United States Patent	12386328
Kind Code	B2
Date of Patent	August 12, 2025
Inventor(s)	Feng; Xiaoming

Parameterized potential functions for equality constraints

Abstract

Prior methods of state estimation, based on a constrained optimization problem with equality and/or inequality constraints, rely on penalty-based heuristics which can produce very large weight values, resulting in ill-conditioning of the gain matrix. Disclosed embodiments of state estimation convert the constrained optimization problem into an unconstrained convex optimization problem in which violated equality and/or inequality constraints are represented as parameterized potential functions, each comprising a center-of-attraction parameter. This unconstrained convex optimization problem can be iteratively formed, using successively updated values for the center-of-attraction parameters, and solved, until no equality and/or inequality constraints are violated, to produce a final estimated state. This final estimated state may then be used to control the system being monitored, such as a power system.

Inventors:	**Feng; Xiaoming (Cary, NC)**
Applicant:	**Hitachi Energy Ltd** (Zürich, CH)
Family ID:	**1000008748360**
Assignee:	**HITACHI ENERGY LTD (Zürich, CH)**
Appl. No.:	**17/879414**
Filed:	**August 02, 2022**

Prior Publication Data

Document Identifier	Publication Date
US 20240045391 A1	Feb. 08, 2024

Publication Classification

Int. Cl.: **G05B19/042** (20060101); **G06F17/11** (20060101); **H02J13/00** (20060101)

U.S. Cl.:

CPC   **G05B19/042** (20130101); **G06F17/11** (20130101); **H02J13/00002** (20200101); H02J2203/20 (20200101)

Field of Classification Search

CPC: G05B (19/02); H02J (13/00002); H02J (2203/20); G06F (17/11)

USPC: 700/291

References Cited

U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
8738191	12/2013	Aivaliotis	700/297	H02J 3/00
2012/0317060	12/2011	Jebara	706/21	G06N 20/00
2021/0143638	12/2020	Cavraro	N/A	G06Q 50/06
2021/0218273	12/2020	Feng	N/A	H02J 3/003
2023/0072795	12/2022	Zeng	N/A	H02J 13/00034

FOREIGN PATENT DOCUMENTS

Patent No.	Application Date	Country	CPC
109390929	12/2018	CN	N/A
113139288	12/2020	CN	N/A
114417414	12/2021	CN	N/A

OTHER PUBLICATIONS

Clements et al., "A New Method for Solving Equality-Constrained Power System Static-State Estimation", IEEE Transactions on Power Systems, vol. 5, No. 4, Nov. 1990, pp. 1260-1266. cited by applicant

Nucera et al., "A Blocked Sparse Matrix Formulation for the Solution of Equality-Constrained State Estimation", IEEE Transactions on Power Systems, vol. 6, No. 1, Feb. 1991, pp. 214-224. cited by applicant

Clements et al. "Power System State Estimation With Inequality Constraints", Worcester, MA USA, CH 3006-4/91/0000-0990@IEEE, pp. 990-993. cited by applicant

Abur, et al., "Least Absolute Value State Estimation with Equality and Inequality Constraints", IEEE Transactions on Power Systems, vol. 8, No. 2, May 1993, pp. 680-686. cited by applicant

Clements et al., "Treatment of Inequality Constraints in Power System State Estimation", IEEE Transactions on Power Systems, vol. 10, No. 2, May 1995, pp. 567-574. cited by applicant

Singh et al., "Constrained LAV State Estimation Using Penalty Functions", on with Equality and Inequality Constraints, IEEE Transactions on Power Systems, vol. 12, No. 1, Feb. 1997, pp. 383-388. cited by applicant

Korres, "A Robust Method for Equality Constrained State Estimation", IEEE Transactions on Power Systems, vol. 17, No. 2, May 2002, pp. 305-314. cited by applicant

De La Villa Jaen, et al, "Implicitly Constrained Substation Model for State Estimation", IEEE Transactions on Power Systems, vol. 17, No. 3, Aug. 2002, pp. 850-856. cited by applicant

Lin et al., "A Decentralized Step-size Technique Used to Solve Large Scale Constrained State Estimation Problem", 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China, 6 pages. cited by applicant

Khokhlov, "Constrained Power System State Estimation on Recurrent Neural Networks", Power Tech, 2005, IEEE Xplore, 7 pages. cited by applicant

Hewett et al., "A Robust Null Space Method for Linear Equality Constrained State Estimation", IEEE

Transactions on Signal Processing, vol. 58, No. 8, Aug. 2010, pp. 3961-3971. cited by applicant

Valverde et al., "A Constrained Formulation for Hybrid State Estimation", IEEE Transactions on Power Systems, vol. 26, No. 3, Aug. 2011, pp. 1102-1109. cited by applicant

Korres et al., "A Constrained Ordering for Solving the Equality Constrained State Estimation", IEEE Transactions on Power Systems, vol. 27, No. 4, Nov. 2012, pp. 1998-2005. cited by applicant

Gomez-Quiles et al., "Equality-Constrained Bilinear State Estimation", IEEE Transactions on Power Systems, vol. 28, No. 2, May 2013, pp. 902-910. cited by applicant

Pires et al., "Constrained Robust Estimation of Power System State Variables and Transformer Tap Positions Under Erroneous Zero-Injections", IEEE Transactions on Power Systems, vol. 29, No. 3, May 2014, pp. 1144-1152. cited by applicant

Kumagai et al., "Constrained State Estimation with Partial Measurements in Unobservable Distribution Network", IEEE Xplore, 2015, 5 pages. cited by applicant

Sodhi et al., "ICS: Incremental Constrained Smoothing for State Estimation", 2020 IEEE International Conference on Robotics and Automation (ICRA), May 31-Aug. 31, 2020. Paris, France, pp. 279-285. cited by applicant

Vanin et al., "A Framework for Constrained Static State Estimation in Unbalanced Distribution Networks", IEEE Transactions on Power Systems, 2021, 11 pages. cited by applicant

Wu et al., "Iterative Method of Maximum likelihood for State Estimation with Inequality Constraints", 2012 International Conference on Computer Science and Electronics Engineering, pp. 576-579. cited by applicant

---

Background/Summary

BACKGROUND
Field of the Invention
(1) The embodiments described herein are generally directed to state estimation, and, more particularly, to state estimation for a power system using parameterized potential functions for equality constraints, and optionally, for inequality constraints as well.
Description of the Related Art
(2) State estimation (SE) is used in management systems to estimate the most likely state of a network of a power system (e.g., power grid) from a sparely redundant set of measurements of nodal and branch quantities (e.g., nodal voltage phasors, power injections, and branch power flows). State estimation is necessary for complex systems in which the state cannot be directly observed and/or measurements of the state are prone to noise or corruption. The state of each node in the network of a power system may be defined, for example, by a nodal voltage magnitude and a nodal phase angle. Examples of management systems that utilize state estimation include, without limitation, a supervisory control and data acquisition (SCADA) system, an energy management system (EMS), a distribution management system (DMS), advanced DMS (ADMS), and the like.
(3) Conventionally, state estimation is solved either as an unconstrained non-linear weighted least squares (WLS) problem or as an equality-constrained non-linear WLS problem, to avoid numerical ill-conditioning for nodes (e.g., buses) with zero power injection. In these cases, the solution does not enforce feasibility constraints on the power injection. For instance, a distributed energy resource (DER) with a maximum output of 10 kilowatts (KW) may be estimated, in the solution, to produce 12 KW. Such conditions may occur in state estimation for both transmission and distribution systems, but are more likely to occur in state estimation for distribution systems due to lower telemetry redundancy and lower data quality. Thus, management systems would benefit from state estimation that constrains

nodal metrics (e.g., power output) to feasible values.

SUMMARY

(4) Accordingly, systems, methods, and non-transitory computer-readable media are disclosed for state estimation for a power system, using parameterized potential functions (e.g., quadratic functions) for at least equality constraints. An objective of embodiments is to constrain the solution to an SE problem to feasible values for nodal metrics by satisfying all equality constraints within a tolerance. A further objective of embodiments is to estimate a state for a system, such as a power system, that can be used to optimize and control the system.

(5) In an embodiment of a method of state estimation for a power system, the method comprises using at least one hardware processor to: acquire a constrained optimization problem comprising a convex first objective function to be minimized subject to one or more equality constraints; convert the constrained optimization problem into an unconstrained convex optimization problem comprising a second objective function that is a sum of the first objective function and a parameterized potential function for each of at least one of the one or more equality constraints, wherein each parameterized potential function is defined by an equality function and a center-of-attraction parameter; and perform a solution process comprising solving the unconstrained convex optimization problem by finding an input to the unconstrained convex optimization problem that minimizes an output of the unconstrained convex optimization problem, wherein one or more center-of-attraction parameters are updated when solving the unconstrained convex optimization problem to ensure that the input satisfies the one or more equality constraints within a tolerance, and wherein the input represents an estimated state of the power system. The method may further comprise using the at least one hardware processor to control the power system based on the estimated state. The method may further comprise providing the estimated state of the power system through a human-to-machine interface of an energy management system. The method may further comprise using the estimated state in one or more of a contingency analysis, a Volt-Var optimization, or optimal power flow, such as distributed energy resource management. The first objective function may comprise an error calculation in which a value of a measurement function, given the input, is subtracted from a value of a system telemetry.

(6) Each parametric potential function may be defined as:

$$w(c(x)-u).\text{sup}.2$$

wherein w is a weight, x is the input, c(x) is the equality function, and u is the center-of-attraction parameter. The second objective function may be defined as:

(7) $$f(x) + \underset{i}{.\text{Math.}} \, w_i(c_i(x) - u_i)^2$$

wherein $f(x)$ is the first objective function, and i is an indexed set of equality constraints for each of which a parameterized potential function is included in the unconstrained convex optimization problem.

(8) The method may further comprise using the at least one hardware processor to perform a first process comprising: performing an initial iteration of the solution process to find an initial optimizing input, wherein the initial iteration of the solution process further comprises initializing a value of each center-of-attraction parameter in each parameterized potential function in the unconstrained convex optimization problem; determining whether or not the one or more equality constraints are violated by the initial optimizing input; when determining that the one or more equality constraints are not violated, outputting the initial optimizing input as an output estimated state of the power system; and when determining that at least one of the one or more equality constraints is violated, until the one or more equality constraints are not violated, over one or more subsequent iterations, performing the solution process, wherein the subsequent iteration of the solution process further comprises updating the value of each center-of-attraction parameter in each parameterized potential function corresponding to the one or more equality constraints, prior to solving the unconstrained convex optimization problem, and outputting the input found by the solution process in a final one of the one or more subsequent iterations as the output estimated state of the power system. The method may further comprise using the at least one hardware processor to control the power system based on the output estimated state. In an embodiment, the one or more equality constraints are determined to be violated

when at least one of the one or more equality constraints is outside of the tolerance, and wherein the one or more equality constraints are determined not to be violated when all of the one or more equality constraints are within the tolerance. In the initial iteration and each of the one or more subsequent iterations, the unconstrained convex optimization problem may comprise a parameterized potential function for all of the one or more equality constraints. In each of the one or more subsequent iterations, an initial input used in solving the unconstrained convex optimization problem in the solution process of the subsequent iteration may be the input found by the solution process in an immediately preceding iteration.

(9) In each of the one or more subsequent iterations, updating the value of each center-of-attraction parameter may comprise calculating the value of each center-of-attraction parameter based on a prior value of that center-of-attraction parameter in an immediately preceding iteration. In each of the one or more subsequent iterations, updating the value of each center-of-attraction parameter may comprise calculating the value of each center-of-attraction parameter further based on a value of the equality function for a corresponding one of the one or more equality constraints given the input found by the solution process in an immediately preceding iteration. The value of each center-of-attraction parameter in a subsequent iteration k may be calculated as follows:

$$u.sup.(k) = u.sup.(k-1) - c(x.sup.(k-1))$$

wherein u.sup.(k) is the value of the center-of-attraction parameter in the subsequent iteration k, u.sup.(k-1) is the value of the center-of-attraction parameter in an immediately preceding iteration k−1, x.sup.(k-1) is the input found in the immediately preceding iteration k−1, and c(x.sup.(k-1)) is a value of the equality function in the parameterized potential function comprising the center-of-attraction parameter, given the input x.sup.(k-1).

(10) The constrained convex optimization problem may further comprise one or more inequality constraints, and the method may further comprise using the at least one hardware processor to: perform the first process, while ignoring the one or more inequality constraints, to find the output estimated state as a first initial optimizing input; determine whether or not the one or more inequality constraints are violated by the first initial optimizing input; when determining that the one or more inequality constraints are not violated, use the first initial optimizing input as a final estimated state of the power system; when determining that at least one of the one or more inequality constraints is violated, until the one or more inequality constraints are not violated, over one or more outer iterations, convert the constrained optimization problem into an inequality-unconstrained convex optimization problem comprising the first objective function and an inequality parameterized potential function for each of at least one of the one or more inequality constraints, wherein each inequality parameterized potential function is defined by an inequality function and a center-of-attraction parameter, determine a value of each center-of-attraction parameter in each inequality parameterized potential function in the inequality-unconstrained convex optimization problem, and perform the first process using the inequality-unconstrained convex optimization problem as the constrained optimization problem in the first process, and output the input found by the first process in a final one of the one or more outer iterations as the final estimated state of the power system. In each of the one or more outer iterations, an inequality parameterized potential function may be added to the inequality-unconstrained convex optimization problem for each of the one or more inequality constraints that are violated and none of the one or more inequality constraints that have not been violated. For an inequality constraint expressed as g(x)≤0, each inequality parameterized potential function component may be defined as:

$$w(g(x)-u).sup.2$$

wherein w is a weight, x is the input, g(x) is the inequality function, and u is the center-of-attraction parameter.

(11) In an embodiment of a method of state estimation for a power system, the method comprises using at least one hardware processor to: acquire a constrained convex optimization problem comprising a first objective function to be minimized subject to one or more equality constraints; convert the constrained convex optimization problem into an unconstrained convex optimization problem comprising a second the objective function that is a sum of the first objective function and a parameterized potential function for each of the one or more equality constraints, wherein each

parameterized potential function is defined by an equality function and a center-of-attraction parameter; define a joint system of equations comprising a first order optimality condition of the unconstrained convex optimization problem and the one or more equality constraints with associated weighting; and determine an estimated state of the power system by solving the joint system of equations using a root-finding algorithm. The method may further comprise using the at least one hardware processor to control the power system based on the estimated state.

(12) It should be understood that any of the features in the methods above may be implemented individually or with any subset of the other features in any combination. Thus, to the extent that the appended claims would suggest particular dependencies between features, disclosed embodiments are not limited to these particular dependencies. Rather, any of the features described herein may be combined with any other feature described herein, or implemented without any one or more other features described herein, in any combination of features whatsoever. In addition, any of the methods, described above and elsewhere herein, may be embodied, individually or in any combination, in executable software modules of a processor-based system, such as a server, and/or in executable instructions stored in a non-transitory computer-readable medium.

Description

BRIEF DESCRIPTION OF THE DRAWINGS

(1) The details of the present invention, both as to its structure and operation, may be gleaned in part by study of the accompanying drawings, in which like reference numerals refer to like parts, and in which:

(2) FIG. **1** illustrates an example infrastructure, in which one or more of the processes described herein, may be implemented, according to an embodiment;

(3) FIG. **2** illustrates an example processing system, by which one or more of the processes described herein, may be executed, according to an embodiment;

(4) FIG. **3** illustrates an example data flow between systems and software modules within the example infrastructure, according to an embodiment;

(5) FIG. **4** illustrates the operation of a virtual measurement that utilizes a center-of-attraction parameter, according to an embodiment;

(6) FIG. **5** illustrates a solution process, according to an embodiment;

(7) FIG. **6** illustrates an overall algorithm for solving a constrained optimization problem, according to an embodiment;

(8) FIG. **7** illustrates an overall algorithm for solving a constrained optimization problem, comprising both equality constraint(s) and inequality constraint(s), according to an embodiment; and

(9) FIG. **8** illustrates an algorithm for solving a constrained optimization problem, comprising equality constraints, according to an embodiment.

DETAILED DESCRIPTION

(10) In an embodiment, systems, methods, and non-transitory computer-readable media are disclosed for state estimation for a power system, using parameterized potential functions for equality constraints, and potentially, for inequality constraints as well. It is generally contemplated that the state estimation will be performed for a power system, such as a power grid (e.g., at any scale, from a large-scale utility grid to a microgrid or smaller). A power system may be any network of electrical components (e.g., power system equipment) configured to generate, store, supply, transmit, distribute, and/or consume electrical power, including, without limitation, power stations configured to produce electricity from combustible fuels (e.g., coal, natural gas, etc.) and/or renewable resources (e.g., wind, solar, nuclear, etc.), transmission systems configured to carry or transmit electricity from sources (e.g., generators) to loads, and distribution systems configured to feed supplied electricity to nearby homes, businesses, and/or other establishments. However, the disclosed approach is not limited to power systems. Rather, the disclosed approach may be applied to any system whose state is estimated using a WLS problem with one or more equality constraints.

(11) As used herein, the term "network" refers to the interconnection of components within a system whose state is to be estimated. In the case of a power system, these components will generally include electrical components, such as power generators, distributed energy resources (e.g., renewable energy sources, battery energy storage (BES) systems, etc.), loads, transformers, transmission and distribution lines, and/or the like. It should be understood that other types of systems may be similarly represented as a network.

(12) As used herein, the term "node" or "bus" generally refers to any point in the network at which a state is to be estimated when solving the SE problem. In the case of a power system, the state of each node may be defined as a voltage magnitude and phase angle at the node. However, the state of a node in a power system could be defined in other manners. It should be understood that the states of nodes in other types of systems will be defined in a manner that is appropriate for that type of system.

(13) After reading this description, it will become apparent to one skilled in the art how to implement the invention in various alternative embodiments and alternative applications. However, although various embodiments of the present invention will be described herein, it is understood that these embodiments are presented by way of example and illustration only, and not limitation. As such, this detailed description of various embodiments should not be construed to limit the scope or breadth of the present invention as set forth in the appended claims.

1. System Overview

1.1. Infrastructure

(14) FIG. **1** illustrates an example infrastructure in which one or more of the disclosed processes may be implemented, according to an embodiment. The infrastructure may comprise a management system **110** (e.g., comprising one or more servers) which hosts and/or executes one or more of the various functions, processes, methods, and/or software modules described herein. Examples of management system **110** include, without limitation, an EMS, DMS, ADMS, SCADA system, and the like. Management system **110** may comprise dedicated servers, or may instead be implemented in a computing cloud, in which the resources of one or more servers are dynamically and elastically allocated to multiple tenants based on demand. In either case, the servers may be collocated (e.g., in a single data center) and/or geographically distributed (e.g., across a plurality of data centers). Management system **110** may also comprise or be communicatively connected to software **112** and/or one or more databases **114**. In addition, Management system **110** may be communicatively connected to one or more user systems **130** and/or power systems **140** (e.g., power grids) via one or more networks **120**.

(15) Network(s) **120** may comprise the Internet, and EMS **110** may communicate with user system(s) **130** and/or power system(s) **140** through the Internet using standard transmission protocols, such as HyperText Transfer Protocol (HTTP), HTTP Secure (HTTPS), File Transfer Protocol (FTP), FTP Secure (FTPS), Secure Shell FTP (SFTP), eXtensible Messaging and Presence Protocol (XMPP), Open Field Message Bus (OpenFMB), IEEE Smart Energy Profile Application Protocol (IEEE 2030.5), and the like, as well as proprietary protocols. While management system **110** is illustrated as being connected to various systems through a single set of network(s) **120**, it should be understood that management system **110** may be connected to the various systems via different sets of one or more networks. For example, management system **110** may be connected to a subset of user systems **130** and/or power systems **140** via the Internet, but may be connected to one or more other user systems **130** and/or power systems **140** via an intranet. Furthermore, while only a few user systems **130** and power systems **140**, one instance of software **112**, and one set of database(s) **114** are illustrated, it should be understood that the infrastructure may comprise any number of user systems, power systems, software instances, and databases.

(16) User system(s) **130** may comprise any type or types of computing devices capable of wired and/or wireless communication, including without limitation, desktop computers, laptop computers, tablet computers, smart phones or other mobile phones, servers, game consoles, televisions, set-top boxes, electronic kiosks, point-of-sale terminals, embedded controllers, programmable logic controllers (PLCs), and/or the like. However, it is generally contemplated that user system(s) **130** would comprise personal computers, mobile devices, or workstations by which agents of an operator of a power system

140 can interact with management system **110**. These interactions may comprise inputting data (e.g., parameters for configuring one or more of the processes described herein) and/or receiving data (e.g., the outputs of one or more processes described herein) via a graphical user interface provided by management system **110** or a system between management system **110** and user system(s) **130**. The graphical user interface may comprise screens (e.g., webpages) that include a combination of content and elements, such as text, images, videos, animations, references (e.g., hyperlinks), frames, inputs (e.g., textboxes, text areas, checkboxes, radio buttons, drop-down menus, buttons, forms, etc.), scripts (e.g., JavaScript), and the like, including elements comprising or derived from data stored in one or more databases (e.g., database(s) **114**).

(17) Management system **110** may execute software **112**, comprising one or more software modules that implement one or more of the disclosed processes. In addition, management system **110** may comprise, be communicatively coupled with, or otherwise have access to one or more database(s) **114** that store the data input into and/or output from one or more of the disclosed processes. Any suitable database may be utilized, including without limitation MySQL™ Oracle™, IBM™, Microsoft SQL™, Access™, PostgreSQL™, and the like, including cloud-based databases, proprietary databases, and unstructured databases.

1.2. Example Processing Device

(18) FIG. **2** is a block diagram illustrating an example wired or wireless system **200** that may be used in connection with various embodiments described herein. For example, system **200** may be used as or in conjunction with one or more of the functions, processes, or methods (e.g., to store and/or execute software **112**) described herein, and may represent components of management system **110**, user system(s) **130**, power system(s) **140**, and/or other processing devices described herein. System **200** can be a server or any conventional personal computer, or any other processor-enabled device that is capable of wired or wireless data communication. Other computer systems and/or architectures may be also used, as will be clear to those skilled in the art.

(19) System **200** preferably includes one or more processors **210**. Processor(s) **210** may comprise a central processing unit (CPU). Additional processors may be provided, such as a graphics processing unit (GPU), an auxiliary processor to manage input/output, an auxiliary processor to perform floating-point mathematical operations, a special-purpose microprocessor having an architecture suitable for fast execution of signal-processing algorithms (e.g., digital-signal processor), a processor subordinate to the main processing system (e.g., back-end processor), an additional microprocessor or controller for dual or multiple processor systems, and/or a coprocessor. Such auxiliary processors may be discrete processors or may be integrated with processor **210**. Examples of processors which may be used with system **200** include, without limitation, any of the processors (e.g., Pentium™, Core i7™, Xeon™, etc.) available from Intel Corporation of Santa Clara, California, any of the processors available from Advanced Micro Devices, Incorporated (AMD) of Santa Clara, California, any of the processors (e.g., A series, M series, etc.) available from Apple Inc. of Cupertino, any of the processors (e.g., Exynos™) available from Samsung Electronics Co., Ltd., of Seoul, South Korea, and/or the like.

(20) Processor **210** is preferably connected to a communication bus **205**. Communication bus **205** may include a data channel for facilitating information transfer between storage and other peripheral components of system **200**. Furthermore, communication bus **205** may provide a set of signals used for communication with processor **210**, including a data bus, address bus, and/or control bus (not shown). Communication bus **205** may comprise any standard or non-standard bus architecture such as, for example, bus architectures compliant with industry standard architecture (ISA), extended industry standard architecture (EISA), Micro Channel Architecture (MCA), peripheral component interconnect (PCI) local bus, standards promulgated by the Institute of Electrical and Electronics Engineers (IEEE) including IEEE 488 general-purpose interface bus (GPIB), IEEE 696/S-100, and/or the like.

(21) System **200** preferably includes a main memory **215** and may also include a secondary memory **220**. Main memory **215** provides storage of instructions and data for programs executing on processor **210**, such as one or more of the functions and/or modules discussed herein (e.g., software **112**). It should be understood that programs stored in the memory and executed by processor **210** may be written and/or compiled according to any suitable language, including without limitation C/C++, Java,

JavaScript, Perl, Visual Basic, NET, and the like. Main memory **215** is typically semiconductor-based memory such as dynamic random access memory (DRAM) and/or static random access memory (SRAM). Other semiconductor-based memory types include, for example, synchronous dynamic random access memory (SDRAM), Rambus dynamic random access memory (RDRAM), ferroelectric random access memory (FRAM), and the like, including read only memory (ROM).

(22) Secondary memory **220** may optionally include an internal medium **225** and/or a removable medium **230**. Removable medium **230** is read from and/or written to in any well-known manner. Removable storage medium **230** may be, for example, a magnetic tape drive, a compact disc (CD) drive, a digital versatile disc (DVD) drive, other optical drive, a flash memory drive, and/or the like. Secondary memory **220** is a non-transitory computer-readable medium having computer-executable code (e.g., software **112**) and/or other data stored thereon. The computer software or data stored on secondary memory **220** is read into main memory **215** for execution by processor **210**.

(23) In alternative embodiments, secondary memory **220** may include other similar means for allowing computer programs or other data or instructions to be loaded into system **200**. Such means may include, for example, a communication interface **240**, which allows software and data to be transferred from external storage medium **245** to system **200**. Examples of external storage medium **245** may include an external hard disk drive, an external optical drive, an external magneto-optical drive, and/or the like. Other examples of secondary memory **220** may include semiconductor-based memory, such as programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable read-only memory (EEPROM), and flash memory (block-oriented memory similar to EEPROM).

(24) As mentioned above, system **200** may include a communication interface **240**. Communication interface **240** allows software and data to be transferred between system **200** and external devices (e.g. printers), networks, or other information sources. For example, computer software or executable code may be transferred to system **200** from a network server (e.g., platform **110**) via communication interface **240**. Examples of communication interface **240** include a built-in network adapter, network interface card (NIC), Personal Computer Memory Card International Association (PCMCIA) network card, card bus network adapter, wireless network adapter, Universal Serial Bus (USB) network adapter, modem, a wireless data card, a communications port, an infrared interface, an IEEE 1394 fire-wire, and any other device capable of interfacing system **200** with a network (e.g., network(s) **120**) or another computing device. Communication interface **240** preferably implements industry-promulgated protocol standards, such as Ethernet IEEE 802 standards, Fiber Channel, digital subscriber line (DSL), asynchronous digital subscriber line (ADSL), frame relay, asynchronous transfer mode (ATM), integrated digital services network (ISDN), personal communications services (PCS), transmission control protocol/Internet protocol (TCP/IP), serial line Internet protocol/point to point protocol (SLIP/PPP), and so on, but may also implement customized or non-standard interface protocols as well.

(25) Software and data transferred via communication interface **240** are generally in the form of electrical communication signals **255**. These signals **255** may be provided to communication interface **240** via a communication channel **250**. In an embodiment, communication channel **250** may be a wired or wireless network (e.g., network(s) **120**), or any variety of other communication links. Communication channel **250** carries signals **255** and can be implemented using a variety of wired or wireless communication means including wire or cable, fiber optics, conventional phone line, cellular phone link, wireless data communication link, radio frequency ("RF") link, or infrared link, just to name a few.

(26) Computer-executable code (e.g., computer programs, such as software **112**) is stored in main memory **215** and/or secondary memory **220**. Computer programs can also be received via communication interface **240** and stored in main memory **215** and/or secondary memory **220**. Such computer programs, when executed, enable system **200** to perform the various functions of the disclosed embodiments as described elsewhere herein.

(27) In this description, the term "computer-readable medium" is used to refer to any non-transitory computer-readable storage media used to provide computer-executable code and/or other data to or

within system **200**. Examples of such media include main memory **215**, secondary memory **220** (including internal memory **225** and/or removable medium **230**), external storage medium **245**, and any peripheral device communicatively coupled with communication interface **240** (including a network information server or other network device). These non-transitory computer-readable media are means for providing executable code, programming instructions, software, and/or other data to system **200**.

(28) In an embodiment that is implemented using software, the software may be stored on a computer-readable medium and loaded into system **200** by way of removable medium **230**, I/O interface **235**, or communication interface **240**. In such an embodiment, the software is loaded into system **200** in the form of electrical communication signals **255**. The software, when executed by processor **210**, preferably causes processor **210** to perform one or more of the processes and functions described elsewhere herein.

(29) In an embodiment, I/O interface **235** provides an interface between one or more components of system **200** and one or more input and/or output devices. Example input devices include, without limitation, sensors, keyboards, touch screens or other touch-sensitive devices, cameras, biometric sensing devices, computer mice, trackballs, pen-based pointing devices, and/or the like. Examples of output devices include, without limitation, other processing devices, cathode ray tubes (CRTs), plasma displays, light-emitting diode (LED) displays, liquid crystal displays (LCDs), printers, vacuum fluorescent displays (VFDs), surface-conduction electron-emitter displays (SEDs), field emission displays (FEDs), and/or the like. In some cases, an input and output device may be combined, such as in the case of a touch panel display (e.g., in a smartphone, tablet, or other mobile device).

(30) System **200** may also include optional wireless communication components that facilitate wireless communication over a voice network and/or a data network (e.g., in the case of user system **130** that is a smart phone or other mobile device). The wireless communication components comprise an antenna system **270**, a radio system **265**, and a baseband system **260**. In system **200**, radio frequency (RF) signals are transmitted and received over the air by antenna system **270** under the management of radio system **265**.

(31) In an embodiment, antenna system **270** may comprise one or more antennae and one or more multiplexors (not shown) that perform a switching function to provide antenna system **270** with transmit and receive signal paths. In the receive path, received RF signals can be coupled from a multiplexor to a low noise amplifier (not shown) that amplifies the received RF signal and sends the amplified signal to radio system **265**.

(32) In an alternative embodiment, radio system **265** may comprise one or more radios that are configured to communicate over various frequencies. In an embodiment, radio system **265** may combine a demodulator (not shown) and modulator (not shown) in one integrated circuit (IC). The demodulator and modulator can also be separate components. In the incoming path, the demodulator strips away the RF carrier signal leaving a baseband receive audio signal, which is sent from radio system **265** to baseband system **260**.

(33) If the received signal contains audio information (e.g., in the case of a user system **130** that is capable of operating as a telephone), then baseband system **260** decodes the signal and converts it to an analog signal. Then the signal is amplified and sent to a speaker. Baseband system **260** also receives analog audio signals from a microphone. These analog audio signals are converted to digital signals and encoded by baseband system **260**. Baseband system **260** also encodes the digital signals for transmission and generates a baseband transmit audio signal that is routed to the modulator portion of radio system **265**. The modulator mixes the baseband transmit audio signal with an RF carrier signal, generating an RF transmit signal that is routed to antenna system **270** and may pass through a power amplifier (not shown). The power amplifier amplifies the RF transmit signal and routes it to antenna system **270**, where the signal is switched to the antenna port for transmission.

(34) Baseband system **260** is also communicatively coupled with processor(s) **210**. Processor(s) **210** may have access to data storage areas **215** and **220**. Processor(s) **210** are preferably configured to execute instructions (i.e., computer programs, such as the disclosed software) that can be stored in main memory **215** or secondary memory **220**. Computer programs can also be received from baseband

processor **260** and stored in main memory **210** or in secondary memory **220**, or executed upon receipt. Such computer programs, when executed, enable system **200** to perform the various functions of the disclosed embodiments.

1.3. Example Data Flow

(35) FIG. **3** illustrates an example data flow between management system **110**, a user system **130**, and a power system **140**, according to an embodiment. Power system **140** may comprise a monitoring module **310** and a control module **320**. Software **112** of management system **110** may comprise a state estimation module **330**, a control and optimization module **340**, and a human-to-machine interface (HMI) **350**. Database **114** of management system **110** may store a system model **360**. It should be understood that communications between the various systems may be performed via network(s) **120**. In addition, communications between a pair of modules may be performed via an application programming interface (API) provided by one of the modules.

(36) Monitoring module **310** may monitor and collect data that are output by one or more sensors in the network of power system **140**. Monitoring module **310** may also derive data from the collected data. Monitoring module **310** may transmit or "push" the collected and/or derived data as system telemetry to state estimation module **330** (e.g., via an API of state estimation module **330**). Alternatively, state estimation module **330** may retrieve or "pull" the system telemetry from monitoring module **310** (e.g., via an API of monitoring module **310**). The system telemetry may include measurements at each of one or more nodes or other points within the network of power system **140**. The system telemetry may be communicated from monitoring module **310** to state estimation module **330** in real time, as data is collected and/or derived, or periodically. As used herein, the term "real time" includes events that occur simultaneously, as well as events that occur contemporaneously, as dictated by ordinary delays resulting from latencies in processing, memory access, communications, and/or the like.

(37) State estimation module **330** receives the system telemetry from monitoring module **310** and uses the system telemetry in conjunction with system model **360** to generate an estimated state of power system **140**. In particular, state estimation module **330** may implement one or more of the processes for state estimation, described herein, to set up an optimization problem based on the system telemetry and system model **360**, and solve the optimization problem. State estimation module **330** may generate the estimated state of power system **140** in real time or periodically (e.g., whenever new system telemetry is received). The estimated state may comprise an estimated voltage magnitude and phase angle for each node within the network of power system **140**. State estimation module **330** may send or "push" the estimated state of power system **140** to optimization and control module **340** (e.g., via an API of optimization and control module **340**). Alternatively, optimization and control module **340** may retrieve or "pull" the estimated state of power system **140** from state estimation module **330** (e.g., via an API of state estimation module **330**). The estimated state may be communicated from state estimation module **330** to optimization and control module **340** in real time, as the estimated state is generated, or periodically.

(38) Optimization and control module **340** receives the estimated state from state estimation module **330**, and uses the estimated state in conjunction with system model **360** to determine an optimal configuration for one or more components of power system **140**, and then control power system **140** to transition to the optimal configuration. In particular, optimization and control module **340** may generate control signals that are transmitted to control module **320** of power system **140**. For example, the control signals may be sent via an API of control module **320**. The control signals may be communicated from optimization and control module **340** of management system **110** to control module **320** of power system **140** in real time, as the estimated states are received and analyzed, periodically, or in response to a user operation. Optimization and control module **340** may control power system **140** automatically (e.g., without any user intervention), semi-automatically (e.g., requiring user approval or confirmation), and/or in response to manual user inputs.

(39) Control module **320** of power system **140** receives the control signals from optimization and control module **340**, and controls one or more components of power system **140** in accordance with the control signals. Examples of such control include, setting a setpoint (e.g., for real and/or reactive

power for distributed energy resources), adjusting the power output of a power generator, adjusting the charging or discharging of a BES system, adjusting the power input to a load, closing or opening a switch (e.g., circuit breaker), and/or the like.

(40) Human-to-machine interface **350** may generate a graphical user interface that is transmitted to user system **130** and receive inputs to the graphical user interface via user system **130**. The graphical user interface may provide information regarding the estimated state of power system **140** determined by state estimation module **330**, an optimal configuration of power system **140** determined by optimization and control module **340**, a control decision or recommendation determined by optimization and control module **340**, a visual representation of system model **360**, and/or the like. In addition, the graphical user interface may provide inputs that enable a user of user system **130** to configure settings of state estimation module **330**, configure settings of optimization and control module **340**, configure system model **360**, specify or approve controls to be transmitted to control module **320** of power system **140**, analyze power system **140**, and/or the like.

(41) System model **360** may be stored in database **114** and accessed by modules, such as state estimation module **330** and optimization and control module **340**, via any known means (e.g., via an API of database **114**, a direct query of database **114**, etc.). Database **114** may store a system model **360** for each power system **140** that is managed by management system **110**. Each system model **360** models the network of power system **140** in any suitable manner. For example, system model **360** may comprise a single-line diagram representing the components of the network and their relationships to each other. It should be understood that the single-line diagram may be implemented as a data structure that is capable of being automatically analyzed by software modules, including state estimation module **330** and optimization and control module **340**.

(42) The estimated state, output by state estimation module **330**, may be used as an input to any downstream function that may benefit from an estimated state of power system **140**. These downstream functions may be implemented by optimization and control module **340**, human-to-machine interface **350**, and/or other modules within management system **110** or an external system. In the event that a downstream function is implemented by another module, state estimation module **330** may send or "push" the estimated state of power system **140** to the other module (e.g., via an API of the implementing module or relayed through optimization and control module **340**). Alternatively, the other module may retrieve or "pull" the estimated state of power system **140** from state estimation module **330** (e.g., via an API of state estimation module **330**) or from optimization and control module **340** (e.g., via an API of optimization and control module **340**). The estimated state may be communicated to the implementing module in real time, as the estimated state is generated, or periodically.

(43) As an example of a downstream function, the estimated state may be stored and, in response to a triggering event, displayed to a user within a graphical user interface of human-to-machine interface **350**. The triggering event may be a user requesting the estimated state, the estimated state satisfying an alert condition, and/or the like. In the event that the estimated state satisfies an alert condition, the user may be prompted via the graphical user interface or other means (e.g., a notification sent via email message, text message, voice message, etc.) to perform a preventative or corrective control operation (e.g., via an input of the graphical user interface, manually, etc.).

(44) In general, the estimated state, which may comprise voltage magnitudes and phase angles for each of the nodes in the network in power system **140**, may be used to produce a baseline model for one or more downstream functions. A downstream function may utilize this baseline model to perform any type of analysis on power system **140**, including optimal power flow, distributed energy resource (DER) management, contingency analysis, and/or the like. The analysis may be performed in response to a user operation, or automatically in real time or periodically. In some cases, the analysis may be provided to a user via human-to-machine interface **350**. In other cases, optimization and control module **340** may, automatically (i.e., without user intervention) or semi-automatically (e.g., with user approval or confirmation), initiate a control operation based on the analysis. Initiation of the control operation may comprise transmitting control commands to control module **320** of power system **140**, which may responsively control power system **140** according to the control commands.

(45) As an example, the estimated state may be used by optimization and control module **340** as an input to contingency analysis. Contingency analysis may utilize the estimated state of power system **140** in a baseline model (e.g., system model **360**) of power system **140** to perform "what-if" analysis for various hypothetical scenarios (e.g., the failure of a component of power system **140**). For example, the estimated state (e.g., voltage magnitude and phase angle) at each node may be used to calculate the power generation or consumption at that node for use in the contingency analysis. The contingency analysis may be executed in real time (e.g., as estimated states are output by state estimation module **330**), periodically, and/or in response to a triggering event (e.g., user request, satisfaction of one or more monitored criteria, etc.). If a problem is detected by the contingency analysis, a user may perform a preventative or corrective control operation (e.g., via an input of a graphical user interface of human-to-machine interface **350**, manually, etc.). Alternatively or additionally, optimization and control module **340** may, automatically or semi-automatically, perform contingency analysis for one or more hypothetical scenarios, and when a problem is detected, initiate a preventative or corrective control operation via communication with control module **320** of power system **140**.

(46) As an additional example, the estimated state may be used by optimization and control module **340** as an input to Volt-Var optimization. Volt-Var optimization may utilize the estimated state (e.g., voltage magnitudes and phase angles) and system model **360** (or other model) as a baseline to determine optimal voltage levels and reactive power to achieve efficient operation of power system **140** (e.g., by reducing system losses, peak demand, and/or energy consumption). The Volt-Var optimization may be executed in real time (e.g., as estimated states are output by state estimation module **330**), periodically, and/or in response to a triggering event (e.g., user request, satisfaction of one or more monitored criteria, etc.). Optimization and control module **340** may, automatically or semi-automatically, initiate a control operation via communication with control module **320** of power system **140** to conform power system **140** to the optimal voltage levels and reactive power, as determined by the Volt-Var optimization. The control operation may comprise controlling switchable capacitors, on-load tap-changers, and/or the like.

(47) As an additional example, the estimated state may be used by optimization and control module **340** as an input to optimal power flow. Optimal power flow may utilize the estimated state and system model **360** (or other model) as a baseline to manage power generation within power system **140**. For example, the optimal power flow may determine the optimal setpoints for power generators within power system **140**, to meet the demand on power system **140**, while minimizing operating costs and/or satisfying one or more other criteria. Optimal power flow may be executed in real time (e.g., as estimated states are output by state estimation module **330**), periodically, and/or in response to a triggering event (e.g., user request, satisfaction of one or more monitored criteria, etc.). Optimization and control module **340** may, automatically or semi-automatically, initiate a control operation via communication with control module **320** of power system **140** to conform power system **140** to the setpoints determined by the optimal power flow.

(48) As an additional example, the estimated state may be used by optimization and control module **340** as an input to distributed energy resource (DER) management, which can be thought of as a species of optimal power flow. DER management may utilize the estimated state and system model **360** (or other model) to manage distributed energy resources within power system **140**. For example, the DER management may manage the setpoints for active and reactive power, power factor, and/or voltage at the points of interconnection with distributed energy resources or other nodes within the network. The DER management may be executed in real time (e.g., as estimated states are output by state estimation module **330**), periodically, and/or in response to a triggering event (e.g., user request, satisfaction of one or more monitored criteria, etc.). Optimization and control module **340** may, automatically or semi-automatically, initiate a control operation via communication with control module **320** of power system **140** to conform power system **140** to the setpoints determined by the DER management.

2. Process Overview

(49) Embodiments of processes for state estimation for a power system, using parameterized potential functions for equality constraints, will now be described in detail. It should be understood that the

described processes may be embodied in one or more software modules that are executed by one or more hardware processors, for example, as software **112** executed by processor(s) **210** of management system **110**. The described processes may be implemented as instructions represented in source code, object code, and/or machine code. These instructions may be executed directly by hardware processor(s) **210**, or alternatively, may be executed by a virtual machine or container operating between the object code and hardware processors **210**. In addition, the disclosed software may be built upon or interfaced with one or more existing systems.

(50) Alternatively, the described processes may be implemented as a hardware component (e.g., general-purpose processor, integrated circuit (IC), application-specific integrated circuit (ASIC), digital signal processor (DSP), field-programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, etc.), combination of hardware components, or combination of hardware and software components. To clearly illustrate the interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps are described herein generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled persons can implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the invention. In addition, the grouping of functions within a component, block, module, circuit, or step is for ease of description. Specific functions or steps can be moved from one component, block, module, circuit, or step to another without departing from the invention.

(51) Furthermore, while the processes, described herein, are illustrated with a certain arrangement and ordering of subprocesses, each process may be implemented with fewer, more, or different subprocesses and a different arrangement and/or ordering of subprocesses. In addition, it should be understood that any subprocess, which does not depend on the completion of another subprocess, may be executed before, after, or in parallel with that other independent subprocess, even if the subprocesses are described or illustrated in a particular order.

2.1. Introduction

(52) Throughout the present disclosure, an equality constraint will be expressed in the form:

$$c(x)=0$$

wherein c(·) is an equality function that relates an input x to an output c(x). The input x represents a state of the system being monitored and controlled, such as power system **140**. In the case of a power system **140**, the state may be defined as the voltage magnitude and phase angle at each node in the network. It should be understood that any equality constraint can be expressed in this manner, as well as other manners. Thus, for ease of understanding and conceptualization, it will be assumed that all equality constraints are expressed in the above form. However, it should be understood that, in an actual implementation, the equality constraint can be expressed in a different manner and either converted to the above form or used in its native form with appropriate modifications to other components of the SE problem.

(53) The constrained optimization problem may be expressed as:

(54) $$\min_{x} f(x)$$

subject to

$$c.sub.i(x)=0 \text{ for } i=1, \ldots, n.sub.EC$$

wherein i is an index of the set of equality constraint(s) to which the objective function $f(x)$ is subject, and n.sub.EC represents the total number of equality constraints to which the objective function $f(x)$ is subject. It should be understood that an "optimization problem" may comprise any problem of finding the best solution from all feasible solutions, and an "objective function" may be any function that is to be maximized or minimized. In addition, the term "constrained" refers to a problem in which the objective function is subject to a respective constraint, whereas the term "unconstrained" refers to a problem in which all respective constraints have been converted into an objective function. The term "convex" refers to any function in which the line segment between any two points on the graph of the function does not lie below the graph between the two points.

(55) The constrained optimization problem may also comprise inequality constraints. Throughout the present disclosure, an inequality constraint will be expressed in the form:

$g(x) \leq 0$

wherein g(·) is an inequality function that relates an input x to an output g(x). It should be understood that any inequality constraint can be expressed in this manner, as well as other manners. Thus, for ease of understanding and conceptualization, it will be assumed that all inequality constraints are expressed in the above form. However, it should be understood that, in an actual implementation, the inequality constraint can be expressed in a different manner and either converted to the above form or used in its native form with appropriate modifications to other components of the SE problem.

(56) Each equality constraint c(x)=0 represents that a measurement function corresponding to an input x must equal zero. However, in practice, a tolerance ε may be allowed, such that −ε≤c(x)≤ε is allowable. The range [−ε, ε] represents a feasible range of values for the equality function c(x). Similarly, inequality constraint g(x)≤0 represents a range of possible values of a measurement corresponding to an input x. In the context of a power system **140**, this range could be the capacity of a power generator (e.g., from zero to a maximum power output), the capacity of a BES system (e.g., based on the current charge level of the BES system), and/or the like. In practice, a tolerance may be allowed, such that the feasible range is represented by g(x)≤ε. The tolerance ε may be the same or different for equality and inequality constraints.

(57) Existing methods for enforcing equality constraints in state estimation include the elimination method, feasible direction method, Lagrange method (e.g., Hachtel's matrix formulation), augmented Lagrange method (using multipliers), and penalty function method. In methods that utilize a penalty function, the weight for each equality constraint generally needs to be set to a very large value, in order to drive a violation of an equality constraint to zero.

(58) In an embodiment, instead of the above methods, a parameterized potential function is used for each equality constraint. In the context of the parameterized potential function, the term "parameterized" refers to the inclusion of an adjustable parameter within the potential function, and the term "potential" refers to the analogy between the unconstrained objective function and a generalized potential field (e.g., like a particle within a potential field is pulled towards a zero-force point). While not necessary to the embodiment, the parameterized potential function may be a quadratic function. Each parameterized potential function may be expressed as:

$w(c(x)-u)^{2}$

wherein u is a center-of-attraction parameter that defines a "center of attraction" that, conceptually, acts like a spring to pull the output of the equality function c(x) towards zero. A value of c(x) violates its respective constraint when c(x)<−ε or c(x)>ε. Conversely, a value of c(x) does not violate its respective constraint when −ε≤c(x)≤ε. The tolerance ε is a practical consideration, since, in a numerical (computer) embodiment with finite precision, it may be impossible to satisfy the equality constraints exactly. The value of ε may be set using any standard means.

(59) Using the above parameterized potential function, the SE problem becomes an unconstrained convex optimization problem UCP(u):

(60) $\text{UCP}(u): \min_{x} P(x, u) \text{ wherein } P(x, u) = f(x) + \sum_{i=1}^{n_{EC}} w_i (c_i(x) - u_i)^2$

wherein $f(x)$ is the objective function of the constrained optimization problem to be minimized. It should be understood that $f(x)$ may be the WLS problem:

(61) $\sum_{l}^{m} w_l (z_l - h_l(x))^2$

wherein z.sub.l represents the system telemetry at a point l, wherein h(·) is a measurement function that relates an input x to an output h(x), representing a predicted measurement given the input x, w.sub.l is a respective weight, and m represents the total number of points being measured in system model **360**. However, more generally, $f(x)$ encompasses any convex objective function to be minimized.

(62) Each parameterized potential function can be thought of as a virtual measurement. FIG. **4** illustrates the operation of a virtual measurement that utilizes a center-of-attraction parameter u,

according to an embodiment. The dotted lines in the figure represent gradients of the quadratic equality function c(x). As illustrated in FIG. **4**, each virtual measurement utilizes a center-of-attraction parameter u. The value of this center-of-attraction parameter u defines a location (i.e., a center of attraction) towards which c(x) is attracted, with the weight w representing the strength of attraction. In other words, the value of the center-of-attraction parameter u can be analogized as an anchor point to which one end of a spring is attached, with the other end of the spring attached to c(x), and weight w analogous to the Hook's constant of the spring.

(63) While solving UCP(u) iteratively, instead of solving the original constrained optimization problem, the value of each center-of-attraction parameter u may be updated to "pull" the value of c(x) towards zero and eventually into the feasible range defined by the tolerance. For example, the value of center-of-attraction parameter u may start at zero (i.e., u.sup.(0)), and the problem UCP(u.sup.(0)) may be solved to convergence, resulting in c(x.sup.(0)). Then, the value of center-of-attraction parameter u may be updated to u.sup.(1), and the problem UCP(u.sup.(1)) may be solved to convergence, resulting in c(x.sup.(1)). Then, the value of center-of-attraction parameter u may be updated to u.sup.(2), and the problem UCP(u.sup.(2)) may be solved to convergence, resulting in c(x.sup.(2)). This iterative process may continue until the value of c(x) is within the feasible range (i.e., zero with a tolerance ε).

(64) The force field associated with the potential field of P(x, u) is defined by the negative gradient of P(x, u):

$$Force(x,u)=-\nabla.sub.xP(x,u)$$

The stationary point of this force field is parametric on the center-of-attraction parameters u of the parameterized potential functions:

(65) $x(u) = \underset{x}{\arg\min} P(x, u)$

(66) If x(u) satisfies the equality constraints, x(u) satisfies the Karush-Kuhn-Tucker (KKT) condition of the constrained optimization problem. In particular, the KKT condition for the constrained optimization problem is:

(67) $\nabla f(x) + \overset{n_{EC}}{\underset{i=1}{.\text{Math}.}} \lambda_i \nabla c_i(x) = 0$

wherein λ.sub.i is the Lagrange multiplier for the equality constraint with index i.

(68) The joint system below has the same solution as the KKT condition of the constrained optimization problem:

(69) $\nabla f(x) + \overset{n_{EC}}{\underset{i=1}{.\text{Math}.}} 2w_i(c_i(x) - u_i)\nabla c_i(x) = 0 c_i(x) = 0, i = 1, .\text{Math}. , n_{EC}$

(70) The parameters of the parameterized potential functions are related to the Lagrange multiplier λ.sub.i as follows:

$$\Delta.sub.i=-2w.sub.iu.sub.i$$

For any KKT point in the constrained optimization problem, there is a related unconstrained convex optimization of P(x, u) in which the center-of-attraction parameter u of each parameterized potential function is uniquely defined for any non-zero weight w. Thus, the constrained optimization problem is converted into a problem of placing the center-of-attraction parameters u, such that the KKT condition of the original constrained optimization problem is satisfied.

2.2. Equality Constraint Algorithm

(71) FIG. **5** illustrates a solution process **500**, according to an embodiment. Solution process **500** may be implemented by state estimation module **330** in software **112** of management system **110**.

(72) In subprocess **510**, an unconstrained convex optimization problem is prepared based on the equality constraints. In particular, a parameterized potential function may be generated for each equality constraint and summed with the objective function from the constrained optimization problem in a second objective function to be minimized. Again, the unconstrained convex optimization problem may be expressed as:

(73) $P: \underset{x}{\min}(f(x) + \overset{n_{Ec}}{\underset{i=1}{.\text{Math}.}} w_i(c_i(x) - u_i)^2)$

wherein w.sub.i(c.sub.i(x)−u.sub.i).sup.2 represents a parameterized quadratic potential function for an equality constraint with index i in the set of equality constraints having size n.sub.EC.

(74) In subprocess **520**, the values of the center-of-attraction parameters u.sub.i in all parameterized potential functions are determined. In an embodiment in which process **500** is performed iteratively, the value of the center-of-attraction parameter u.sub.i.sup.(k) for a current iteration k may be calculated based on a value of that center-of-attraction parameter u.sub.i.sup.(k-1) that was determined in an immediately preceding iteration k−1. The value of a center-of-attraction parameter u.sub.i in a parameterized potential function may also be calculated based on a value of the equality function c.sub.i(x) in that same parameterized potential function. In other words, the value of the center-of-attraction parameter u may be adjusted based on the amount of violation.

(75) In an embodiment, the value of each center-of-attraction parameter u.sub.i may be determined as follows:

(76) TABLE-US-00001 if u.sub.i.sup.(k−1) is null, u.sub.i.sup.(k) = u.sup.initial else, u.sub.i.sup.(k) = u.sub.i.sup.(k−1) − α * c.sub.i(x.sup.(k-1)) $0 < \alpha \le 1$

wherein u.sup.initial is a predefined value, a is a scaling factor, which may be determined empirically or in any appropriate manner, or may be omitted or set to $\alpha=1$ so as to be effectively omitted. In an embodiment, u.sup.initial=0.

(77) In subprocess **530**, the unconstrained convex optimization problem

(78) $\min\limits_{x} P(x, u)$

is solved by finding an input x that minimizes P(x, u), given the center-of-attraction parameters u determined in subprocess **520**. In an embodiment in which process **500** is performed iteratively, the initial input x.sup.(k),ini in iteration k may be set to the final input x.sup.(k-1),fin that was output as the optimizing input (i.e., the input that minimizes the second objective function) from the immediately preceding iteration k−1. The result of subprocess **530** and process **500** will be a final and optimizing input x.sup.(k),fin for the current iteration k of process **500**. This final input x.sup.(k),fin represents an estimated state of power system **140**.

(79) FIG. **6** illustrates an overall equality constraint (EC) algorithm **600** for solving a constrained optimization problem, comprising equality constraint(s), according to an embodiment. EC algorithm **600** may be implemented by state estimation module **330** in software **112** of management system **110**. EC algorithm **600** operates to convert a constrained optimization problem, comprising an objective function to be minimized subject to one or more equality constraints, to an unconstrained convex optimization problem, and then solve the unconstrained convex optimization problem. In solving the unconstrained convex optimization problem, EC algorithm **600** may iteratively execute solution process **500**.

(80) Initially, in subprocess **610**, the constrained optimization problem is acquired. This acquisition process may comprise retrieving a representation of the constrained optimization problem or retrieving representations of one or more components of the constrained optimization problem, such as the objective function to be minimized, and the equality functions of the equality constraint(s) to which the objective function is subject. Again, the constrained optimization problem may be represented as:

(81) $0\min\limits_{x} f(x)$

subject to

c.sub.i(x)=0,i=1, . . . ,n.sub.EC

Components of the constrained optimization problem, such as the objective function and any equality constraints or equality functions, may be stored in memory that persists over iterations of solution process **500**, such that they can be easily accessed in each iteration of solution process **500**, without having to be regenerated or redetermined in each iteration.

(82) In subprocess **620**, the constrained optimization problem is converted to an unconstrained convex optimization problem with each of the equality constraint(s) represented as a parameterized potential function. The unconstrained convex optimization problem may be represented as:

(83) $P^{(0)}: \min\limits_{x}(f(x) + \sum\limits_{i=1}^{n_{EC}} w_i(c_i(x) - u_i^{(0)})^2)u_i^{(0)} = u^{initial}, \forall i$

wherein i is an index of the set of equality constraint(s), and n.sub.EC is the size of the set of equality constraint(s). In an embodiment, u.sup.initial=0.

(84) Solution process **500** is performed to find an optimizing input x.sup.(0),fin that minimizes the unconstrained convex optimization problem. Over each subsequent iteration k of solution process **500**, if any, the optimizing input x.sup.(k-1),fin found by the immediately preceding iteration k−1 may be used as the initial input x.sup.(k),fin to the unconstrained convex optimization problem to be solved in iteration k of solution process **500** (i.e., in subprocess **530**). In an embodiment, in each iteration of solution process **500**, a parameterized potential function is included in the unconstrained convex optimization problem for every equality constraint:

(85) $P^{(k)} : \min_{x}(f(x) + \overset{n_{EC}}{\underset{i=1}{.Math.}} w_i(c_i(x) - u_i^{(k)})^2)$

(86) Each center-of-attraction parameter u.sup.(k) in an iteration k of solution process **500** may be set, as discussed elsewhere herein, based on the value of the corresponding center-of-attraction parameter u.sup.(k-1) in the immediately preceding iteration k−1 of solution process **500**, or initialized to u.sup.initial (e.g., zero) if there was no preceding iteration k−1. The output of each iteration of solution process **500** will be a final input x.sup.(k),fin, representing the optimizing input, given the parameters of the unconstrained convex optimization problem in the current iteration k.

(87) In subprocess **630**, it is determined whether or not the most recently found input x.sup.(k),fin violates any of the equality constraint(s). In an embodiment, an equality constraint is violated when the value of its equality function is not within a tolerance ε of the constraint value (e.g., c(x)<−ε or c(x)>ε, if the constraint value is zero). Conversely, in such an embodiment, an equality constraint is not violated when the value of its equality function is within a tolerance ε of the constraint value (i.e., −ε≤c(x)≤ε, if the constraint value is zero). If no equality constraints are violated (i.e., "No" in subprocess **630**), EC algorithm **600** proceeds to subprocess **640**. Otherwise, if at least one equality constraint is violated (i.e., "Yes" in subprocess **630**), algorithm executes another iteration of solution process **500**.

(88) In subprocess **640**, once no equality constraints are violated (i.e., "No" in subprocess **630**), the most recently found (i.e., last) optimizing input is output as the final estimated state of power system **140**. This will be the input x.sup.(k),fin, output by the last iteration of solution process **500**. It should be understood that this final estimated state of power system **140**, output by subprocess **640**, may be sent to optimization and control module **340**, which may use the final estimated state to optimize a configuration of power system **140** and/or otherwise control power system **140**.

(89) An example of EC pseudocode, implementing an embodiment of EC algorithm **600**, is provided below:

(90) TABLE-US-00002 01. k = 0. 02. Initialize problem to P.sup.(0). 03. Initialize P.sup.(0) with x.sup.(0),ini and u.sub.i.sup.(0) = 0, ∀i. 04. Solve P.sup.(k) to convergence to find x.sup.(k),fin. 05. If |c.sub.i(x.sup.(k),fin)| ≤ ε, ∀i then output x.sup.(k),fin and stop. 06. u.sub.i.sup.(k+1) = u.sub.i.sup.(k) − α * c.sub.i(x.sup.(k)), ∀i 07. x.sup.(k+1),ini = x.sup.(k),fin. 08. k = k + 1. 09. Go to 04.

(91) It should also be understood that how an iteration k is defined or delineated is arbitrary. For example, in the above EC pseudocode, the center-of-attraction parameters u.sub.i are determined, after solving the unconstrained convex optimization problem P.sup.(k), for a future iteration k+1. However, an iteration k can alternatively be delineated as including the determination of the center-of-attraction parameters u.sub.i prior to solving the unconstrained convex optimization problem P.sup.(k), as illustrated in solution process **500**. In any case, the particular delineation of the iterations k, in any of the disclosed examples, should not be construed as limiting in any respect.

2.3. Inequality Constraint Algorithm

(92) FIG. **7** illustrates an overall inequality constraint (IC) algorithm **700** for solving a constrained optimization problem, comprising both equality constraint(s) and inequality constraint(s), according to an embodiment. IC algorithm **700** may be implemented by state estimation module **330** in software **112** of management system **110**. IC algorithm **700** operates, over one or more iterations, to convert an IC-EC optimization problem, comprising an objective function to be minimized subject to one or more equality constraints and one or more inequality constraints, to an EC convex optimization problem (i.e., a problem that is subject to equality constraints, but not subject to any inequality constraints), and then solve the EC convex optimization problem using EC algorithm **600**, which converts the EC

convex optimization problem to a fully unconstrained convex optimization problem (i.e., a problem that is not subject to any inequality or equality constraints).

(93) Initially, in subprocess **710**, the IC-EC optimization problem is acquired. This acquisition process may comprise retrieving a representation of the IC-EC optimization problem or retrieving representations of one or more components of the IC-EC optimization problem, such as the objective function to be minimized, the inequality functions of the inequality constraint(s) to which the objective function is subject, and the equality functions of the equality constraint(s) to which the objective function is subject. The IC-EC optimization problem may be represented as:

(94) $\min_x f(x)$

subject to

$c.\text{sub}.i(x)=0, i=1, \ldots, n.\text{sub.EC}$

$g.\text{sub}.j(x)\leq0, j=1, \ldots, n.\text{sub.IC}$ and equivalently:

(95) $\min_x \{f(x), \text{subjectto} c_i(x) = 0, i = 1, .\text{Math.}, n_{EC}\}$

subject to

$g.\text{sub}.j(x)\leq0, j=1, \ldots, n.\text{sub.IC}$

wherein i is an index of the set of equality constraint(s), j is an index of the set of inequality constraint(s), n.sub.EC is the size of the set of equality constraint(s), and n.sub.IC is the size of the set of inequality constraint(s). Components of the constrained optimization problem, such as the objective function and any inequality and equality constraints or inequality and equality functions, may be stored in memory that persists over iterations of the loop formed by EC algorithm **600** and subprocesses **730-750** of IC algorithm **700**, such that they can be easily accessed in each iteration without having to be regenerated or redetermined in each iteration.

(96) In subprocess **720**, a convex optimization problem is generated from the constrained optimization problem without any inequality constraints. In other words, the IC-EC optimization problem is converted into an EC-only optimization problem, without any representation of any inequality constraints. That is to say, no parameterized potential functions are included in the EC optimization problem for any inequality constraints. Thus, the convex optimization problem may be represented as:

(97) $P_{EC}^{(noIC)}: \min_x \{f(x), \text{subjectto} c_i(x) = 0, i = 1, .\text{Math.}, n_{EC}\}$

but not subject to any inequality constraints. It should be understood that P.sub.EC.sup.(noIC) is an inequality-unconstrained optimization problem, but is still an EC optimization problem.

(98) Next, an initial iteration of EC algorithm **600** is executed to solve the EC optimization problem P.sub.EC.sup.(noIC) by converting the inequality-unconstrained optimization problem into a fully unconstrained optimization problem (e.g., in subprocess **510**):

(99) $P_{EC}^{(noIC)}: \min_x (f(x) + \sum_j^{n_{EC}} w_j (c_j(x) - u_j)^2)$

subject to no equality and inequality constraints. Next, P.sub.EC.sup.(noIC) is solved to convergence in order to find an input x.sup.(noIC),fin that minimizes P.sup.(noIC) (e.g., in subprocess **530**). This input x.sup.(noIC),fin is output (e.g., in subprocess **640**) as a potential solution to the optimization problem which may or may not satisfy the equality and inequality constraints. This solution process **500** may be performed iteratively, while updating each center-of-attraction parameter u.sub.i (e.g., in subprocess **520**) until no equality constraints are violated (i.e., "No" in subprocess **630**).

(100) In subprocess **730**, it is determined whether or not the most recently found input x violates any of the inequality constraint(s). It should be understood that, in the first iteration of subprocess **730**, the most recently found input to be evaluated in subprocess **730** will be x.sup.(noIC),fin In all subsequent iterations of subprocess **720**, the most recently found input to be evaluated in subprocess **720** will be the final input x.sup.(k-1),fin found by the most recent iteration k−1 of EC algorithm **600**. In an embodiment, an inequality constraint is violated when the value of its inequality function is not within a tolerance ε of the feasible range (i.e., g(x)>ε). Conversely, in such an embodiment, an inequality constraint is not violated when the value of its inequality function is within a tolerance ε of the feasible range (i.e., g(x)≤ε). If no inequality constraints are violated (i.e., "No" in subprocess **730**), IC algorithm **700** proceeds to subprocess **760**. Otherwise, if at least one inequality constraint is violated

(i.e., "Yes" in subprocess **730**), IC algorithm **700** executes an iteration of subprocess **740**, followed by subprocess **750**, followed by EC algorithm **600**.

(101) In subprocess **740**, an inequality-unconstrained or EC convex optimization problem is generated based on the set of inequality constraint(s) determined to have been violated in subprocess **730**. In particular, a parameterized potential function may be generated for each inequality constraint in the set of violated inequality constraint(s) and summed with the objective function from the constrained optimization problem in a second objective function to be minimized. While not necessary to the embodiment, the parameterized potential function may be a quadratic function. The EC convex optimization problem may be expressed as:

(102) $P_{EC} : \min_{X}(f(x) + \sum_{j=1}^{n_{VIC}} w_j(g_j(x) - u_j)^2, \text{subject to } c_i(x) = 0, i = 1, \text{.Math. }, n_{EC})$

wherein w.sub.j(g.sub.j (x)−u.sub.j).sup.2 represents a parameterized quadratic potential function for an inequality constraint j in the set of violated inequality constraints having size n.sub.VIC. Notably, this inequality-unconstrained convex optimization problem is still constrained by equality constraints.

(103) In subprocess **750**, the values of the center-of-attraction parameters u.sub.j in all parameterized potential functions for violated inequality constraints are determined. The value of the center-of-attraction parameter u.sub.j.sup.(k) for a current iteration k may be calculated based on a value of that center-of-attraction parameter u.sub.j.sup.(k-1) that was determined in an immediately preceding iteration k−1. The value of a center-of-attraction parameter u.sub.j in a parameterized potential function may also be calculated based on a value of the inequality function g.sub.j(x) in that same parameterized potential function. In other words, the value of the center-of-attraction parameter u may be adjusted based on the amount of violation.

(104) In an embodiment, the value of each center-of-attraction parameter u.sub.j, for the parameterized potential functions representing inequality constraints, may be determined as follows:

(105) when the set inequality constraints that are determined to be violated by the most recently found optimizing input is not empty,

u.sub.j.sup.(k)=u.sub.j.sup.(k-1),$j \in$V.sup.(k) and $j.Math.A$.sup.(k),

u.sub.j.sup.(k)=u.sup.initial,$j \in$A.sup.(k); and

when the set inequality constraints that are determined to be violated by the most recently found optimizing input is empty,

if g.sub.j(x.sup.(k-1))$\geq$0,u.sub.j.sup.(k)=u.sub.j.sup.(k-1)$-\alpha$*g.sub.j(x.sup.(k-1))

if g.sub.j(x.sup.(k-1))<0,u.sub.j.sup.(k)=u.sub.j.sup.(k-1)$+\alpha$*(g.sub.j(x.sup.(k-1)$-$u.sub.j.sup.(k-1))

wherein

wherein u.sup.initial is a predefined value, and α is a scaling factor, which may be determined empirically or in any appropriate manner, or may be omitted or set to α=1 so as to be effectively omitted. In an embodiment, u.sup.initial=0.

(106) After the EC convex optimization problem P.sub.EC has been defined in subprocesses **740** and **750**, another iteration of EC algorithm **600** is executed. Whereas the EC optimization problem P.sup. (noIC) is used as the problem to be converted into a fully unconstrained optimization problem in the initial iteration of EC algorithm **600**, the EC optimization problem P c) is used as the problem to be converted into a fully unconstrained optimization problem P.sub.EC.sup.(k) in each subsequent iteration k of EC algorithm **600**. In each subsequent iteration k of EC algorithm **600**, the fully unconstrained convex optimization problem P.sup.(k) may be expressed as:

(107) $P^{(k)} : \min_{X}(f(x) + \sum_{j=1}^{n_{VIC}} w_j(g_j(x) - u_j)^2 + \sum_{i=1}^{n_{EC}} w_i(c_i(x) - u_i)^2)$

(108) In subprocess **760**, once no inequality constraints are violated (i.e., "No" in subprocess **730**), the most recently found (i.e., last) optimizing input is output as the final estimated state of power system **140**. This may be the input x.sup.(noIC),fin, output by the initial iteration of EC algorithm **600**, if no inequality violations were determined in the first, and therefore, only iteration of subprocess **730**. Otherwise, this will be an input x.sup.(k),fin, output by the last iteration of EC algorithm **600**. It should be understood that this final estimated state of power system **140**, output by subprocess **760**, may be sent to optimization and control module **340**, which may use the final estimated state to optimize a

configuration of power system **140** and/or otherwise control power system **140** as discussed elsewhere herein.

(109) In IC algorithm **700**, EC algorithm **600** is utilized to solve the inequality-unconstrained or EC optimization problem, which is still subject to equality constraints. However, in an alternative embodiment, a different algorithm may be used to solve the EC optimization problem and/or convert the EC optimization problem to a fully unconstrained convex optimization problem. Thus, EC algorithm **600** in IC algorithm **700** could be replaced by any other suitable algorithm for solving an EC optimization problem.

(110) An example of IC pseudocode, implementing an embodiment of IC algorithm **700**, is provided below:

(111) TABLE-US-00003 01. Initialize problem to P.sub.EC.sup.(noIC). 02. Initialize P.sub.EC.sup. (noIC) with x.sup.(noIC),ini. 03. Solve P.sub.EC.sup.(noIC) to convergence to find x.sup.(noIC),fin (e.g., using EC pseudocode). 04. If g.sub.j(x.sup.(noIC),fin) ≤ ε, ∀j then output x.sup.(noIC),fin and stop. 05. k = 0. 06. V.sub.IC = {j | g.sub.j(x.sup.(noIC),fin) > 0} 07. Prepare and solve P.sub.EC.sup.(k) convergence to find x.sup.(k),fin (e.g., using EC pseudocode). 08. A.sup.(k) = {j | j .Math. V.sub.IC.sup.(k) and g.sub.j(x.sup.(k),fin) > 0}. 09. If A.sup.(k) = { } then go to 11. 10. If A.sup.(k) ≠ { } then V.sub.IC.sup.(k+1) = V.sub.IC.sup.(k) + A.sup.(k); update P.sub.EC.sup.(k+1); k = k + 1; and go to 07. 11. If g.sub.j(x.sup.(k),fin) ≤ ε, ∀j then output x.sup.(k),fin and stop. 12. A.sup.(k) = {j | j .Math. V.sub.IC and g.sub.j(x.sup.(k),fin) > 0}. 13. If A.sup.(k) ≠ { } then   V.sub.IC.sup.(k+1) = V.sub.IC.sup.(k) + A.sup.(k);   u.sub.j.sup.(k+1) = u.sub.j.sup.(k) for j ∈ V.sub.IC.sup.(k); and   u.sub.j.sup.(k+1) = u.sup.initial for j ∈ A.sup.(k). 14. If A.sup.(k) = { } then   u.sub.j.sup. (k+1) = u.sub.j.sup.(k) − α * g.sub.j(x.sup.(k)) for j for which g.sub.j(x.sup.(k)) ≥ 0; and u.sub.j.sup.(k+1) = u.sub.j.sup.(k) + α * (g.sub.j(x.sup.(k)) − u.sub.j.sup.(k)) for j for which g.sub.j(x.sup.(k)) < 0. 15. x.sup.(k+1),ini = x.sup.(k),fin. 16. k = k + 1. 17. Prepare and solve P.sub.EC.sup.(k) to convergence to find x.sup.(k),fin (e.g., using EC pseudocode). 18. Go to 11.

(112) Notably, in the above pseudocode, the set V.sub.IC.sup.(k) consists of all inequality constraints that have been determined to be violated in all prior iterations up to and including the current iteration k, and the set A.sup.(k) consists of any inequality constraints that are determined to be violated by the most recently found optimizing input. In an embodiment, the initialization value u.sup.initial for each center-of-attraction parameter u is a predefined value, such as zero. However, it should be understood that other initialization values (i.e., non-zero initialization values) could be used for the center-of-attraction parameters u, and that the same or different initialization values may be used for center-of-attraction parameters u.sub.i (i.e., for equality constraints) and u.sub.j (i.e., for inequality constraints).

(113) It should also be understood that how an iteration k is defined or delineated is arbitrary. For example, in the above pseudocode, the sets V.sub.IC and A and the center-of-attraction parameters u.sub.j are determined, after solving the unconstrained convex optimization problem P.sub.EC.sup.(k), for a future iteration k+1. However, an iteration k can alternatively be delineated as including the determination of the sets V.sub.IC and A and the center-of-attraction parameters u.sub.j prior to solving the fully unconstrained convex optimization problem P.sub.EC.sup.(k), as illustrated in EC algorithm **600**. In any case, the particular delineation of the iterations k, in any of the disclosed examples, should not be construed as limiting in any respect. It should also be understood that, within IC algorithm **700**, EC algorithm **600** executes an inner loop of iterations, which are also referred to herein as iterations k, and which are distinct from the outer iterations k executed by IC algorithm **700**.

2.4. Alternative Embodiment of Equality Constraint Algorithm

(114) FIG. **8** illustrates an EC algorithm **800** for solving a constrained optimization problem, comprising equality constraint(s), according to an alternative embodiment. EC algorithm **800** may be implemented by state estimation module **330** in software **112** of management system **110**. Again, the constrained optimization problem may be represented as:

(115) $\min_{x} f(x)$

subject to

c.sub.i(x)=0,i=1, . . . ,n.sub.EC

(116) In subprocess **810**, an unconstrained parametric convex optimization problem is generated based on the equality constraints, thereby converting the constrained optimization problem into an unconstrained convex optimization problem. In particular, a parameterized potential function (e.g., quadratic function) may be generated for each equality constraint and summed with the objective function from the constrained optimization problem in a second objective function to be minimized. The unconstrained convex optimization problem may be expressed as:

(117) $0 P_{UC}(u): \min_x (f(x) + \sum_{i=1}^{n_{EC}} w_i(c_i(x) - u_i)^2)$

wherein w.sub.i(c.sub.i(x)−u.sub.i).sup.2 represents a parameterized quadratic potential function for an equality constraint within index i in the determined set of inequality constraints having size n.sub.EC. The solution to the parametric problem P.sub.UC(u) is x(u), which is parametric on u and must also satisfy the equality constraints.

(118) In subprocess **820**, a joint system of equations is defined. The joint system of equations comprises the first order optimality condition of problem P.sub.UC(u) and the equality constraints:

(119) $\left\{ \begin{array}{l} \nabla(f(x) + \sum_{i=1}^{n_{EC}} w_i(c_i(x) - u_i)^2) = 0 \\ w_i c_i(x) = 0, i = 1, \ .\text{Math}. \ , n_{EC} \end{array} \right.$

(120) In subprocess **830**, this joint system of equations, defined in subprocess **820**, is solved to find an optimizing input x. The solution may be found using any root-finding algorithm, such as Newton's method, also known as the Newton-Raphson method, to yield a solution that satisfies the KKT condition of the constrained optimization problem. In Newton's method, an initial guess of input x.sup.ini is made, the function is approximated by its tangent line, and the x-intercept of this tangent line is computed, which will generally be a better approximation of the original function's root than x.sup.ini. This process may be repeated with the computed x-intercept of the prior iteration as the new guess in each new iteration, until the solution converges to within a tolerance.

(121) In subprocess **840**, the solution from subprocess **830** is output. For example, the solution may be output as an estimated state to optimization and control module **340**. Optimization and control module **340** may then utilize the estimated state to optimize and/or control power system **140**, as described elsewhere herein.

(122) Using $f(x) = w.\text{sub}.l(z.\text{sub}.l − h.\text{sub}.l(x)).\text{sup}.2$, the constrained SE problem can be expressed as:

(123) $\min_x (z - h(x))^T W_h (z - h(x))$

subject to

$c(x) = 0$

(124) The unconstrained SE problem, generated in subprocess **810**, can be expressed as:

(125) $\min_x (z - h(x))^T W_h (z - h(x)) + (u - c(x))^T W_c (u - c(x))$

$\min_x ([\begin{smallmatrix} z \\ u \end{smallmatrix}] - [\begin{smallmatrix} h(x) \\ c(x) \end{smallmatrix}])^T [\begin{smallmatrix} W_h & \\ & W_c \end{smallmatrix}]([\begin{smallmatrix} z \\ u \end{smallmatrix}] - [\begin{smallmatrix} h(x) \\ c(x) \end{smallmatrix}])$

(126) The joint system of equations, defined in subprocess **820**, can be expressed as:

(127) $\left\{ \begin{array}{l} [\begin{smallmatrix} H(x) \\ C(x) \end{smallmatrix}]^T [\begin{smallmatrix} W_h & \\ & W_c \end{smallmatrix}]([\begin{smallmatrix} z \\ u \end{smallmatrix}] - [\begin{smallmatrix} h(x) \\ c(x) \end{smallmatrix}]) = 0 \\ W_c c(x) = 0 \end{array} \right.$

(128) The Newton-Raphson iterative equations, used in subprocess **830**, for the joint system can be expressed as:

(129)
$[\begin{smallmatrix} H^T W_h H + C^T W_c C & -C^T W_c \\ -W_c C & \end{smallmatrix}][\begin{smallmatrix} \Delta x^{(k)} \\ \Delta u^{(k)} \end{smallmatrix}] = [\begin{smallmatrix} H^T W_h(z - h(x^{(k)})) + C^T W_c(u^{(k)} - c(x^{(k)})) \\ W_c c(x) \end{smallmatrix}]$

(130) The upper left corner of the submatrix is H.sup.TW.sub.hH+C.sup.TW.sub.cC, which differs from H.sup.T W.sub.hH in the Hachtel's matrix in the Lagrange method, and results in a better-

conditioned gain matrix.

3. Example Usage

(131) In an embodiment, state estimation module **330** of a management system **110** is used to estimate the state of a system, such as power system **140**, using algorithms **600** and/or **700**. In particular, state estimation module **330** may receive system telemetry from a monitoring module **310** of the system, and use the system telemetry and system model **360** (e.g., retrieved from database **114**) to define an optimization problem, subject to equality constraints, and optionally inequality constraints, as an unconstrained convex optimization problem. The optimization problem is formed as an unconstrained optimization problem by generating a parameterized potential function $w(c(x)-u).sup.2$ for each of the equality constraints. The unconstrained optimization problem may also include a parameterized potential function $w(g(x)-u).sup.2$ for each of the inequality constraints. The sum of any generated parameterized potential functions and an objective function $f(x)$, derived from the system telemetry and system model **360** and representing the system (e.g., power system **140**) is then minimized to find an optimizing input x. The use of the parameterized potential functions, representing virtual measurements, shapes the potential field of the resulting unconstrained optimization problem, such that its stationary stationarity condition is identical to the KKT condition of the constrained optimization problem.

(132) The resulting unconstrained optimization problem may be iteratively prepared, using successively updated values for the center-of-attraction parameters u corresponding to the constraints, and solved, until no constraints are violated (i.e., all constraints are satisfied within a tolerance), to produce a final estimated state. The final estimated state may be output to an optimization and control module **340** of management system **110**, which may input the final estimated state into one or more downstream functions, such as the generation of a graphical user interface, contingency analysis, power flow analysis (e.g., dispatcher or optimal power flow analysis), Volt-Var optimization, DER management, and/or the like.

(133) In some cases, one or more of these downstream functions may, automatically (e.g., without user intervention) or semi-automatically (e.g., after user approval or confirmation), issue control commands to a control module **320** of the system, to thereby control the system. Thus, the state estimation may be used to monitor and control a system, such as a power system **140**. For example, if the voltage at a measured point is too low, a corresponding power generator in power system **140** may be controlled to increase its reactive power output to increase the voltage at that measured point.

(134) Notably, the disclosed algorithms **600** and **700** are robust and easy to implement, and provide fast convergence. In an embodiment, the weights w in the parameterized potential functions and/or the objective function $f(x)$ may be adaptively adjusted based on the rate of convergence. In addition, while discussed in the context of state estimation, the disclosed algorithms **600** and **700** could be used in other applications. One such application is training a machine-learning model, such as a neural network, and particularly a physics-informed neural network (PINN).

(135) The above description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the invention. Various modifications to these embodiments will be readily apparent to those skilled in the art, and the general principles described herein can be applied to other embodiments without departing from the spirit or scope of the invention. Thus, it is to be understood that the description and drawings presented herein represent a presently preferred embodiment of the invention and are therefore representative of the subject matter which is broadly contemplated by the present invention. It is further understood that the scope of the present invention fully encompasses other embodiments that may become obvious to those skilled in the art and that the scope of the present invention is accordingly not limited.

(136) Combinations, described herein, such as "at least one of A, B, or C," "one or more of A, B, or C," "at least one of A, B, and C," "one or more of A, B, and C," and "A, B, C, or any combination thereof" include any combination of A, B, and/or C, and may include multiples of A, multiples of B, or multiples of C. Specifically, combinations such as "at least one of A, B, or C," "one or more of A, B, or C," "at least one of A, B, and C," "one or more of A, B, and C," and "A, B, C, or any combination thereof" may be A only, B only, C only, A and B, A and C, B and C, or A and B and C, and any such

combination may contain one or more members of its constituents A, B, and/or C. For example, a combination of A and B may comprise one A and multiple B's, multiple A's and one B, or multiple A's and multiple B's.

Claims

1. A method of state estimation for a power system, the method comprising using at least one hardware processor to: acquire a constrained optimization problem comprising a convex first objective function to be minimized subject to one or more equality constraints; convert the constrained optimization problem into an unconstrained convex optimization problem comprising a second objective function that is a sum of the first objective function and a parameterized potential function for each of at least one of the one or more equality constraints, wherein each parameterized potential function is defined by an equality function and a center-of-attraction parameter; and perform a first process comprising performing an initial iteration of a solution process to find an initial optimizing input, wherein the solution process comprises solving the unconstrained convex optimization problem by finding an input to the unconstrained convex optimization problem that minimizes an output of the unconstrained convex optimization problem, wherein one or more center-of-attraction parameters are updated when solving the unconstrained convex optimization problem to ensure that the input satisfies the one or more equality constraints within a tolerance, wherein the input represents an estimated state of the power system, and wherein the initial iteration of the solution process further comprises initializing a value of each center-of-attraction parameter in each parameterized potential function in the unconstrained convex optimization problem, determining whether or not the one or more equality constraints are violated by the initial optimizing input, when determining that the one or more equality constraints are not violated, outputting the initial optimizing input as an output estimated state of the power system, and when determining that at least one of the one or more equality constraints is violated, until the one or more equality constraints are not violated, over one or more subsequent iterations, performing the solution process, wherein the subsequent iteration of the solution process further comprises updating the value of each center-of-attraction parameter in each parameterized potential function corresponding to the one or more equality constraints, prior to solving the unconstrained convex optimization problem, and outputting the input found by the solution process in a final one of the one or more subsequent iterations as the output estimated state of the power system.
2. The method of claim 1, further comprising using the at least one hardware processor to control the power system based on the estimated state.
3. The method of claim 1, wherein each parametric potential function is defined as: $w(c(x)-u).\text{sup}.2$ wherein w is a weight, x is the input, $c(x)$ is the equality function, and u is the center-of-attraction parameter.
4. The method of claim 3, wherein the second objective function is defined as: $f(x) + \underset{i}{.\text{Math.}} \, w_i(c_i(x) - u_i)^2$ wherein $f(x)$ is the first objective function, and i is an indexed set of equality constraints for each of which a parameterized potential function is included in the unconstrained convex optimization problem.
5. The method of claim 1, further comprising using the at least one hardware processor to control the power system based on the output estimated state.
6. The method of claim 1, wherein the one or more equality constraints are determined to be violated when at least one of the one or more equality constraints is outside of the tolerance, and wherein the one or more equality constraints are determined not to be violated when all of the one or more equality constraints are within the tolerance.
7. The method of claim 1, wherein, in the initial iteration and each of the one or more subsequent iterations, the unconstrained convex optimization problem comprises a parameterized potential function for all of the one or more equality constraints.
8. The method of claim 1, wherein, in each of the one or more subsequent iterations, an initial input used in solving the unconstrained convex optimization problem in the solution process of the

subsequent iteration is the input found by the solution process in an immediately preceding iteration.

9. The method of claim 1, wherein, in each of the one or more subsequent iterations, updating the value of each center-of-attraction parameter comprises calculating the value of each center-of-attraction parameter based on a prior value of that center-of-attraction parameter in an immediately preceding iteration.

10. The method of claim 9, wherein, in each of the one or more subsequent iterations, updating the value of each center-of-attraction parameter comprises calculating the value of each center-of-attraction parameter further based on a value of the equality function for a corresponding one of the one or more equality constraints given the input found by the solution process in an immediately preceding iteration.

11. The method of claim 10, wherein the value of each center-of-attraction parameter in a subsequent iteration k is calculated as follows:

$u(k)=u.\sup.(k-1)−C(x.\sup.(k-1))$ wherein u.sup.(k) is the value of the center-of-attraction parameter in the subsequent iteration k, u.sup.(k-1) is the value of the center-of-attraction parameter in an immediately preceding iteration k−1, x.sup.(k-1) is the input found in the immediately preceding iteration k−1, and c(x.sup.(-1) is a value of the equality function in the parameterized potential function comprising the center-of-attraction parameter, given the input x.sup.(k-1).

12. The method of claim 1, wherein the constrained convex optimization problem further comprises one or more inequality constraints, and the method further comprises using the at least one hardware processor to: perform the first process, while ignoring the one or more inequality constraints, to find the output estimated state as a first initial optimizing input; determine whether or not the one or more inequality constraints are violated by the first initial optimizing input; when determining that the one or more inequality constraints are not violated, use the first initial optimizing input as a final estimated state of the power system; when determining that at least one of the one or more inequality constraints is violated, until the one or more inequality constraints are not violated, over one or more outer iterations, convert the constrained optimization problem into an inequality-unconstrained convex optimization problem comprising the first objective function and an inequality parameterized potential function for each of at least one of the one or more inequality constraints, wherein each inequality parameterized potential function is defined by an inequality function and a center-of-attraction parameter, determine a value of each center-of-attraction parameter in each inequality parameterized potential function in the inequality-unconstrained convex optimization problem, and perform the first process using the inequality-unconstrained convex optimization problem as the constrained optimization problem in the first process, and output the input found by the first process in a final one of the one or more outer iterations as the final estimated state of the power system.

13. The method of claim 12, wherein, in each of the one or more outer iterations, an inequality parameterized potential function is added to the inequality-unconstrained convex optimization problem for each of the one or more inequality constraints that are violated and none of the one or more inequality constraints that have not been violated.

14. The method of claim 12, wherein, for an inequality constraint expressed as g(x)≤0, each inequality parameterized potential function component is defined as:

$w(g(x)−u).\sup.2$ wherein w is a weight, x is the input, g(x) is the inequality function, and u is the center-of-attraction parameter.

15. The method of claim 1, further comprising using the estimated state in one or more of a contingency analysis, a Volt-Var optimization, or an optimal power flow.

16. The method of claim 1, wherein the first objective function comprises an error calculation in which a value of a measurement function, given the input, is subtracted from a value of a system telemetry.

17. A system comprising: at least one hardware processor; and software configured to, when executed by the at least one hardware processor, acquire a constrained optimization problem comprising a convex first objective function to be minimized subject to one or more equality constraints, convert the constrained optimization problem into an unconstrained convex optimization problem comprising a second objective function that is a sum of the first objective function and a parameterized potential function for each of at least one of the one or more equality constraints, wherein each parameterized

potential function is defined by an equality function and a center-of-attraction parameter, and perform a first process comprising performing an initial iteration of a solution process to find an initial optimizing input, wherein the solution process comprises solving the unconstrained convex optimization problem by finding an input to the unconstrained convex optimization problem that minimizes an output of the unconstrained convex optimization problem, wherein one or more center-of-attraction parameters are updated when solving the unconstrained convex optimization problem to ensure that the input satisfies the one or more equality constraints within a tolerance, wherein the input represents an estimated state of the power system, and wherein the initial iteration of the solution process further comprises initializing a value of each center-of-attraction parameter in each parameterized potential function in the unconstrained convex optimization problem, determining whether or not the one or more equality constraints are violated by the initial optimizing input, when determining that the one or more equality constraints are not violated, outputting the initial optimizing input as an output estimated state of the power system, and when determining that at least one of the one or more equality constraints is violated, until the one or more equality constraints are not violated, over one or more subsequent iterations, performing the solution process, wherein the subsequent iteration of the solution process further comprises updating the value of each center-of-attraction parameter in each parameterized potential function corresponding to the one or more equality constraints, prior to solving the unconstrained convex optimization problem, and outputting the input found by the solution process in a final one of the one or more subsequent iterations as the output estimated state of the power system.