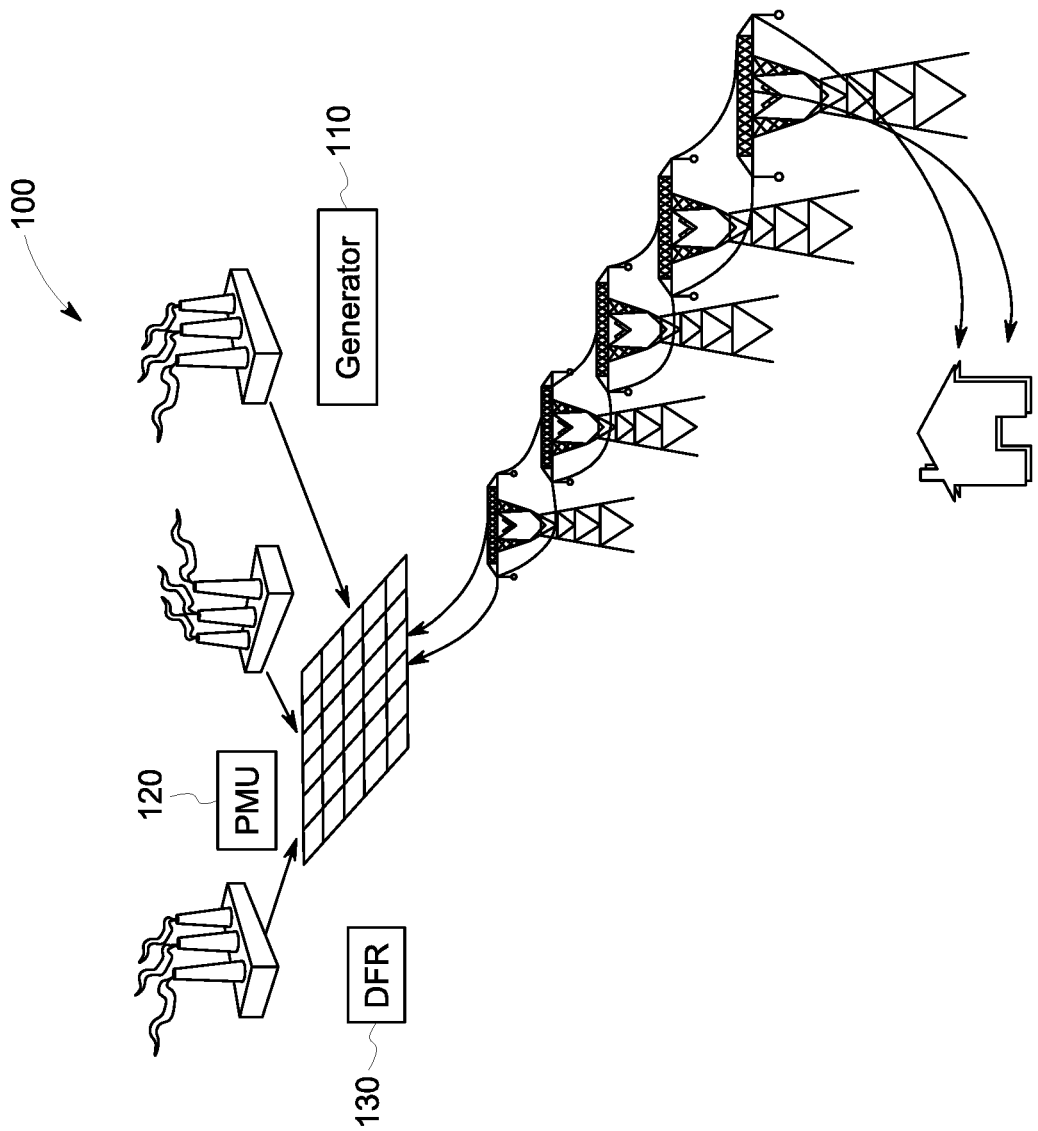


FIG. 1

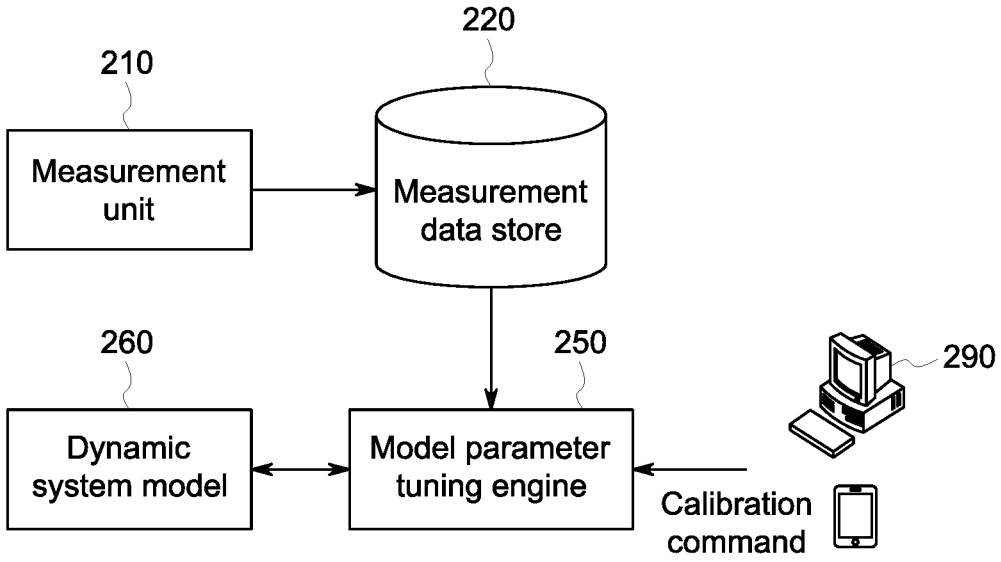

220
210
Measurement unit
Measurement data store
260
250
290
Dynamic system model
Model parameter tuning engine
Calibration command

FIG. 2

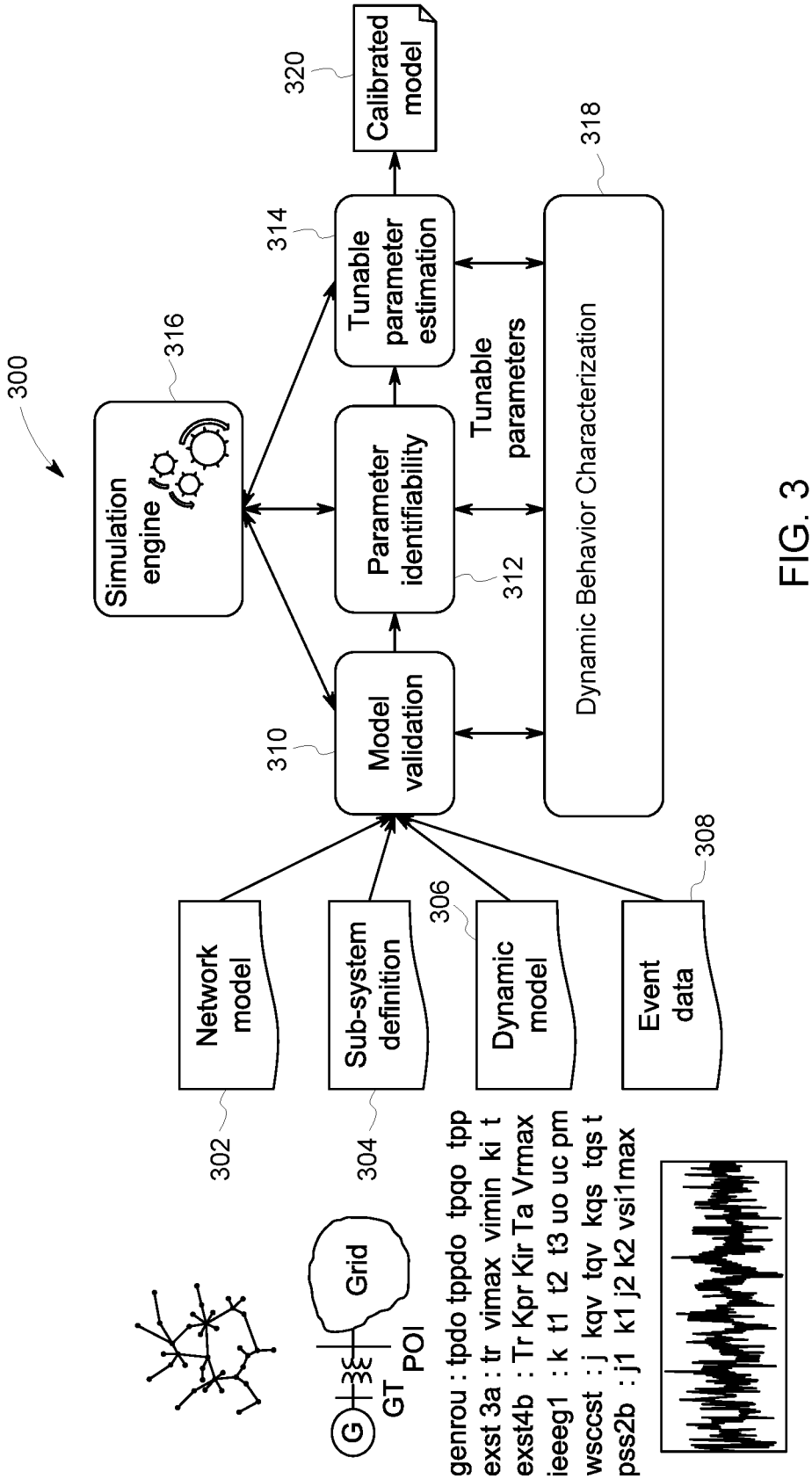

300
316
Simulation engine
310
Model validation
312
Parameter identifiability
314
Tunable parameter estimation
320
Calibrated model
318
Dynamic Behavior Characterization
Tunable parameters
302
Network model
304
Sub-system definition
306
Dynamic model
308
Event data
Grid
GT
POI
genrou : tpdo tppdo tpqo tpp
exst 3a : tr vimax vimin ki t
exst4b : Tr Kpr Kir Ta Vrmax
ieeeg1 : k t1 t2 t3 uo uc pm
wsccst : j kqv tqv kqs tqs t
pss2b : j1 k1 j2 k2 vsi1max

FIG. 3

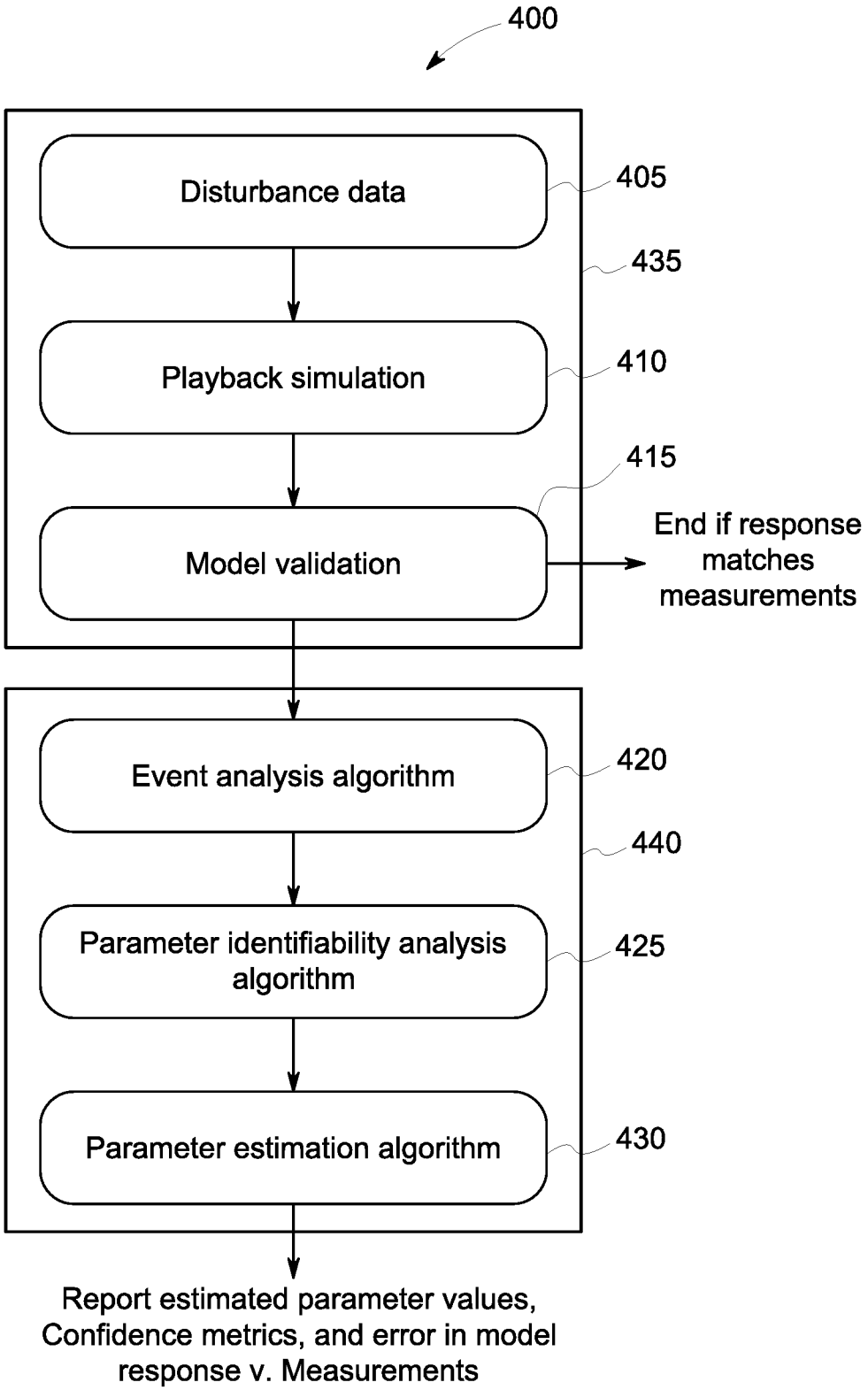

400
Disturbance data
405
435
Playback simulation
410
415
Model validation
End if response matches measurements
Event analysis algorithm
420
440
Parameter identifiability analysis algorithm
425
Parameter estimation algorithm
430
Report estimated parameter values, Confidence metrics, and error in model response v. Measurements

FIG. 4

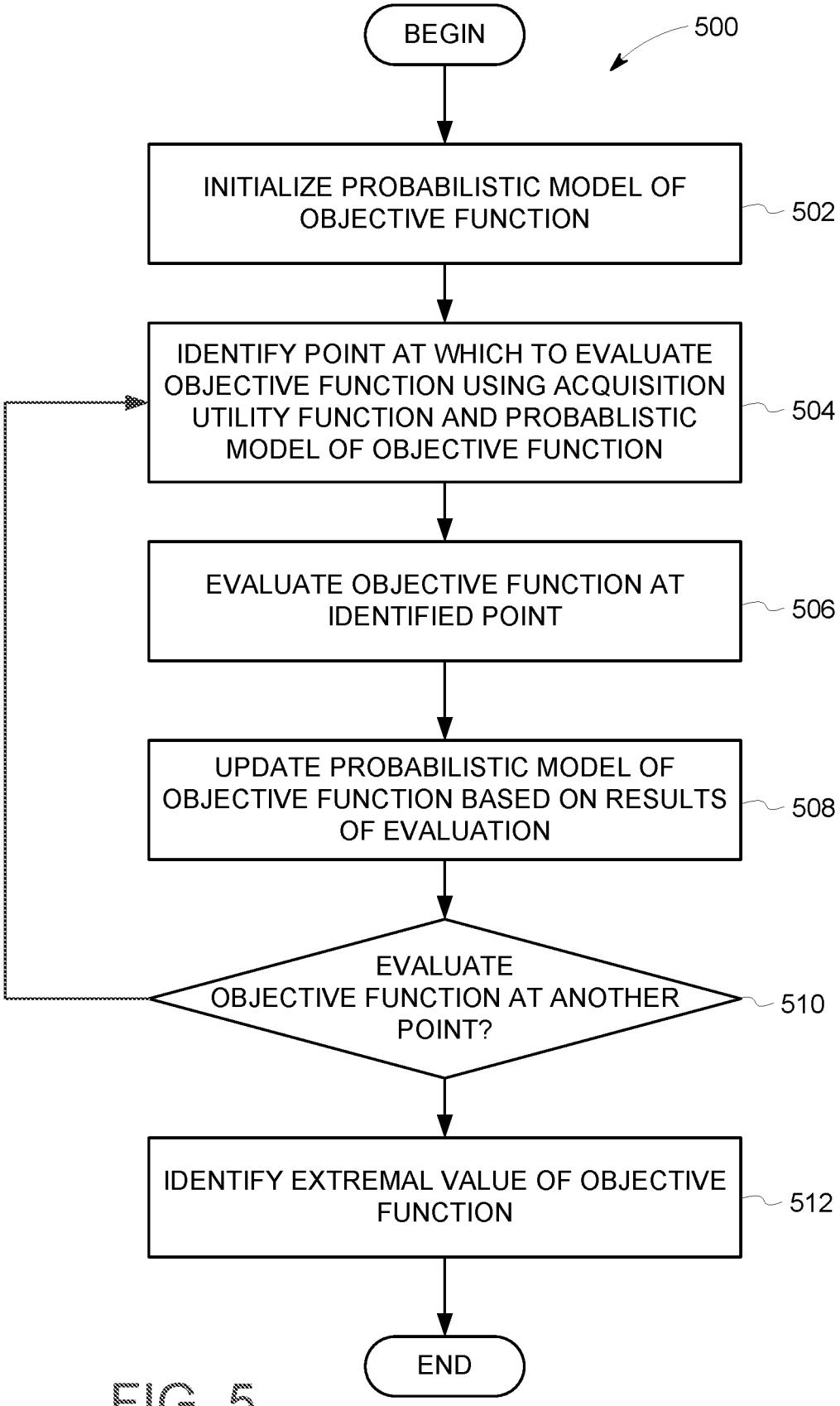

BEGIN
500
INITIALIZE PROBABILISTIC MODEL OF OBJECTIVE FUNCTION
502
IDENTIFY POINT AT WHICH TO EVALUATE OBJECTIVE FUNCTION USING ACQUISITION UTILITY FUNCTION AND PROBABLISTIC MODEL OF OBJECTIVE FUNCTION
504
EVALUATE OBJECTIVE FUNCTION AT IDENTIFIED POINT
506
UPDATE PROBABILISTIC MODEL OF OBJECTIVE FUNCTION BASED ON RESULTS OF EVALUATION
508
EVALUATE OBJECTIVE FUNCTION AT ANOTHER POINT?
510
IDENTIFY EXTREMAL VALUE OF OBJECTIVE FUNCTION
512
END

FIG. 5

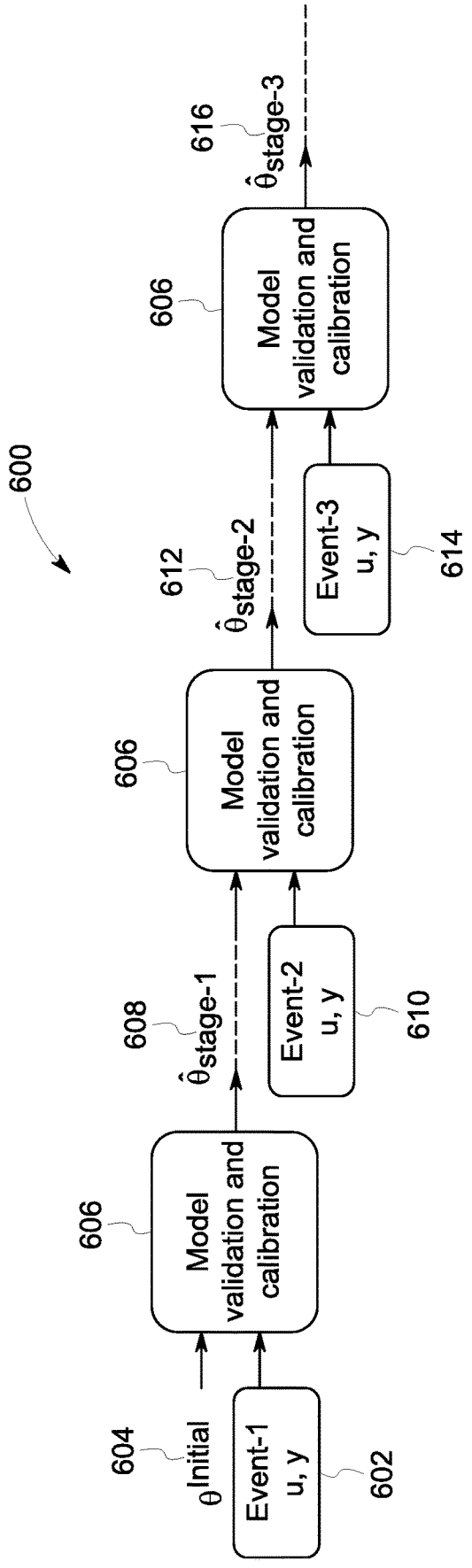

600
604
$\theta^{Initial}$
Event-1
u, y
602
606
Model validation and calibration
608
$\hat{\theta}_{stage-1}$
Event-2
u, y
610
606
Model validation and calibration
612
$\hat{\theta}_{stage-2}$
Event-3
u, y
614
606
Model validation and calibration
616
$\hat{\theta}_{stage-3}$

FIG. 6

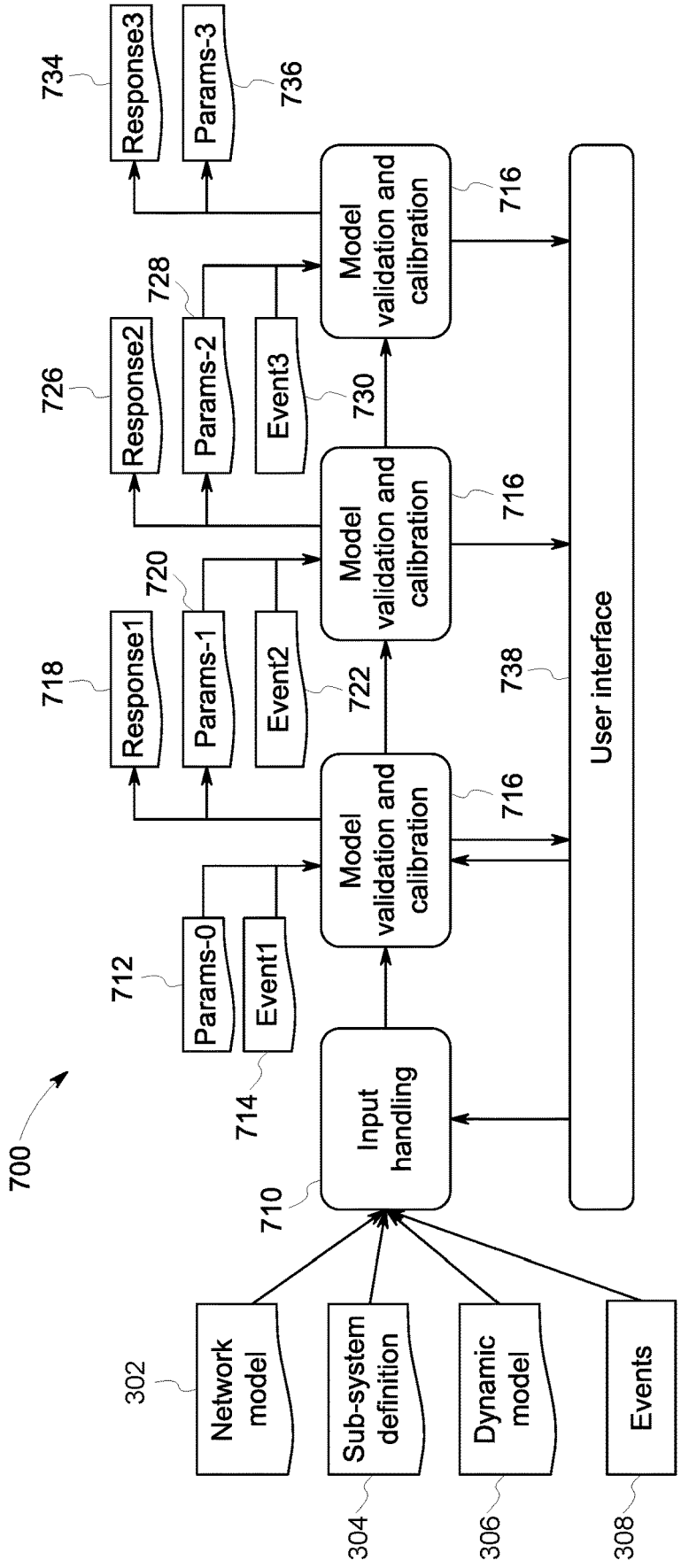

700
302
Network model
304
Sub-system definition
306
Dynamic model
308
Events
710
Input handling
712
Params-0
714
Event1
716
Model validation and calibration
718
Response1
720
Params-1
722
Event2
726
Response2
728
Params-2
730
Event3
734
Response3
736
Params-3
738
User interface

FIG. 7

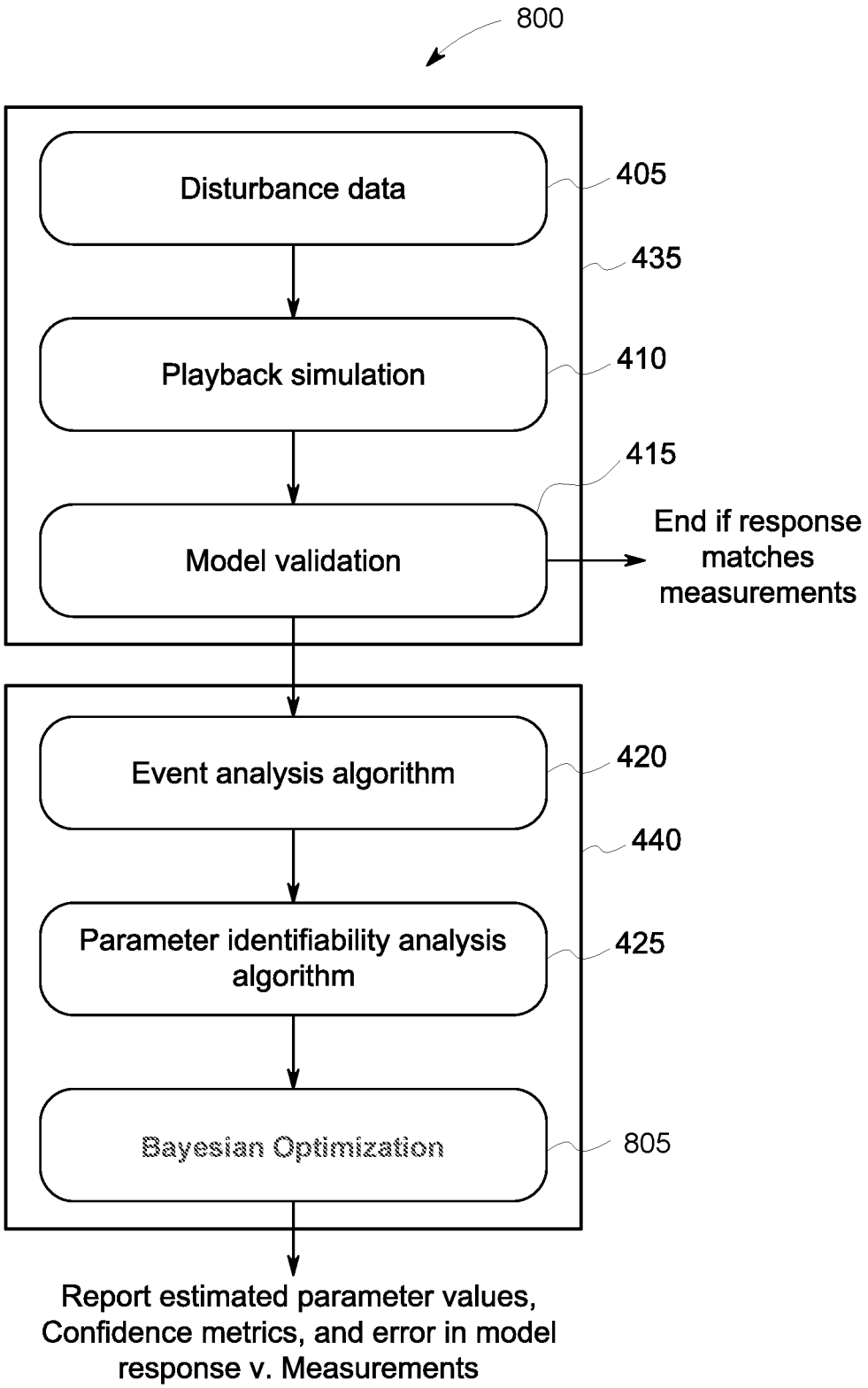

800
Disturbance data
405
435
Playback simulation
410
415
Model validation
End if response matches measurements
Event analysis algorithm
420
440
Parameter identifiability analysis algorithm
425
Bayesian Optimization
805
Report estimated parameter values, Confidence metrics, and error in model response v. Measurements

FIG. 8

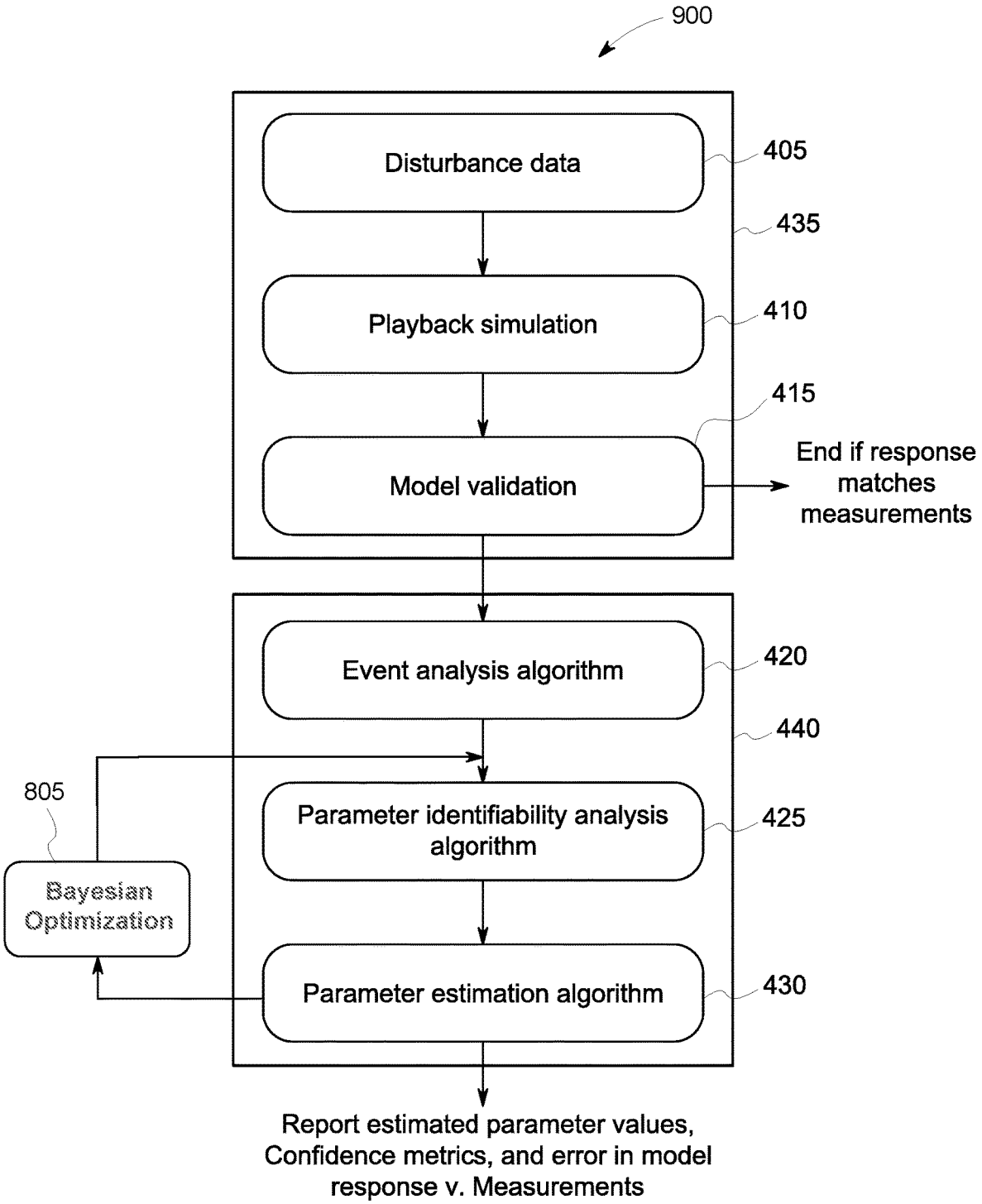

900
Disturbance data
405
435
Playback simulation
410
415
Model validation
End if response matches measurements
Event analysis algorithm
420
440
805
Parameter identifiability analysis algorithm
425
Bayesian Optimization
Parameter estimation algorithm
430
Report estimated parameter values, Confidence metrics, and error in model response v. Measurements

FIG. 9

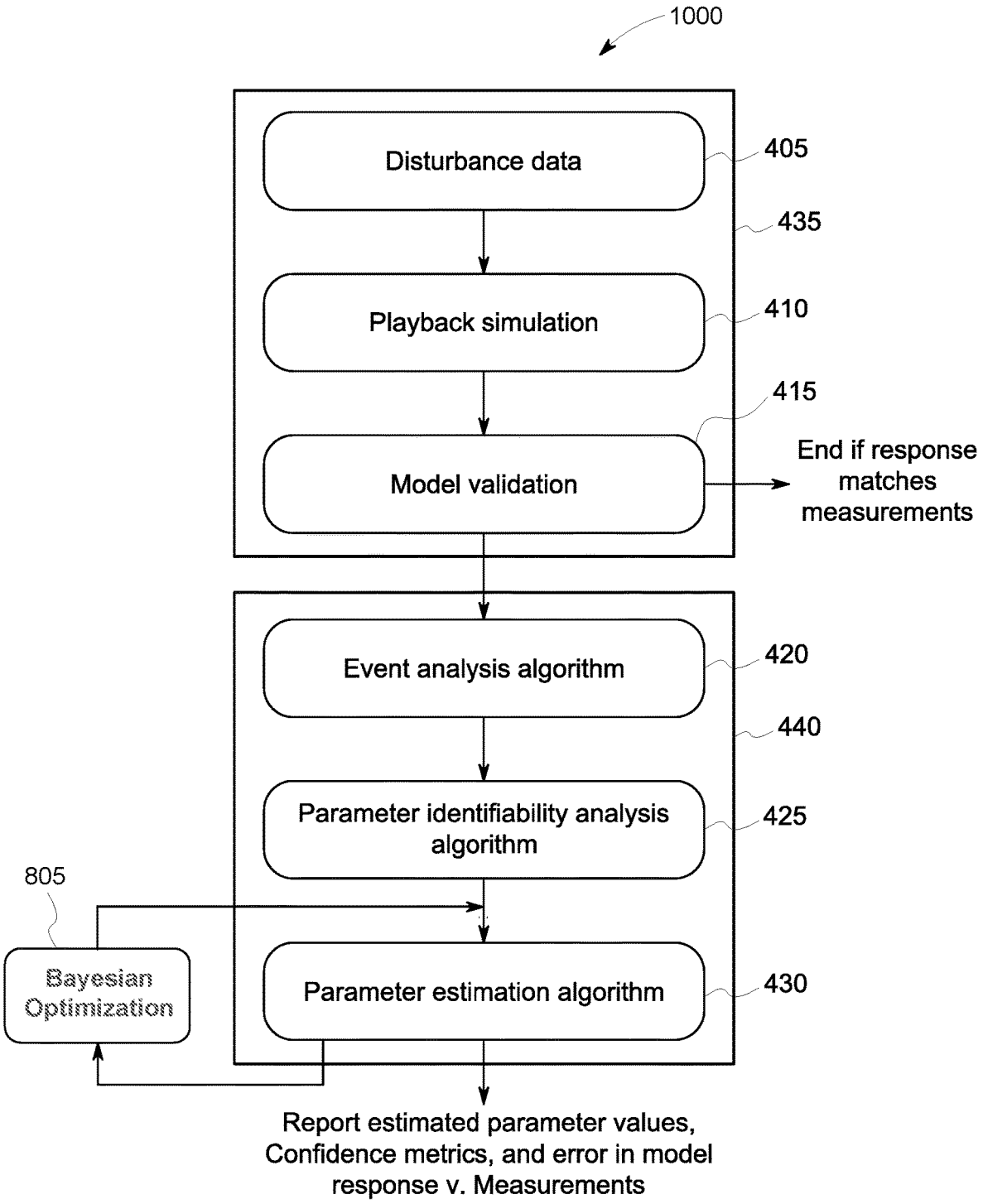

1000
Disturbance data
405
435
Playback simulation
410
415
Model validation
End if response matches measurements
Event analysis algorithm
420
440
Parameter identifiability analysis algorithm
425
805
Bayesian Optimization
Parameter estimation algorithm
430
Report estimated parameter values, Confidence metrics, and error in model response v. Measurements

FIG. 10

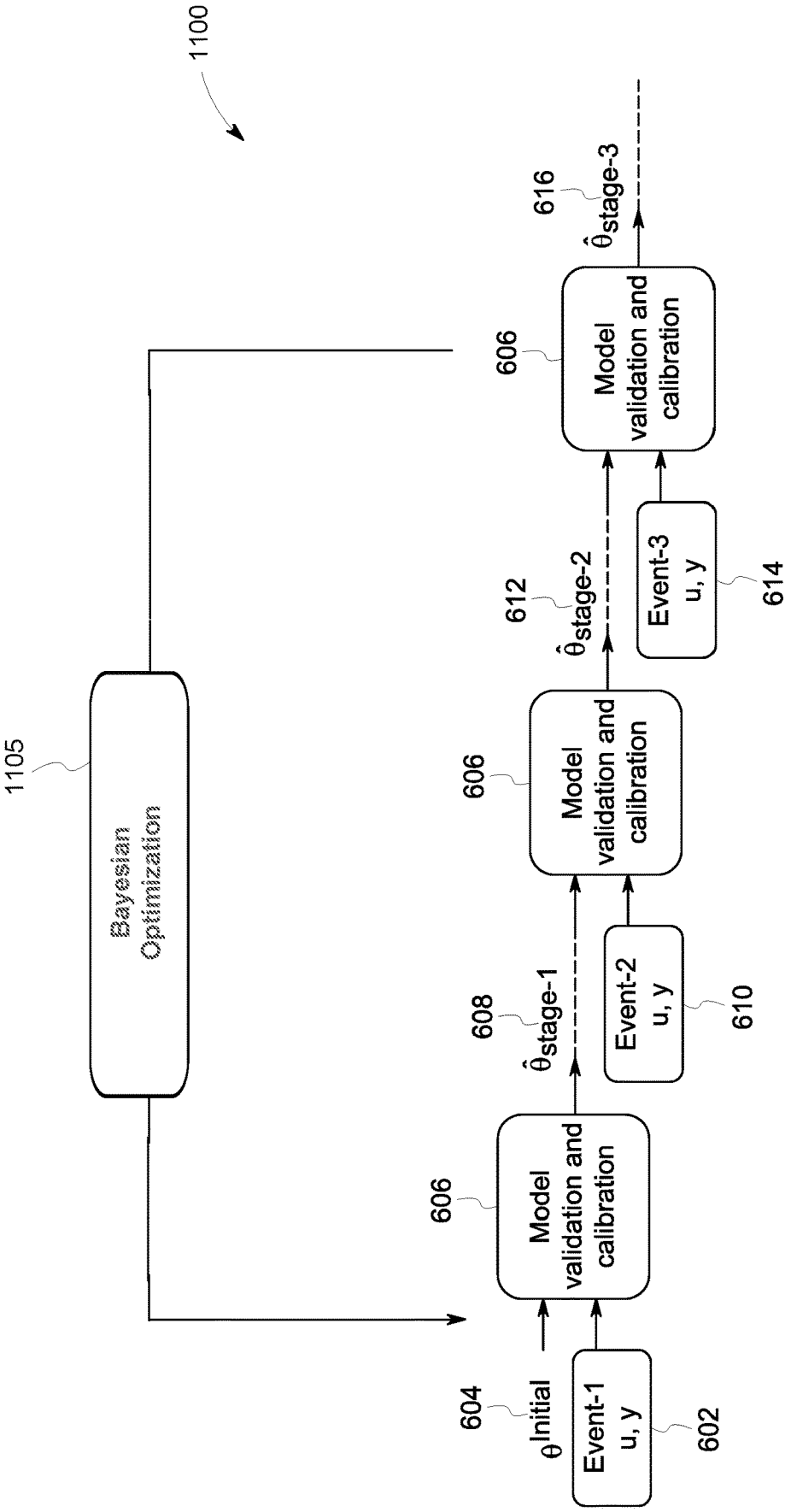

1100
1105
Bayesian Optimization
604
$\theta^{Initial}$
Event-1 u, y
602
606
Model validation and calibration
608
$\hat{\theta}_{stage-1}$
Event-2 u, y
610
606
Model validation and calibration
612
$\hat{\theta}_{stage-2}$
Event-3 u, y
614
606
Model validation and calibration
616
$\hat{\theta}_{stage-3}$

FIG. 11

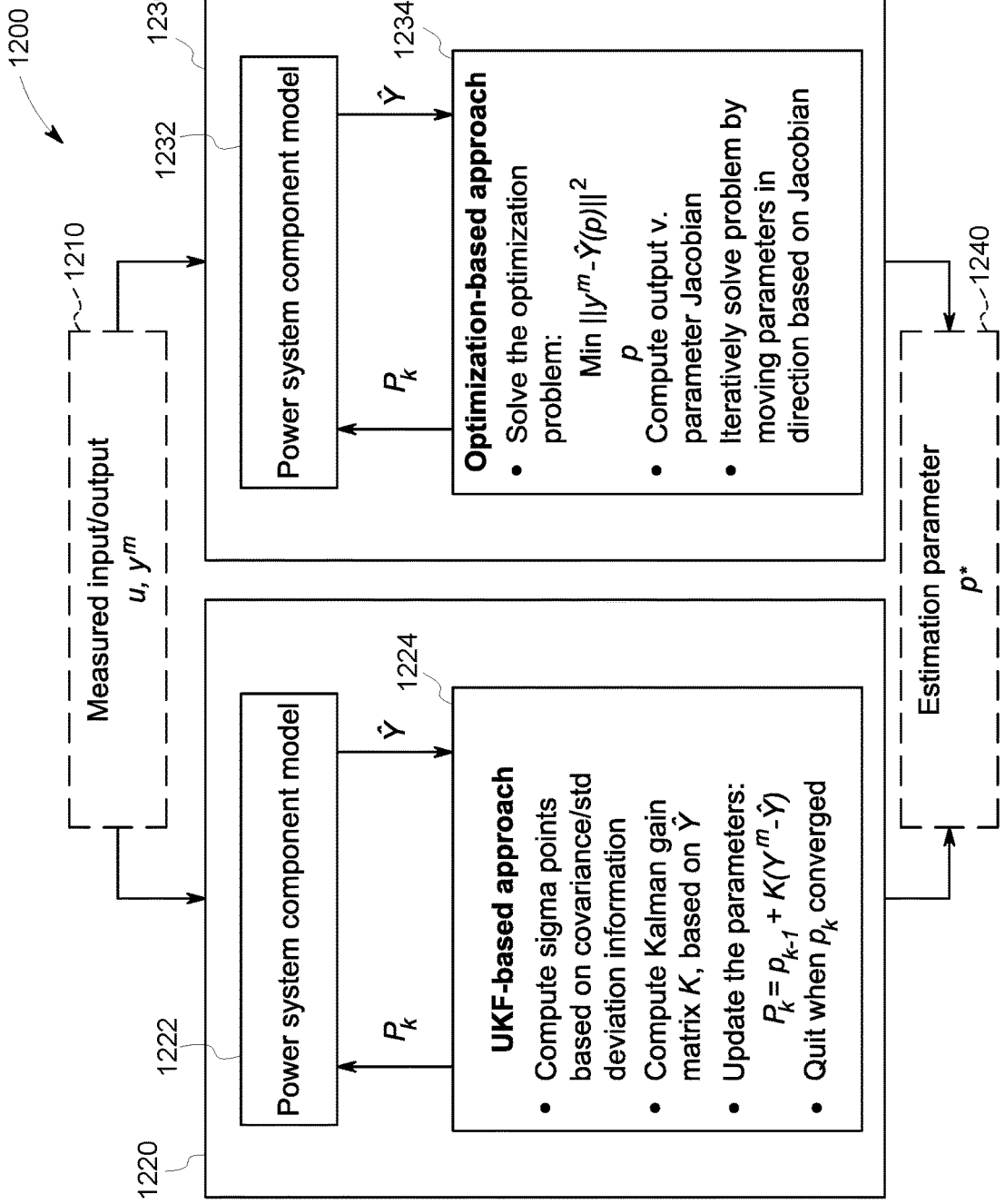

1200
1210
Measured input/output
u, y^m
1220
1222
Power system component model
P_k
$\hat{Y}$
1224
UKF-based approach
• Compute sigma points based on covariance/std deviation information
• Compute Kalman gain matrix K, based on $\hat{Y}$
• Update the parameters: $P_k = p_{k-1} + K(Y^m - \hat{Y})$
• Quit when p_k converged
1230
1232
Power system component model
P_k
$\hat{Y}$
1234
Optimization-based approach
• Solve the optimization problem: $\min_p \|y^m - \hat{Y}(p)\|^2$
• Compute output v. parameter Jacobian
• Iteratively solve problem by moving parameters in direction based on Jacobian
1240
Estimation parameter
p^*

FIG. 12

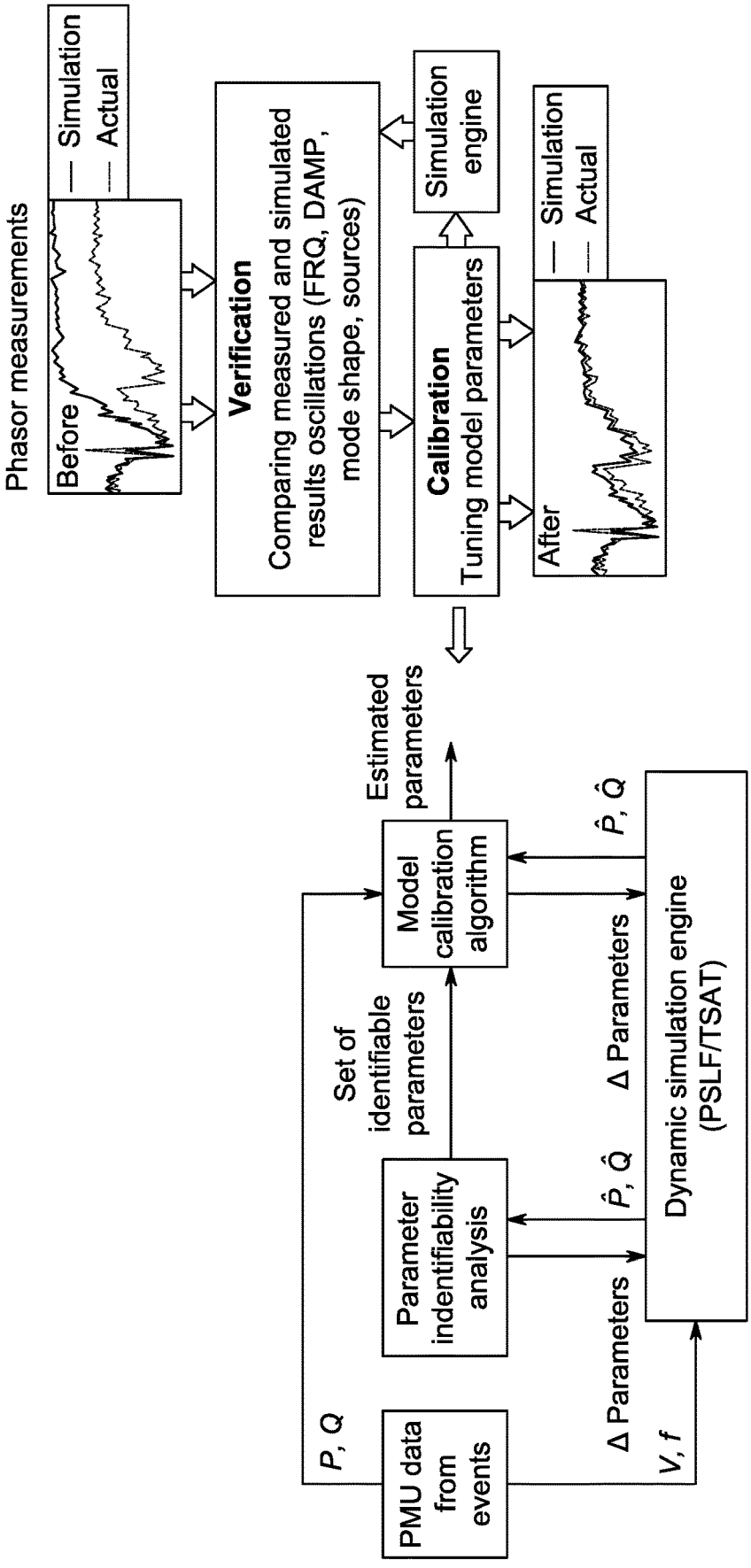

Phasor measurements
Before
Simulation
Actual
Verification
Comparing measured and simulated results oscillations (FRQ, DAMP, mode shape, sources)
Simulation engine
Calibration
Tuning model parameters
After
Simulation
Actual
Estimated parameters
Model calibration algorithm
Set of identifiable parameters
Parameter indentifiability analysis
PMU data from events
P, Q
Δ Parameters
V, f
P̂, Q̂
Δ Parameters
P̂, Q̂
Dynamic simulation engine (PSLF/TSAT)

FIG. 13

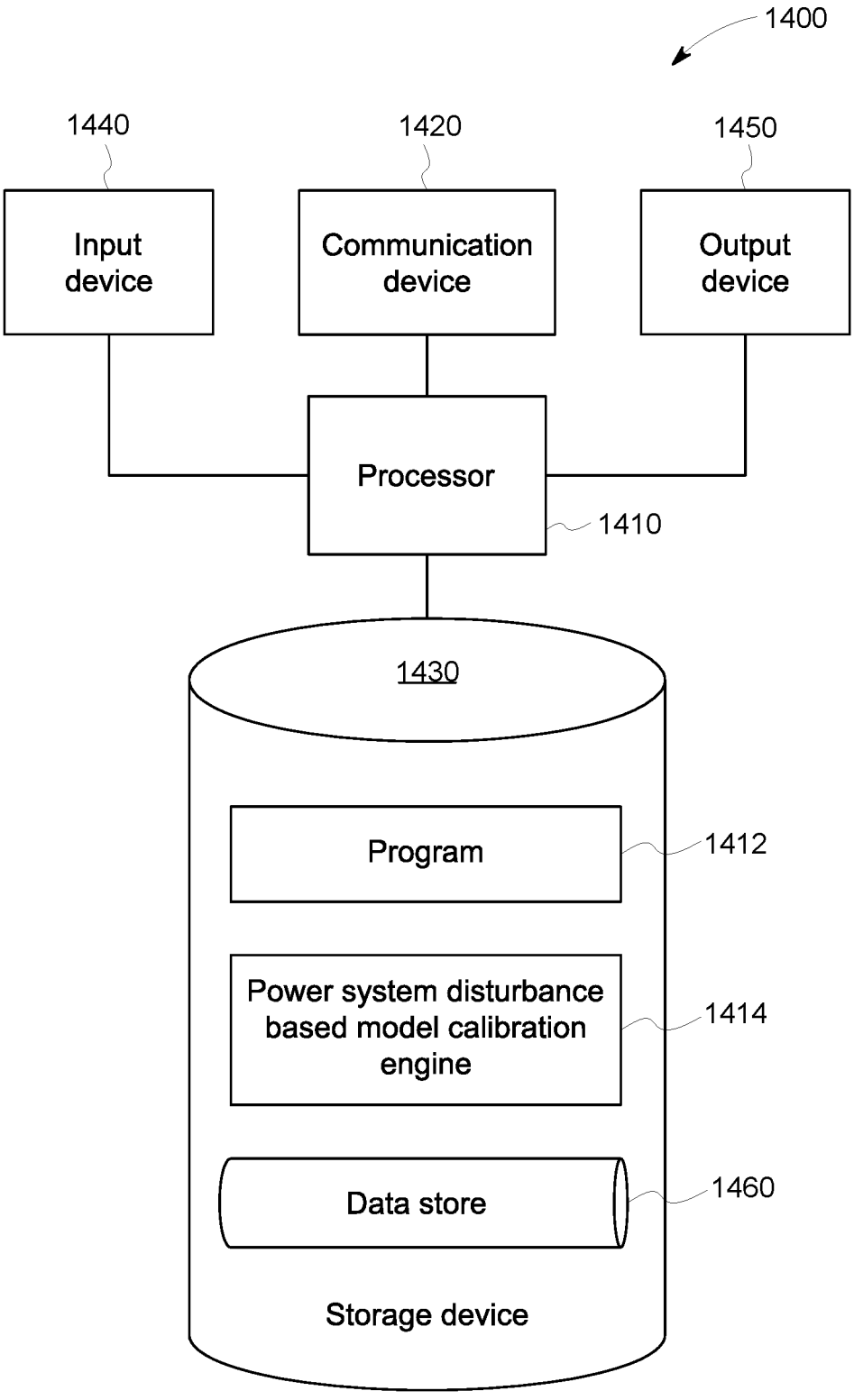

1400
1440
1420
1450
Input device
Communication device
Output device
Processor
1410
1430
Program
1412
Power system disturbance based model calibration engine
1414
Data store
1460
Storage device

FIG. 14

SYSTEMS AND METHODS FOR ENHANCED POWER SYSTEM MODEL CALIBRATION

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of priority to U.S. Provisional Patent Application No. 62/833,492, filed Apr. 12, 2019, entitled "SYSTEMS AND METHODS FOR SEQUENTIAL POWER SYSTEM MODEL PARAMETER ESTIMATION," the entire contents and disclosure of which are incorporated by reference in its entirety.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH & DEVELOPMENT

[0002] This invention was made with government support under U.S. Government Contract Number: DE-0E0000858 awarded by the Department of Energy. The government has certain rights in the invention.

BACKGROUND

[0003] The field of the invention relates generally to enhanced power system model calibration, and more particularly, to a system for modeling sequential power systems based on multiple events with Bayesian Optimization.

[0004] During the 1996 Western System Coordinating Council (WSCC) blackout, the planning studies conducted using dynamic models had predicted stable system operation, whereas the real system became unstable in a few minutes with severe swings. To ensure the models represent the real system accurately, North American Electric Reliability Coordinator (NERC) requires generators above 20 MVA to be tested every 5 years or 10 years (depending on their interconnection) to check the accuracy of dynamic models and update the power plant dynamic models as necessary.

[0005] Some of the methods of performing calibration on the model include performing staged tests and direct measurement of disturbances. In a staged test, a generator is first taken offline from normal operation. While the generator is offline, the testing equipment is connected to the generator and its controllers to perform a series of predesigned tests to derive the desired model parameters. This method may cost $15,000-$35,000 per generator per test in the United States and includes both the cost of performing the test and the cost of taking the generator off-line. Phasor Measurement Units (PMUs) and Digital Fault Recorders (DFRs) have seen dramatic increases in installation in recent years, which allows for non-invasive model validation by using the sub-second-resolution dynamic data. Varying types of disturbances across locations in the power system along with the large installed base of PMUs makes it possible to validate the dynamic models of the generators frequently at different operating conditions.

[0006] As more and more disturbances in power systems are being recorded by PMUs every day, the North American Electric Reliability Corporation (NERC) has pointed out that the analysis of multiple system events is beneficial for model calibration. A generator or load model built from one or two field tests may not be a good model, since it may overfit some specific event and lack the ability to fit the new, fresh measured load curves. Thus far, the primary questions in the community have been associated with how to calibrate the model parameters to make maximal use of the multiple events. Furthermore, depending on disturbance length, one typical simulation may take up to 200-300 seconds. This could be very time consuming when it comes to multiple events. Furthermore, another challenge is determining the exploration and exploitations, also known as breadth and depth of parameters to be analyzed. In addition, the method used may be sensitive to the initial values used. Accordingly, there exists a need for additional speed and accuracy in model calibration.

BRIEF DESCRIPTION

[0007] In one aspect, a system for enhanced power system model calibration is provided. The system includes a computing device including at least one processor in communication with at least one memory device. The at least one processor is programmed to store a model of a device. The model includes a plurality of parameters. The at least one processor is also programmed to receive a plurality of events associated with the device. The at least one processor is further programmed to receive a first set of input calibration values for the plurality of parameters. In addition, the at least one processor is programmed to sequentially analyze the plurality of events in a first sequence to determine a set of calibrated parameter values for the model. Moreover, the at least one processor is programmed to validate the set of calibrated parameter values for the model to determine fit. Furthermore, the at least one processor is programmed to perform Bayesian optimization on the determined fit, the set of calibrated parameter values for the model, and the plurality of events.

[0008] In another aspect, a system for enhanced power system model calibration is provided. The system includes a computing device including at least one processor in communication with at least one memory device. The at least one processor is programmed to store a model of a device. The model includes a plurality of parameters. The at least one processor is also programmed to receive a first event associated with the device. The at least one processor is further programmed to analyze the first event to identify a subset of important parameters from the plurality of parameters. In addition, the at least one processor is programmed to perform Bayesian optimization on the subset of important parameters to determine a set of calibrated parameter values for the model.

[0009] In a further aspect, a system for enhanced sequential power system model calibration is provided. The system includes a computing device including at least one processor in communication with at least one memory device. The at least one processor is programmed to store a model of a device. The model includes a plurality of parameters. The at least one processor is also programmed to receive a first event associated with the device. The at least one processor is further programmed to analyze the first event to identify a subset of important parameters from the plurality of parameters. In addition, the at least one processor is programmed to determine at least one hyperparameter based on the analysis. Moreover, the at least one processor is programmed to perform Bayesian optimization on the hyperparameter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The Figures described below depict various aspects of the systems and methods disclosed therein. It should be

understood that each Figure depicts an embodiment of a particular aspect of the disclosed systems and methods, and that each of the Figures is intended to accord with a possible embodiment thereof. Further, wherever possible, the following description refers to the reference numerals included in the following Figures, in which features depicted in multiple Figures are designated with consistent reference numerals.

[0011] There are shown in the drawings arrangements which are presently discussed, it being understood, however, that the present embodiments are not limited to the precise arrangements and are instrumentalities shown, wherein:

[0012] FIG. **1** illustrates a block diagram of a power distribution grid.

[0013] FIG. **2** illustrates a high-level block diagram of a system for performing sequential calibration in accordance with some embodiments.

[0014] FIG. **3** illustrates a block diagram of an exemplary system architecture for model calibration, in accordance with one embodiment of the disclosure.

[0015] FIG. **4** illustrates a process for power system model parameter conditioning in accordance with some embodiments.

[0016] FIG. **5** illustrates a process for performing optimization using an objective function in at least part by using an integrated acquisition function and a probabilistic model of the objective function, in accordance with some embodiments.

[0017] FIG. **6** illustrates a process for sequential calibration using the system architecture shown in FIG. **3**.

[0018] FIG. **7** is a data flow diagram illustrating the architecture system shown in FIG. **3** executing the sequential calibration process shown in FIG. **6**.

[0019] FIG. **8** illustrates a process for using Bayesian Optimization to optimize model parameters in accordance with the process shown in FIG. **4**.

[0020] FIG. **9** illustrates a process for using Bayesian Optimization to optimize parameter identifiability analysis in accordance with the process shown in FIG. **4**.

[0021] FIG. **10** illustrates a process for using Bayesian Optimization to optimize a hyperparameter in accordance with the process shown in FIG. **4**.

[0022] FIG. **11** illustrates a process for using Bayesian Optimization to optimize event sequences for sequential model calibration, such as shown in the process shown in FIG. **6**.

[0023] FIG. **12** is a diagram illustrating candidate parameter estimation algorithms in accordance with some embodiments.

[0024] FIG. **13** illustrates a two-stage approach of the process for model calibration.

[0025] FIG. **14** is a diagram illustrating an exemplary apparatus or platform according to some embodiments.

DETAILED DESCRIPTION

[0026] In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of embodiments. However, it will be understood by those of ordinary skill in the art that the embodiments may be practiced without these specific details. In other instances, well-known methods, procedures, components and circuits have not been described in detail so as not to obscure the embodiments.

[0027] One or more specific embodiments are described below. In an effort to provide a concise description of these embodiments, all features of an actual implementation may not be described in the specification. It should be appreciated that in the development of any such actual implementation, as in any engineering or design project, numerous implementation-specific decisions must be made to achieve the developers' specific goals, such as compliance with system-related and business-related constraints, which may vary from one implementation to another. Moreover, it should be appreciated that such a development effort might be complex and time consuming, but would nevertheless be a routine undertaking of design, fabrication, and manufacture for those of ordinary skill having the benefit of this disclosure.

[0028] As used herein, the term "Power System Simulation" refers to power system modeling and network simulation in order to analyze electrical power systems using design/offline or real-time data. Power system simulation software is a class of computer simulation programs that focus on the operation of electrical power systems. These types of computer programs are used in a wide range of planning and operational situations, for example: Electric power generation—Nuclear, Conventional, Renewable, Commercial facilities, Utility transmission, and Utility distribution. Applications of power system simulation include, but are not limited to: long-term generation and transmission expansion planning, short-term operational simulations, and market analysis (e.g. price forecasting). A traditional simulation engine relies on differential algebraic equations (DAEs) therein to represent the relationship between voltage, frequency, active power, and reactive power. Those mathematically relationships may be used to study different power systems applications including, but not limited to: Load flow, Short circuit or fault analysis, Protective device coordination, Discrimination or selectivity, Transient or dynamic stability, Harmonic or power quality analysis, and Optimal power flow.

[0029] As used herein, the term "Power System Devices" refers to devices that the simulation engine or simulation model represents, the devices may include: Transmission Systems, Generating Units, and Loads. Transmission Systems include, but are not limited to, transmission lines, power transformers, mechanically switched shunt capacitors and reactors, phase-shifting transformers, static VAR compensators (SVC), flexible AC transmission systems (FACTS), and high-voltage dc (HVDC) transmission systems. The models may include equipment controls such as voltage pick-up and drop-out levels for shunt reactive devices. Generating Units include the entire spectrum of supply resources—hydro-, steam-, gas-, and geothermal generation along with rapidly emerging wind and solar power plants. The Load represents the electrical load in the system, which range from simple light-bulbs to large industrial facilities.

[0030] As used herein, the term "Model Validation" is defined within regulatory guidance as "the set of processes and activities intended to verify that models are performing as expected, in line with their design objectives, and business uses." It also identifies "potential limitations and assumptions, and assesses their possible impact." In the power system context, the Model Validation assures that the model accurately represents the operation of the real system—including model structure, correct assumptions, and that the output matches actual events. There is a reason behind Model Validation for power system asset. The behav-

ior of power plants and electric grids changes over time and should be monitored and updated to ensure that they remain accurate.

[0031] The purpose of model validation is to understand the underlying power system phenomena so they can be appropriately represented in power system studies. The eventual goal of the systems described herein is to generate a total system model that can reasonably predict the outcome of an event. However, to achieve this, the individual constituents of the system model need to be valid. The process of model validation and the eventual "validity" of the model require sound "engineering judgment" rather than being based on a simple pass/fail of the model determined by some rigid criteria. This is because any modeling activity necessitates certain assumptions and compromises, which can only be determined by a thorough understanding of the process being modeled and the purpose for which the model is to be used. Component level Model Validation can be done either through staged tests or on-line disturbance based model validation.

[0032] As used herein, the term "Model Calibration" refers to adjustments of the model parameters to improve the model so that the model's response will match the real, actual, or measured response, given the same model input. Once the model is validated, a calibration process is used to make minor adjustments to the model and its parameters so that the model continues to provide accurate outputs. High-speed, time synchronized data, collected using phasor measurement units (PMUs), are used for model validation of the dynamic response to grid events.

[0033] As used herein, the term "Phasor Measurement Unit" (PMU) refers to a device used to estimate the magnitude and phase angle of an electrical phasor quantity (such as voltage or current) in the electricity grid using a common time source for synchronization. Time synchronization is usually provided by GPS and allows synchronized real-time measurements of multiple remote points on the grid. PMUs are capable of capturing samples from a waveform in quick succession and reconstructing the phasor quantity, made up of an angle measurement and a magnitude measurement. The resulting measurement is known as a synchrophasor. These time synchronized measurements are important because if the grid's supply and demand are not perfectly matched, frequency imbalances can cause stress on the grid, which is a potential cause for power outages.

[0034] PMUs may also be used to measure the frequency in the power grid. A typical commercial PMU may report measurements with very high temporal resolution in the order of 30-60 measurements per second. Engineers use this in analyzing dynamic events in the grid which is not possible with traditional SCADA measurements that generate one measurement every 2 or 4 seconds. Therefore, PMUs equip utilities with enhanced monitoring and control capabilities and are considered to be one of the most important measuring devices in the future of power systems. A PMU can be a dedicated device, or the PMU function can be incorporated into a protective relay or other device.

[0035] As used herein, the terms "Power Grid Disturbance" and "Power Grid Event" refer to outages, forced or unintended disconnection, or failed re-connection of breaker as a result of faults in the power grid. A grid disturbance starts with a primary fault and may also consist of one or more secondary faults or latent faults. A grid disturbance may, for example, be: a tripping of breaker because of lightning striking a line; a failed line connection when repairs or adjustments need to be carried out before the line can be connected to the network; an emergency disconnection due to fire; an undesired power transformer disconnection because of faults due to relay testing; or tripping with a successful high-speed automatic reclosing of a circuit breaker.

[0036] PMU recordings of almost any noticeable grid event may be used for model validation. During grid disturbances, a device operates outside of its normal steady-state condition, providing an opportunity to observe the dynamic behavior of the asset during transients. The PMU data from these transient grid disturbances provides information that cannot be captured with SCADA. These transient disturbances often pose the most risk for grid stability and reliability. Some of the grid events that may generate valuable PMU data for model validation purposes include, but are not limited to:

[0037] Frequency excursion events—In a frequency excursion event, a substantial loss of load or generation causes a significant shift in electrical frequency, typically outside an interconnection's standard. PMU data on a generator's response to a frequency excursion may be used to examine the settings and performance of models of governor and automatic generation control (used to adjust the power output of a generator in response to changes in frequency).

[0038] Voltage excursion events—A fault on the system, a significant change in load or generation (including intermittent renewables), or the loss of a significant load or generation asset may cause voltage shifts. PMU data on a generator's response to a voltage excursion may be used to validate models of its excitation system, reactive capabilities, and automated voltage regulation settings (used to control the input voltage for the exciter of a generator to stabilize generator output voltage).

[0039] Device trips—Transmission devices and lines routinely trip out of service. They cause less severe impacts than a frequency or voltage excursion, but can provide similar data sets useful for model validation.

[0040] Remedial Action Scheme (RAS) activations—Useful data events for model validation can be caused by a reaction to mitigate grid disturbances. Certain grid disturbances may cause a RAS activation, which will attempt to regulate the grid back to a normal operating condition. In some systems, the RAS may include switching on devices such as shunt reactors, changing FACTS devices, or inserting braking resistance. Activation of the RAS may create additional discrete disturbance events on the system, providing frequency and voltage events that can also be used for model validation.

[0041] Probing signals—In the WECC, the high-voltage direct current (HVDC) station at Celilo, Oreg., has the ability to modulate its output power to a known signal, effectively serving as a signal generator into the western power system. These signals can be used to verify and calibrate system-level and generator models' frequency responses, particularly for small-signal-stability analysis.

[0042] A dynamic power system model calibration or tuning using Bayesian optimization is disclosed herein. The system 1) receives a dynamic model, measurement data as dynamic model input and output, initial parameter value for the dynamic model. The system then 2) defines an objective function which represents the deviation between the simulated response using the parameter value and the measured

response. The system also 3) conducts parameter screening to ensure the number of tunable parameter is less than ten. The system further 4) dynamically tunes the parameter value to an undated value by using a Bayesian optimization method.

[0043] In some embodiments, the system may conduct a local search based on the updated value to generate a further undated parameter value. The system may also perform a post evaluation to evaluate the reasonableness of the tuned parameter value.

[0044] The Bayesian Optimization described herein maintains a probabilistic surrogate model and an acquisition function. The objective function represents the goal of the model and the acquisition function is an intermediate function that allows the system to achieve the goal and to identify the next point to analyze next. The Bayesian Optimization performs the following steps. First the Bayesian Optimization initializes a probabilistic model of the objective function using initial parameter points, the probabilistic model of the objective function comprising a stationary probabilistic model composed with a non-linear one-to-one mapping of the values of the parameters from a first domain to a second domain. In the exemplary embodiment, the first domain includes the dynamic model parameters and/or the hyperparameters. The second domain includes the measurement of the similarity between the simulation response generated from the model parameter and/or hyperparameters and the measured response. The Bayesian Optimization then repeats the following steps until reaching a fixed number of iteration or time or a stopping criterion is reached. The Bayesian Optimization generates a new set of parameter values corresponding to at least one parameter of the power system model calibration system, by optimizing an acquisition function, which depends at least in part on the current set of parameter values and the probabilistic model of the objective function. Then the Bayesian Optimization augments the data set with the new set of parameter values and evaluated the objective function value using the power system model operated at the identified set of parameter values. Further, the Bayesian Optimization updates the probabilistic model of the objective function to obtain an updated probabilistic model of the objective function, based on the augmented data set. Because Bayesian optimization is a global technique, unlike many other algorithms, to search for a global solution the system does not have to start the algorithm from various initial points.

[0045] FIG. **1** illustrates a power distribution grid **100**. The grid **100** includes a number of components, such as power generators **110**. In some cases, planning studies conducted using dynamic models predict stable grid **100** operation, but the actual grid **100** may become unstable in a few minutes with severe swings (resulting in a massive blackout). To ensure that the models represent the real system accurately, the North American Electric Reliability Coordinator ("NERC") requires generators **110** above 10 MVA to be tested every five years to check the accuracy of dynamic models and let the power plant dynamic models be updated as necessary. The systems described herein consider not only active power (P) and reactive power (Q), but also voltage (U) and frequency (F).

[0046] In a typical staged test, a generator **110** is first taken offline from normal operation. While the generator **110** is offline, testing equipment is connected to the generator **110** and its controllers to perform a series of pre-designed tests to derive the desired model parameters. Recently, PMUs **120** and Digital Fault Recorders ("DFRs") **130** have seen dramatic increase in installation in recent years, which may allow for non-invasive model validation by using the sub-second-resolution dynamic data. Varying types of disturbances across locations in the grid **100** along with the large installed base of PMUs **120** may, according to some embodiments, make it possible to validate the dynamic models of the generators **110** frequently at different operating conditions. There is a need for a production-grade software tool generic enough to be applicable to wide variety of models (traditional generating plant, wind, solar, dynamic load, etc. with minimal changes to existing simulation engines. Note that model calibration is a process that seeks multiple (dozens or hundreds) of model parameters, which could suffer from local minimum and multiple solutions. There is need for an algorithm to enhance the quality of a solution within a reasonable amount time and computation burdens.

[0047] Online performance monitoring of power plants using synchrophasor data or other high-resolution disturbance monitoring data acts as a recurring test to ensure that the modeled response to system events matches actual response of the power plant or generating unit. From the Generator Owner (GO)'s perspective, online verification using high resolution measurement data can provide evidence of compliance by demonstrating the validity of the model by online measurement. Therefore, it is a cost-effective approach for GO as they may not have to take the unit offline for testing of model parameters. Online performance monitoring requires that disturbance monitoring equipment such as a PMU be located at the terminals of an individual generator or Point of Interconnection (POI) of a power plant.

[0048] The disturbance recorded by PMU normally consists of four variables: voltage, frequency, active power, and reactive power. To use the PMU data for model validation, the playback simulation has been developed and is now available in many major grid simulators. The simulated output including active power and reactive power will be generated and can be further compared with the measured active power and reactive power.

[0049] To achieve such results, FIG. **2** is a high-level block diagram of a system **200** in accordance with some embodiments. The system **200** includes one or more measurement units **210** (e.g., PMUs, DFRs, or other devices to measure frequency, voltage, current, or power phasors) that store information into a measurement data store **220**. As used herein, the term "PMU" might refer to, for example, a device used to estimate the magnitude and phase angle of an electrical phasor quantity like voltage or current in an electricity grid using a common time source for synchronization. The term "DFR" might refer to, for example, an Intelligent Electronic Device ("TED") that can be installed in a remote location, and acts as a termination point for field contacts. According to some embodiments, the measurement data might be associated with disturbance event data and/or data from deliberately performed unit tests. According to some embodiments, a model parameter tuning engine **250** may access this data and use it to tune parameters for a dynamic system model **260**. The process might be performed automatically or be initiated via a calibration command from a remote operator interface device **290**. As used

herein, the term "automatically" may refer to, for example, actions that can be performed with little or no human intervention.

[0050] Note that power systems may be designed and operated using mathematical models (power system models) that characterize the expected behavior of power plants, grid elements, and the grid as a whole. These models support decisions about what types of equipment to invest in, where to put it, and how to use it in second-to-second, minute-to-minute, hourly, daily, and long-term operations. When a generator, load, or other element of the system does not act in the way that its model predicts, the mismatch between reality and model-based expectations can degrade reliability and efficiency. Inaccurate models have contributed to a number of major North American power outages.

[0051] The behavior of power plants and electric grids may change over time and should be checked and updated to assure that they remain accurate. Engineers use the processes of validation and calibration to make sure that a model can accurately predict the behavior of the modeled object. Validation assures that the model accurately represents the operation of the real system—including model structure, correct assumptions, and that the output matches actual events. Once the model is validated, a calibration process may be used to make minor adjustments to the model and its parameters so that the model continues to provide accurate outputs. High-speed, time-synchronized data, collected using PMUs may facilitate model validation of the dynamic response to grid events. Grid operators may use, for example, PMU data recorded during normal plant operations and grid events to validate grid and power plant models quickly and at lower cost.

[0052] The transmission operators or Regional reliability coordinators, or Independent System Operators, like MISO, ISO-New England, PG&E, can use this calibrated generator or power system model for power system stability study based on N-k contingencies, in every 5 to 10 minutes. If there is a stability issue (transient stability) for some specific contingency, the power flow will be redirected to relieve the stress-limiting factors. For example, the output of some power generators will be adjusted to redirect the power flow. Alternatively, adding more capacity (more power lines) to the existing system can be used to increase the transmission capacity.

[0053] With a model that accurately reflects oscillations and their causes, the grid operator can also diagnose the causes of operating events, such as wind-driven oscillations, and identify appropriate corrective measures before those oscillations spread to harm other assets or cause a loss of load.

[0054] As used herein, devices, including those associated with the system **200** and any other device described herein, may exchange information via any communication network which may be one or more of a Local Area Network ("LAN"), a Metropolitan Area Network ("MAN"), a Wide Area Network ("WAN"), a proprietary network, a Public Switched Telephone Network ("PSTN"), a Wireless Application Protocol ("WAP") network, a Bluetooth network, a wireless LAN network, and/or an Internet Protocol ("IP") network such as the Internet, an intranet, or an extranet. Note that any devices described herein may communicate via one or more such communication networks.

[0055] The model parameter tuning engine **250** may store information into and/or retrieve information from various data stores, which may be locally stored or reside remote from the model parameter tuning engine **250**. Although a single model parameter tuning engine **250** is shown in FIG. **2**, any number of such devices may be included. Moreover, various devices described herein might be combined according to embodiments of the present invention. For example, in some embodiments, the measurement data store **220** and the model parameter tuning engine **250** might comprise a single apparatus. The system **200** functions may be performed by a constellation of networked apparatuses, such as in a distributed processing or cloud-based architecture.

[0056] A user may access the system **200** via the device **290** (e.g., a Personal Computer ("PC"), tablet, or smartphone) to view information about and/or manage operational information in accordance with any of the embodiments described herein. In some cases, an interactive graphical user interface display may let an operator or administrator define and/or adjust certain parameters (e.g., when a new electrical power grid component is calibrated) and/or provide or receive automatically generated recommendations or results from the system **200**.

[0057] The example embodiments provide a predictive model which can be used to replace the dynamic simulation engine when performing the parameter identification and the parameter calibration. This is described in U.S. patent application Ser. No. 15/794769, filed 26 Oct. 2017, the contents of which are incorporated by reference in their entirety. The model can be trained based on historical behavior of a dynamic simulation engine thereby learning patterns between inputs and outputs of the dynamic simulation engine. The model can emulate the functionality performed by the dynamic simulation engine without having to perform numerous rounds of simulation. Instead, the model can predict (e.g., via a neural network, or the like) a subset of parameters for model calibration and also predict/estimate optimal parameter values for the subset of parameters in association with a power system model that is being calibrated. According to the examples herein, the model may be used to capture both input-output function and first derivative of a dynamic simulation engine used for model calibration. The model may be updated based on its confidence level and prediction deviation against the original simulation engine.

[0058] Here, the model may be a surrogate for a dynamic simulation engine and may be used to perform model calibration without using DAE equations. The system described herein may be a model parameter tuning engine, which is configured to receive the power system data and model calibration command, and search for the optimal model parameters using the surrogate model until the closeness between simulated response and the real response from the power system data meet a predefined threshold. In the embodiments described herein, the model operates on disturbance event data that includes one or more of device terminal real power, reactive power, voltage magnitude, and phase angle data. The model calibration may be triggered by user or by automatic model validation step. In some aspects, the model may be trained offline when there is no grid event calibration task. The model may represent a set of different models used for different kinds of events. In some embodiments, the model's input may include at least one of voltage, frequency and other model tunable parameters. The model may be a neural network model, fuzzy logic, a polynomial function, and the like. Other model tunable parameters may

include a parameter affecting dynamic behavior of machine, exciter, stabilizer and governor. Also, the surrogate model's output may include active power, reactive power or both. In some cases, the optimizer may be gradient based method including Newton-like methods. For example, the optimizer may be gradient free method including pattern search, genetic algorithm, simulated annealing, particle swarm optimizer, differential evolution, and the like.

[0059] FIG. **3** illustrates a block diagram of exemplary system architecture **300** for power system model calibration, in accordance with one embodiment of the disclosure. In the exemplary embodiment, the system architecture **300** receives network models **302**, sub-system definitions **304**, dynamic models **306**, and event data **308**.

[0060] Steady state network models **302** (sometimes referred to as power-flow data) can be either EMS or system planning models. In some embodiments, they may be in e-terra NETMOM or CIM13 format. Dynamic models **306** can be in either PSS/E or PSLF or TSAT format. The system **300** can also accept more than one dynamic data file when data is distributed among multiple files. In the exemplary embodiment, the network models **302** and the dynamic models **306** use the same naming convention for the network elements.

[0061] In the exemplary embodiment, the sub-system definitions **304** are based on the network model **302** and one or more maps of the power plant. A sub-system identification module combines the network model **302** and the one or more maps to generate the sub-system definition **304**. In some embodiments, the sub-system definition **304** is provided via an XML file that defines the POI(s) and generators that makes up a power plant. Power plants are defined by generators in the plant with its corresponding POI(s). A few examples of power plant sub-system definitions are listed below in TABLE 1.

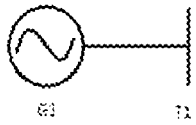
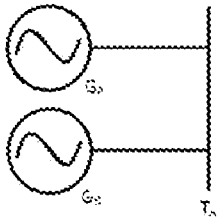
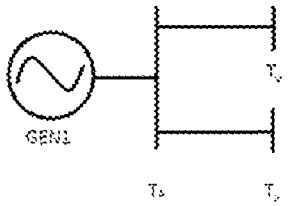
Sub-system name	Sub-system	POI(s)	Generator(s)
S1		T1	G1
S2		T_A	G_A
			G_B
S3		T_X	GEN1
		T_Y	

TABLE 1

[0062] In the exemplary embodiment, the system **300** provides a user interface to facilitate defining the power plant starting from a potential POI. Potential POIs are identified as terminals/buses in the system having all required measurements (V, f, P, Q) to perform model validation and calibration. A measurement mapping module identifies terminals with V, f, P, Q measurements and lets the user search for radially connected generators starting from potential POIs. Sub-system definitions **304** may also be saved for future use. In some embodiments, a sub-system definition **304** is defined for each event **308**.

[0063] Events **308** are situations where the voltage and/or the frequency of the power system changes. For example, an event **308** may be a generator turning on. In some embodiments, the event **308** has the same or similar attributes to a previous event **308**, such as that same generator turning on; the event **308** is skipped to reduce redundant processing. In the exemplary embodiment, the event data or Phasor data **308** will be imported from a variety of sources, such as, but not limited to, e-terraphasorpoint, openPDC, CSV files, COMTRADE files and PI historian. In the exemplary embodiment, the POIs will have at least voltage, frequency, real power and reactive power measurements. In some embodiments, voltage angle is substituted for frequency.

[0064] The network models **302**, sub-system definitions **304**, dynamic models **306**, and event data **308** are analyzed by the system **300** as described herein. In the exemplary embodiment disclosed herein, the model utilizes multiple disturbance events to validate and calibrate power system models for compliance with NERC mandated grid reliability requirements. The interactive model calibration system described herein may include three steps. The first step is an interactive user console to allow a user to select a local region for emphasis or de-emphasis. The next step is a parameter identifiability module configured to analyze the mutual information between the measurement value and the Jacobian matrix. The third step in an integrated approach where the parameter identifiability module and the nonlinear least square optimization for parameter estimation automatically assign the weights based on the user's selection on the user console.

[0065] More specifically, the network models **302**, sub-system definitions **304**, dynamic models **306**, and event data **308** are analyzed and validated by the model validation component **310**. If the models are validated, then the corresponding data is sent to a parameter identifiability component **312**. This component **312** analyzes the event and models to determine which parameters are significant for this event **308**. Then, the tunable parameters are transmitted to a tunable parameter estimation component **314**, which further analyzes the significant parameters to calibrate the parameters in the model being executed by the simulation engine **316**. In the exemplary embodiment, the model validation component **310**, the parameter identifiability component **312**, and the tunable parameter estimation component **314** are all in communication with a dynamic behavior characterization component **318**, which extracts features from the events **308**, generates weights for those features, and provides the user the ability to fine tune the model calibration and add subject matter expert knowledge to the model calibration process. The end result is a fully calibrated model **320**. The steps in this process are further described below.

[0066] In the exemplary embodiment, the model validation component **310** validates the models **302** and **306** and definitions **304** that are being input into the system **300**. In at least one embodiment, a typical synchronous generator model has four parts: machine model, turbine-governor model, excitation model, and power system stabilizer (PSS) model. The model validation component **310** validates the provided models based on a collection of published NERC List of Acceptable Models, user preferences, and historical data. In some embodiments, there may also be prohibited model lists that are evaluated. Furthermore, units with a power system stabilizer (PSS) should have an excitation system model.

[0067] In the exemplary embodiment, the user will be notified if any prohibited model or missing excitation model has been identified. Based on this information, the user can further correct the dynamic model **306** if there is human error, or to use the model conversion module to convert any prohibited model to the valid models before evaluating the curve fitting performance. Of course, the user can also ignore the warning and continue the model validation and calibration process.

[0068] The second step is parameter identifiability. The goal of this step is to perform a comprehensive identifiability study across the models **302** and **306**, the definitions **304**, and the events **308** and provide an identifiable parameter set for the simultaneous calibration which tunes the most identifiable parameters. The parameter identifiability component **312** analyzes the parameters to identify potential parameters for use based on the dot product (or scalar product) of the columns of J and r as defined below. In the exemplary embodiment, r is referred to as residual which is the difference between the measured response data series and the simulated response data series where:

$$r(p)=y_t^m-y_t(x) \quad \text{EQ. 1}$$

where y_t^m is the measured response of active and reactive power provided in the event data **308**, $y_t(x)$is the simulated response of active and reactive power based on dynamic simulation engine, including but not limited to, GE's PSLF, Siemens PTI's PSS/E, etc. x represents the model parameters.

[0069] The parameter identifiability component **312** uses the sum of squares (SOS) objective: $\|r(x)\|_2^2$. Then the parameter identifiability component **312** uses the Quadratic Model (QM) of the objective at (x_k+d) to approximate the next step like $r(x_{k+1})$.

$$QM(J_k, r_k, d)=\|r(x)+J_k d\|_2^2 \quad \text{EQ. 2}$$

where J_k is the Jacobian vector, which is equal to

$$J_k = \frac{dr}{dx}\Big|_{xk},$$

and $r_k=r(x_k)$ which is the sensitivity result. This leads to:

$$r(x_{k+1})=r(x_k)+J_k(x_k+d) \quad \text{EQ. 3}$$

[0070] The ultimate goal is to get $r(x_{k+1})=0$. This leads to $r(x_k)=-J_k(x_k+d)$.

[0071] In the exemplary embodiment, the vector $r(x_k)$ is compared to the Jacobian vector J_k to determine the θ (angle) between them. In some embodiments, each vector J_k may have up to 1000 values each, where the number of values in

the Jacobian vector depends on the number of sampling points in the event. The θ is calculated by generating the dot product of the vector $r(x_k)$ to the Jacobian vector J_k.

$$r(x_k)*J_k=\|r(x_k)\|\|J_k\|\cos\theta \quad \text{EQ. 4}$$

[0072] The resulting θ is compared to a threshold. Parameters with a corresponding θ below the threshold are sent to the pool of parameters that are selected. The ideal θ is zero, but that is generally unachievable. In some embodiments, any parameter with a θ of less than 5° is selected by the parameter identifiability component **312**. This threshold is configurable by the user, such as through an interactive user interface. The key idea is that the more orthogonal the angles are between the vectors of J and r, the less likely changes to that parameter moves the response in the desired way. This approach can be extended to a weighted version, by scaling both the measured response and simulated response with a weight vector w_t. The weight factor w_t has the same length of the data samples in the event of interest. In this way, given a defined weight factor, it can affect the above calculated angles are between the vectors of J and r. Where r and J_k may be calculated as:

$$r(x_k)= \sum_{t=1}^{T} w_p(t)*\left(\frac{y_p^m(t)-y_p(x,t)}{y_p^{base}}\right)^2+w_q(t)*\left(\frac{y_q^m(t)-y_q(x,t)}{y_q^{base}}\right)^2 \quad \text{EQ. 5}$$

$$J_k=\frac{r(x_k+\Delta x)-r(x_k)}{\Delta x} \quad \text{EQ. 6}$$

where t represents each point of time in the event, where T is the event time length, and where $w_p(t)$ is a weight vector assigned along the time axis to the active power p, $w_q(t)$ is a weight vector assigned along the time axis to the reactive power q, $y_p^m(t)$ represents the measured active power at time stamp t, $y_p(x, t)$ represents the simulation result at time stamp t with parameter x, y_p^{base} represents the base value of the active power p.

[0073] In the exemplary embodiment, the parameter identifiability component **312** receives a plurality of raw parameters x. The parameter identifiability component **312** analyzes each of the parameters using the above equations to determine the θ between the J_k and the $r(x_k)$ for each of the parameters. If the θ meets or is below a predetermined threshold, the parameter identifiability component **312** stores that parameter in a pool of parameters. In the exemplary embodiment, the parameter identifiability component **312** presents the parameters in the pool to the user for approval or adjustment via an interactive user interface.

[0074] Once selected or confirmed by the user, the tunable parameters are provided to the tunable parameter estimation component **314**. The tunable parameter estimation component **314** adjusts the models based on the tunable parameters selected or confirmed by the user. The parameter estimation component **314** also performs weighted non-linear least squares optimizations for estimating the parameters. The goal is to identify the right parameter to minimize the difference between the $y_t(x)$ and y_t^m so that the estimation matches the measured response.

$$\min_{x_l\le x\le x_u}\sum_{t=1}^{T} w_p(t)*\left(\frac{y_p^m(t)-y_p(x,t)}{y_p^{base}}\right)^2+ w_q(t)*\left(\frac{y_q^m(t)-y_q(x,t)}{y_q^{base}}\right)^2 \quad \text{EQ. 7}$$

where t represents each point of time in the event, where T is the event time length, and where $w_p(t)$ is a weight vector assigned along the time axis to the active power p, $w_q(t)$ is a weight vector assigned along the time axis to the reactive power q, $y_p^m(t)$ represents the measured active power at time stamp t, $y_p(x, t)$ represents the simulation result at time stamp t with parameter x, y_p^{base} represents the base value of the active power p, which could be 100 MVA for example. x_l, x_u represent the low bound and high bound for parameter x.

[0075] In reality, there are around 60~120 parameters for one typical generator simulation model. Tuning all of them given one event is not realistic and not desirable. The industry expects as few as possible of the parameters being tuned given one event or multiple events. One approach is to use the above mentioned sensitivity analysis (or parameter identification) to down select only those parameter subset which lead to high sensitive response change. An alternative approach is to use the sparse optimization by adding a L1 norm as a regularization term in the objective function, and the optimization solver will determine the parameter value while minimizing the number of parameters tuned. This can be stated as:

$$\min_{x_l\le x\le x_u}\sum_{t=1}^{T} w_p(t)*\left(\frac{y_p^m(t)-y_p(x,t)}{y_p^{base}}\right)^2+ w_q(t)*\left(\frac{y_q^m(t)-y_q(x,t)}{y_q^{base}}\right)^2+\alpha\cdot\|x-x_0\|_1 \quad \text{EQ. 8}$$

where α is how important the tunable parameter is, x_0 is the initial parameter, x is the parameter, and $\|x-x_0\|$ is a penalty term. This is considered weighted sparse nonlinear least square optimization.

[0076] In the exemplary embodiment, the system defines regions or segments (which are portions or time slices of the event) and their corresponding weights (as shown in FIG. **4**). The system also allows the user to adjust the regions and weights through the user interface. The user may then assign different weights to each region. For example, a user may assign a first weight for times 0 to 0.3 seconds in the event and a second weight for times 0.3 to 1 second into the event. In addition, the user may define two different weights for the active power curve and the reactive power curve. In some embodiments, the system defines a default weight that is used for sections or regions that do not have user defined weights.

[0077] In the exemplary embodiment, the parameter estimation component **314** performs multiple iterations of the calculations until the residual error between the measure values and the estimated values is reduced to below a threshold.

[0078] In some embodiments, the user accesses a user interface to set the total number of events **308** that will be analyzed, set the stored file locations, and set the sequence

that the events **308** will be analyzed in. The user interface may also be used for other adjustments as described herein.

[0079] The feature of an event may include peak value, bottom value, overshoot percentage, rising time, settling time, delay time, peak time, steady state error, phase shift, damping ratio, energy function, cumulative deviation in energy, Fourier transformation spectrum information, frequency response, principal component, minimum volume ellipsoid, and/or steady state gain (P, Q, u, f) of the event. The feature is extracted from the time series of active power, reactive power, voltage, and frequency.

[0080] Alternatively, the system **300** may use Bayesian Optimization to tune the parameters. Bayesian Optimization is a general framework for the global optimization of noisy, expensive, blackbox functions. The strategy is based on the notion that one can use a relatively cheap probabilistic model to query as a surrogate for the financially, computationally or physically expensive function that is subject to the optimization. Bayes' rule is used to derive the posterior estimate of the true function given observations, and the surrogate is then used to determine the next most promising point to query. Bayesian Optimization methods maintain a surrogate that models the objective function, which the methods then use to choose where to evaluate. Bayesian Optimization distinguishes itself from other surrogate methods by using surrogates developed using Bayesian statistics, and in deciding where to evaluate the objective using a Bayesian interpretation of these surrogates. Bayesian Optimization consists of two main components: a Bayesian statistical model for modeling the objective function, and an acquisition function for deciding where to sample next. After evaluating the objective according to an initial space-filling experimental design, often consisting of points chosen uniformly at random, the model and acquisition function are used iteratively to allocate the remainder of a budget of N function evaluations.

[0081] A sample Bayesian Optimization algorithm is as follows: a) place a Gaussian process prior on f; b) observe f at n_0 points according to an initial space-filling experimental design; c) set $n=n_0$; d) while $n \leq N$, 1) update the posterior probability distribution on f using all available data, 2) let x_n be a maximizer of the acquisition function over x, where the acquisition function is computed using the current posterior distribution, 3) observe $y_n=f(x_n)$, and 4) increment n; and e) return a solution. The solution is either the point evaluated with the largest $f(x)$, or the point with the largest posterior mean.

[0082] A common approach is to use a GP to define a distribution over objective functions from the input space to a loss that one wishes to minimize. That is, given observation pairs of the form $\{x_n, y_n\}$ N n=1, where $x_n \in X$ and $y_n \in R$, and assuming that the function $f(x)$ is drawn from a Gaussian process prior where $y_n \sim N(f(x_n)v)$, where v is the function observation noise variance.

[0083] In some embodiments, there is a tradeoff, with the largest expected improvement occurring where the posterior standard deviation is high (far away from previously evaluated points), and where the posterior mean is also high. The smallest expected improvement is 0, at points that were previously evaluated. The posterior standard deviation is 0 at this point, and the posterior mean is necessarily no larger than the best previously evaluated point. The expected improvement algorithm would evaluate next at the point indicated with an x where the function is maximized.

[0084] Because grid disturbances occur intermittently, the user of the calibration tool may be required to re-calibrate model parameters in a sequential manner as new disturbances come in. In this scenario, the user has a model that was calibrated to some observed grid disturbances to start with, and observes a larger that acceptable mismatch with a newly encountered disturbance. The task is to tweak the model parameters so that the model explains the new disturbance without detrimentally affecting the match with earlier disturbances. One potential solution is to run calibration simultaneously on all events of interest strung together; however, this comes at the cost of significant computational expense and engineering involved in enabling running a batch of events simultaneously. One more efficient method may be to carry some essential information from the earlier calibrations runs and guide the subsequent calibration run that helps explain the new disturbance without losing earlier calibration matches.

[0085] In the exemplary embodiment, the framework of Bayesian estimation may be used to develop a sequential estimation capability into the existing calibration framework. The true posterior distribution of parameters (assuming Gaussian priors) after the calibration process may be quite complicated due to the nonlinearity of the models. One approach in sequential estimation is to consider a Gaussian approximation of this posterior as is done in Kalman filtering approaches to sequential nonlinear estimation. In a nonlinear least squares approach, this simplifies down to a quadratic penalty term for deviations from the previous estimates, and the weights for this quadratic penalty come from a Bayesian argument.

$$\min \sum_{t=1}^{T} w_t * \left(\frac{y_t^i - y_t(x)}{y_{base}} \right) + (x - x_{mean})^T * \left(\sum_b^k \right)^{-1} * (x - x_{mean}) \quad \text{EQ. 9}$$

[0086] The measured values of P and Q may be represented by a simulated value plus an error term.

$$y_i = y(x_i|b) + e_i \quad \text{EQ. 10}$$

$$\Sigma_b^k = \Sigma_b^{k-1} + J^T * J \quad \text{EQ. 11}$$

[0087] In some embodiments, the errors may be subject to Normal distribution, either independently or else with errors correlated in some known way, such as, but not limited to, multivariate Normal distribution.

$$e_i \sim N(0, \sigma_i)$$

$$e \sim N(0, \Sigma)$$

[0088] The above may be used to find the parameters of a model b from the data.

$$P(b|\{y_i\}) \propto P(\{y_i\}|b)P(b) \quad \text{EQ. 12}$$

$$\propto \prod_i \exp\left[-\frac{1}{2} \left(\frac{y_i - y(x_i|b)}{\sigma_i} \right)^2 \right] P(b)$$

$$\propto \exp\left[-\frac{1}{2} \sum_i \left(\frac{y_i - y(x_i|b)}{\sigma_i} \right)^2 \right] P(b)$$

$$\propto \exp\left[-\frac{1}{2} x^2(b) \right] P(b)$$

[0089] Alternatively, the parameter value b_0 that minimizes x^2 may be calculated using a Taylor series approximation.

$$-\frac{1}{2}x^2(b) \approx -\frac{1}{2}x^2_{min} - \frac{1}{2}(b-b_0)^T\left[\frac{1}{2}\frac{\partial^2 x^2}{\partial b \partial b}\right](b-b_0) \quad \text{EQ. 13}$$

$$P(B \mid \{y_i\}) \propto \exp\left[-\frac{1}{2}(b-b_0)^T\sum\nolimits_b^{-1}(b-b_0)\right]P(b) \quad \text{EQ. 14}$$

$$\sum\nolimits_b = \left[\frac{1}{2}\frac{\partial^2 x^2}{\partial b \partial b}\right]^{-1} \quad \text{EQ. 15}$$

where Σ_b is the covariance of "standard error" matrix of the fitted parameters.

[0090] FIG. **4** is a process **400** for power system model parameter conditioning according to some embodiments. At Step **405**, disturbance data may be obtained (e.g., from a PMU or DFR) to obtain, for example, V, f, P, and Q measurement data at a Point Of Interest ("POI"). At Step **410**, a playback simulation may run load model benchmarking using default model parameters (e.g., associated with a Positive Sequence Load Flow ("PSLF") or Transient Security Assessment Tool ("TSAT")). At Step **415**, model validation may compare measurements to default model response. If the response matches the measurements, the framework may end (e.g., the existing model is sufficiently correct and does not need to be updated). At Step **420**, an event analysis algorithm may determine if event is qualitatively different from previous events. At Step **425**, a parameter identifiability analysis algorithm may determine most identifiable set of parameters across all events of interest. For example, a first event may have 90 to 100 parameters. For that event, Step **425** uses the parameter identifiability algorithm to select 1 to 20 of those parameters.

[0091] Finally, at Step **430** an Unscented Kalman Filter ("UKF")/optimization-based parameter estimation algorithm/process may be performed. As a result, the estimated parameter values, confidence metrics, and error in model response (as compared to measurements) may be reported. In some embodiments, Steps **405-415** are considered model validation **435** and Steps **420-430** are considered model calibration **440**. As described elsewhere herein, the systems may use one or both of model validation **435** and model calibration **440**. In some embodiments, Steps **405-430** are considered a model validation and calibration (MVC) process **400**.

[0092] Disturbance data may be monitored by one or more PMUs coupled to an electrical power distribution grid may be received. The disturbance data can include voltage ("V"), frequency ("f"), and/or active and nonactive reactive ("P", and "Q") power measurements from one or more points of interest (POI) on the electrical power grid. A power system model may include model parameters. These model parameters may be the current parameters incorporated in the power system model. The current parameters may be stored in a model parameter record. Model calibration involves identifying a subset of parameters that can be "tuned" and modifying/adjusting the parameters such that the power system model behaves identically or almost identically to the actual power component being represented by the power system model.

[0093] In accordance with some embodiments, the model calibration can implement model calibration with three functionalities. The first functionality is an event screening tool to select characteristics of a disturbance event from a library of recorded event data. This functionality can simulate the power system responses when the power system is subjected to different disturbances. The second functionality is a parameter identifiability study. When implementing this functionality, the can simulate the response(s) of a power system model. The third functionality is simultaneous tuning of models using event data to adjust the identified model parameters. According to various embodiments, the second functionality (parameter identifiability) and the third functionality (tuning of model parameters) may be done using a surrogate model in place of a dynamic simulation engine **316**.

[0094] Here, the model calibration algorithm attempts to find a parameter value (θ^*) for a parameter (or parameters) of the power system model that creates a matching output between the simulated active power ($\hat{P}$) and the simulated reactive power ($\hat{Q}$) predicted by the model with respect to the actual active power (P) and actual reactive power (Q) of the component on the electrical grid.

[0095] As grid disturbances occur intermittently, the user of the calibration tool described herein may be required to re-calibrate model parameters in a sequential manner as new disturbances come in. In this scenario, the user has a model that was calibrated to some observed grid disturbances to start with, and observes a larger that acceptable mismatch with a newly encountered disturbance. The task now is to tweak the model parameters so that the model explains the new disturbance without detrimentally affecting the match with earlier disturbances. One solution would be to run calibration simultaneously on all events of interest strung together, but this comes at the cost of significant computational expense and engineering involved in enabling running a batch of events simultaneously. Instead, it may be desirable to carry some essential information from the earlier calibrations runs and guide the subsequent calibration run that helps explain the new disturbance without losing earlier calibration matches.

[0096] Event screening can be implemented during the simulation to provide computational efficiency. If hundreds of events are stitched together and fed into the calibration algorithm unselectively, the algorithm may not be able to converge. To maintain the number of events manageable and still keep an acceptable representation of all the events, a screening procedure may be performed to select the most characteristic events among all. Depending on the type of events, the measurement data could have different characteristics. For example, if an event is a local oscillation, the oscillation frequency in the measurement data would be much faster as compared to an inter-area oscillation event. In some implementations, a K-medoids clustering algorithm can be utilized to group events with similar characteristic together, thus reducing the number of events to be calibrated.

[0097] Instead of using the time consuming simulation engine, the surrogate model or models (such as Neural Networks) with equivalent function of dynamic simulation engine, may be used for both identifiability and calibration. The surrogate model may be built offline while there is no request for model calibration. Once built, the surrogate model comprising a set of weights and bias in learned

structure of network will be used to predict the active power ($\hat{P}$) and reactive ($\hat{Q}$) given different set of parameters together with time stamped voltage (V) and frequency (f).

[0098] The parameter identifiability analysis addresses two aspects: (a) magnitude of sensitivity of output to parameter change; and (b) dependencies among different parameter sensitivities. For example, if the sensitivity magnitude of a particular parameter is low, the parameter would appear in a row being close to zero in the parameter estimation problem's Jacobian matrix. Also, if some of the parameter sensitivities have dependencies, it reflects that there is a linear dependence among the corresponding rows of the Jacobian. Both these scenarios lead to singularity of the Jacobian matrix, making the estimation problem infeasible. Therefore, it may be important to select a subset of parameters which are highly sensitive as well as result in no dependencies among parameter sensitivities. Once the subset of parameters is identified, values in the active power system model for the parameters may be updated, and the system may generate a report and/or display of the estimated parameter values(s), confidence metrics, and the model error response as compared to measured data.

[0099] FIG. **5** illustrates a process **500** for performing optimization using an objective function at least in part by using an integrated acquisition function and a probabilistic model of the objective function, in accordance with some embodiments. Process **500** may be used to identify the best or optimal generator model parameters, as well as a hyperparameter in either the parameter identifiability algorithm **425** or the parameter estimation algorithm **430** (shown in FIG. **4**), which contributes to achieving a global minimum of the objective function. For the purposes of this disclosure, a hyperparameter is a parameter whose value is used to control the learning process. The hyperparameter for the parameter identifiability algorithm **425** may be the threshold for the single value decomposition (SVD) approach and a dot product angle (DPA). The hyperparameter for the parameter estimation algorithm **430** may be the maximum number of iterations, algorithm types (Levenberg-Marquardt algorithm, Gauss-Newton algorithm, Trust Region algorithm, Kalman filter algorithm, particle swarm optimization algorithm, differential evolution algorithm and Bayesian Optimization), residual tolerance, etc. The objective function maps the parameter or hyperparameter to performance or accuracy of the model prediction compared to the real measurement.

[0100] Process **400** begins at Step **502**, where a probabilistic model of the objective function is initialized. In some embodiments, the probabilistic model of the objective function may comprise a Gaussian process, a neural network, and an adaptive basis function regression model (linear or non-linear).

[0101] Next, process **400** proceeds to Step **504**, where a parameter or hyperparameter at which to evaluate the objective function is identified. The identification may be performed, at least in part, by using an acquisition utility function and a probabilistic model of the objective function. In some embodiments, an acquisition utility function that depends on parameters of the probabilistic model may be used at Step **504** such as, for example, a probability of improvement acquisition utility function, an expected improvement acquisition utility function, a regret minimization acquisition utility function, and an entropy-based acquisition utility function.

[0102] In some embodiments, the point at which to evaluate the objective function may be identified as the point (or as approximation to the point) at which the acquisition utility function attains its maximum value. In some embodiments, Markov chain Monte Carlo methods may be used to identify or approximate the point at which the integrated acquisition utility function attains its maximum value.

[0103] After the point at which to evaluate the objective function is identified in Step **504**, process **500** proceeds to Step **506**, where the objective function is evaluated at the identified point. Then process **500** proceeds to Step **508**, where the probabilistic model of the objective function is updated based on results of the evaluation. The probabilistic model of the objective function may be updated in any of numerous ways based on results of the new evaluation obtained in Step **506**. As one non-limiting example, updating the probabilistic model of the objective function may comprise updating (e.g., re-estimating) one or more parameters of the probabilistic model based on results of the evaluation performed in Step **506**. As another non-limiting example, updating the probabilistic model of the objective function may comprise updating the covariance kernel of the probabilistic model (e.g., when the probabilistic model comprises a Gaussian process, the covariance kernel of the Gaussian process may be updated based on results of the new evaluation).

[0104] Process **500** proceeds to decision block **510**, where it is determined whether the objective function is to be evaluated at another point, also known as the terminating criteria. This includes a threshold number of evaluations of the objective function or stagnation where the values of the objective function have not increased by more than a threshold value of iterations, such as 4+[D/2], where D is the number of parameters to be estimated.

[0105] When it is determined, at decision block **510**, that the objective function is to be evaluated again, process **500** returns, via the YES branch, to Step **504**, so that Steps **504-508** are repeated. On the other hand, when it is determined at decision block **510** that the objective function is not to be evaluated again, process **500** proceeds to Step **512**, where an extremal value of the objective function may be identified based on the one or more values of the objective function obtained during process **500**.

[0106] The Bayesian Optimization constructs a prior distribution about $f(x)$ based on input and output values of the function, and updates the distribution iteratively with new values derived by the Bayesian Optimization. For example, new input values to black-box function are derived from the prior distribution of input and output values, in an acquisition function optimization. The new input values are then used to evaluate the black-box function to generate a new output to be included in the prior distribution of values for a next iteration of the optimization. The process is repeated until a termination criteria is met (e.g., the input values to the black-box function are optimized within a desired threshold, or a maximum number of iterations, specified by the user, have been reached).

[0107] FIG. **6** illustrates a process **600** for sequential calibration using the system architecture **300** (shown in FIG. **3**). In the exemplary embodiment, the system **300** receives a plurality of events **308** (shown in FIG. **3**) and events **602**, **610**, and **614**. In some embodiments, process **600** is performed by one or more of the system architecture **300**, the

processor **1410**, and the power system disturbance based model calibration engine **1414** (both shown in FIG. **14**).

[0108] In the exemplary embodiment, process **600** receives initial parameters **604** and choses a first event **602**. In some embodiments, the first event **602** is one of the received plurality of events. In other embodiments, the first event **602** is a historical event or an event designated for testing purposes. The first event **602** and the initial parameters **604** are used as inputs for a model validation and calibration (MVC) process **606**, also known as MVC engine **606**. In the exemplary embodiment, the MVC process **606** is similar to the MVC **400**. In the exemplary embodiment, the first event **602** includes at least the actual voltage, frequency, active power, reactive power for the event. The MVC process **606** generates a first updated set of parameters **608** based on how the initial parameters **604** matched up with the first event **602**. In some embodiments, the MVC process **606** uses the initial parameters **604** and the voltage and frequency to predict the active and reactive power for the first event **602**. Then the MVC process **606** compares the predicted active and reactive power to the actual active and reactive power for the first event **602**. The MVC process **606** adjusts the initial parameters **604** based on that comparison to generate an updated parameter set **608**.

[0109] In process **600**, the first updated set of parameters **608** are then used with a second event **610** as inputs into the MVC process **606** to generate a second updated set of parameters **612**. The second updated set of parameters **612** and then used with a third event **614** to be another set of inputs for the MVC process **606** to generate a third updated set of parameters **616**.

[0110] In the exemplary embodiment, the process **600** continues to serially analyze events to generate updated parameter sets. For example, if the process **600** receives 25 events, then each event will be analyzed in order to determine updated parameters based on that event and MVC process **606**, with the goal being that the parameters allow the MVC process **606** to generate adjusted parameters to accurately predict the outcome of the plurality of events.

[0111] By analyzing each event individually and serially rather than as a group or in parallel, process **600** allows for the parameters that affect each event to be analyzed, rather than have events that cancel out the effect of different parameters. For example, considering three different events, event-1, event-2, event-3, the sequential approach shown in process **600** will generated three down-selected parameters subsets, say P-1, P-2 and P-3, corresponding to the three events. Each parameter subset is determined to be the best subset which can describe the corresponding event based on the parameter identifiability algorithm **425**. Then the parameter subset P-1, P-2, P-3 may be further used for the parameter estimation process **430** based on the corresponding event. However, the parameter identifiability in a group calibration approach may not reach such an optimality. Furthermore, as the important parameters are identified for each event, and the parameters for each of these events are analyzed overall for the entire set of events. In this way, the parameters for each event contribute to the final parameters and allow the system to find the ideal parameters for the entire set while still taking into account each individual event.

[0112] FIG. **7** is a data flow diagram illustrating a subsection **700** of the system architecture **300** (shown in FIG. **3**) executing the sequential calibration process **600** (shown in FIG. **6**). In the exemplary embodiment, the system architecture **700** receives network models **302**, sub-system definitions **304**, dynamic models **306**, and event data **308** at an input handling component **710**. In some embodiments, input handling component **710** includes an event screening component.

[0113] The network models **302**, sub-system definitions **304**, dynamic models **306**, and event data **308** are analyzed by the system **700** as described herein. In the exemplary embodiment disclosed herein, the model utilizes multiple disturbance events to validate and calibrate power system models for compliance with NERC mandated grid reliability requirements.

[0114] In some embodiments, the user accesses the user interface **738** to set the total number of events **308** that will be used in process **600**, set the stored file locations, and set the sequence that the events **308** will be analyzed in.

[0115] In the exemplary embodiment, system **700** includes a set of initial parameters **712**. In some embodiments, the set of initial parameters **712** are based on the dynamic model **706**. The initial parameters **712** and a first event **714** are set as inputs and a model validation and calibration (MVC) **716** is performed using those parameters **712** and that first event **714**. In some embodiments, the MVC **716** is performed by the simulation engine **316** (shown in FIG. **3**). In some embodiments, the MVC **716** is associated with the MVC process **606** (shown in FIG. **6**) and/or the MVC process **400** (shown in FIG. **4**). The MVC **716** generates a response **718**, which includes statistics about how the initial parameters **712** performed in matching up to the first event **714** based on the MVC process **606**. The MVC **716** also generates a first set of updated parameters **720** based on the event's performance in the MVC process **606**.

[0116] In some embodiments, the MVC **716** uses the initial parameters **712** and the voltage and frequency of the first event **714** to predict the active and reactive power for the first event **714**. Then, the MVC **716** compares the predicted active and reactive power to the actual active and reactive power for the first event **714**. The MVC **716** adjusts the parameters **712** into the first set of updated parameters **720** based on that comparison and also uses the comparison to generate the first response **718**.

[0117] In the exemplary embodiment, the system **700** uses the first set of updated parameters **720** with the second event **722** into the MVC process **606** to generate a second updated set of parameters **728** and a second response **726**. The second updated set of parameters **728** is then used with a third event **730** to be another set of inputs for the MVC process **606** to generate a third updated set of parameters **736** and a third response **734**.

[0118] In the exemplary embodiment, the system **700** continues to serially analyze events **308** to generate updated parameter sets. For example, if the system **700** receives 25 events **308**, then each event **308** will be analyzed in order to determine updated parameters based on that event **308** and the MVC process **606**, with the goal being that the parameters allow the MVC process **606** to generate adjusted parameters to accurately predict the outcome of the plurality of events.

[0119] In some embodiments, the user may use the user interface **738** to review the responses and the updated parameters. Furthermore, the user interface **738** may allow the user to determine the order that the events **308** are analyzed. In other embodiments, the system **700** may seri-

ally analyze the events **308** in a plurality of orders to determine the ideal set of updated parameters.

[0120] FIG. **8** illustrates a process **800** for using Bayesian Optimization to optimize model parameters in accordance with the process **400** (shown in FIG. **4**). Process **800** may be executed by system **300** (shown in FIG. **3**) and platform **1400** (shown in FIG. **14**).

[0121] At Step **405**, disturbance data may be obtained (e.g., from a PMU or DFR) to obtain, for example, V, f, P, and Q measurement data at a Point Of Interest ("POI"). At Step **410**, a playback simulation may run load model benchmarking using default model parameters (e.g., associated with a Positive Sequence Load Flow ("PSLF") or Transient Security Assessment Tool ("TSAT")). At Step **415**, model validation may compare measurements to default model response. If the response matches the measurements, the framework may end (e.g., the existing model is sufficiently correct and does not need to be updated). At Step **420**, an event analysis algorithm may determine if event is qualitatively different from previous events. At Step **425**, a parameter identifiability analysis algorithm may determine most identifiable set of parameters across all events of interest. For example, a first event may have 90 to 100 parameters. For that event, Step **425** uses the parameter identifiability algorithm to select 1 to 10 of those parameters.

[0122] Finally, Step **430** (shown in FIG. **4**) is replaced with Bayesian optimization **805**. The Bayesian optimization **805** performs well in problems for functions with a small number of dimensions (e.g., less than 10 unknown variables), but may not scale well to higher dimensions. In the exemplary embodiment, the parameter selected for Bayesian optimization should be less than 10, and preferably 1~5. The parameter identifiability analysis may be single value decomposition approach, Dot Product Angle (DPA), user selection, etc.

[0123] Note the Bayesian optimization **805** in this approach is configured to estimate parameters of dynamic models (e.g., gains, transfer functions, integrators, derivative, time constants, limiters, saturation constants, dead zones, delay).

[0124] Events are situations where the voltage and/or the frequency of the power system changes. For each event, the event screening component determines whether the event is novel enough. For example, an event may be a generator turning on. If the event has the same or similar attributes to a previous event, such as that same generator turning on, then the event screening component skips this event. In the exemplary embodiment, the event screening component compares the event to those events stored in a database. If the event is novel enough, then the event is stored in the database. Then the event is sent to the parameter identifiability component. This component analyzes the event in combination with past events and the parameters identified as significant with those events to determine which parameters are significant for this event. Then the tunable parameters are transmitted to the Bayesian Optimization component, which further analyzes the significant parameters to calibrate the parameters in the model being executed by the simulation engine.

[0125] Disturbance data may be monitored by one or more PMUs coupled to an electrical power distribution grid may be received. The disturbance data can include voltage ("V"), frequency ("f"), and/or active and nonactive reactive ("P" and "Q") power measurements from one or more points of interest (POI) on the electrical power grid. A power system model may include model parameters. These model parameters can be the current parameters incorporated in the power system model. The current parameters can be stored in a model parameter record. Model calibration involves identifying a subset of parameters that can be "tuned" and modifying/adjusting the parameters such that the power system model behaves identically or almost identically to the actual power component being represented by the power system model.

[0126] In accordance with some embodiments, the model calibration can implement model calibration with three functionalities. The first functionality is an event screening tool to select characteristics of a disturbance event from a library of recorded event data. This functionality may simulate the power system responses when the power system is subjected to different disturbances. The second functionality is a parameter identifiability study. This functionality may simulate the response(s) of a power system model. The third functionality is simultaneous tuning of models using event data to adjust the identified model parameters. According to various embodiments, the second functionality (parameter identifiability) and the third functionality (tuning of model parameters) may be implemented using a surrogate model in place of a dynamic simulation engine.

[0127] Instead of using the time consuming simulation engine, the surrogate model or models (such as Neural Networks) with equivalent function of dynamic simulation engine, may be used for both identifiability and calibration. The surrogate model may be built offline when there is no request for model calibration. Once built, the surrogate model includes a set of weights and bias in a learned structure of the network will be used to predict the active power ($\hat{P}$) and reactive ($\hat{Q}$) given different set of parameters together with time stamped voltage (V) and frequency (f).

[0128] The parameter identifiability analysis addresses two aspects: (a) magnitude of sensitivity of output to parameter change; and (b) dependencies among different parameter sensitivities. For example, if the sensitivity magnitude of a particular parameter is low, the parameter would appear in a row being close to zero in the parameter estimation problem's Jacobian matrix. Also, if some of the parameter sensitivities have dependencies, it reflects that there is a linear dependence among the corresponding rows of the Jacobian. Both these scenarios lead to singularity of the Jacobian matrix, making the estimation problem infeasible. Therefore, it may be important to select a subset of parameters which are highly sensitive as well as result in no dependencies among parameter sensitivities. Once the subset of parameters is identified, values in the active power system model for the parameters may be updated, and the system may generate a report and/or display of the estimated parameter values(s), confidence metrics, and the model error response as compared to measured data.

[0129] In FIG. **8**, parameter identifiability analysis algorithm **425** may be performed to generate a trajectory sensitivities matrix for an electrical power system using a dynamic model of the electrical power system that includes a plurality of system parameters. Two embodiments for parameter identifiability are singular-value decomposition (SVD) based approach and Dot Product Angle (DPA) based approach.

[0130] "SVD," as used herein, refers to a matrix decomposition method for reducing a matrix to its constituent

parts. For example, by reducing a matrix to its constituent parts, certain subsequent matrix calculations may be simplified. For example, SVD includes a factorization of a real or complex matrix. SVD includes a generalization of an eigen-decomposition of a positive semidefinite normal matrix (e.g., a symmetric matrix with positive eigenvalues) to any m×n matrix via an extension of polar decomposition. SVD has many useful applications in signal processing and statistics, for example.

[0131] "DPA," as used herein, refers to an algebraic operation that takes two equal-length sequences of numbers, such as, e.g., coordinate vectors, and returns a single number. In Euclidean geometry, a dot product of Cartesian coordinates of two vectors commonly used and is often referred to as "the" inner product (or rarely projection product) of Euclidean space even though it is not the only inner product that can be defined on Euclidean space. Algebraically, a dot product is the sum of the products of the corresponding entries of the two sequences of numbers. Geometrically, it is the product of the Euclidean magnitudes of the two vectors and the cosine of the angle between them. These definitions are equivalent when using Cartesian coordinates. In modern geometry, Euclidean spaces are often defined by using vector spaces. In this case, the dot product is used for defining lengths (e.g., the length of a vector is the square root of the dot product of the vector by itself) and angles (e.g., the cosine of the angle of two vectors is the quotient of their dot product by the product of their lengths).

[0132] In one particular embodiment, an issue of parameter identifiability may be considered or addressed. For example, a relatively simple linear 2-parameter estimation problem may include:

$$y = C\begin{bmatrix}\theta_1\\ \theta_2\end{bmatrix}, \text{ with } C = [\mathcal{C}\mathcal{C}] \rightarrow y = \mathcal{C}(\theta_1 + \theta_2) \qquad \text{EQ. 16}$$

[0133] In Equation 16, if $(\theta_1{}^*+\theta_2{}^*)$ is a true solution $(\tilde{c}(\theta_1{}^*+\theta_2{}^*)=y^m)$, then any $\theta_1{}^*-\delta$, $\theta_2{}^*+\delta$) equally explain the measurements, for example. A failure to identify parameters uniquely may be due to a rank deficiency of output matrix C. An analogous quantity n a nonlinear case may comprise a Jacobian matrix as shown below in Equation 17:

$$S = \begin{bmatrix} \frac{dP(t_1)}{d\theta_1} & \frac{dP(t_1)}{d\theta_2} & \cdots & \frac{dP(t_1)}{d\theta_k} \\ \frac{dP(t_2)}{d\theta_1} & \frac{dP(t_2)}{d\theta_2} & \cdots & \frac{dP(t_2)}{d\theta_k} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{dP(t_N)}{d\theta_1} & \frac{dP(t_N)}{d\theta_2} & \cdots & \frac{dP(t_N)}{d\theta_k} \\ \frac{dQ(t_1)}{d\theta_1} & \frac{dQ(t_1)}{d\theta_2} & \cdots & \frac{dQ(t_1)}{d\theta_k} \\ \frac{dQ(t_2)}{d\theta_1} & \frac{dQ(t_2)}{d\theta_2} & \cdots & \frac{dQ(t_2)}{d\theta_k} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{dQ(t_N)}{d\theta_1} & \frac{dQ(t_N)}{d\theta_2} & \cdots & \frac{dQ(t_N)}{d\theta_k} \end{bmatrix} \qquad \text{EQ. 17}$$

[0134] A rank deficiency of Jacobian matrix S may result from (a) a relatively small number of entries in columns of S; and/or (b) columns of Jacobian matrix S being nearly linearly dependent. Such factors may show the following, qualitatively: (a) low parameter sensitivity—a successful estimation of that parameter is unlikely because its effect cannot be observed; and/or (b) a nearly linear dependency—a successful estimation of these parameters may therefore be unlikely because of the individual parameter effects. Moreover, a presence of parameters with weak and/or nearly linearly dependent effects may be reflected as non-unique solutions. Accordingly, it is important to determine the right set of parameters to be tuned.

[0135] In accordance with one particular example of parameter identifiability for multiple events, an average identifiability ranking across disturbances may be calculated. Because sensitivity studies are conducted at the parameters' default values, for example, a parameter conditioning tool may also perform a global sensitivity consistency study when the parameters' values deviate far away from their default values. Such a study may portray a geometry of the parameter sensitivity in the entire parameter space, for example.

[0136] Different events may have different characteristics, such that conventional identifiability analysis corresponding to each single event may not be applicable to other events. For example, a set of most-identifiable parameters for event A may not be identifiable for event B. Accordingly, for a single event calibration, the value of this set of parameters may only be tuned by a conventional approach to make the output match event A's measurement data. However, if the tuned parameter values are used to simulate event B, there may still be discrepancy between simulation output from the power system model and measurement data from PMUs.

[0137] In accordance with embodiments, because there is availability of measurement data from multiple events, a comprehensive identifiability analysis or study across multiple events may be performed. Such a comprehensive study may provide a most-identifiable parameter set for simultaneous calibration of multiple disturbances. In accordance with embodiments, this parameter set may be used to tune a power system model to better match (as compared to conventionally-tuned power system models) measurement data of multiple events simultaneously.

[0138] If a quantity of N events is considered, applying singular-value decomposition (SVD) analysis to the sensitivity trajectory matrices may result in a quantity of null spaces equal to the value of N. The null space for one event also may be interpreted as a system of homogeneous algebraic equations with parameter sensitivities being the unknowns. Because the null space from one event has a rank lower than the number of parameters, the number of equations is less than the number of unknowns.

[0139] Considering more events is equivalent to adding more equations to the system. After the event number N exceeds a certain value, the system would have more equations than unknowns. Characteristics of events should be diverse in accordance with an embodiment in order to tune parameters of the system. In an implementation, a numerical rank should be greater than the number of unknowns. A solution which minimizes the difference between the left and right hands of the equation system may represent a comprehensive sensitivity magnitude of all parameters across all the considered events. For sensitivity dependency, accounting for the null spaces of all considered events, a comprehensive dependency index may also be calculated.

[0140] In accordance with one or more embodiments, if the number of events is not large enough to construct a null space with higher rank than the number of parameters, the identifiability for each single event may be analyzed, and then the average identifiability may be used as the identifiability across all events.

[0141] In accordance with one or more embodiments, a model calibration algorithm may implement the sample Bayesian Optimization algorithm described above to perform a sensitivity dependency calculation using a null space of the trajectory sensitivity matrix to calculate sensitivity dependency. The dependency index may be defined by counting the large elements in the right singular vectors in null space.

[0142] Another parameter identifiability approach is Dot Product Angle (DPA) based approach. The performance of a parameter identifiability analysis may analyze parameters to identify potential parameters for use based on the dot product (or scalar product) of the columns of J and r as defined below. In the exemplary embodiment, r comprises a residual which comprises the difference between the measured response data series and the simulated response data series where:

$$r(\theta)=y_t^m-y_t(\theta) \quad \text{EQ. 18}$$

where y_t^m is the measured response of active and reactive power provided in event data, $y_t(x)$ is the simulated response of active and reactive power based on dynamic simulation engine, including but not limited to, General Electric®'s PSLF, Siemens® PTI's PSS/E, etc. In Equation 18, θ represents the model parameters.

[0143] An equivalent expression for the above residual is the sum of squares (SOS) objective: $\|r(x)\|_2^2$. The parameter identifiability analysis may use the Quadratic Model (QM) of the objective at (θ_k+d) to approximate the residual at the next step $r(\theta_{k+1})$.

$$QM(J_k, r_k, d)=\|r(\theta)+J_k d\|_2^2 \quad \text{EQ. 19}$$

where J_k is the Jacobian vector, which is equal to

$$J_k = \frac{dr}{d\theta}\Big|_{\theta k},$$

and $r_k=r(\theta_k)$ is the sensitivity result. This leads to:

$$r(\theta_{k+1})=r(\theta_k)+J_k(\theta_k+d) \quad \text{EQ. 20}$$

[0144] The ultimate goal is to get $r(\theta_{k+1})$ equal to zero. This leads to:

$$r(\theta_k)=-J_k(\theta_k\ d) \quad \text{EQ. 21}$$

[0145] In the exemplary embodiment, the vector $r(\theta_k)$ is compared to the Jacobian vector J_k to determine the θ between them. In some embodiments, each vector J_k may have up to 1000 values, where the number of values in the Jacobian vector depends on the number of sampling points in the event. The θ is calculated by generating the dot product of the vector $r(\theta_k)$to the Jacobian vector J_k.

$$r(\theta_k)^* J_k=\|r(\theta_k)\|\|J_k\|\cos\alpha \quad \text{EQ. 22}$$

[0146] The resulting dot product angle α is compared to a threshold. Parameters with a corresponding α below the threshold are sent to the pool of parameters that are selected. The ideal α is zero, but that is generally unachievable. In some embodiments, any parameter with an angle α of less than 5° is selected by of parameter identifiability analysis. This threshold may be configurable by the user, such as through an interactive user interface. The threshold may be treated as a hyperparameter to be tuned by the Bayesian optimization. A key idea is that the more orthogonal the angles are between the vectors of J and r, the less likely changes to that parameter moves the response in the desired way. This approach may be extended to a weighted version, by scaling both the measured response and simulated response with a weight vector w_t. The weight factor w_t has the same length of the data samples in the event of interest. In this way, given a defined weight factor, it can affect the above calculated dot product angles are between the vectors of J and r, and hence the parameter screening result.

[0147] Given the down selected parameter subset generated from Step **425**, the Bayesian optimization may be utilized to tune the value of down selected parameter subset to improve the residual defined in the optimization objective function. For example, a first event may have 90 to 100 parameters. For that event, Step **425** uses the parameter identifiability algorithm **425** to select 1 to 10 of those parameters. Bayesian optimization performs well in problems for functions with a small number of dimensions (e.g., less than 10 unknown variables), but does not scale well to higher dimensions. The parameter selected for Bayesian optimization should be less than 10, and preferably 1~5. Note Bayesian optimization **805** in FIG. **8** is configured to estimate dynamic model parameters (e.g., gains, transfer functions, integrators, derivative, time constants, limiters, saturation constants, dead zones, delay).

[0148] The process of using Bayesian optimization to tune the dynamic model parameters is given. The process begins from an initialization of probabilistic model of the objective function, such as the objective function defined in EQ. 7, EQ. 8 or EQ. 9. In some embodiments, the probabilistic model of the objective function may comprise a Gaussian process, a neural network, adaptive basis function regression model (linear or non-linear).

[0149] Next, a dynamic model parameter to evaluate the objective function is identified. The identification may be performed, at least in part by, using an acquisition utility function and a probabilistic model of the objective function. In some embodiments, an acquisition utility function that depends on parameters of the probabilistic model may be used such as, for example, a probability of improvement acquisition utility function, an expected improvement acquisition utility function, a regret minimization acquisition utility function, and an entropy-based acquisition utility function.

[0150] In some embodiments, the point at which to evaluate the objective function may be identified as the point (or as approximation to the point) at which the acquisition utility function attains its maximum value. In some embodiments, Markov chain Monte Carlo methods may be used to identify or approximate the point at which the integrated acquisition utility function attains its maximum value.

[0151] After the point at which the objective function is identified, the probabilistic model of the objective function is updated based on results of the evaluation. The probabilistic model of the objective function may be updated in any of numerous ways based on results of the new evaluation of the objective function at the identified point. In one non-limiting example, updating the probabilistic model of the objective function may comprise updating the covariance

kernel of the probabilistic model (e.g., when the probabilistic model comprises a Gaussian process, the covariance kernel of the Gaussian process may be updated based on results of the new evaluation).

[0152] The above process may be repeated until it meets the terminating criteria, including stagnation wherein the values of the objective function have not increased by more than a threshold value of iterations, such as 4+[D/2], D is the number of parameters to be estimated. Once the terminating criteria are met, the optimal value for the dynamic model parameters may be generated and stored for users' review.

[0153] FIG. **9** illustrates a process **900** for using Bayesian Optimization to optimize parameter identifiability analysis in accordance with the process **400** (shown in FIG. **4**). Process **900** is similar to process **800** (shown in FIG. **8**) and based on process **400**.

[0154] Process **900** is configured to optimize not only the generator model parameters (e.g., gains, transfer functions, integrators, derivative, time constants, limiters, saturation constants, dead zones, delay, etc.), but also the hyperparameter in the parameter identifiability algorithm **425**, including threshold for the SVD or angle. The parameter estimation algorithm **430** in this case may be Kalman filter, non-linear least square optimization solver. The hyper parameter may also include the max number of iterations, algorithm type (Levenberg-Marquardt algorithm, Gauss-Newton algorithm, Trust Region algorithm, Kalman filter algorithm, particle swarm optimization algorithm, differential evolution algorithm and Bayesian Optimization), residual tolerance, and weight in objective functions in the parameter estimation algorithm **430**. As another embodiment, the parameter to be estimated in Bayesian Optimization **805** may be a combination of both a parameter and a hyperparameter. In the exemplary embodiment the hyperparameter will affect the algorithm performance of the parameter identifiability analysis algorithm **425** and the parameter estimation algorithm **430**, but not the model itself. In some embodiments, the hyperparameters include the weight parameters w as described above. In these embodiments, the Bayesian Optimization **805** is used to find the ideal weights for one or more parameters.

[0155] In process **900**, the Bayesian Optimization **805** oversees the parameter identifiability analysis algorithm **425** and the parameter estimation algorithm **430**.

[0156] In some embodiments, the Bayesian Optimization **805** may also replace the parameter estimation algorithm **430** as shown in FIG. **8**. In these embodiments, the Bayesian Optimization **805** analyzes both the parameters and the hyperparameter.

[0157] FIG. **10** illustrates a process **1000** for using Bayesian Optimization to optimize a hyperparameter in accordance with the process **400** (shown in FIG. **4**). The process **1000** is similar to the process **900** (shown in FIG. **9**) and based on the process **400**.

[0158] The process **1000** is configured to optimize not only the generator model parameters (e.g., gains, transfer functions, integrators, derivative, time constants, limiters, saturation constants, dead zones, delay), but also the hyper parameter in the parameter estimation algorithm **430**, including max number of iterations, algorithm type (Levenberg-Marquardt algorithm, Gauss-Newton algorithm, Trust Region algorithm, Kalman filter algorithm, particle swarm optimization algorithm, differential evolution algorithm and Bayesian Optimization), and residual tolerance in the parameter estimation algorithm. In some embodiments, the hyperparameters include the weight parameters w as described above. In these embodiments, the Bayesian Optimization **805** is used to find the ideal weights for one or more parameters.

[0159] FIG. **11** illustrates a process **1100** for using Bayesian Optimization to optimize event sequences for sequential model calibration, such as shown in the process **600** (shown in FIG. **6**). The process **1100** may be executed by the system **300** (shown in FIG. **3**), the system **700** (shown in FIG. **7**), and the platform **1400** (shown in FIG. **14**).

[0160] In the process **1100**, a Bayesian Optimization component **1105** is configured to optimize the sequence of events for the sequential model calibration process **600** (shown in FIG. **6**). The Bayesian Optimization component **1105** uses the best fitting error and the average fitting error to determine the optimal event sequence. In the exemplary embodiment, the system **700** analyzes a first sequence of events, such as event 1 **602**, event 2 **610**, and event 3 **614**. The system **700** then calculates the average fitting error (also known as average prediction residual) or the best fitting error from the analysis of the sequence. The average fitting error may be calculated by performing model validation **435** (shown in FIG. **4**) over the three events **602**, **610**, and **614** with the third updated set of parameters **616**. The best fitting error may be calculated by determining the minimum over all fitting error. Based on the average fitting error or the best fitting error, the Bayesian The optimization component **1105** determines the optimal event sequence for analysis. In some embodiments, the system **700** then analyzes the events **602**, **610**, and **614** in that sequence to get the parameter set. This parameter set is used to calculate the average fitting error and/or the best fitting error. If the calculated average fitting error and/or the best fitting error meets a threshold, then process **1100** ends. Otherwise the Bayesian Optimization component **1105** is called to determine another event sequence for analysis and the process **1100** is re-executed. The process **1100** may be continually executed until a terminating condition is reached, such as a minimum fitting error across all of the events **602**, **610**, and **614**.

[0161] In some further embodiments, the Bayesian Optimization component **1105** is used to determine the optimal initial values **604** that least to the least fitting error. In some additional embodiments, the Bayesian Optimization component **1105** is used to both determine the ideal event sequence and the optimal initial values **604**.

[0162] FIG. **12** illustrates candidate parameter estimation algorithms **1200** according to some embodiments. In one approach **1220**, measured input/output data **1210** (u, y^m) may be used by a power system component model **1222** and an UKF based approach **1224** to create an estimation parameter (p*) **1240**.

[0163] In particular, the system may compute sigma points based on covariance and standard deviation information. The Kalman Gain matrix K may be computed based on $\hat{Y}$ and the parameters may be updated based on:

$$p_k = p_{k-1} + K(y^m - \hat{y})$$

until p_k converges. According to another approach **1230**, the measured input/output data **1210** (u, y^m) may be used by a power system component model **1232** and an optimization-based approach **1234** to create the estimation parameter (p*) **1240**. In this case, the following optimization problem may be solved:

$$\min_{p}\left\|y^{m}-\hat{Y}(p)\right\|^{2}$$

[0164] The system may then compute output as compared to parameter Jacobian information and iteratively solve the above optimization problem by moving parameters in directions indicated by the Jacobian information.

[0165] FIG. **13** illustrates a two-stage approach of the process for model calibration. In this approach, PMU data from events is fed into a dynamic simulation engine. The dynamic simulation engine communicates with a parameter identifiability analysis component and returns the changes to the parameters. The parameter identifiability analysis component also transmits a set of identifiable parameters to a model calibration algorithm component. The model calibration algorithm component uses the set of identifiable parameters, PMU data from events, and other data from the dynamic simulation engine to generate estimated parameters. This approach may be used to calibrate the tuning model parameters.

[0166] With the playback simulation capability, the user can compare the response (active power and reactive power) of system models with dynamics observed during disturbances in the system, which is called model validation. The grid disturbance (aka. events) can also be used to correct the system model when simulated response is significantly different from the measured values, which is called model calibration. As shown in the right side of FIG. **14**, the goal is to achieve a satisfactory match between the measurement data and simulated response. If obvious a discrepancy is observed, then the model calibration process may be employed.

[0167] The first step of the model calibration process is parameter identification, which aims to identify a subset of parameters with strong sensitivity to the observed event. In the exemplary embodiment, the model calibration process requires a balance on matching in measurement space and reasonableness in the model parameter space. Numerical curve fitting without adequate engineering guidance tends to provide overfitted parameter results, and leads to non-unique sets of parameters (leading to same curve fitting performance), which should be avoided.

[0168] The embodiments described herein may also be implemented using any number of different hardware configurations. For example, FIG. **14** is a block diagram of an apparatus or platform **1400** that may be, for example, associated with the system **200** of FIG. **2** and/or any other system described herein. The platform **1400** includes a processor **1410**, such as one or more commercially available Central Processing Units ("CPUs") in the form of one-chip microprocessors, coupled to a communication device **1420** configured to communicate via a communication network (not shown in FIG. **14**). The communication device **1420** may be used to communicate, for example, with one or more remote measurement units, components, user interfaces, etc. The platform **1400** further includes an input device **1440** (e.g., a computer mouse and/or keyboard to input power grid and/or modeling information) and/an output device **1450** (e.g., a computer monitor to render a display, provide alerts, transmit recommendations, and/or create reports). According to some embodiments, a mobile device, monitoring physical system, and/or PC may be used to exchange information with the platform **1400**.

[0169] The processor **1410** also communicates with a storage device **1430**. The storage device **1430** may include any appropriate information storage device, including combinations of magnetic storage devices (e.g., a hard disk drive), optical storage devices, mobile telephones, and/or semiconductor memory devices. The storage device **1430** stores a program **1412** and/or a power system disturbance based model calibration engine **1414** for controlling the processor **1410**. The processor **1410** performs instructions of the programs **1412**, **1414**, and thereby operates in accordance with any of the embodiments described herein. For example, the processor **1410** may calibrate a dynamic simulation engine, having system parameters, associated with a component of an electrical power system (e.g., a generator, wind turbine, etc.). The processor **1410** may receive, from a measurement data store **1460**, measurement data measured by an electrical power system measurement unit (e.g., a phasor measurement unit, digital fault recorder, or other means of measuring frequency, voltage, current, or power phasors). The processor **1410** may then pre-condition the measurement data and set-up an optimization problem based on a result of the pre-conditioning. The system parameters of the dynamic simulation engine may be determined by solving the optimization problem with an iterative method until at least one convergence criteria is met. According to some embodiments, solving the optimization problem includes a Jacobian approximation that does not call the dynamic simulation engine if an improvement of residual meets a pre-defined criterion.

[0170] The programs **1412**, **1414** may be stored in a compressed, uncompiled and/or encrypted format. The programs **1412**, **1414** may furthermore include other program elements, such as an operating system, clipboard application, a database management system, and/or device drivers used by the processor **1410** to interface with peripheral devices.

[0171] As used herein, information may be "received" by or "transmitted" to, for example: (i) the platform **1400** from another device; or (ii) a software application or module within the platform **1400** from another software application, module, or any other source.

[0172] In some other embodiments, the system **700** (shown in FIG. **7**) stores a model of a device, such as generator **110**. The model includes a plurality of parameters. The system **700** receives a plurality of events **602**, **610**, and **614** (shown in FIG. **6**) associated with the device. In some embodiments, the events **602**, **610**, and **614** include sensor information of the event **602**, **610**, and **614** occurring at the device. In other embodiments, the sensor information is associated with a similar device.

[0173] The system **700** also receives a first set of input calibration values **604** (shown in FIG. **6**) for the plurality of parameters. The system **700** sequentially analyzes the plurality of events **602**, **610**, and **614** in a first sequence to determine a set of calibrated parameter values **616** (shown in FIG. **6**) for the model. The system **700** validates **435** (shown in FIG. **4**) the set of calibrated parameter values **616** for the model to determine fit. The system **700** then performs Bayesian optimization **1105** (shown in FIG. **11**) on the determined fit, the set of calibrated parameter values **616** for the model, and the plurality of events **602**, **610**, and **614**.

[0174] In some embodiments, the system **700** determines a second sequence of events based on the Bayesian optimization **1105**. The system **700** sequentially analyzes the

plurality of events **602**, **610**, and **614** based on the second sequence to determine a second fit. The system **700** performs Bayesian optimization **1105** on the second fit, the set of calibrated parameter values **616** for the model, and the plurality of events **602**, **610**, and **614** to determine a third sequence. The system **700** sequentially analyzes the plurality of events based **602**, **610**, and **614** on the third sequence.

[0175] In other embodiments, the system **700** determines a second set of input calibration values **604** based on the Bayesian optimization **1105**. The system **700** sequentially analyzes the plurality of events **602**, **610**, and **614** based on the second set of input calibration values **604** to determine a second fit. The system **700** performs Bayesian optimization on the second fit, the set of calibrated parameter values **616** for the model, and the plurality of events **602**, **610**, and **614** to determine a third set of input calibration values **604**. The system **700** sequentially analyzes the plurality of events **602**, **610**, and **614** based on the third set of input calibration values **604**.

[0176] In some embodiments, the system **700** compares the fit to a terminating condition. When the terminating condition is reached, the system **700** updates the model to include the set of calibrated parameter values **616**.

[0177] In some embodiments, the fit is based on an average fitting error of the set of calibrated parameter values **616** across the plurality of events **602**, **610**, and **614**. In other embodiments, the fit is based on a best fitting error of the set of calibrated parameter values **616** across the plurality of events **602**, **610**, and **614**.

[0178] In some embodiments, the model is a power system model and the Bayesian optimization maintains a probabilistic surrogate model and an acquisition function. In these embodiments, the system **700** initializes the probabilistic surrogate model of an objective function using a plurality of initial parameter points. The probabilistic surrogate model of the objective function includes a stationary probabilistic model including a non-linear one-to-one mapping of values of the plurality of parameters from a first domain to a second domain. The system **700** also generates a new set of parameter values corresponding to at least one parameter of the plurality of parameters by optimizing an acquisition function. The acquisition function is based at least in part on the set of calibrated parameter values and the probabilistic surrogate model of the objective function. The system **700** further evaluates the objective function using the power system model operated with the new set of parameter values. In addition, the system **700** updates the probabilistic surrogate model of the objective function to obtain an updated probabilistic surrogate model of the objective function. Moreover, the system **700** repeats until reaching at least one of a predetermined number of iterations, a predetermined period of time, and a termination condition.

[0179] In the exemplary embodiment, the system **300** (shown in FIG. **3**) stores a model of a device. The model includes a plurality of parameters. The system **300** receives a first event **308** associated with the device. The system **300** analyzes the first event **308** to identify a subset of important parameters from the plurality of parameters. The system **300** performs Bayesian optimization **805** (shown in FIG. **8**) on the subset of important parameters to determine a set of calibrated parameter values for the model.

[0180] In some embodiments, the system **300** analyzes the first event using at least one of a single value decomposition (SVD) approach and a dot product angle (DPA) approach. In some embodiments, the subset of important parameters includes less than ten parameters.

[0181] In some further embodiments, the system **300** receives a second event **308** associated with the device. The system **300** analyzes the second event **308** to determine a second subset of important parameters from the plurality of parameters based on the set of calibrated parameter values. The system **300** performs Bayesian optimization **805** on the second subset of important parameters to determine a second set of calibrated parameter values for the model.

[0182] In the exemplary embodiment, the system **300** stores a model of a device. The model includes a plurality of parameters. The system **300** receive a first event **308** associated with the device. The system **300** analyzes the first event **308** to identify a subset of important parameters from the plurality of parameters. The system **300** determines at least one hyperparameter based on the analysis. The system **300** performs Bayesian optimization **805** on the hyperparameter.

[0183] In some embodiments, the system **300** analyzes the first event **308** using at least one of a single value decomposition (SVD) approach and a dot product angle (DPA) approach. In some embodiments, the at least one hyperparameter includes at least one of a maximum number of iterations, a residual tolerance, and one or more parameter weights.

[0184] In some further embodiments, the system **300** reanalyzes the first event **308** to identify the subset of important parameters from the plurality of parameters based on the hyperparameter. The system **300** determines a set of calibrated parameter values for the model based on the subset of important parameters.

[0185] In still further embodiments, the system **300** determines a set of calibrated parameter values for the model based on the subset of important parameters and the hyperparameter. In other embodiments, the system **300** performs Bayesian optimization **805** on the subset of important parameters to determine a set of calibrated parameter values for the model.

[0186] At least one of the technical solutions to the technical problems provided by this system may include: (i) improved speed in modeling parameters; (ii) more robust models in response to measurement noise; (iii) compliance with NERC mandated grid reliability requirements; (iv) reduce the chance that an important parameter is not updated; (v) improved accuracy in parameter identifiability; (vi) improved accuracy in parameter estimation; and (vii) improved optimization of parameters based on event training.

[0187] The methods and systems described herein may be implemented using computer programming or engineering techniques including computer software, firmware, hardware, or any combination or subset thereof, wherein the technical effects may be achieved by performing at least one of the following steps: a) store a model of a device, wherein the model includes a plurality of parameters; b) receive a plurality of events associated with the device; c) receive a first set of input calibration values for the plurality of parameters; d) sequentially analyze the plurality of events in a first sequence to determine a set of calibrated parameter values for the model; e) validate the set of calibrated parameter values for the model to determine fit; f) perform Bayesian optimization on the determined fit, the set of calibrated parameter values for the model, and the plurality

of events; g) determine a second sequence of events based on the Bayesian optimization; h) sequentially analyze the plurality of events based on the second sequence to determine a second fit; i) perform Bayesian optimization on the second fit, the set of calibrated parameter values for the model, and the plurality of events to determine a third sequence; j) sequentially analyze the plurality of events based on the third sequence; k) determine a second set of input calibration values based on the Bayesian optimization; l) sequentially analyze the plurality of events based on the second set of input calibration values to determine a second fit; m) perform Bayesian optimization on the second fit, the set of calibrated parameter values for the model, and the plurality of events to determine a third set of input calibration values; n) sequentially analyze the plurality of events based on the third set of input calibration values; o) compare the fit to a terminating condition; and p) when the terminating condition is reached, update the model to include the set of calibrated parameter values.

[0188] In other embodiments, the technical effects may be achieved by performing at least one of the following steps: a) store a model of a device, wherein the model includes a plurality of parameters; b) receive a first event associated with the device; c) analyze the first event to identify a subset of important parameters from the plurality of parameters, wherein the subset of important parameters includes less than ten parameters; d) perform Bayesian optimization on the subset of important parameters to determine a set of calibrated parameter values for the model; e) analyze the first event using at least one of a single value decomposition approach and a dot product angle approach; f) receive a second event associated with the device; g) analyze the second event to determine a second subset of important parameters from the plurality of parameters based on the set of calibrated parameter values; and h) perform Bayesian optimization on the second subset of important parameters to determine a second set of calibrated parameter values for the model.

[0189] In still other embodiments, the technical effects may be achieved by performing at least one of the following steps: a) store a model of a device, wherein the model includes a plurality of parameters; b) receive a first event associated with the device; c) analyze the first event to identify a subset of important parameters from the plurality of parameters; d) determine at least one hyperparameter based on the analysis, wherein the at least one hyperparameter includes at least one of a maximum number of iterations, a residual tolerance, and one or more parameter weights; e) perform Bayesian optimization on the hyperparameter; f) analyze the first event using at least one of a single value decomposition approach and a dot product angle approach; g) reanalyze the first event to identify the subset of important parameters from the plurality of parameters based on the hyperparameter; h) determine a set of calibrated parameter values for the model based on the subset of important parameters; i) determine a set of calibrated parameter values for the model based on the subset of important parameters and the hyperparameter; and j) perform Bayesian optimization on the subset of important parameters to determine a set of calibrated parameter values for the model.

[0190] The computer-implemented methods discussed herein may include additional, less, or alternate actions, including those discussed elsewhere herein. The methods may be implemented via one or more local or remote processors, transceivers, servers, and/or sensors, and/or via computer-executable instructions stored on non-transitory computer-readable media or medium.

[0191] Additionally, the computer systems discussed herein may include additional, less, or alternate functionality, including that discussed elsewhere herein. The computer systems discussed herein may include or be implemented via computer-executable instructions stored on non-transitory computer-readable media or medium.

[0192] A processor or a processing element may employ artificial intelligence and/or be trained using supervised or unsupervised machine learning, and the machine learning program may employ a neural network, which may be a convolutional neural network, a deep learning neural network, or a combined learning module or program that learns in two or more fields or areas of interest. Machine learning may involve identifying and recognizing patterns in existing data in order to facilitate making predictions for subsequent data. Models may be created based upon example inputs in order to make valid and reliable predictions for novel inputs.

[0193] Additionally or alternatively, the machine learning programs may be trained by inputting sample data sets or certain data into the programs, such as image data, text data, report data, and/or numerical analysis. The machine learning programs may utilize deep learning algorithms that may be primarily focused on pattern recognition, and may be trained after processing multiple examples. The machine learning programs may include Bayesian program learning (BPL), voice recognition and synthesis, image or object recognition, optical character recognition, and/or natural language processing—either individually or in combination. The machine learning programs may also include natural language processing, semantic analysis, automatic reasoning, and/or machine learning.

[0194] In supervised machine learning, a processing element may be provided with example inputs and their associated outputs, and may seek to discover a general rule that maps inputs to outputs, so that when subsequent novel inputs are provided the processing element may, based upon the discovered rule, accurately predict the correct output. In unsupervised machine learning, the processing element may be required to find its own structure in unlabeled example inputs. In one embodiment, machine learning techniques may be used to extract data about the computer device, the user of the computer device, the computer network hosting the computer device, services executing on the computer device, and/or other data.

[0195] Based upon these analyses, the processing element may learn how to identify characteristics and patterns that may then be applied to training models, analyzing sensor data, and detecting abnormalities.

[0196] As will be appreciated based upon the foregoing specification, the above-described embodiments of the disclosure may be implemented using computer programming or engineering techniques including computer software, firmware, hardware or any combination or subset thereof. Any such resulting program, having computer-readable code means, may be embodied or provided within one or more computer-readable media, thereby making a computer program product, i.e., an article of manufacture, according to the discussed embodiments of the disclosure. The computer-readable media may be, for example, but is not limited to, a fixed (hard) drive, diskette, optical disk, magnetic tape,

semiconductor memory such as read-only memory (ROM), and/or any transmitting/receiving medium, such as the Internet or other communication network or link. The article of manufacture containing the computer code may be made and/or used by executing the code directly from one medium, by copying the code from one medium to another medium, or by transmitting the code over a network.

[0197] These computer programs (also known as programs, software, software applications, "apps", or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms "machine-readable medium" and "computer-readable medium" refer to any computer program product, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The "machine-readable medium" and "computer-readable medium," however, do not include transitory signals. The term "machine-readable signal" refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0198] As used herein, a processor may include any programmable system including systems using micro-controllers, reduced instruction set circuits (RISC), application specific integrated circuits (ASICs), logic circuits, and any other circuit or processor capable of executing the functions described herein. The above examples are example only, and are thus not intended to limit in any way the definition and/or meaning of the term "processor."

[0199] As used herein, the terms "software" and "firmware" are interchangeable, and include any computer program stored in memory for execution by a processor, including RAM memory, ROM memory, EPROM memory, EEPROM memory, and non-volatile RAM (NVRAM) memory. The above memory types are example only, and are thus not limiting as to the types of memory usable for storage of a computer program.

[0200] In another embodiment, a computer program is provided, and the program is embodied on a computer-readable medium. In an example embodiment, the system is executed on a single computer system, without requiring a connection to a server computer. In a further example embodiment, the system is being run in a Windows® environment (Windows is a registered trademark of Microsoft Corporation, Redmond, Wash.). In yet another embodiment, the system is run on a mainframe environment and a UNIX® server environment (UNIX is a registered trademark of X/Open Company Limited located in Reading, Berkshire, United Kingdom). In a further embodiment, the system is run on an iOS® environment (iOS is a registered trademark of Cisco Systems, Inc. located in San Jose, Calif.). In yet a further embodiment, the system is run on a Mac OS® environment (Mac OS is a registered trademark of Apple Inc. located in Cupertino, Calif.). In still yet a further embodiment, the system is run on Android® OS (Android is a registered trademark of Google, Inc. of Mountain View, Calif.). In another embodiment, the system is run on Linux® OS (Linux is a registered trademark of Linus Torvalds of Boston, Mass.). The application is flexible and designed to run in various different environments without compromising any major functionality.

[0201] In some embodiments, the system includes multiple components distributed among a plurality of computer devices. One or more components may be in the form of computer-executable instructions embodied in a computer-readable medium. The systems and processes are not limited to the specific embodiments described herein. In addition, components of each system and each process can be practiced independent and separate from other components and processes described herein. Each component and process can also be used in combination with other assembly packages and processes. The present embodiments may enhance the functionality and functioning of computers and/or computer systems.

[0202] As used herein, an element or step recited in the singular and preceded by the word "a" or "an" should be understood as not excluding plural elements or steps, unless such exclusion is explicitly recited. Furthermore, references to "example embodiment," "exemplary embodiment," or "one embodiment" of the present disclosure are not intended to be interpreted as excluding the existence of additional embodiments that also incorporate the recited features.

[0203] The patent claims at the end of this document are not intended to be construed under 35 U.S.C. § 112(f) unless traditional means-plus-function language is expressly recited, such as "means for" or "step for" language being expressly recited in the claim(s).

[0204] This written description uses examples to disclose the disclosure, including the best mode, and also to enable any person skilled in the art to practice the disclosure, including making and using any devices or systems and performing any incorporated methods. The patentable scope of the disclosure is defined by the claims, and may include other examples that occur to those skilled in the art. Such other examples are intended to be within the scope of the claims if they have structural elements that do not differ from the literal language of the claims, or if they include equivalent structural elements with insubstantial differences from the literal language of the claims.

In the claims:

1. A system for power system model calibration comprising a computing device comprising at least one processor in communication with at least one memory device, wherein said at least one processor is programmed to:

store a model of a device, wherein the model includes a plurality of parameters;

receive a plurality of events associated with the device;

receive a first set of input calibration values for the plurality of parameters;

sequentially analyze the plurality of events in a first sequence to determine a set of calibrated parameter values for the model;

validate the set of calibrated parameter values for the model to determine a fit; and

perform Bayesian optimization on the determined fit, the set of calibrated parameter values for the model, and the plurality of events.

2. The system in accordance with claim **1**, wherein the model is a power system model, wherein the Bayesian optimization maintains a probabilistic surrogate model and an acquisition function, and wherein to perform Bayesian optimization said at least one processor is further programmed to:

initialize the probabilistic surrogate model of an objective function using a plurality of initial parameter points,

wherein the probabilistic surrogate model of the objective function comprises a stationary probabilistic model including a non-linear one-to-one mapping of values of the plurality of parameters from a first domain to a second domain;

generate a new set of parameter values corresponding to at least one parameter of the plurality of parameters by optimizing an acquisition function, wherein the acquisition function is based at least in part on the set of calibrated parameter values and the probabilistic surrogate model of the objective function;

evaluate the objective function using the power system model operated with the new set of parameter values;

update the probabilistic surrogate model of the objective function to obtain an updated probabilistic surrogate model of the objective function; and

repeat until reaching at least one of a predetermined number of iterations, a predetermined period of time, and a termination condition.

3. The system in accordance with claim **1**, wherein said at least one processor is further programmed to determine a second sequence of events based on the Bayesian optimization.

4. The system in accordance with claim **3**, wherein said at least one processor is further programmed to sequentially analyze the plurality of events based on the second sequence to determine a second fit.

5. The system in accordance with claim **4**, wherein said at least one processor is further programmed to:

perform Bayesian optimization on the second fit, the set of calibrated parameter values for the model, and the plurality of events to determine a third sequence; and

sequentially analyze the plurality of events based on the third sequence.

6. The system in accordance with claim **1**, wherein said at least one processor is further programmed to determine a second set of input calibration values based on the Bayesian optimization.

7. The system in accordance with claim **6**, wherein said at least one processor is further programmed to sequentially analyze the plurality of events based on the second set of input calibration values to determine a second fit.

8. The system in accordance with claim **7**, wherein said at least one processor is further programmed to:

perform Bayesian optimization on the second fit, the set of calibrated parameter values for the model, and the plurality of events to determine a third set of input calibration values; and

sequentially analyze the plurality of events based on the third set of input calibration values.

9. The system in accordance with claim **1**, wherein said at least one processor is further programmed to:

compare the fit to a terminating condition; and

when the terminating condition is reached, update the model to include the set of calibrated parameter values.

10. The system in accordance with claim **1**, wherein the fit is based on one of an average fitting error of the set of calibrated parameter values across the plurality of events and a best fitting error of the set of calibrated parameter values across the plurality of events.

11. A system for power system model calibration comprising a computing device comprising at least one processor in communication with at least one memory device, wherein said at least one processor is programmed to:

store a model of a device, wherein the model includes a plurality of parameters;

receive a first event associated with the device;

analyze the first event to identify a subset of important parameters from the plurality of parameters; and

perform Bayesian optimization on the subset of important parameters to determine a set of calibrated parameter values for the model.

12. The system in accordance with claim **11**, wherein said at least one processor is further programmed to analyze the first event using at least one of a single value decomposition approach and a dot product angle approach.

13. The system in accordance with claim **11**, wherein the subset of important parameters includes less than ten parameters.

14. The system in accordance with claim **11**, wherein said at least one processor is further programmed to:

receive a second event associated with the device;

analyze the second event to determine a second subset of important parameters from the plurality of parameters based on the set of calibrated parameter values; and

perform Bayesian optimization on the second subset of important parameters to determine a second set of calibrated parameter values for the model.

15. A system for power system model calibration comprising a computing device comprising at least one processor in communication with at least one memory device, wherein said at least one processor is programmed to:

store a model of a device, wherein the model includes a plurality of parameters;

receive a first event associated with the device;

analyze the first event to identify a subset of important parameters from the plurality of parameters;

determine at least one hyperparameter based on the analysis; and

perform Bayesian optimization on the hyperparameter.

16. The system in accordance with claim **15**, wherein said at least one processor is further programmed to analyze the first event using at least one of a single value decomposition approach and a dot product angle approach.

17. The system in accordance with claim **15**, wherein said at least one hyperparameter includes at least one of a maximum number of iterations, a residual tolerance, and one or more parameter weights.

18. The system in accordance with claim **15**, wherein said at least one processor is further programmed to:

re-analyze the first event to identify the subset of important parameters from the plurality of parameters based on the hyperparameter; and

determine a set of calibrated parameter values for the model based on the subset of important parameters.

19. The system in accordance with claim **15**, wherein said at least one processor is further programmed to determine a set of calibrated parameter values for the model based on the subset of important parameters and the hyperparameter.

20. The system in accordance with claim **15**, wherein said at least one processor is further programmed to perform Bayesian optimization on the subset of important parameters to determine a set of calibrated parameter values for the model.

* * * * *