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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 July 2018 (12.07.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/128652 A1

(51) International Patent Classification:
G06Q 10/06 (2012.01) *G06Q 50/08* (2012.01)
G06N 5/04 (2006.01)

(21) International Application Number:
PCT/US2017/052702

(22) International Filing Date:
21 September 2017 (21.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/400,926 06 January 2017 (06.01.2017) US

(71) Applicant: **JOHNSON CONTROLS TECHNOLOGY COMPANY** [US/US]; 5757 North Green Bay Avenue, Milwaukee, Wisconsin 53209 (US).

(72) Inventors: **SINHA, Sudhi**; 1918 E. Lafayette Place, Apt. 1601, Milwaukee, Wisconsin 53202 (US). **PARK, Youngchoon**; 2715 Chadwick Ct., Brookfield, Wisconsin 53045 (US).

(74) Agent: **ZIEBERT, Joseph N.** et al.; Foley & Lardner LLP, 3000 K Street NW Suite 600, Washington, District of Columbia 20007 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: HVAC SYSTEM WITH AUTOMATED DEVICE PAIRING

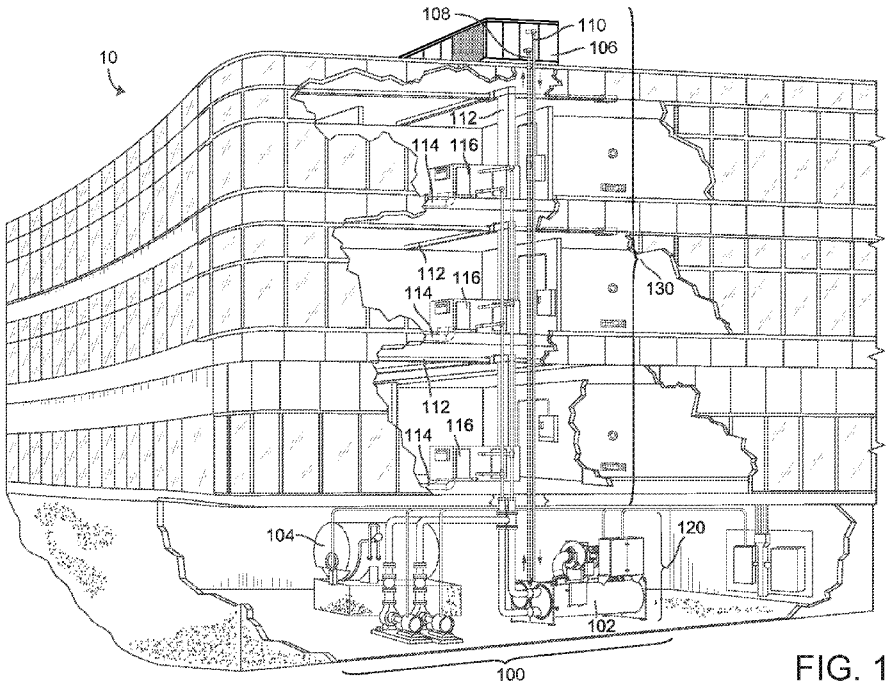


FIG. 1

(57) Abstract: A heating, ventilation, or air conditioning (HVAC) system for a building includes a plurality of actuation devices operable to affect one or more variables in the building, a plurality of sensors configured to measure the variables affected by the actuation devices, and a controller. The controller is configured to operate the actuation devices to affect one or more of the measured variables by providing an actuation signal to the actuation devices and to receive sensor response signals from the sensors. The sensor response signals indicate an effect of the actuation signal on the measured variables. For each of the sensor response signals, the controller is configured to calculate a similarity metric indicating a similarity between the sensor response signal and the actuation signal. The controller is configured to automatically establish a device pairing including one of the actuation devices and one of the sensors based on the similarity metrics.

WO 2018/128652 A1

[Continued on next page]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

HVAC SYSTEM WITH AUTOMATED DEVICE PAIRING

CROSS-REFERENCE TO RELATED PATENT APPLICATION

[0001] This application claims the benefit of and priority to U.S. Patent Application No. 15/400,926 filed January 6, 2017, the entire disclosure of which is incorporated by reference herein.

BACKGROUND

[0002] The present disclosure relates generally to heating, ventilation, or air conditioning (HVAC) systems for a building. The present disclosure relates more particularly to systems and methods for automatically establishing relationships between sensors and actuation devices in a building HVAC system.

[0003] Building HVAC systems typically include many different sensors and actuation devices. The sensors measure various environmental variables (e.g., temperature, humidity, air flow, etc.) and provide sensor readings to a controller. In a feedback control system, a controller uses the sensor readings to generate appropriate control signals for the actuation devices (e.g., chillers, boilers, valves, actuators, etc.) which operate to affect the environmental variables measured by the sensors. Some buildings have hundreds of sensors and actuation devices. It can be cumbersome, error prone, and time consuming to manually identify the associations between sensors and their corresponding actuation devices. Additionally, buildings evolve over time which can change existing relationships between sensors and actuation devices. For example, remodeling a building can break existing relationships between paired sensors and actuation devices or create new relationships that did not previously exist.

[0004] Some building HVAC systems record timeseries data (e.g., trend data, historical data, etc.) for various measured or calculated variables. Timeseries data can be used for many purposes including, for example, fault detection, benchmarking, executing queries, and other data analysis applications. Some data analysis applications compare two or more timeseries as part of the analysis. However, it can be difficult to compare timeseries to each other due to dimensional mismatches resulting from different sampling rates, different units of measurement, and other factors. Without the ability to accurately compare timeseries data, performing actions such as finding similar trends, executing queries on a broad data

set, running diagnostic algorithms, conducting benchmarking, reporting on compliance across portfolio of systems or buildings, and other actions can be challenging.

SUMMARY

[0005] One implementation of the present disclosure is a heating, ventilation, or air conditioning (HVAC) system for a building. The HVAC system includes a number of actuation devices operable to affect one or more variables in the building, a number of sensors configured to measure the variables affected by the actuation devices, and a controller. The controller is configured to operate the actuation devices to affect one or more of the measured variables by providing an actuation signal to the actuation devices and to receive sensor response signals from the sensors. The sensor response signals indicate an effect of the actuation signal on the measured variables. For each of the sensor response signals, the controller is configured to calculate a similarity metric indicating a similarity between the sensor response signal and the actuation signal. The controller is configured to automatically establish a device pairing including one of the actuation devices and one of the sensors based on the similarity metrics.

[0006] In some embodiments, the device pairing defines a control relationship between the actuation device in the device pairing and the sensor in the device pairing. The control relationship can indicate that the actuation device in the device pairing is operable to control a variable measured by the sensor in the device pairing.

[0007] In some embodiments, the controller is configured to automatically create a feedback control loop including the actuation device in the device pairing and the sensor in the device pairing. The controller can use the feedback control loop to generate and provide control signals to the actuation device in the device pairing based on measurements received from the sensor in the device pairing.

[0008] In some embodiments, the controller is configured to calculate the similarity metrics based on differences between samples of the actuation signal and corresponding samples of each of the sensor response signals.

[0009] In some embodiments, the controller is configured to determine, for each of the sensor response signals, a delay time of the sensor response signal relative to the actuation signal. The controller can identify a sensor corresponding to the sensor response signal

having a minimum of the delay times and can establish the device pairing such that the identified sensor is included in the device pairing.

[0010] In some embodiments, the controller is configured to generate an actuation signal timeseries including a plurality of samples of the actuation signal and generate a sensor response timeseries for each of the sensor response signals. Each sensor response timeseries may include a plurality of samples of one of the measured variables. In some embodiments, the controller calculates the similarity metrics by comparing the actuation signal timeseries to each of the sensor response timeseries.

[0011] In some embodiments, the controller is configured to detect a dimensional mismatch between the actuation signal timeseries and one or more of the sensor response timeseries and correct the dimensional mismatch by modifying at least one of the actuation signal timeseries and one or more of the sensor response timeseries.

[0012] In some embodiments, the controller is configured to apply a discrete cosine transformation (DCT) to the actuation signal and each of the sensor response signals. Each DCT may generate a plurality of DCT coefficients. The controller can calculate the similarity metrics by comparing the DCT coefficients resulting from the DCT of the actuation signal to DCT coefficients resulting from the DCT of each sensor response signal.

[0013] In some embodiments, the controller is configured to receive baseline sensor signals from each of the plurality of sensors. The baseline sensor signals may indicate values of the measured variables during a time period before the actuation signal is provided to the actuation devices. For each of the baseline sensor signals, the controller can calculate a similarity metric indicating a similarity between the baseline sensor signal and the actuation signal.

[0014] In some embodiments, the controller is configured to determine, for each of the plurality of sensors, whether the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal. In some embodiments, the controller establishes the device pairing in response to a determination that the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal.

[0015] Another implementation of the present disclosure is a method for establishing device pairings in a heating, ventilation, or air conditioning (HVAC) system for a building.

The method includes operating one or more actuation devices to affect one or more measured variables in the building by providing an actuation signal to the actuation devices and receiving sensor response signals from a plurality of sensors configured to measure the variables affected by the actuation devices. The sensor response signals indicate an effect of the actuation signal on the measured variables. The method includes calculating a similarity metric for each of the sensor response signals. Each similarity metric indicates a similarity between the actuation signal and one of the sensor response signals. The method includes automatically establishing a device pairing including one of the actuation devices and one of the sensors based on the similarity metrics.

[0016] In some embodiments, the device pairing defines a control relationship between the actuation device in the device pairing and the sensor in the device pairing. The control relationship may indicate that the actuation device in the device pairing is operable to control a variable measured by the sensor in the device pairing.

[0017] In some embodiments, the method includes automatically creating a feedback control loop including the actuation device in the device pairing and the sensor in the device pairing. The method may include using the feedback control loop to generate and provide control signals to the actuation device in the device pairing based on measurements received from the sensor in the device pairing.

[0018] In some embodiments, the similarity metrics are calculated based on differences between samples of the actuation signal and corresponding samples of each of the sensor response signals.

[0019] In some embodiments, the method includes determining, for each of the sensor response signals, a delay time of the sensor response signal relative to the actuation signal. The method may include identifying a sensor corresponding to the sensor response signal having a minimum of the delay times and establishing the device pairing such that the identified sensor is included in the device pairing.

[0020] In some embodiments, the method includes generating an actuation signal timeseries including a plurality of samples of the actuation signal and generating a sensor response timeseries for each of the sensor response signals. Each sensor response timeseries may include a plurality of samples of one of the measured variables. The method may include calculating the similarity metrics by comparing the actuation signal timeseries to each of the sensor response timeseries.

[0021] In some embodiments, the method includes detecting a dimensional mismatch between the actuation signal timeseries and one or more of the sensor response timeseries and correcting the dimensional mismatch by modifying at least one of the actuation signal timeseries and one or more of the sensor response timeseries.

[0022] In some embodiments, the method includes applying a discrete cosine transformation (DCT) to the actuation signal and each of the sensor response signals. Each DCT may generate a plurality of DCT coefficients. The method may include calculating the similarity metrics by comparing the DCT coefficients resulting from the DCT of the actuation signal to DCT coefficients resulting from the DCT of each sensor response signal.

[0023] In some embodiments, the method includes receiving baseline sensor signals from each of the plurality of sensors. The baseline sensor signals may indicate values of the measured variables during a time period before the actuation signal is provided to the actuation devices. The method may include, for each of the baseline sensor signals, calculating a similarity metric indicating a similarity between the baseline sensor signal and the actuation signal.

[0024] In some embodiments, the method includes determining, for each of the plurality of sensors, whether the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal. The method may include establishing the device pairing in response to a determination that the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal.

[0025] Those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects, inventive features, and advantages of the devices and/or processes described herein, as defined solely by the claims, will become apparent in the detailed description set forth herein and taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] Some embodiments will become more fully understood from the following detailed description, taken in conjunction with the accompanying drawings, wherein like reference numerals refer to like elements, in which:

[0027] FIG. 1 is a drawing of a building equipped with a heating, ventilation, or air conditioning (HVAC) system, according to an exemplary embodiment;

[0028] FIG. 2 is a drawing of a waterside system which can be used in combination with the HVAC system of FIG. 1, according to an exemplary embodiment;

[0029] FIG. 3 is a drawing of an airside system which can be used in combination with the HVAC system of FIG. 1, according to an exemplary embodiment;

[0030] FIG. 4 is a block diagram of a building management system which can be used to monitor and control the building and HVAC system of FIG. 1, according to an exemplary embodiment;

[0031] FIG. 5 is a block diagram of another building management system which can be used to monitor and control the building and HVAC system of FIG. 1, according to an exemplary embodiment;

[0032] FIG. 6A is a block diagram of a HVAC system including sensors, actuation devices, and a controller, which can be implemented in the building of FIG. 1, according to an exemplary embodiment;

[0033] FIG. 6B is a block diagram illustrating a portion of the HVAC system of FIG. 6A in greater detail including a smart actuator which can be configured to perform automated device pairing, according to an exemplary embodiment;

[0034] FIG. 6C is a block diagram illustrating a portion of the HVAC system of FIG. 6A in greater detail including a smart chiller which can be configured to perform automated device pairing, according to an exemplary embodiment;

[0035] FIG. 6D is a block diagram illustrating a portion of the HVAC system of FIG. 6A in greater detail including a smart thermostat which can be configured to perform automated device pairing, according to an exemplary embodiment;

[0036] FIG. 7 is a graph illustrating different types of signals and timeseries evaluated by the controller of FIG. 6A, according to an exemplary embodiment;

[0037] FIG. 8 is a graph illustrating dimensional mismatch handling performed by the controller of FIG. 6A, according to an exemplary embodiment;

[0038] FIG. 9 is a flowchart of a process for establishing device pairings between sensors and actuation devices, which can be performed by the controller of FIG. 6A, according to an exemplary embodiment; and

[0039] FIG. 10 is a flowchart of a process for handling dimensional mismatches between actuation signal timeseries and sensor response timeseries, which can be performed by the controller of FIG. 6A, according to an exemplary embodiment.

DETAILED DESCRIPTION

[0040] Referring generally to the FIGURES, a heating, ventilation, or air conditioning (HVAC) system with automated device pairing and dimensional mismatch handling are shown according to various exemplary embodiments. The HVAC system includes a plurality of sensors and actuation devices (e.g., chillers, boilers, fans, dampers, actuators, valves, etc.). The sensors measure various environmental variables in the building (e.g., zone temperature, humidity, air flow, etc.). The actuation devices operate to affect the measured variables by providing heating, cooling, airflow, etc. to the building. A controller provides actuation signals to the actuation devices and receives sensor response signals from the sensors. The controller uses the sensor response signals to determine an effect of the actuation signals on the measured variables.

[0041] In some embodiments, the controller automatically establishes device pairings between sensors and actuation devices based on the sensor response signals. For each combination of an actuation signal and a sensor response signal, the controller can calculate a similarity metric. The similarity metric indicates the similarity or closeness between the sensor response signal and the actuation signal. The controller can use the similarity metrics to identify which of the sensor response signals most closely matches each actuation signal. The controller can then establish a device pairing between the actuation device and the sensor corresponding to the matching actuation signal and sensor response signal.

[0042] In some embodiments, the controller stores timeseries data for the actuation signals and sensor response signals. Different variables can be measured at different sampling rates, which can lead do dimensional mismatches between two or more timeseries that span the same range of times. For example, a timeseries sampled at a rate of ten samples per second may include twice the number of samples as a different timeseries sampled at a rate of five samples per second. The controller can automatically handle dimensional

mismatches between two or more timeseries by performing a discrete cosine transformation for each timeseries.

[0043] A discrete cosine transformation (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. Performing the DCT may result in a set of DCT coefficients for each timeseries. The DCT coefficients represent the magnitudes of the cosine functions in the summation. The controller can apply a quantization process to the DCT coefficients in each set such that only a predetermined number of the DCT coefficients in each set are retained. The remaining DCT coefficients can be discarded or replaced with zeros, which has the effect of removing some of the higher frequency cosine functions from the summation. The controller can compare two or more timeseries by comparing the DCT coefficients resulting from each DCT. Advantageously, this allows for direct comparison between the transformed timeseries without requiring decompression, interpolation, synchronization, or other processing steps. Other features and advantages of the HVAC system and controller are described in greater detail below.

Building HVAC Systems and Building Management Systems

[0044] Referring now to FIGS. 1-5, several building management systems (BMS) and HVAC systems in which the systems and methods of the present disclosure can be implemented are shown, according to some embodiments. In brief overview, FIG. 1 shows a building 10 equipped with a HVAC system 100. FIG. 2 is a block diagram of a waterside system 200 which can be used to serve building 10. FIG. 3 is a block diagram of an airside system 300 which can be used to serve building 10. FIG. 4 is a block diagram of a BMS which can be used to monitor and control building 10. FIG. 5 is a block diagram of another BMS which can be used to monitor and control building 10.

Building and HVAC System

[0045] Referring particularly to FIG. 1, a perspective view of a building 10 is shown. Building 10 is served by a BMS. A BMS is, in general, a system of devices configured to control, monitor, and manage equipment in or around a building or building area. A BMS can include, for example, a HVAC system, a security system, a lighting system, a fire alerting system, any other system that is capable of managing building functions or devices, or any combination thereof.

[0046] The BMS that serves building 10 includes a HVAC system 100. HVAC system 100 can include a plurality of HVAC devices (e.g., heaters, chillers, air handling units, pumps, fans, thermal energy storage, etc.) configured to provide heating, cooling, ventilation, or other services for building 10. For example, HVAC system 100 is shown to include a waterside system 120 and an airside system 130. Waterside system 120 may provide a heated or chilled fluid to an air handling unit of airside system 130. Airside system 130 may use the heated or chilled fluid to heat or cool an airflow provided to building 10. An exemplary waterside system and airside system which can be used in HVAC system 100 are described in greater detail with reference to FIGS. 2-3.

[0047] HVAC system 100 is shown to include a chiller 102, a boiler 104, and a rooftop air handling unit (AHU) 106. Waterside system 120 may use boiler 104 and chiller 102 to heat or cool a working fluid (e.g., water, glycol, etc.) and may circulate the working fluid to AHU 106. In various embodiments, the HVAC devices of waterside system 120 can be located in or around building 10 (as shown in FIG. 1) or at an offsite location such as a central plant (e.g., a chiller plant, a steam plant, a heat plant, etc.). The working fluid can be heated in boiler 104 or cooled in chiller 102, depending on whether heating or cooling is required in building 10. Boiler 104 may add heat to the circulated fluid, for example, by burning a combustible material (e.g., natural gas) or using an electric heating element. Chiller 102 may place the circulated fluid in a heat exchange relationship with another fluid (e.g., a refrigerant) in a heat exchanger (e.g., an evaporator) to absorb heat from the circulated fluid. The working fluid from chiller 102 and/or boiler 104 can be transported to AHU 106 via piping 108.

[0048] AHU 106 may place the working fluid in a heat exchange relationship with an airflow passing through AHU 106 (e.g., via one or more stages of cooling coils and/or heating coils). The airflow can be, for example, outside air, return air from within building 10, or a combination of both. AHU 106 may transfer heat between the airflow and the working fluid to provide heating or cooling for the airflow. For example, AHU 106 can include one or more fans or blowers configured to pass the airflow over or through a heat exchanger containing the working fluid. The working fluid may then return to chiller 102 or boiler 104 via piping 110.

[0049] Airside system 130 may deliver the airflow supplied by AHU 106 (i.e., the supply airflow) to building 10 via air supply ducts 112 and may provide return air from building 10 to AHU 106 via air return ducts 114. In some embodiments, airside system 130 includes

multiple variable air volume (VAV) units 116. For example, airside system 130 is shown to include a separate VAV unit 116 on each floor or zone of building 10. VAV units 116 can include dampers or other flow control elements that can be operated to control an amount of the supply airflow provided to individual zones of building 10. In other embodiments, airside system 130 delivers the supply airflow into one or more zones of building 10 (e.g., via supply ducts 112) without using intermediate VAV units 116 or other flow control elements. AHU 106 can include various sensors (e.g., temperature sensors, pressure sensors, etc.) configured to measure attributes of the supply airflow. AHU 106 may receive input from sensors located within AHU 106 and/or within the building zone and may adjust the flow rate, temperature, or other attributes of the supply airflow through AHU 106 to achieve setpoint conditions for the building zone.

Waterside System

[0050] Referring now to FIG. 2, a block diagram of a waterside system 200 is shown, according to some embodiments. In various embodiments, waterside system 200 may supplement or replace waterside system 120 in HVAC system 100 or can be implemented separate from HVAC system 100. When implemented in HVAC system 100, waterside system 200 can include a subset of the HVAC devices in HVAC system 100 (e.g., boiler 104, chiller 102, pumps, valves, etc.) and may operate to supply a heated or chilled fluid to AHU 106. The HVAC devices of waterside system 200 can be located within building 10 (e.g., as components of waterside system 120) or at an offsite location such as a central plant.

[0051] In FIG. 2, waterside system 200 is shown as a central plant having a plurality of subplants 202-212. Subplants 202-212 are shown to include a heater subplant 202, a heat recovery chiller subplant 204, a chiller subplant 206, a cooling tower subplant 208, a hot thermal energy storage (TES) subplant 210, and a cold thermal energy storage (TES) subplant 212. Subplants 202-212 consume resources (e.g., water, natural gas, electricity, etc.) from utilities to serve thermal energy loads (e.g., hot water, cold water, heating, cooling, etc.) of a building or campus. For example, heater subplant 202 can be configured to heat water in a hot water loop 214 that circulates the hot water between heater subplant 202 and building 10. Chiller subplant 206 can be configured to chill water in a cold water loop 216 that circulates the cold water between chiller subplant 206 building 10. Heat recovery chiller subplant 204 can be configured to transfer heat from cold water loop 216 to hot water loop 214 to provide additional heating for the hot water and additional cooling for

the cold water. Condenser water loop 218 may absorb heat from the cold water in chiller subplant 206 and reject the absorbed heat in cooling tower subplant 208 or transfer the absorbed heat to hot water loop 214. Hot TES subplant 210 and cold TES subplant 212 may store hot and cold thermal energy, respectively, for subsequent use.

[0052] Hot water loop 214 and cold water loop 216 may deliver the heated and/or chilled water to air handlers located on the rooftop of building 10 (e.g., AHU 106) or to individual floors or zones of building 10 (e.g., VAV units 116). The air handlers push air past heat exchangers (e.g., heating coils or cooling coils) through which the water flows to provide heating or cooling for the air. The heated or cooled air can be delivered to individual zones of building 10 to serve thermal energy loads of building 10. The water then returns to subplants 202-212 to receive further heating or cooling.

[0053] Although subplants 202-212 are shown and described as heating and cooling water for circulation to a building, it is understood that any other type of working fluid (e.g., glycol, CO2, etc.) can be used in place of or in addition to water to serve thermal energy loads. In other embodiments, subplants 202-212 may provide heating and/or cooling directly to the building or campus without requiring an intermediate heat transfer fluid. These and other variations to waterside system 200 are within the teachings of the present disclosure.

[0054] Each of subplants 202-212 can include a variety of equipment configured to facilitate the functions of the subplant. For example, heater subplant 202 is shown to include a plurality of heating elements 220 (e.g., boilers, electric heaters, etc.) configured to add heat to the hot water in hot water loop 214. Heater subplant 202 is also shown to include several pumps 222 and 224 configured to circulate the hot water in hot water loop 214 and to control the flow rate of the hot water through individual heating elements 220. Chiller subplant 206 is shown to include a plurality of chillers 232 configured to remove heat from the cold water in cold water loop 216. Chiller subplant 206 is also shown to include several pumps 234 and 236 configured to circulate the cold water in cold water loop 216 and to control the flow rate of the cold water through individual chillers 232.

[0055] Heat recovery chiller subplant 204 is shown to include a plurality of heat recovery heat exchangers 226 (e.g., refrigeration circuits) configured to transfer heat from cold water loop 216 to hot water loop 214. Heat recovery chiller subplant 204 is also shown to include several pumps 228 and 230 configured to circulate the hot water and/or cold water through

heat recovery heat exchangers 226 and to control the flow rate of the water through individual heat recovery heat exchangers 226. Cooling tower subplant 208 is shown to include a plurality of cooling towers 238 configured to remove heat from the condenser water in condenser water loop 218. Cooling tower subplant 208 is also shown to include several pumps 240 configured to circulate the condenser water in condenser water loop 218 and to control the flow rate of the condenser water through individual cooling towers 238.

[0056] Hot TES subplant 210 is shown to include a hot TES tank 242 configured to store the hot water for later use. Hot TES subplant 210 may also include one or more pumps or valves configured to control the flow rate of the hot water into or out of hot TES tank 242. Cold TES subplant 212 is shown to include cold TES tanks 244 configured to store the cold water for later use. Cold TES subplant 212 may also include one or more pumps or valves configured to control the flow rate of the cold water into or out of cold TES tanks 244.

[0057] In some embodiments, one or more of the pumps in waterside system 200 (e.g., pumps 222, 224, 228, 230, 234, 236, and/or 240) or pipelines in waterside system 200 include an isolation valve associated therewith. Isolation valves can be integrated with the pumps or positioned upstream or downstream of the pumps to control the fluid flows in waterside system 200. In various embodiments, waterside system 200 can include more, fewer, or different types of devices and/or subplants based on the particular configuration of waterside system 200 and the types of loads served by waterside system 200.

Airside System

[0058] Referring now to FIG. 3, a block diagram of an airside system 300 is shown, according to some embodiments. In various embodiments, airside system 300 may supplement or replace airside system 130 in HVAC system 100 or can be implemented separate from HVAC system 100. When implemented in HVAC system 100, airside system 300 can include a subset of the HVAC devices in HVAC system 100 (e.g., AHU 106, VAV units 116, ducts 112-114, fans, dampers, etc.) and can be located in or around building 10. Airside system 300 may operate to heat or cool an airflow provided to building 10 using a heated or chilled fluid provided by waterside system 200.

[0059] In FIG. 3, airside system 300 is shown to include an economizer-type air handling unit (AHU) 302. Economizer-type AHUs vary the amount of outside air and return air used by the air handling unit for heating or cooling. For example, AHU 302 may receive return air 304 from building zone 306 via return air duct 308 and may deliver supply air 310 to

building zone 306 via supply air duct 312. In some embodiments, AHU 302 is a rooftop unit located on the roof of building 10 (e.g., AHU 106 as shown in FIG. 1) or otherwise positioned to receive both return air 304 and outside air 314. AHU 302 can be configured to operate exhaust air damper 316, mixing damper 318, and outside air damper 320 to control an amount of outside air 314 and return air 304 that combine to form supply air 310. Any return air 304 that does not pass through mixing damper 318 can be exhausted from AHU 302 through exhaust damper 316 as exhaust air 322.

[0060] Each of dampers 316-320 can be operated by an actuator. For example, exhaust air damper 316 can be operated by actuator 324, mixing damper 318 can be operated by actuator 326, and outside air damper 320 can be operated by actuator 328. Actuators 324-328 may communicate with an AHU controller 330 via a communications link 332. Actuators 324-328 may receive control signals from AHU controller 330 and may provide feedback signals to AHU controller 330. Feedback signals can include, for example, an indication of a current actuator or damper position, an amount of torque or force exerted by the actuator, diagnostic information (e.g., results of diagnostic tests performed by actuators 324-328), status information, commissioning information, configuration settings, calibration data, and/or other types of information or data that can be collected, stored, or used by actuators 324-328. AHU controller 330 can be an economizer controller configured to use one or more control algorithms (e.g., state-based algorithms, extremum seeking control (ESC) algorithms, proportional-integral (PI) control algorithms, proportional-integral-derivative (PID) control algorithms, model predictive control (MPC) algorithms, feedback control algorithms, etc.) to control actuators 324-328.

[0061] Still referring to FIG. 3, AHU 302 is shown to include a cooling coil 334, a heating coil 336, and a fan 338 positioned within supply air duct 312. Fan 338 can be configured to force supply air 310 through cooling coil 334 and/or heating coil 336 and provide supply air 310 to building zone 306. AHU controller 330 may communicate with fan 338 via communications link 340 to control a flow rate of supply air 310. In some embodiments, AHU controller 330 controls an amount of heating or cooling applied to supply air 310 by modulating a speed of fan 338.

[0062] Cooling coil 334 may receive a chilled fluid from waterside system 200 (e.g., from cold water loop 216) via piping 342 and may return the chilled fluid to waterside system 200 via piping 344. Valve 346 can be positioned along piping 342 or piping 344 to control a flow rate of the chilled fluid through cooling coil 334. In some embodiments, cooling coil

334 includes multiple stages of cooling coils that can be independently activated and deactivated (e.g., by AHU controller 330, by BMS controller 366, etc.) to modulate an amount of cooling applied to supply air 310.

[0063] Heating coil 336 may receive a heated fluid from waterside system 200(e.g., from hot water loop 214) via piping 348 and may return the heated fluid to waterside system 200 via piping 350. Valve 352 can be positioned along piping 348 or piping 350 to control a flow rate of the heated fluid through heating coil 336. In some embodiments, heating coil 336 includes multiple stages of heating coils that can be independently activated and deactivated (e.g., by AHU controller 330, by BMS controller 366, etc.) to modulate an amount of heating applied to supply air 310.

[0064] Each of valves 346 and 352 can be controlled by an actuator. For example, valve 346 can be controlled by actuator 354 and valve 352 can be controlled by actuator 356. Actuators 354-356 may communicate with AHU controller 330 via communications links 358-360. Actuators 354-356 may receive control signals from AHU controller 330 and may provide feedback signals to controller 330. In some embodiments, AHU controller 330 receives a measurement of the supply air temperature from a temperature sensor 362 positioned in supply air duct 312 (e.g., downstream of cooling coil 334 and/or heating coil 336). AHU controller 330 may also receive a measurement of the temperature of building zone 306 from a temperature sensor 364 located in building zone 306.

[0065] In some embodiments, AHU controller 330 operates valves 346 and 352 via actuators 354-356 to modulate an amount of heating or cooling provided to supply air 310 (e.g., to achieve a setpoint temperature for supply air 310 or to maintain the temperature of supply air 310 within a setpoint temperature range). The positions of valves 346 and 352 affect the amount of heating or cooling provided to supply air 310 by cooling coil 334 or heating coil 336 and may correlate with the amount of energy consumed to achieve a desired supply air temperature. AHU 330 may control the temperature of supply air 310 and/or building zone 306 by activating or deactivating coils 334-336, adjusting a speed of fan 338, or a combination of both.

[0066] Still referring to FIG. 3, airside system 300 is shown to include a building management system (BMS) controller 366 and a client device 368. BMS controller 366 can include one or more computer systems (e.g., servers, supervisory controllers, subsystem controllers, etc.) that serve as system level controllers, application or data servers, head

nodes, or master controllers for airside system 300, waterside system 200, HVAC system 100, and/or other controllable systems that serve building 10. BMS controller 366 may communicate with multiple downstream building systems or subsystems (e.g., HVAC system 100, a security system, a lighting system, waterside system 200, etc.) via a communications link 370 according to like or disparate protocols (e.g., LON, BACnet, etc.). In various embodiments, AHU controller 330 and BMS controller 366 can be separate (as shown in FIG. 3) or integrated. In an integrated implementation, AHU controller 330 can be a software module configured for execution by a processor of BMS controller 366.

[0067] In some embodiments, AHU controller 330 receives information from BMS controller 366 (e.g., commands, setpoints, operating boundaries, etc.) and provides information to BMS controller 366 (e.g., temperature measurements, valve or actuator positions, operating statuses, diagnostics, etc.). For example, AHU controller 330 may provide BMS controller 366 with temperature measurements from temperature sensors 362-364, equipment on/off states, equipment operating capacities, and/or any other information that can be used by BMS controller 366 to monitor or control a variable state or condition within building zone 306.

[0068] Client device 368 can include one or more human-machine interfaces or client interfaces (e.g., graphical user interfaces, reporting interfaces, text-based computer interfaces, client-facing web services, web servers that provide pages to web clients, etc.) for controlling, viewing, or otherwise interacting with HVAC system 100, its subsystems, and/or devices. Client device 368 can be a computer workstation, a client terminal, a remote or local interface, or any other type of user interface device. Client device 368 can be a stationary terminal or a mobile device. For example, client device 368 can be a desktop computer, a computer server with a user interface, a laptop computer, a tablet, a smartphone, a PDA, or any other type of mobile or non-mobile device. Client device 368 may communicate with BMS controller 366 and/or AHU controller 330 via communications link 372.

Building Management Systems

[0069] Referring now to FIG. 4, a block diagram of a building management system (BMS) 400 is shown, according to some embodiments. BMS 400 can be implemented in building 10 to automatically monitor and control various building functions. BMS 400 is shown to include BMS controller 366 and a plurality of building subsystems 428. Building

subsystems 428 are shown to include a building electrical subsystem 434, an information communication technology (ICT) subsystem 436, a security subsystem 438, a HVAC subsystem 440, a lighting subsystem 442, a lift/escalators subsystem 432, and a fire safety subsystem 430. In various embodiments, building subsystems 428 can include fewer, additional, or alternative subsystems. For example, building subsystems 428 may also or alternatively include a refrigeration subsystem, an advertising or signage subsystem, a cooking subsystem, a vending subsystem, a printer or copy service subsystem, or any other type of building subsystem that uses controllable equipment and/or sensors to monitor or control building 10. In some embodiments, building subsystems 428 include waterside system 200 and/or airside system 300, as described with reference to FIGS. 2-3.

[0070] Each of building subsystems 428 can include any number of devices, controllers, and connections for completing its individual functions and control activities. HVAC subsystem 440 can include many of the same components as HVAC system 100, as described with reference to FIGS. 1-3. For example, HVAC subsystem 440 can include a chiller, a boiler, any number of air handling units, economizers, field controllers, supervisory controllers, actuators, temperature sensors, and other devices for controlling the temperature, humidity, airflow, or other variable conditions within building 10. Lighting subsystem 442 can include any number of light fixtures, ballasts, lighting sensors, dimmers, or other devices configured to controllably adjust the amount of light provided to a building space. Security subsystem 438 can include occupancy sensors, video surveillance cameras, digital video recorders, video processing servers, intrusion detection devices, access control devices and servers, or other security-related devices.

[0071] Still referring to FIG. 4, BMS controller 366 is shown to include a communications interface 407 and a BMS interface 409. Interface 407 may facilitate communications between BMS controller 366 and external applications (e.g., monitoring and reporting applications 422, enterprise control applications 426, remote systems and applications 444, applications residing on client devices 448, etc.) for allowing user control, monitoring, and adjustment to BMS controller 366 and/or subsystems 428. Interface 407 may also facilitate communications between BMS controller 366 and client devices 448. BMS interface 409 may facilitate communications between BMS controller 366 and building subsystems 428 (e.g., HVAC, lighting security, lifts, power distribution, business, etc.).

[0072] Interfaces 407, 409 can be or include wired or wireless communications interfaces (e.g., jacks, antennas, transmitters, receivers, transceivers, wire terminals, etc.) for conducting data communications with building subsystems 428 or other external systems or devices. In various embodiments, communications via interfaces 407, 409 can be direct (e.g., local wired or wireless communications) or via a communications network 446 (e.g., a WAN, the Internet, a cellular network, etc.). For example, interfaces 407, 409 can include an Ethernet card and port for sending and receiving data via an Ethernet-based communications link or network. In another example, interfaces 407, 409 can include a Wi-Fi transceiver for communicating via a wireless communications network. In another example, one or both of interfaces 407, 409 can include cellular or mobile phone communications transceivers. In one embodiment, communications interface 407 is a power line communications interface and BMS interface 409 is an Ethernet interface. In other embodiments, both communications interface 407 and BMS interface 409 are Ethernet interfaces or are the same Ethernet interface.

[0073] Still referring to FIG. 4, BMS controller 366 is shown to include a processing circuit 404 including a processor 406 and memory 408. Processing circuit 404 can be communicably connected to BMS interface 409 and/or communications interface 407 such that processing circuit 404 and the various components thereof can send and receive data via interfaces 407, 409. Processor 406 can be implemented as a general purpose processor, an application specific integrated circuit (ASIC), one or more field programmable gate arrays (FPGAs), a group of processing components, or other suitable electronic processing components.

[0074] Memory 408 (e.g., memory, memory unit, storage device, etc.) can include one or more devices (e.g., RAM, ROM, Flash memory, hard disk storage, etc.) for storing data and/or computer code for completing or facilitating the various processes, layers and modules described in the present application. Memory 408 can be or include volatile memory or non-volatile memory. Memory 408 can include database components, object code components, script components, or any other type of information structure for supporting the various activities and information structures described in the present application. According to some embodiments, memory 408 is communicably connected to processor 406 via processing circuit 404 and includes computer code for executing (e.g., by processing circuit 404 and/or processor 406) one or more processes described herein.

[0075] In some embodiments, BMS controller 366 is implemented within a single computer (e.g., one server, one housing, etc.). In various other embodiments BMS controller 366 can be distributed across multiple servers or computers (e.g., that can exist in distributed locations). Further, while FIG. 4 shows applications 422 and 426 as existing outside of BMS controller 366, in some embodiments, applications 422 and 426 can be hosted within BMS controller 366 (e.g., within memory 408).

[0076] Still referring to FIG. 4, memory 408 is shown to include an enterprise integration layer 410, an automated measurement and validation (AM&V) layer 412, a demand response (DR) layer 414, a fault detection and diagnostics (FDD) layer 416, an integrated control layer 418, and a building subsystem integration later 420. Layers 410-420 can be configured to receive inputs from building subsystems 428 and other data sources, determine optimal control actions for building subsystems 428 based on the inputs, generate control signals based on the optimal control actions, and provide the generated control signals to building subsystems 428. The following paragraphs describe some of the general functions performed by each of layers 410-420 in BMS 400.

[0077] Enterprise integration layer 410 can be configured to serve clients or local applications with information and services to support a variety of enterprise-level applications. For example, enterprise control applications 426 can be configured to provide subsystem-spanning control to a graphical user interface (GUI) or to any number of enterprise-level business applications (e.g., accounting systems, user identification systems, etc.). Enterprise control applications 426 may also or alternatively be configured to provide configuration GUIs for configuring BMS controller 366. In yet other embodiments, enterprise control applications 426 can work with layers 410-420 to optimize building performance (e.g., efficiency, energy use, comfort, or safety) based on inputs received at interface 407 and/or BMS interface 409.

[0078] Building subsystem integration layer 420 can be configured to manage communications between BMS controller 366 and building subsystems 428. For example, building subsystem integration layer 420 may receive sensor data and input signals from building subsystems 428 and provide output data and control signals to building subsystems 428. Building subsystem integration layer 420 may also be configured to manage communications between building subsystems 428. Building subsystem integration layer 420 translate communications (e.g., sensor data, input signals, output signals, etc.) across a plurality of multi-vendor/multi-protocol systems.

[0079] Demand response layer 414 can be configured to optimize resource usage (e.g., electricity use, natural gas use, water use, etc.) and/or the monetary cost of such resource usage in response to satisfy the demand of building 10. The optimization can be based on time-of-use prices, curtailment signals, energy availability, or other data received from utility providers, distributed energy generation systems 424, from energy storage 427 (e.g., hot TES 242, cold TES 244, etc.), or from other sources. Demand response layer 414 may receive inputs from other layers of BMS controller 366 (e.g., building subsystem integration layer 420, integrated control layer 418, etc.). The inputs received from other layers can include environmental or sensor inputs such as temperature, carbon dioxide levels, relative humidity levels, air quality sensor outputs, occupancy sensor outputs, room schedules, and the like. The inputs may also include inputs such as electrical use (e.g., expressed in kWh), thermal load measurements, pricing information, projected pricing, smoothed pricing, curtailment signals from utilities, and the like.

[0080] According to some embodiments, demand response layer 414 includes control logic for responding to the data and signals it receives. These responses can include communicating with the control algorithms in integrated control layer 418, changing control strategies, changing setpoints, or activating/deactivating building equipment or subsystems in a controlled manner. Demand response layer 414 may also include control logic configured to determine when to utilize stored energy. For example, demand response layer 414 may determine to begin using energy from energy storage 427 just prior to the beginning of a peak use hour.

[0081] In some embodiments, demand response layer 414 includes a control module configured to actively initiate control actions (e.g., automatically changing setpoints) which minimize energy costs based on one or more inputs representative of or based on demand (e.g., price, a curtailment signal, a demand level, etc.). In some embodiments, demand response layer 414 uses equipment models to determine an optimal set of control actions. The equipment models can include, for example, thermodynamic models describing the inputs, outputs, and/or functions performed by various sets of building equipment. Equipment models may represent collections of building equipment (e.g., subplants, chiller arrays, etc.) or individual devices (e.g., individual chillers, heaters, pumps, etc.).

[0082] Demand response layer 414 may further include or draw upon one or more demand response policy definitions (e.g., databases, XML files, etc.). The policy definitions can be edited or adjusted by a user (e.g., via a graphical user interface) so that

the control actions initiated in response to demand inputs can be tailored for the user's application, desired comfort level, particular building equipment, or based on other concerns. For example, the demand response policy definitions can specify which equipment can be turned on or off in response to particular demand inputs, how long a system or piece of equipment should be turned off, what setpoints can be changed, what the allowable set point adjustment range is, how long to hold a high demand setpoint before returning to a normally scheduled setpoint, how close to approach capacity limits, which equipment modes to utilize, the energy transfer rates (e.g., the maximum rate, an alarm rate, other rate boundary information, etc.) into and out of energy storage devices (e.g., thermal storage tanks, battery banks, etc.), and when to dispatch on-site generation of energy (e.g., via fuel cells, a motor generator set, etc.).

[0083] Integrated control layer 418 can be configured to use the data input or output of building subsystem integration layer 420 and/or demand response later 414 to make control decisions. Due to the subsystem integration provided by building subsystem integration layer 420, integrated control layer 418 can integrate control activities of the subsystems 428 such that the subsystems 428 behave as a single integrated supersystem. In some embodiments, integrated control layer 418 includes control logic that uses inputs and outputs from a plurality of building subsystems to provide greater comfort and energy savings relative to the comfort and energy savings that separate subsystems could provide alone. For example, integrated control layer 418 can be configured to use an input from a first subsystem to make an energy-saving control decision for a second subsystem. Results of these decisions can be communicated back to building subsystem integration layer 420.

[0084] Integrated control layer 418 is shown to be logically below demand response layer 414. Integrated control layer 418 can be configured to enhance the effectiveness of demand response layer 414 by enabling building subsystems 428 and their respective control loops to be controlled in coordination with demand response layer 414. This configuration may advantageously reduce disruptive demand response behavior relative to conventional systems. For example, integrated control layer 418 can be configured to assure that a demand response-driven upward adjustment to the setpoint for chilled water temperature (or another component that directly or indirectly affects temperature) does not result in an increase in fan energy (or other energy used to cool a space) that would result in greater total building energy use than was saved at the chiller.

[0085] Integrated control layer 418 can be configured to provide feedback to demand response layer 414 so that demand response layer 414 checks that constraints (e.g., temperature, lighting levels, etc.) are properly maintained even while demanded load shedding is in progress. The constraints may also include setpoint or sensed boundaries relating to safety, equipment operating limits and performance, comfort, fire codes, electrical codes, energy codes, and the like. Integrated control layer 418 is also logically below fault detection and diagnostics layer 416 and automated measurement and validation layer 412. Integrated control layer 418 can be configured to provide calculated inputs (e.g., aggregations) to these higher levels based on outputs from more than one building subsystem.

[0086] Automated measurement and validation (AM&V) layer 412 can be configured to verify that control strategies commanded by integrated control layer 418 or demand response layer 414 are working properly (e.g., using data aggregated by AM&V layer 412, integrated control layer 418, building subsystem integration layer 420, FDD layer 416, or otherwise). The calculations made by AM&V layer 412 can be based on building system energy models and/or equipment models for individual BMS devices or subsystems. For example, AM&V layer 412 may compare a model-predicted output with an actual output from building subsystems 428 to determine an accuracy of the model.

[0087] Fault detection and diagnostics (FDD) layer 416 can be configured to provide on-going fault detection for building subsystems 428, building subsystem devices (i.e., building equipment), and control algorithms used by demand response layer 414 and integrated control layer 418. FDD layer 416 may receive data inputs from integrated control layer 418, directly from one or more building subsystems or devices, or from another data source. FDD layer 416 may automatically diagnose and respond to detected faults. The responses to detected or diagnosed faults can include providing an alert message to a user, a maintenance scheduling system, or a control algorithm configured to attempt to repair the fault or to work-around the fault.

[0088] FDD layer 416 can be configured to output a specific identification of the faulty component or cause of the fault (e.g., loose damper linkage) using detailed subsystem inputs available at building subsystem integration layer 420. In other exemplary embodiments, FDD layer 416 is configured to provide "fault" events to integrated control layer 418 which executes control strategies and policies in response to the received fault events. According to some embodiments, FDD layer 416 (or a policy executed by an integrated control engine

or business rules engine) may shut-down systems or direct control activities around faulty devices or systems to reduce energy waste, extend equipment life, or assure proper control response.

[0089] FDD layer 416 can be configured to store or access a variety of different system data stores (or data points for live data). FDD layer 416 may use some content of the data stores to identify faults at the equipment level (e.g., specific chiller, specific AHU, specific terminal unit, etc.) and other content to identify faults at component or subsystem levels. For example, building subsystems 428 may generate temporal (i.e., time-series) data indicating the performance of BMS 400 and the various components thereof. The data generated by building subsystems 428 can include measured or calculated values that exhibit statistical characteristics and provide information about how the corresponding system or process (e.g., a temperature control process, a flow control process, etc.) is performing in terms of error from its setpoint. These processes can be examined by FDD layer 416 to expose when the system begins to degrade in performance and alert a user to repair the fault before it becomes more severe.

[0090] Referring now to FIG. 5, a block diagram of another building management system (BMS) 500 is shown, according to some embodiments. BMS 500 can be used to monitor and control the devices of HVAC system 100, waterside system 200, airside system 300, building subsystems 428, as well as other types of BMS devices (e.g., lighting equipment, security equipment, etc.) and/or HVAC equipment.

[0091] BMS 500 provides a system architecture that facilitates automatic equipment discovery and equipment model distribution. Equipment discovery can occur on multiple levels of BMS 500 across multiple different communications busses (e.g., a system bus 554, zone buses 556-560 and 564, sensor/actuator bus 566, etc.) and across multiple different communications protocols. In some embodiments, equipment discovery is accomplished using active node tables, which provide status information for devices connected to each communications bus. For example, each communications bus can be monitored for new devices by monitoring the corresponding active node table for new nodes. When a new device is detected, BMS 500 can begin interacting with the new device (e.g., sending control signals, using data from the device) without user interaction.

[0092] Some devices in BMS 500 present themselves to the network using equipment models. An equipment model defines equipment object attributes, view definitions,

schedules, trends, and the associated BACnet value objects (e.g., analog value, binary value, multistate value, etc.) that are used for integration with other systems. Some devices in BMS 500 store their own equipment models. Other devices in BMS 500 have equipment models stored externally (e.g., within other devices). For example, a zone coordinator 508 can store the equipment model for a bypass damper 528. In some embodiments, zone coordinator 508 automatically creates the equipment model for bypass damper 528 or other devices on zone bus 558. Other zone coordinators can also create equipment models for devices connected to their zone busses. The equipment model for a device can be created automatically based on the types of data points exposed by the device on the zone bus, device type, and/or other device attributes. Several examples of automatic equipment discovery and equipment model distribution are discussed in greater detail below.

[0093] Still referring to FIG. 5, BMS 500 is shown to include a system manager 502; several zone coordinators 506, 508, 510 and 518; and several zone controllers 524, 530, 532, 536, 548, and 550. System manager 502 can monitor data points in BMS 500 and report monitored variables to various monitoring and/or control applications. System manager 502 can communicate with client devices 504 (e.g., user devices, desktop computers, laptop computers, mobile devices, etc.) via a data communications link 574 (e.g., BACnet IP, Ethernet, wired or wireless communications, etc.). System manager 502 can provide a user interface to client devices 504 via data communications link 574. The user interface may allow users to monitor and/or control BMS 500 via client devices 504.

[0094] In some embodiments, system manager 502 is connected with zone coordinators 506-510 and 518 via a system bus 554. System manager 502 can be configured to communicate with zone coordinators 506-510 and 518 via system bus 554 using a master-slave token passing (MSTP) protocol or any other communications protocol. System bus 554 can also connect system manager 502 with other devices such as a constant volume (CV) rooftop unit (RTU) 512, an input/output module (IOM) 514, a thermostat controller 516 (e.g., a TEC5000 series thermostat controller), and a network automation engine (NAE) or third-party controller 520. RTU 512 can be configured to communicate directly with system manager 502 and can be connected directly to system bus 554. Other RTUs can communicate with system manager 502 via an intermediate device. For example, a wired input 562 can connect a third-party RTU 542 to thermostat controller 516, which connects to system bus 554.

[0095] System manager 502 can provide a user interface for any device containing an equipment model. Devices such as zone coordinators 506-510 and 518 and thermostat controller 516 can provide their equipment models to system manager 502 via system bus 554. In some embodiments, system manager 502 automatically creates equipment models for connected devices that do not contain an equipment model (e.g., IOM 514, third party controller 520, etc.). For example, system manager 502 can create an equipment model for any device that responds to a device tree request. The equipment models created by system manager 502 can be stored within system manager 502. System manager 502 can then provide a user interface for devices that do not contain their own equipment models using the equipment models created by system manager 502. In some embodiments, system manager 502 stores a view definition for each type of equipment connected via system bus 554 and uses the stored view definition to generate a user interface for the equipment.

[0096] Each zone coordinator 506-510 and 518 can be connected with one or more of zone controllers 524, 530-532, 536, and 548-550 via zone buses 556, 558, 560, and 564. Zone coordinators 506-510 and 518 can communicate with zone controllers 524, 530-532, 536, and 548-550 via zone busses 556-560 and 564 using a MSTP protocol or any other communications protocol. Zone busses 556-560 and 564 can also connect zone coordinators 506-510 and 518 with other types of devices such as variable air volume (VAV) RTUs 522 and 540, changeover bypass (COBP) RTUs 526 and 552, bypass dampers 528 and 546, and PEAK controllers 534 and 544.

[0097] Zone coordinators 506-510 and 518 can be configured to monitor and command various zoning systems. In some embodiments, each zone coordinator 506-510 and 518 monitors and commands a separate zoning system and is connected to the zoning system via a separate zone bus. For example, zone coordinator 506 can be connected to VAV RTU 522 and zone controller 524 via zone bus 556. Zone coordinator 508 can be connected to COBP RTU 526, bypass damper 528, COBP zone controller 530, and VAV zone controller 532 via zone bus 558. Zone coordinator 510 can be connected to PEAK controller 534 and VAV zone controller 536 via zone bus 560. Zone coordinator 518 can be connected to PEAK controller 544, bypass damper 546, COBP zone controller 548, and VAV zone controller 550 via zone bus 564.

[0098] A single model of zone coordinator 506-510 and 518 can be configured to handle multiple different types of zoning systems (e.g., a VAV zoning system, a COBP zoning system, etc.). Each zoning system can include a RTU, one or more zone controllers, and/or

a bypass damper. For example, zone coordinators 506 and 510 are shown as Verasys VAV engines (VVEs) connected to VAV RTUs 522 and 540, respectively. Zone coordinator 506 is connected directly to VAV RTU 522 via zone bus 556, whereas zone coordinator 510 is connected to a third-party VAV RTU 540 via a wired input 568 provided to PEAK controller 534. Zone coordinators 508 and 518 are shown as Verasys COBP engines (VCEs) connected to COBP RTUs 526 and 552, respectively. Zone coordinator 508 is connected directly to COBP RTU 526 via zone bus 558, whereas zone coordinator 518 is connected to a third-party COBP RTU 552 via a wired input 570 provided to PEAK controller 544.

[0099] Zone controllers 524, 530-532, 536, and 548-550 can communicate with individual BMS devices (e.g., sensors, actuators, etc.) via sensor/actuator (SA) busses. For example, VAV zone controller 536 is shown connected to networked sensors 538 via SA bus 566. Zone controller 536 can communicate with networked sensors 538 using a MSTP protocol or any other communications protocol. Although only one SA bus 566 is shown in FIG. 5, it should be understood that each zone controller 524, 530-532, 536, and 548-550 can be connected to a different SA bus. Each SA bus can connect a zone controller with various sensors (e.g., temperature sensors, humidity sensors, pressure sensors, light sensors, occupancy sensors, etc.), actuators (e.g., damper actuators, valve actuators, etc.) and/or other types of controllable equipment (e.g., chillers, heaters, fans, pumps, etc.).

[0100] Each zone controller 524, 530-532, 536, and 548-550 can be configured to monitor and control a different building zone. Zone controllers 524, 530-532, 536, and 548-550 can use the inputs and outputs provided via their SA busses to monitor and control various building zones. For example, a zone controller 536 can use a temperature input received from networked sensors 538 via SA bus 566 (e.g., a measured temperature of a building zone) as feedback in a temperature control algorithm. Zone controllers 524, 530-532, 536, and 548-550 can use various types of control algorithms (e.g., state-based algorithms, extremum seeking control (ESC) algorithms, proportional-integral (PI) control algorithms, proportional-integral-derivative (PID) control algorithms, model predictive control (MPC) algorithms, feedback control algorithms, etc.) to control a variable state or condition (e.g., temperature, humidity, airflow, lighting, etc.) in or around building 10.

HVAC System With Automated Device Pairing and Dimensional Mismatch Handling

[0101] Referring now to FIG. 6A, a block diagram of a HVAC system 600 is shown, according to an exemplary embodiment. HVAC system 600 is shown to include a controller 602, an input module 630, an output module 632, several sensors 640, and several actuation devices 650. In brief overview, controller 602 receives sensor readings from sensors 640 via input module 630 and uses the sensor readings to generate actuation signals (e.g., control signals, setpoints, operating commands, etc.) for actuation devices 650. Controller 602 provides the actuation signals to actuation devices 650 via output module 632. Actuation devices 650 operate to affect an environmental condition in a building (e.g., temperature, humidity, airflow, etc.), which can be measured by sensors 640 and provided as a feedback to controller 602.

[0102] Controller 602 can be any type of controller in a HVAC system or BMS. In some embodiments, controller 602 is a zone controller configured to monitor and control a building zone. For example, controller 602 can be a zone temperature controller, a zone humidity controller, a zone lighting controller, a VAV zone controller (e.g., VAV zone controllers 524, 532, 536, 550), a COBP zone controller (e.g., COPB controller 530, 548), or any other type of controller for a building zone. In other embodiments, controller 602 is a system controller or subsystem controller. For example, controller 602 can be a BMS controller (e.g., BMS controller 366), a central plant controller, a subplant controller, a supervisory controller for a HVAC system or any other type of building subsystem (e.g., a controller for any of building subsystems 428). In some embodiments, controller 602 is a field controller or device controller configured to monitor and control the performance of a set of HVAC devices or other building equipment. For example, controller 602 can be an AHU controller (e.g., AHU controller 330), a thermostat controller (e.g., thermostat controller 516), a rooftop unit controller, a chiller controller, a damper controller, or any other type of controller in a HVAC system or BMS.

[0103] Sensors 640 can include any of a variety of physical sensors configured to measure a variable state or condition in a building. For example, sensors 640 are shown to include temperature sensors 641, humidity sensors 642, airflow sensors 643, lighting sensors 644, pressure sensors 645, and voltage sensors 646. Sensors 640 can be distributed throughout a building and configured to measure various environmental conditions at different locations in the building. For example, one of temperature sensors 641 can be located in a first zone of the building and configured to measure the temperature of the first zone, whereas another

of temperature sensors 641 can be located in a second zone of the building and configured to measure the temperature of the second zone. Similarly, sensors 640 can be distributed throughout a HVAC system and configured to measure conditions at different locations in the HVAC system. For example, one of temperature sensors 641 can be a supply air temperature sensor configured to measure the temperature of the airflow provided to a building zone from an AHU, whereas another of temperature sensors 641 can be a return air temperature sensor configured to measure the temperature of the airflow returning from the building zone to the AHU.

[0104] Sensors 640 are shown providing sensor readings to controller 602 via input module 630. The sensor readings can include analog inputs, digital inputs, measurements, data samples, and/or other types of data generated by sensors 640. In some embodiments, sensors 640 provide analog inputs to input module 630 and input module 630 converts the analog inputs to digital data samples. Each data sample can include a data point and associated metadata. The data point can include a measured value attribute indicating the value of the measured variable and a time attribute indicating the time at which the measured value was observed. The metadata can include a unit of measure (e.g., degrees C, degrees F, kPa, volts, Watts, m/s, etc.), a sampling rate, a source description, a location, a purpose, or other attributes describing the associated data point. Controller 602 can receive the sensor readings from input module 630 and store the sensor readings as timeseries data in a timeseries database 616 (described in greater detail below). Controller 602 can use the sensor readings and/or timeseries data to generate appropriate actuation signals for actuation devices 650.

[0105] Actuation devices 650 can include any of a variety of physical devices configured to affect a variable state or condition in a building. For example, actuation devices 650 are shown to include chillers 651, heaters 652, valves 653, air handling units (AHUs) 654, dampers 655, and actuators 656. Although only a few types of actuation devices 650 are shown, it should be understood that actuation devices 650 can include any type of equipment or device configured to affect building conditions. For example, actuation devices 650 can power relays, switches, lights, pumps, fans, cooling towers, or other types of building equipment or central plant equipment. Actuation devices 650 can include some or all of the equipment in building 10, HVAC system 100, waterside system 200, airside system 300, BMS 400, and/or BMS 500, as described with reference to FIGS. 1-5. Actuation devices 650 can operate to affect various building conditions including

temperature, humidity, airflow, lighting, air quality, power consumption, or any other variable state or condition in a building.

[0106] Actuation devices 650 are shown receiving actuation signals from controller 602 via output module 632. In some embodiments, the actuation signals are control signals for actuation devices 650 (e.g., operating setpoints, on/off commands, etc.). For example, the actuation signals can include commands to activate or deactivate individual chillers 651 or heaters 652 and/or commands to operate chillers 651 or heaters 652 at a variable capacity (e.g., operate at 20% capacity, 40% capacity, etc.). The actuation signals can include position setpoints for valves 653, dampers 655, or actuators 656. The position setpoints can include commands to move to a fully closed position, a 50% open position, a fully open position, or any intermediate position.

[0107] In some embodiments, the actuation signals are provided directly to actuation devices 650 from controller 602 and used to adjust a physical operation of actuation devices 650 (e.g., if controller 602 directly controls actuation devices 650). In other embodiments, the actuation signals are provided to an intermediate controller for actuation devices 650. For example, controller 602 can provide a setpoint to a local controller for one or more of actuation devices 650. The local controller can then generate control signals for actuation devices 650 to achieve the setpoint received from controller 602.

[0108] Controller 602 can use the sensor readings from sensors 640 as feedback to determine appropriate actuation signals for actuation devices 650. Controller 602 can be configured to use one or more feedback control algorithms (e.g., state-based algorithms, extremum seeking control (ESC) algorithms, proportional-integral (PI) control algorithms, proportional-integral-derivative (PID) control algorithms, model predictive control (MPC) algorithms, etc.) to control actuation devices 650 based on the sensor readings. For example, if the sensor reading from one of temperature sensors 641 indicates that the temperature of a particular building zone is below a temperature setpoint for the building zone, controller 602 can provide an actuation signal to one of heaters 652, dampers 655, or AHUs 654 to increase the amount of heating provided to the building zone.

[0109] In some embodiments, the feedback control actions performed by controller 602 require knowledge of the relationships between sensors 640 and actuation devices 650. For example, in order to drive the temperature measured by a particular temperature measured toward a setpoint, controller 602 may need to identify which of actuation devices 650 is

configured to affect the measured temperature. In other words, controller 602 may need to identify causal relationships between various sensors 640 and actuation devices 650. If such relationships are not already known, controller 602 can perform an automated device pairing process to establish associations between various sensors 640 and actuation devices 650.

[0110] Some buildings have hundreds of sensors 640 and actuation devices 650. It can be cumbersome, error prone, and time consuming to manually identify the associations between sensors 640 and their corresponding actuation devices 650. Additionally, building evolve over time which can change existing relationships between sensors 640 and actuation devices 650. For example, remodeling a building can break existing relationships between paired sensors 640 and actuation devices 650 or create new relationships that did not previously exist. Advantageously, the automated device pairing process performed by controller 602 can automatically identify causal relationships between various sensors 640 and actuation devices 650 (e.g., heater A affects temperature sensor B, damper C affects flow sensor D, etc.). Once the causal relationships have been identified, controller 602 can store associations between related sensors 640 and actuation devices 650 and use the stored associations to perform control actions.

[0111] Still referring to FIG. 6A, controller 602 is shown to include a communications interface 604 and a processing circuit 606. Communications interface 604 can include wired or wireless communications interfaces (e.g., jacks, antennas, transmitters, receivers, transceivers, wire terminals, etc.) for conducting data communications with external systems or devices (e.g., input module 630, output module 632, sensors 640, actuation devices 650, etc.). Data communications via communications interface 604 can be direct (e.g., local wired or wireless communications) or via a communications network (e.g., a LAN, a WAN, the Internet, a cellular network, etc.). For example, communications interface 604 can include an Ethernet card and port for sending and receiving data via an Ethernet-based communications link or network, a Wi-Fi transceiver for communicating via a wireless communications network, and/or cellular or mobile phone communications transceivers for communicating via a cellular communications network.

[0112] Processing circuit 606 is shown to include a processor 608 and memory 610. Processing circuit 606 can be communicably connected to communications interface 604 such that processing circuit 606 and the various components thereof can send and receive data via communications interface 604. Processor 608 can be implemented as a general

purpose processor, an application specific integrated circuit (ASIC), one or more field programmable gate arrays (FPGAs), a group of processing components, or other suitable electronic processing components.

[0113] Memory 610 (e.g., memory, memory unit, storage device, etc.) can include one or more devices (e.g., RAM, ROM, Flash memory, hard disk storage, etc.) for storing data and/or computer code for completing or facilitating the various processes, layers and modules described in the present application. Memory 610 can be or include volatile memory or non-volatile memory. Memory 610 can include database components, object code components, script components, or any other type of information structure for supporting the various activities and information structures described in the present application. In some embodiments, memory 610 is communicably connected to processor 608 via processing circuit 606 and includes computer code for executing (e.g., by processing circuit 606 and/or processor 608) one or more processes described herein.

[0114] Still referring to FIG. 6A, controller 602 is shown to include virtual sensors 614 and virtual actuation devices 612. Virtual sensors 614 can include logical representations of one or more physical sensors 640. For example, virtual sensors 614 can include data objects (e.g., BACnet objects, JSON objects, etc.), each of which corresponds to a particular physical sensor 640 and functions as a logical representation of the corresponding physical sensor 640 within the memory 610 of controller 602. Virtual sensors 614 can include various attributes which describe the corresponding physical sensors 640 and include the sensor readings from the corresponding physical sensors 640. An example of a virtual sensor 614 shown below:

```
{
        "unique identifier" : "ucis-2327-2127-sded-iesa",
        "signature" : "7b1c018bf975c88fbe9df6292bf370b1",
        "BACnet object" : {
                "object identifier" : "analog input #1101",
                "object name" : "507_SP2.RET_AIR",
                "description" : "return air temp",
                "device type" : "thermistor",
                "object type" : "analog input",
                "units" : "DEG_F",
                "present value" : "68"
                "update interval" : "15 min",
                "status flags" : [
                        "in alarm",
                        "fault",
                        "overridden",
```

```
                "out of service"
            ],
            "event status" : "normal",
            "reliability" : "no fault detected",
            "out of service" : "false",

        }
}
```


[0115] Virtual sensors 614 can include a unique identifier attribute and a signature attribute. In some embodiments, the unique identifier attribute is a text string uniquely identifying a particular virtual sensor 614 (e.g., ucis-2327-2127-sded-iesa). The signature attribute can be generated by controller 602 as a function of the BACnet object attribute values. For example, controller 602 can generate the signature attribute using the following function:

$$Signature = MD5(CONCAT(\text{"analog input \#1101"}, \text{"507_SP2.RET_AIR"}, \text{"return air temp"}, \text{"thermistor"}, \text{"analog input"}, \text{"DEG_F"}, \text{"15 min"}, \text{"68"})$$

where the $CONCAT$ ( ) function is a string concatenation function of the attribute values and the $MD5$ ( ) function is a hashing function producing a hash value (e.g., a 128 bit hash value). When any of the attribute values change, the signature attribute will also change. Accordingly, the signature attribute enables capturing of source data changes, including changes in the present value of the sensor reading.

[0116] Virtual sensors 614 can include a BACnet object attribute with various sub-attributes describing the corresponding physical sensors 640. For example, virtual sensors 614 can include an object identifier attribute which identifies the type of input (e.g., analog input, digital input, enumerated value input, binary input, etc.), an object name attribute which names the corresponding physical sensor (e.g., 507_SP2.RET_AIR), a description attribute which provides a description of the measured value (e.g., return air temperature, supply air temperature, relative humidity, etc.), and a device type attribute which indicates the type of physical sensor 640 represented by the virtual sensor 614 (e.g., thermistor, thermocouple, limit switch, piezoelectric, etc.).

[0117] Virtual sensors 614 can include a present value attribute which indicates the present value of the sensor reading (e.g., 68), a units attribute which indicates the unit of measure of the present value attribute (e.g., degrees F, degrees C, kPa, volts, etc.), and an update interval attribute which indicates how often the present value attribute is updated

(e.g., 15 minutes, 1 minute, 1 second, etc.). In some embodiments, virtual sensors 614 include status attributes (e.g., status flags, event status, reliability, etc.) which indicate the current status of the corresponding physical sensor 640. The attributes of virtual sensors 614 can be updated in real time (e.g., continuously or periodically as defined by the update interval) to reflect the current sensor readings and/or the status of the corresponding physical sensors 640. In some embodiments, virtual sensors 614 include software agents which monitor the sensor readings and other information (e.g., metadata) received from sensors 640 and update the corresponding attribute values accordingly.

[0118] Similarly, virtual actuation devices 612 can include logical representations of one or more physical actuation devices 650. For example, virtual actuation devices 612 can include data objects (e.g., BACnet objects, JSON objects, etc.), each of which corresponds to a particular physical actuation device 650 and functions as a logical representation of the corresponding physical actuation device 650 within the memory 610 of controller 602. Virtual actuation devices 612 can include various attributes which describe the corresponding physical actuation devices 650. In some embodiments, virtual actuation devices 612 include some or all of the same attributes of virtual sensors 614, as previously described. In some embodiments, virtual actuation devices 612 include a present value attribute which includes the most recent value of the actuation signal provided to the virtual actuation device.

[0119] Still referring to FIG. 6A, controller 602 is shown to include a timeseries database 616. Virtual sensors 614 are shown providing sensor readings to timeseries database 616, whereas virtual actuation devices 612 are shown providing actuation signals to timeseries database 616. The sensor readings provided by virtual sensors 614 can include a series of sensor readings collected by each of sensors 640 over time. Similarly, the actuation signals provided by virtual actuation devices 612 can include a series of values of the actuation signals provided to each of actuation devices 650 over time. Timeseries database 616 can store the sensor readings and actuation signals as timeseries data for each of sensors 640 and actuation devices 650. Each timeseries corresponds to one of sensors 640 or actuation devices 650 and includes a series of data values received from the corresponding sensor 640 or provided to the corresponding actuation device 650.

[0120] In some embodiments, each timeseries is a partially ordered tuple of a particular data point and associated metadata, as shown in the following equation:

$$Timeseries = < Data\ Point, Metadata >$$

Each data point may include a series of data values (e.g., sensor reading values or actuation signal values) and corresponding times at which those values were measured or provided. An example of a $Data\ Point$ is shown in the following equation:

$$Data\ Point = < time, value >$$

where each of $time$ and $value$ can include a vector of timeseries values. $Metadata$ can include various attributes of the corresponding data point, as shown in the following equation:

$$Metadata = < unit, sampling\ rate, source\ description, location, purpose, \dots >$$

where each item of $Metadata$ represents one of the attributes (and corresponding attribute values) of the virtual sensor 614 or virtual actuation device 612 from which the timeseries values of the associated data point were received. Timeseries database 616 can store each timeseries for use by other components of controller 602.

[0121] In some embodiments, timeseries database 616 stores each timeseries of actuation signal values as an actuation signal timeseries $t(x)$ as shown in the following equation:

$$t(x) = \{t_1, t_2, t_3, \dots, t_{N-1}, t_N\}$$

where each element t_i of the actuation signal timeseries $t(x)$ is the value of the actuation signal at a particular time (i.e., a sample of the actuation signal) and N is the total number of elements in the actuation signal timeseries $t(x)$. Similarly, timeseries database 616 can store each timeseries of sensor reading values as a sensor response timeseries $r(x)$ as shown in the following equation:

$$r(x) = \{r_1, r_2, r_3, \dots, r_{M-1}, r_M\}$$

where each element r_i of the sensor response timeseries $r(x)$ is the value of the sensor response signal at a particular time (i.e., a sample of the sensor response signal) and M is the total number of elements in the sensor response timeseries $r(x)$.

[0122] Controller 602 can automatically identify causal relationships between various sensors 640 and actuation devices 650 based on the timeseries data associated therewith. For example, for each actuation signal timeseries $t(x)$, controller 602 can identify one or more of the sensor response timeseries $r(x)$ which closely match the actuation signal timeseries. In some embodiments, controller 602 uses a distance function to determine

which of the sensor response timeseries $r(x)$ most closely match the actuation signal timeseries $t(x)$. As described in detail below, the distance function may compare corresponding values t_i and r_i of each timeseries to calculate a similarity metric or similarity score for pairs of actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$.

[0123] In some embodiments, the similarity metric calculation performed by controller 602 requires the actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$ to have the same number of samples (i.e., $N = M$) and/or sampling rate to allow the corresponding values t_i and r_i of each timeseries to be identified and compared. However, some timeseries can have different numbers of samples (i.e., $N \neq M$), which can be collected at different sampling rates. This is referred to as a dimensional mismatch between timeseries. A dimensional mismatch between timeseries can complicate the similarity metric calculation since it can be difficult to determine the corresponding values t_i and r_i of each timeseries. However, controller 602 can automatically identify and compensate for dimensional mismatches between timeseries.

[0124] Still referring to FIG. 6A, controller 602 is shown to include a dimensional mismatch identifier 618. Dimensional mismatch identifier 618 is configured to identify dimensional mismatches between various actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. As described above, a dimensional mismatch may occur when two timeseries have a different number of samples and/or sampling rates. In some embodiments, dimensional mismatch identifier 618 determines the size of each timeseries. For example, dimensional mismatch identifier 618 can determine the number of samples N in the actuation signal timeseries $t(x)$ and the number of samples M in the sensor response timeseries $r(x)$. Dimensional mismatch identifier 618 can detect a dimensional mismatch in response to a determination that the number of samples N in the actuation signal timeseries $t(x)$ is different from the number of samples M in the sensor response timeseries $r(x)$ (i.e., $N \neq M$).

[0125] In some embodiments, dimensional mismatch identifier 618 determines the sampling rate of each timeseries. In some embodiments, the sampling rate of a timeseries may be stored as metadata associated with the timeseries in timeseries database 616. Dimensional mismatch identifier 618 can determine the sampling rate of a timeseries by reading the sampling rate from the metadata in timeseries database 616. In other

embodiments, dimensional mismatch identifier 618 calculates the sampling rate for one or more timeseries based on the size of the timeseries and the range of time spanned by the timeseries.

[0126] Dimensional mismatch identifier 618 can identify a start time and an end time for the timeseries by reading the timestamps associated with the first and last data samples in the timeseries. Dimensional mismatch identifier 618 can then calculate the sampling rate by dividing the size of the timeseries by the difference between the end time and the start time, as shown in the following equation:

$$sampling_rate = \frac{size_of_timeseries}{end_time - start_time}$$

where $size_of_timeseries$ is the number of samples M or N in the timeseries, end_time is the timestamp associated with the last sample in the timeseries, $start_time$ is the timestamp associated with the first sample in the timeseries, and $sampling_rate$ is the sampling rate of the timeseries, expressed as the number of samples per unit time (e.g., 0.8 samples/hour). Dimensional mismatch identifier 618 can detect a dimensional mismatch in response to a determination that two timeseries have different sampling rates.

[0127] Dimensional mismatch identifier 618 can be configured to correct a dimensional mismatch between two timeseries. In some embodiments, dimensional mismatch identifier 618 corrects dimensional mismatch by increasing the number of samples of the timeseries with the fewer number of samples (e.g., by interpolating between samples). In other embodiments, dimensional mismatch identifier 618 corrects dimensional mismatch by reducing the number of samples of the timeseries with the greater number of samples (e.g., by discarding extra samples). In other embodiments, dimensional mismatch identifier 618 merely identifies a dimensional mismatch to other components of controller 602 which are configured to address the dimensional mismatch. For example, dimensional mismatch identifier 618 is shown reporting a dimensional mismatch to discrete cosine transformer 620.

[0128] Still referring to FIG. 6A, controller 602 is shown to include a discrete cosine transformer 620. Discrete cosine transformer 620 can be configured to perform a discrete cosine transform (DCT) for each actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. A DCT expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. In particular, a DCT is a Fourier-

related transform similar to the discrete Fourier transform (DFT), but using only real numbers. There are eight standard DCT variants, commonly referred to as DCT-I, DCT-II, DCT-III, DCT-IV, DCT-V, DCT-VI, DCT-VII, and DCT-VIII. One of these variants (i.e., DCT-II) is discussed in detail below. However, it should be understood that discrete cosine transformer 620 can use any standard or non-standard DCT variant in other embodiments.

[0129] In some embodiments, discrete cosine transformer 620 performs a DCT for each actuation signal timeseries $t(x)$ using the following equation:

$$T(k) = \sum_{i=0}^{N-1} t_i \cos\left[\frac{\pi}{N}\left(i + \frac{1}{2}\right)k\right] \quad k = 0, \dots, N-1$$

where $T(k)$ is the kth coefficient of the DCT of the actuation signal timeseries $t(x)$, t_i is the ith sample of the actuation signal timeseries $t(x)$, and N is the number of samples of the actuation signal timeseries $t(x)$. Discrete cosine transformer 620 can generate an array T of the DCT coefficients (e.g., $T = [T(0), T(1), T(2), \dots, T(N-2), T(N-1)]$) where the length of the array T is the same as the number of samples N of the actuation signal timeseries $t(x)$.

[0130] Similarly, discrete cosine transformer 620 can perform a DCT for each sensor response timeseries $r(x)$ using the following equation:

$$R(k) = \sum_{i=0}^{M-1} r_i \cos\left[\frac{\pi}{M}\left(i + \frac{1}{2}\right)k\right] \quad k = 0, \dots, M-1$$

where $R(k)$ is the kth coefficient of the DCT of the sensor response timeseries $r(x)$, r_i is the ith sample of the sensor response timeseries $r(x)$, and M is the number of samples of the sensor response timeseries $r(x)$. Discrete cosine transformer 620 can generate an array R of the DCT coefficients (e.g., $R = [R(0), R(1), R(2), \dots, R(M-2), R(M-1)]$) where the length of the array R is the same as the number of samples M of the sensor response timeseries $r(x)$.

[0131] The following example illustrates the result of applying DCT to an input timeseries $X(n)$. The input timeseries $X(n)$ can be an actuation signal timeseries $t(x)$ or a sensor response timeseries $r(x)$ as previously described. The samples of the input timeseries $X(n)$ are shown in the following array:

$$X(n) = [1.00, 1.70, 2.00, 2.00, 4.30, 4.50, 3.00, 3.00, 2.30, 2.20, 2.20, 2.30]$$

where the input timeseries $X(n)$ includes twelve timeseries values $X(1), \ldots, X(12)$. Applying DCT to the input timeseries $X(n)$ results in a set of DCT coefficients, shown in the following array:

$$Y(k) = [8.80, -0.57, -2.65, -1.15, 0.81, 0.52, -0.20, -0.93, -0.63, 0.32, 0.50, -0.06]$$

where the array of DCT coefficients $Y(k)$ includes twelve DCT coefficients $Y(1), \ldots, Y(12)$.

[0132] Still referring to FIG. 6A, controller 602 is shown to include a DCT quantizer 622. DCT quantizer 622 can be configured to apply a quantization process to the sets of DCT coefficients generated by discrete cosine transformer 620. As described above, the DCT process performed by discrete cosine transformer 620 converts an input data timeseries $X(n)$ into a sum of cosine functions which oscillate at different frequencies. The cosine function with the lowest frequency is typically first in the summation, followed by cosine functions with successively higher frequencies. Accordingly, the DCT coefficient which occurs first in the array of DCT coefficients $Y(k)$ represents the magnitude of the lowest frequency cosine function. Each of the following DCT coefficients represents the magnitude of a cosine function with a successively higher oscillation frequency.

[0133] DCT quantizer 622 can apply a quantization process to the sets of DCT coefficients by filling some of the higher frequency DCT coefficients with zeros. This has the effect of removing some of the higher frequency components (i.e., cosine functions) from the summation while retaining the lower frequency components. In some embodiments, DCT quantizer 622 performs the quantization process using a predetermined quantization level. The quantization level may define the number of the DCT coefficients which are retained (i.e., not filled with zeros). For example, a quantization level of six may retain the DCT coefficients applied to the six lowest frequency cosine functions (e.g., the first six DCT coefficients in the array) while the remaining DCT coefficients are filled with zeros.

[0134] The following example illustrates the result of a quantization process which can be performed by DCT quantizer 622. DCT quantizer 622 can modify the array of DCT coefficients $Y(k)$ shown above to form the following quantized array $QY(k)$:

$$QY(k) = [8.80, -0.57, -2.65, -1.15, 0.81, 0.52, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]$$

In this example, a quantization level of six is applied, meaning that only the first six DCT coefficients are retained from the original array of DCT coefficients $Y(k)$. The remaining DCT coefficients are filled with zeros. True compression can be achieved by not storing the zeros. For example, DCT quantizer 622 can store the following compressed array $C(k)$:

$$C(k) = [8.80, -0.57, -2.65, -1.15, 0.81, 0.52]$$

in which the coefficients filled with zeros are discarded to produce a compressed array with a length equal to the quantization level applied. In this example, the compressed array $C(k)$ has a length of six resulting from the use of a quantization level of six. In various embodiments, DCT quantizer 622 can use a quantization level of six or any other quantization level to produce compressed arrays of various lengths.

[0135] In some embodiments, DCT quantizer 622 automatically determines the quantization level to apply based on the number of samples in each of the original actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. As described above, the number of DCT coefficients produced by discrete cosine transformer 620 for a given input timeseries $X(n)$ may be equal to the number of samples in the timeseries $X(n)$ prior to performing DCT. For example, an input timeseries $X_1(n)$ with twelve samples may result in twelve DCT coefficients in the resultant DCT coefficient array $Y_1(k)$, whereas an input timeseries $X_2(n)$ with ten samples may result in ten DCT coefficients in the resultant DCT coefficient array $Y_2(k)$. In some embodiments, DCT quantizer 622 identifies the actuation signal timeseries $t(x)$ or sensor response timeseries $r(x)$ with the fewest samples and applies a quantization level equal to the number of samples in the identified timeseries.

[0136] In some embodiments, DCT quantizer 622 applies the same quantization level to the sets of DCT coefficients corresponding to each of the original actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. Using the same quantization level for each of the original timeseries may result in the same number of compressed DCT coefficients being stored for each of the original actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. In some embodiments, the number of stored DCT coefficients is equal to the number of samples in the original timeseries with the fewest samples. Advantageously, this allows for direct comparison of the DCT coefficients in the compressed arrays $C(k)$ generated for each of the original timeseries without requiring decompression, interpolation, synchronization, or other processing steps after the compressed arrays $C(k)$ are generated.

[0137] In some embodiments, DCT quantizer 622 generates a compressed timeseries T_α based on each compressed array of DCT coefficients. DCT quantizer 622 can store the compressed timeseries T_α using the following data structure:

$$T_\alpha = \langle \alpha, \delta, \rho, \kappa, \langle \psi \rangle, \langle v_1, v_2, \ldots v_\rho \rangle \rangle$$

where α is the timeseries ID of the source timeseries (e.g., the actuation signal timeseries $t(x)$ or sensor response timeseries $r(x)$), δ is the dimension of the source timeseries (e.g., the number of samples in the source timeseries), ρ is the quantization level applied by DCT quantizer 622, κ is a pointer for metadata, $\langle \psi \rangle$ indicates the start time and end time of samples in the source timeseries, and $\langle v_1, v_2, \ldots v_\rho \rangle$ is the array of compressed DCT coefficients. An example of a compressed timeseries stored using this data structure is as follows:

$$T_{203} = \langle 203,12,6, \langle 2016{:}10{:}05{:}12{:}00{:}00, 2016{:}10{:}05{:}13{:}00{:}00 \rangle, \langle 8.80, -0.57, -2.65, -1.15, 0.81, 0.52 \rangle \rangle$$

where 203 is the timeseries ID of the source timeseries, 12 is size of the source timeseries (e.g., 12 samples in the source timeseries), 6 is the quantization level applied by DCT quantizer 622, 2016:10:05:12:00:00 is the start time of the source timeseries (e.g., the timestamp of the earliest sample in the source timeseries), 2016:10:05:13:00:00 is the end time of the source timeseries (e.g., the timestamp of the latest sample in the source timeseries), and the array ⟨8.80, -0.57, -2.65, -1.15, 0.81, 0.52⟩ includes the compressed DCT coefficients generated by DCT quantizer 622.

[0138] Still referring to FIG. 6A, controller 602 is shown to include a similarity calculator 624. Similarity calculator 624 can be configured to determine whether two timeseries are similar to each other based on the compressed DCT coefficients and/or compressed timeseries generated by DCT quantizer 622. In some embodiments, similarity calculator 624 determines whether any of the sensor response timeseries $r(x)$ are similar to a given actuation signal timeseries $t(x)$. Similarity calculator 624 can repeat this process for each actuation signal timeseries $t(x)$ to determine whether any of the sensor response timeseries $r(x)$ are similar to each actuation signal timeseries $t(x)$.

[0139] In some embodiments, similarity calculator 624 determines whether two timeseries are similar to each other by calculating a similarity metric for the two timeseries. The similarity metric can be based on the compressed DCT coefficients generated by DCT

quantizer 622 for the two timeseries. For example, the compressed DCT coefficients generated for a given actuation signal timeseries $t(x)$ can be represented by an array T, whereas the compressed DCT coefficients generated for a given sensor response timeseries $r(x)$ can be represented by an array R. The arrays T and R can be particular instances of the compressed array $C(k)$ generated by DCT quantizer 622 for the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$, respectively. Each array T and R can include a predetermined number N of DCT coefficients, defined by the quantization level applied by DCT quantizer 622. Examples of arrays T and R are as follows:

$$T = \langle t_1, t_2, \ldots, t_N \rangle$$
$$R = \langle r_1, r_2, \ldots, r_N \rangle$$

[0140] Similarity calculator 624 can calculate a similarity metric for the source timeseries $t(x)$ and $r(x)$ based on the corresponding arrays T and R of compressed DCT coefficients. In some embodiments, similarity calculator 624 calculates the similarity metric using the following equation:

$$d(T,R) = \sum_{i=1}^{i=N} \frac{\sqrt{(t_i - r_i)^2}}{\delta_i}$$

where t_i is the ith DCT coefficient in the array T based on the actuation signal timeseries $t(x)$, r_i is the ith DCT coefficient in the array R based on the sensor response timeseries $r(x)$, δ_i is the standard deviation of the ith DCT coefficients, and N is the number of DCT coefficients in each array T and R. Low values of the similarity metric $d(T,R)$ indicate a greater similarity, whereas high values of the similarity metric $d(T,R)$ indicate a lesser similarity. Similarity calculator 624 can calculate a similarity metric for each pairing of an actuation signal timeseries $t(x)$ and a sensor response timeseries $r(x)$.

[0141] Still referring to FIG. 6A, controller 602 is shown to include a device pairing generator 626. Device pairing generator 626 is shown receiving the similarity metrics from similarity calculator 624. Device pairing generator 626 can be configured to generate device pairings based on the similarity metrics. Each device pairing may include one of sensors 640 and one of actuation devices 650. A device pairing may indicate that the actuation device 650 in the device pairing is configured to affect the variable measured by the sensor 640 in the device pairing. For each potential device pairing (i.e., for each combination of a sensor 640 and an activation device 650), device pairing generator 626 can

identify the corresponding arrays T and R of compressed DCT coefficients and the calculated similarity metric $d(T, R)$ based on the arrays T and R. Device pairing generator 626 can use the similarity metric $d(T, R)$ to determine whether to generate a device pairing between the given sensor 640 and the given actuation device 650.

[0142] In some embodiments, device pairing generator 626 generates device pairings by comparing the similarity metric $d(T, R)$ to a threshold value. Device pairing generator 626 can be configured to generate a device pairing between a sensor 640 and an actuation device 650 if the similarity metric $d(T, R)$ is less than the threshold value (e.g., $d(T, R) < threshold$). The threshold value can be a predefined value or a calculated value (e.g., a standard deviation of the DCT coefficients).

[0143] In some embodiments, the threshold value is a similarity metric between the actuation signal timeseries $t(x)$ and a baseline (e.g., average) sensor signal timeseries $a(x)$ over a predetermined time period. The baseline sensor signal timeseries $a(x)$ can indicate the average sensor response from a particular sensor 640 before the actuation signal is applied to the actuation device 650 (e.g., baseline sensor readings), whereas the sensor response timeseries $r(x)$ can indicate the sensor response from the same sensor 640 after the actuation signal is applied to the actuation device 650. If the actuation device 650 affects the sensor 640, the actuation signal timeseries $t(x)$ is expected to be more similar to the sensor response timeseries $r(x)$ than the baseline sensor signal timeseries $a(x)$. Accordingly, the similarity metric $d(T, R)$ between the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ is expected to be lower (i.e., more similar) than the similarity metric $d(T, A)$ between the actuation signal timeseries $t(x)$ and the baseline sensor signal timeseries $a(x)$.

[0144] In some embodiments, timeseries database 616 stores a baseline sensor signal timeseries $a(x)$ for each of sensors 640 based on sensor readings from sensors 640 before the actuation signal is applied to actuation devices 650. Timeseries database 616 can also store a sensor response timeseries $r(x)$ for each of sensors 640 based on sensor readings from sensors 640 while the actuation signal is applied to actuation devices 650 or after the actuation signal is applied to actuation devices 650. Discrete cosine transformer 620 and DCT quantizer 622 can generate DCT coefficients and compressed DCT coefficients for each baseline sensor signal timeseries $a(x)$, sensor response timeseries $r(x)$, and actuation signal timeseries $t(x)$. Similarity calculator 624 can then calculate a similarity metric

$d(T, R)$ between each actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$ and a similarity metric $d(T, A)$ between each actuation signal timeseries $t(x)$ and baseline sensor signal timeseries $a(x)$. Device pairing generator 626 can generate a device pairing between a sensor 640 and an actuation device 650 if the similarity metric $d(T, R)$ for a given combination of a sensor 640 and an actuation device 650 is less than the similarity metric $d(T, A)$ for the sensor 640 and the actuation device 650.

[0145] In some embodiments, device pairing generator 626 generates device pairings by comparing the similarity metrics $d(T, R)$ for various combinations of sensors 640 and actuation devices 650. For each actuation device 650, device pairing generator 626 can identify the similarity metrics $d(T, R)$ calculated for the actuation device 650 in combination with each of sensors 640. Each similarity metric $d(T, R)$ indicates the similarity (i.e., the closeness) between the actuation signal timeseries $t(x)$ associated with the actuation device 650 and the sensor response timeseries $r(x)$ associated with one of sensors 640. For example, the similarity metric $d(T_1, R_1)$ may indicate the similarity between a first actuation device 650 (corresponding to array T_1) and a first sensor 640 (corresponding to array R_1), whereas the similarity metric $d(T_1, R_2)$ may indicate the similarity between the first actuation device 650 and a second sensor 640 (corresponding to array R_2). Device paring generator 626 can identify all of the similarity metrics associated with a given actuation device 650 (e.g., $d(T_1, R_1), \ldots, d(T_1, R_P)$, where P is the total number of sensors 640 and/or sensor response timeseries $r(x)$).

[0146] Device pairing generator 626 can determine which of the identified similarity metrics is the lowest for a given actuation device 650. The lowest similarity metric indicates the closest match between the actuation signal timeseries $t(x)$ associated with the actuation device 650 and the sensor response timeseries $r(x)$ associated with one of sensors 640. Device pairing generator 626 can generate a device pairing between the actuation device 650 and the sensor 640 having the lowest similarity metric $d(T, R)$ with the actuation device 650. If the actuation device 650 has the same similarity metric with multiple sensors 640 (e.g., $d(T_1, R_1) = d(T_1, R_2)$), device pairing generator 626 can examine the time delay Δw between the actuation signal timeseries $t(x)$ associated with the actuation device 650 and sensor response timeseries $r(x)$ associated with each of sensors 640. The time delay Δw may indicate the delay between the time w_1 at which the actuation signal is applied to the actuation device 650 and the time w_2 at which the effects of the

actuation signal are evident in the sensor response (e.g., $\Delta w = w_2 - w_1$). Device pairing generator 626 can determine which of the sensors 640 has the lowest time delay Δw and can generate a device pairing between the actuation device 650 and the sensor 640 with the lowest time delay Δw.

[0147] Device pairing generator 626 can generate one or more device pairings for each of actuation devices 650. Each device pairing can identify one of actuation devices 650 and one of sensors 640. A device pairing between an actuation device 650 and a sensor 640 indicates that the actuation device 650 is capable of affecting the value measured by the sensor. Device pairing generator 626 can provide the device pairings to device controller 628. Device controller 628 can use the device pairings to generate the actuation signals for actuation devices 650. In some embodiments, device controller 628 uses the device pairings to automatically generate and store causal relationships between various sensors 640 and actuation devices 650.

[0148] In some embodiments, device controller 628 uses the device pairings to create kits of causally related devices. Each kit may be a logical grouping or set of devices in HVAC system 600 which includes one or more of sensors 640 and one or more of actuation devices 650. In some embodiments, each kit includes all of the sensors 640 and actuation devices 650 that are linked to each other by the device pairings generated by device pairing generator 626. Each device in a given kit may have a device pairing with at least one other device in the kit. For example, a temperature sensor for a building zone may have a device pairing with a chiller which is operable to affect the temperature of the building zone. The temperature sensor may also have device pairings with an air handling unit and an airflow damper which operate to provide airflow to the building zone. The kit generated by device controller 628 may include the temperature sensor and all of the actuation devices which operate to affect the temperature of the building zone (e.g., the chiller, the air handling unit, the airflow damper, etc.).

[0149] In some embodiments, device controller 628 uses the kits to detect and diagnose faults or performance issues in the building. For example, a temperature fault for a building zone (e.g., temperature out of range) can be detected by a temperature sensor located in the building zone. However, the temperature fault may originate from a fault in one or more of the actuation devices 650 which affect the temperature measured by the temperature sensor. Device controller 628 can use the kits of causally related devices to identify one or more actuation devices 650 (e.g., a chiller, a heater, an air handling unit, a damper, etc.) which

operate to affect the variable in fault (e.g., the temperature measured by the temperature sensor). Device controller 628 can then test the actuation devices 650 in the kit to determine whether any of the actuation devices 650 are operating abnormally and diagnose the cause of the fault.

[0150] In some embodiments, device controller 628 uses the kits to generate recommendations for resolving the fault. For example, device controller 628 may recommend that all of the actuation devices 650 in a kit be tested or investigated in order to determine which of the actuation devices 650 is contributing to a detected fault associated with a sensor 640 in the kit. In some embodiments, the recommendation is based on a duration of the fault. For example, if the fault has been occurring for an amount of time which is less than a duration threshold, device controller 628 may recommend that one or more of the devices in the kit be investigated or tested. However, if the fault has been occurring for an amount of time which exceeds the duration threshold, device controller 628 may recommend that one or more of the devices in the kit be replaced, repaired, or changed in order to resolve the fault.

[0151] In some embodiments, device controller 628 provides actuation devices 650 with test signals as part of the device pairing process. The test signals may be predetermined signals or sequences of control operations which differ from the control signals provided to actuation devices 650 during normal operation. In some embodiments, the test signals are the actuation signals $t(x)$ used by other components of controller 602 to generate the device pairings. Device controller 628 can provide the test signals to actuation devices 650 via communications interface 604 and to virtual actuation devices 612. Virtual actuation devices 612 can update their status in real time based on the test signals.

[0152] In some embodiments, device controller 628 uses the device pairings to create feedback control loops for HVAC system 600. Each feedback control loop can receive a feedback signal from one or more of sensors 640 and can provide a control signal to one or more of actuation devices 650. Device controller 628 can use the device pairings to define the sensors 640 and actuation devices 650 in each control loop. For example, device controller 628 can create a control loop which receives a feedback signal from the sensor 640 in a device pairing and provides a control signal to the actuation device 650 in the device pairing. Device controller 628 can map the sensor readings from the sensor 640 in the device pairing to the feedback signal in the control loop. Similarly, device controller

628 can map the actuation signals provided to the actuation device 650 in the device pairing to the control signal in the control loop.

[0153] Device controller 628 can use the feedback control loops to generate the actuation signals for actuation devices 650. Device controller 628 can use state-based algorithms, extremum seeking control (ESC) algorithms, proportional-integral (PI) control algorithms, proportional-integral-derivative (PID) control algorithms, model predictive control (MPC) algorithms, or any other type of control methodology to generate the actuation signals for actuation devices 650 based on the sensor readings. For example, if the sensor reading from one of temperature sensors 641 indicates that the temperature of a particular building zone is below a temperature setpoint for the building zone, device controller 628 can provide an actuation signal to one of heaters 652, dampers 655, or AHUs 654 to increase the amount of heating provided to the building zone. Advantageously, the relationships between actuation devices 650 and sensors 640 can be identified automatically based on the device pairings to allow device controller 628 to determine which of actuation devices 650 can be operated to affect a given sensor reading.

[0154] Referring now to FIG. 6B, a block diagram illustrating a portion of HVAC system 600 in greater detail is shown, according to an exemplary embodiment. HVAC system 600 is shown to include a smart actuator 658, sensors 640, and a valve/damper 672. Smart actuator 658 may be one of actuation devices 650 or a separate actuator in HVAC system 600. Valve/damper 672 may be an airflow damper, a fluid control valve, an expansion valve, or any other type of flow control device in HVAC system 600. Smart actuator 658 can be configured to operate valve/damper 672 (e.g., by opening and closing valve/damper 672) based on sensor readings received from sensors 640. Advantageously, smart actuator 658 can automatically determine which of sensors 640 is affected by valve/damper 672 and can operate valve/damper 672 based on the sensor readings from the affected sensor or sensors 640.

[0155] Smart actuator 658 is shown to include an actuation device 674 having a motor 676 and a drive device 678. Drive device 678 may be mechanically coupled to valve/damper 672 and configured to open and close valve/damper 672 when operated by motor 676. Motor 676 may be mechanically coupled to drive device 678 and configured to operate drive device 678 based on actuation signals received from processing circuit 606. Unlike conventional actuators, smart actuator 658 can independently and automatically

determine appropriate actuation signals for actuation device 674 without requiring input from an external controller.

[0156] Smart actuator 658 is shown to include a communications interface 680 and a processing circuit 606. Communications interface 680 may be the same or similar to communications interface 604, as described with reference to FIG. 6A. Communications interface 680 can include wired or wireless communications interfaces (e.g., jacks, antennas, transmitters, receivers, transceivers, wire terminals, etc.) for conducting data communications with external systems or devices (e.g., sensors 640, user devices, supervisory controllers, etc.). Data communications via communications interface 680 can be direct (e.g., local wired or wireless communications) or via a communications network (e.g., a LAN, a WAN, the Internet, a cellular network, etc.). For example, communications interface 680 can include an Ethernet card and port for sending and receiving data via an Ethernet-based communications link or network, a Wi-Fi transceiver for communicating via a wireless communications network, and/or cellular or mobile phone communications transceivers for communicating via a cellular communications network.

[0157] Processing circuit 606 may include some or all of the components of processing circuit 606 shown in FIG. 6A. For example, processing circuit 606 is shown to include a processor 608 and memory 608. Processing circuit 606 can be communicably connected to communications interface 680 such that processing circuit 606 and the various components thereof can send and receive data via communications interface 680. Processor 608 can be implemented as a general purpose processor, an application specific integrated circuit (ASIC), one or more field programmable gate arrays (FPGAs), a group of processing components, or other suitable electronic processing components.

[0158] Memory 610 (e.g., memory, memory unit, storage device, etc.) can include one or more devices (e.g., RAM, ROM, Flash memory, hard disk storage, etc.) for storing data and/or computer code for completing or facilitating the various processes, layers and modules described in the present application. Memory 610 can be or include volatile memory or non-volatile memory. Memory 610 can include database components, object code components, script components, or any other type of information structure for supporting the various activities and information structures described in the present application. In some embodiments, memory 610 is communicably connected to processor 608 via processing circuit 606 and includes computer code for executing (e.g., by processing circuit 606 and/or processor 608) one or more processes described herein.

[0159] Memory 610 may include some or all of the components of memory 610 shown in FIG. 6A. For example, memory 610 may include virtual actuation devices 612, virtual sensors 614, timeseries database 616, dimensional mismatch identifier 618, discrete cosine transformer 620, DCT quantizer 622, similarity calculator 624, device pairing generator 626, and device controller 628. When implemented in smart actuator 658, device pairing generator 626 can generate pairings between actuation device 674 and one or more of sensors 640 using the techniques described with reference to FIG. 6A. In other words, device pairing generator 626 can determine which of sensors 640 is/are affected by actuation device 674. Device controller 628 can use the device pairings and the sensor readings from the affected sensors 640 to generate actuation signals for actuation device 674.

[0160] Referring now to FIG. 6C, a block diagram illustrating another portion of HVAC system 600 in greater detail is shown, according to an exemplary embodiment. HVAC system 600 is shown to include a smart chiller 659, sensors 640, and a building zone 670. Smart chiller 659 may be one of actuation devices 650 or a separate chiller in HVAC system 600. Smart chiller 659 can be configured to provide cooling for building zone 670 based on sensor readings received from sensors 640. For example, smart chiller 659 is shown to include a refrigeration circuit 660 having a compressor 662, a condenser 664, an expansion device 666, and an evaporator 668. Compressor 662 can be configured to circulate a refrigerant between condenser 664 and evaporator 668 based on actuation signals received from processing circuit 606. Evaporator 668 can provide cooling for an airflow provided to building zone 670 either directly (e.g., by directly chilling the airflow) or via an intermediate coolant (e.g., by chilling a coolant which is used to chill the airflow). Unlike conventional chillers, smart chiller 659 can independently and automatically determine appropriate actuation signals for refrigeration circuit 660 without requiring input from an external controller.

[0161] Smart chiller 659 can automatically determine which of sensors 640 is/are affected by refrigeration circuit 660 and can operate refrigeration circuit 660 based on the sensor readings from the affected sensor(s) 640. For example, sensors 640 are shown to include a zone A temperature sensor 641a, a zone B temperature sensor 641b, and a zone C temperature sensor 641C. One or more of temperature sensors 641a-641c may be located in building zone 670 and configured to measure the temperature of building zone 670. However, the locations of temperature sensors 641a-641c may be unknown to smart chiller

659 when smart chiller 659 is first installed. Smart chiller 659 can use the device pairing techniques described with reference to FIG. 6A to determine which of temperature sensors 641a-641c is affected by refrigeration circuit 660.

[0162] Smart chiller 659 is shown to include a communications interface 680 and a processing circuit 606. Communications interface 680 may be the same or similar to communications interface 604, as described with reference to FIG. 6A. Communications interface 680 can include wired or wireless communications interfaces (e.g., jacks, antennas, transmitters, receivers, transceivers, wire terminals, etc.) for conducting data communications with external systems or devices (e.g., sensors 640, user devices, supervisory controllers, etc.). Data communications via communications interface 680 can be direct (e.g., local wired or wireless communications) or via a communications network (e.g., a LAN, a WAN, the Internet, a cellular network, etc.). For example, communications interface 680 can include an Ethernet card and port for sending and receiving data via an Ethernet-based communications link or network, a Wi-Fi transceiver for communicating via a wireless communications network, and/or cellular or mobile phone communications transceivers for communicating via a cellular communications network.

[0163] Processing circuit 606 may include some or all of the components of processing circuit 606 shown in FIG. 6A. For example, processing circuit 606 is shown to include a processor 608 and memory 608. Processing circuit 606 can be communicably connected to communications interface 680 such that processing circuit 606 and the various components thereof can send and receive data via communications interface 680. Processor 608 can be implemented as a general purpose processor, an application specific integrated circuit (ASIC), one or more field programmable gate arrays (FPGAs), a group of processing components, or other suitable electronic processing components.

[0164] Memory 610 (e.g., memory, memory unit, storage device, etc.) can include one or more devices (e.g., RAM, ROM, Flash memory, hard disk storage, etc.) for storing data and/or computer code for completing or facilitating the various processes, layers and modules described in the present application. Memory 610 can be or include volatile memory or non-volatile memory. Memory 610 can include database components, object code components, script components, or any other type of information structure for supporting the various activities and information structures described in the present application. In some embodiments, memory 610 is communicably connected to processor

608 via processing circuit 606 and includes computer code for executing (e.g., by processing circuit 606 and/or processor 608) one or more processes described herein.

[0165] Memory 610 may include some or all of the components of memory 610 shown in FIG. 6A. For example, memory 610 may include virtual actuation devices 612, virtual sensors 614, timeseries database 616, dimensional mismatch identifier 618, discrete cosine transformer 620, DCT quantizer 622, similarity calculator 624, device pairing generator 626, and device controller 628. When implemented in smart chiller 659, device pairing generator 626 can generate pairings between smart chiller 659 and one or more of sensors 640. In other words, device pairing generator 626 can determine which of sensors 640 is/are affected by smart chiller 659. Device controller 628 can use the device pairings and the sensor readings from the affected sensors 640 to generate actuation signals for compressor 662 and/or other components of refrigeration circuit 660.

[0166] Referring now to FIG. 6D, a block diagram illustrating another portion of HVAC system 600 in greater detail is shown, according to an exemplary embodiment. HVAC system 600 is shown to include a smart thermostat 648, actuation devices 650, and a building zone 670. Smart thermostat 648 is shown to include sensors 640. Sensors 640 can include a temperature sensor 641, humidity sensor 642, or any other type of sensor 640, as described with reference to FIG. 6A. Sensors 640 can be configured to measure various environmental conditions or variables within building zone 670. For example, temperature sensor 641 can measure the temperature of building zone 670, whereas humidity sensor 642 can measure the humidity of building zone 670. Smart thermostat 648 can independently and automatically determine appropriate actuation signals for actuation devices 650 without requiring input from an external controller.

[0167] Smart thermostat 648 can automatically determine which of actuation devices 650 affect the environmental conditions of building zone 670 and can operate actuation devices 650 to control the environmental conditions measured by sensors 640. For example, actuation devices 650 are shown to include several chillers 651, several heaters 652, and several air handling units 654. One or more of actuation devices 650 may operate to affect conditions within building zone 670. However, smart thermostat 648 may be unaware of such causal relationships when smart thermostat 648 is initially installed. Smart thermostat 648 can use the device pairing techniques described with reference to FIG. 6A to determine which of actuation devices 650 can be operated to control conditions within building zone 670.

[0168] Smart thermostat 648 is shown to include a communications interface 680 and a processing circuit 606. Communications interface 680 may be the same or similar to communications interface 604, as described with reference to FIG. 6A. Communications interface 680 can include wired or wireless communications interfaces (e.g., jacks, antennas, transmitters, receivers, transceivers, wire terminals, etc.) for conducting data communications with external systems or devices (e.g., actuation devices 650, user devices, supervisory controllers, etc.). Data communications via communications interface 680 can be direct (e.g., local wired or wireless communications) or via a communications network (e.g., a LAN, a WAN, the Internet, a cellular network, etc.). For example, communications interface 680 can include an Ethernet card and port for sending and receiving data via an Ethernet-based communications link or network, a Wi-Fi transceiver for communicating via a wireless communications network, and/or cellular or mobile phone communications transceivers for communicating via a cellular communications network.

[0169] Processing circuit 606 may include some or all of the components of processing circuit 606 shown in FIG. 6A. For example, processing circuit 606 is shown to include a processor 608 and memory 608. Processing circuit 606 can be communicably connected to communications interface 680 such that processing circuit 606 and the various components thereof can send and receive data via communications interface 680. Processor 608 can be implemented as a general purpose processor, an application specific integrated circuit (ASIC), one or more field programmable gate arrays (FPGAs), a group of processing components, or other suitable electronic processing components.

[0170] Memory 610 (e.g., memory, memory unit, storage device, etc.) can include one or more devices (e.g., RAM, ROM, Flash memory, hard disk storage, etc.) for storing data and/or computer code for completing or facilitating the various processes, layers and modules described in the present application. Memory 610 can be or include volatile memory or non-volatile memory. Memory 610 can include database components, object code components, script components, or any other type of information structure for supporting the various activities and information structures described in the present application. In some embodiments, memory 610 is communicably connected to processor 608 via processing circuit 606 and includes computer code for executing (e.g., by processing circuit 606 and/or processor 608) one or more processes described herein.

[0171] Memory 610 may include some or all of the components of memory 610 shown in FIG. 6A. For example, memory 610 may include virtual actuation devices 612, virtual

sensors 614, timeseries database 616, dimensional mismatch identifier 618, discrete cosine transformer 620, DCT quantizer 622, similarity calculator 624, device pairing generator 626, and device controller 628. When implemented in smart thermostat 648, device pairing generator 626 can generate pairings between sensors 640 and one or more of actuation devices 650. In other words, device pairing generator 626 can determine which of actuation devices 650 affect the variables or conditions measured by sensors 640. Device controller 628 can use the device pairings and the sensor readings from sensors 640 to generate actuation signals for actuation devices 650.

Example Graphs

[0172] Referring now to FIG. 7 a graph 700 illustrating the different types of signals evaluated by controller 602 is shown, according to an exemplary embodiment. In graph 700, line 702 represents an actuation signal $t(x)$, line 706 represents a sensor response signal $r(x)$, and line 710 represents a baseline sensor signal $a(x)$. The actuation signal $t(x)$ can be provided to one or more of actuation devices 650, as previously described. In the example shown in FIG. 7, the actuation signal $t(x)$ is a setpoint signal (e.g., a temperature setpoint) which can be provided as an input to actuation devices 650 (e.g., controllable HVAC equipment) which operate to affect the temperature of a building zone. In some embodiments, the actuation signal $t(x)$ is a predetermined test signal which differs from the normal actuation signals provided to actuation devices 650. For example, the actuation signal $t(x)$ is shown as a step increase from 68 °F to 72 °F, which is held for a predetermined time period, followed by a step decrease from 72 °F back to 68 °F. The actuation signal $t(x)$ causes actuation devices 650 to increase an amount of heating provided to the building zone, which causes the temperature of the building zone to increase.

[0173] The sensor response signal $r(x)$ can be received from one or more of sensors 640, as previously described. In the example shown in FIG. 7, the sensor response signal $r(x)$ is a measured temperature signal received from a temperature sensor located in the building zone controlled by actuation devices 650. The temperature of the building zone begins to increase shortly after the actuation signal $t(x)$ is increased, which results in an increase in the sensor response signal $r(x)$. Similarly, the temperature of the building zone begins to decrease shortly after the actuation signal $t(x)$ is decreased, which results in a decrease in

the sensor response signal $r(x)$. The time delay between the actuation signal $t(x)$ and the sensor response signal $r(x)$ is shown as delay time Δw.

[0174] The baseline sensor signal $a(x)$ can be received from the same sensor 640 which provides the sensor response signal $r(x)$. However, the baseline sensor signal $a(x)$ indicates the sensor readings from the sensor 640 from a time period before the actuation signal $t(x)$ is provided to actuation devices 650. The signals shown in graph 700 are characteristic of a sensor 640 and an actuation device 650 which have a causal relationship with each other. For example, the sensor response signal $r(x)$ closely matches the actuation signal $t(x)$, whereas the baseline sensor signal $a(x)$ is significantly different from both the actuation signal $t(x)$ and the sensor response signal $r(x)$. In other words, the sensor response signal $r(x)$ is correlated with the actuation signal $t(x)$, which indicates a causal relationship between the corresponding actuation device 650 and sensor 640.

[0175] Still referring to FIG. 7, graph 700 is shown to include a line 704 representing the discrete cosine transform (DCT) of the actuation signal $t(x)$, a line 708 representing the DCT of the sensor response signal $r(x)$, and a line 712 representing the DCT of the baseline sensor signal $a(x)$. Discrete cosine transformer 620 can be configured to generate DCTs of the various timeseries signals received or generated by controller 602, as previously described. In graph 700, the DCTs represented by lines 704, 708, and 712 are continuous functions (i.e., summations of cosine functions) which results in a smooth curves for each of lines 704, 708, and 712. Advantageously, the DCT functions reduce any noise present in the signals $t(x)$, $r(x)$, and $a(x)$, and can be evaluated at any point. In some embodiments, controller 602 compares the DCTs of the actuation signal $t(x)$, sensor response signal $r(x)$, and baseline sensor signal $a(x)$ rather than the original signals to determine whether a causal relationship exists between a particular sensor 640 and a particular actuation device 650.

[0176] Referring now to FIG. 8 a graph 800 illustrating the dimensional mismatch handling performed by controller 602 is shown, according to an exemplary embodiment. In graph 800, line 802 represents samples of actuation signal timeseries $t(x)$ provided to one or more of actuation devices 650, whereas line 804 represents samples of a sensor response timeseries $r(x)$ received from one or more of sensors 640. The actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ cover the same time period. However, sampling rate used to generate the samples of the sensor response timeseries $r(x)$ is twice

the sampling rate used to generate the samples of the actuation timeseries $t(x)$, which results in twice as many samples of the sensor response timeseries $r(x)$ being collected during the same time period. In other words, the sensor response timeseries $r(x)$ includes twice as many samples (e.g., 24 samples) as the actuation signal timeseries $t(x)$ (e.g., 12 samples).

[0177] In order to compare the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$, controller 602 can generate DCTs of each timeseries. As described with reference to FIG. 6A, discrete cosine transformer 620 can generate DCT coefficients for the actuation signal timeseries $t(x)$ using the following equation:

$$T(k) = \sum_{i=0}^{N-1} t_i \cos\left[\frac{\pi}{N}\left(i + \frac{1}{2}\right)k\right] \quad k = 0, \dots, N-1$$

where $T(k)$ is the kth coefficient of the DCT of the actuation signal timeseries $t(x)$, t_i is the ith sample of the actuation signal timeseries $t(x)$, and N is the number of samples of the actuation signal timeseries $t(x)$. Discrete cosine transformer 620 can generate an array T of the DCT coefficients (e.g., $T = [T(0), T(1), T(2), \dots, T(N-2), T(N-1)]$) where the length of the array T is the same as the number of samples N of the actuation signal timeseries $t(x)$ (e.g., 12 samples).

[0178] Similarly, discrete cosine transformer 620 can generate DCT coefficients for the sensor response timeseries $r(x)$ using the following equation:

$$R(k) = \sum_{i=0}^{M-1} r_i \cos\left[\frac{\pi}{M}\left(i + \frac{1}{2}\right)k\right] \quad k = 0, \dots, M-1$$

where $R(k)$ is the kth coefficient of the DCT of the sensor response timeseries $r(x)$, r_i is the ith sample of the sensor response timeseries $r(x)$, and M is the number of samples of the sensor response timeseries $r(x)$. Discrete cosine transformer 620 can generate an array R of the DCT coefficients (e.g., $R = [R(0), R(1), R(2), \dots, R(M-2), R(M-1)]$) where the length of the array R is the same as the number of samples M of the sensor response timeseries $r(x)$ (e.g., 24 samples).

[0179] To make the arrays T and R the same dimension, DCT quantizer 622 can perform a quantization process. In some embodiments, DCT quantizer 622 performs the quantization process using a predetermined quantization level. The quantization level may define the number of the DCT coefficients in each array T and R which are retained (i.e.,

not filled with zeros). For example, a quantization level of twelve may retain the first twelve DCT coefficients in each array T and R while the remaining DCT coefficients are filled with zeros. In the example shown in FIG. 8, DCT quantizer 622 applies a quantization level of twelve to each array T and R of DCT coefficients. This quantization level has no effect on the array T since the array T only contains 12 DCT coefficients. However, applying a quantization level of twelve to the array R has the effect of filling the final twelve DCT coefficients with zeros such that only the first twelve DCT coefficients in array R are retained.

[0180] Still referring to FIG. 8, graph 800 is shown to include a line 806 representing the inverse DCT of the actuation signal timeseries $t(x)$ and a line 808 representing the inverse DCT of the sensor response timeseries $r(x)$. Lines 806-808 illustrate the similarity between the quantized DCT functions based on the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$. As discussed with reference to FIG. 6A, similarity calculator 624 can evaluate the similarity between two or more DCT functions without reconstructing the inverse DCT (e.g., by calculating a distance between the DCT coefficients). However, the inverse DCTs are shown in graph 800 to illustrate how controller 602 can reduce the dimension of the sensor response timeseries $r(x)$ to match the dimension of the actuation signal timeseries $t(x)$ (or vice versa) by performing a discrete cosine transform and applying a quantization process.

Flow Diagrams

[0181] Referring now to FIG. 9, a flowchart of a process 900 for establishing device pairings between sensors and actuation devices is shown, according to an exemplary embodiment. In some embodiments, process 900 is performed by one or more components of controller 602, as described with reference to FIG. 6A. Process 900 can be used in a building and/or a building HVAC system to automatically establish device pairings between various sensors and causally-related actuation devices. The device pairings can then be used to automatically generate control loops for use in controlling the actuation devices.

[0182] Process 900 is shown to include collecting baseline sensor measurements from a plurality of sensors (step 902). The sensors can include some or all of sensors 640, as described with reference to FIG. 6A. For example, the sensors can include temperature sensors, humidity sensors, airflow sensors, lighting sensors, pressure sensors, voltage sensors, or any other type of sensor in a building and/or a building HVAC system. The

sensors can be distributed throughout a building and configured to measure various environmental conditions at different locations in the building. For example, one temperature sensor can be located in a first zone of the building and configured to measure the temperature of the first zone, whereas another temperature sensor can be located in a second zone of the building and configured to measure the temperature of the second zone. Similarly, the sensors can be distributed throughout a HVAC system and configured to measure conditions at different locations in the HVAC system. For example, one of temperature sensor can be a supply air temperature sensor configured to measure the temperature of the airflow provided to a building zone from an AHU, whereas another temperature sensor can be a return air temperature sensor configured to measure the temperature of the airflow returning from the building zone to the AHU.

[0183] The baseline sensor measurements may be received from the plurality of sensors during a baseline time period before an actuation signal or test signal is provided to actuation devices 650. In some embodiments, the baseline temperature measurements are used to generate a baseline sensor signal timeseries $a(x)$. The baseline sensor signal timeseries $a(x)$ may indicate the average sensor readings from the plurality of sensors before the actuation signal or test signal is provided to the actuation devices 650.

[0184] In some embodiments, step 902 includes storing the baseline sensor signal timeseries $a(x)$ in a timeseries database (e.g., timeseries database 616). The timeseries database can store the baseline sensor measurements from each of the plurality of sensors as separate baseline sensor signal timeseries $a(x)$ as shown in the following equation:

$$a(x) = \{a_1, a_2, a_3, \ldots, a_{P-1}, t_P\}$$

where each element a_i of the baseline sensor signal timeseries $a(x)$ is the value of a baseline sensor signal at a particular time (i.e., a sample of the baseline sensor signal) and P is the total number of elements in the baseline sensor signal timeseries $a(x)$.

[0185] Still referring to FIG. 9, process 900 is shown to include providing an actuation signal to an actuation device (step 904). The actuation device can include any of actuation devices 650, as described with reference to FIG. 6A. For example, the actuation device can include a chiller, heater, valve, air handling unit (AHU), damper, actuator, and/or any other physical device configured to affect a variable state or condition in a building or building HVAC system. The actuation device can include any of the equipment in building 10, HVAC system 100, waterside system 200, airside system 300, BMS 400, and/or BMS 500,

as described with reference to FIGS. 1-5. The actuation device can operate to affect various building conditions including temperature, humidity, airflow, lighting, air quality, power consumption, or any other variable state or condition in a building. In some embodiments, the actuation device receives the actuation signal from a controller (e.g., controller 602) via an output module.

[0186] In some embodiments, the actuation signal is a control signal for the actuation devices (e.g., operating setpoints, on/off commands, etc.). For example, the actuation signal can include commands to activate or deactivate the actuation device and/or commands to operate the actuation device a variable capacity (e.g., operate at 20% capacity, 40% capacity, etc.). If the actuation device is a device with a variable position (e.g., a valve, a damper, an actuator, etc.) the actuation signal can include position setpoints for the actuation device. The position setpoints can include commands to move to a fully closed position, a 50% open position, a fully open position, or any intermediate position. In some embodiments, the actuation signal is a predetermined test signal which differs from the normal actuation signals provided to the actuation device.

[0187] In some embodiments, the actuation signal is provided directly to the actuation device from the controller and used to adjust a physical operation of the actuation device (e.g., if the controller directly controls the actuation device). In other embodiments, the actuation signal is provided to an intermediate controller for the actuation devices. For example, a supervisory controller can provide a setpoint to a local controller for the actuation device. The local controller can then generate control signals for the actuation devices to achieve the setpoint received from the supervisory controller.

[0188] In some embodiments, step 904 includes using the actuation signal to generate an actuation signal timeseries $t(x)$. The actuation signal timeseries $t(x)$ can be stored in the timeseries database. The timeseries database can store each timeseries of actuation signal values as an actuation signal timeseries $t(x)$ as shown in the following equation:

$$t(x) = \{t_1, t_2, t_3, \ldots, t_{N-1}, t_N\}$$

where each element t_i of the actuation signal timeseries $t(x)$ is the value of the actuation signal at a particular time (i.e., a sample of the actuation signal) and N is the total number of elements in the actuation signal timeseries $t(x)$.

[0189] Still referring to FIG. 9, process 900 is shown to include recording sensor response signals from the plurality of sensors in response to the actuation signal (step 906). The

sensor response signals indicate the effect of the actuation device on the variables measured by the plurality of sensors. If a causal relationship exists between the actuation device and a particular sensor (i.e., the actuation device can affect the value measured by the sensor), the sensor response signal may change in response to providing the actuation signal to the actuation device. However, if no causal relationship exists between the actuation device and the sensor (i.e., the actuation device is not capable of affecting the value measured by the sensor), the sensor response signal may not change in response to providing the actuation signal to the actuation device.

[0190] In some embodiments, step 906 includes generating timeseries of sensor response values. Each timeseries of sensor response values can be stored in the timeseries database as a sensor response timeseries $r(x)$ as shown in the following equation:

$$r(x) = \{r_1, r_2, r_3, \ldots, r_{M-1}, r_M\}$$

where each element r_i of the sensor response timeseries $r(x)$ is the value of the sensor response signal at a particular time (i.e., a sample of the sensor response signal) and M is the total number of elements in the sensor response timeseries $r(x)$.

[0191] Process 900 is shown to include determining a similarity between the actuation signal and each sensor response signal (step 908). In some embodiments, step 908 includes calculating a similarity metric or similarity score indicating a similarity (e.g., a distance) between the actuation signal timeseries $t(x)$ and each of the sensor response timeseries $r(x)$. The following equation can be used to calculate the similarity metric between the actuation signal timeseries $t(x)$ and a given sensor response timeseries $r(x)$:

$$d(t(x), r(x)) = \sum_{i=1}^{i=N} \frac{\sqrt{(t_i - r_i)^2}}{\delta_i}$$

[0192] where t_i is the ith sample in the actuation signal timeseries $t(x)$, r_i is the ith sample in the sensor response timeseries $r(x)$, δ_i is the standard deviation of the ith samples, and N is the number of samples in each timeseries. Low values of the similarity metric $d(t(x), r(x))$ indicate a greater similarity (i.e., a lower distance between timeseries), whereas high values of the similarity metric $d(t(x), r(x))$ indicate a lesser similarity (i.e., a greater distance between timeseries). Step 908 can include calculating a similarity metric for each pairing of the actuation signal timeseries $t(x)$ with one of the sensor response timeseries $r(x)$.

[0193] The previous equation for calculating the similarity metric can be used if both the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ have the same number of samples. However, if the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ have different numbers of samples, additional processing may be required. For example, step 908 may include performing a discrete cosine transformation (DCT) of the actuation signal timeseries $t(x)$ and each sensor response timeseries $r(x)$ to generate sets of DCT coefficients for each timeseries. The sets of DCT coefficients can then be quantized, as described with reference to FIG. 6A, to produce arrays of compressed DCT coefficients for each timeseries. The similarity metrics can then be calculated based on the compressed DCT coefficients generated for each timeseries.

[0194] The compressed DCT coefficients generated for the actuation signal timeseries $t(x)$ can be represented by an array T, and the compressed DCT coefficients generated for a given sensor response timeseries $r(x)$ can be represented by an array R. The arrays T and R can be particular instances of the compressed array $C(k)$ generated by DCT quantizer 622 for the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$, respectively. Each array T and R can include a predetermined number N of DCT coefficients, defined by the quantization level applied by DCT quantizer 622. Examples of arrays T and R are as follows:

$$T = \langle t_1, t_2, \ldots, t_N \rangle$$
$$R = \langle r_1, r_2, \ldots, r_N \rangle$$

Step 908 can include generating an array T of DCT coefficients for the actuation signal timeseries $t(x)$ and an array R of DCT coefficients for each of the sensor response timeseries $r(x)$.

[0195] In some embodiments, step 908 includes calculating a similarity metric for the source timeseries $t(x)$ and $r(x)$ based on the corresponding arrays T and R of compressed DCT coefficients. Step 908 can include calculating the similarity metric using the following equation:

$$d(T,R) = \sum_{i=1}^{i=N} \frac{\sqrt{(t_i - r_i)^2}}{\delta_i}$$

where t_i is the ith DCT coefficient in the array T based on the actuation signal timeseries $t(x)$, r_i is the ith DCT coefficient in the array R based on the sensor response timeseries

$r(x)$, δ_i is the standard deviation of the ith DCT coefficients, and N is the number of DCT coefficients in each array T and R. Low values of the similarity metric $d(T, R)$ indicate a greater similarity, whereas high values of the similarity metric $d(T, R)$ indicate a lesser similarity. Step 908 can include calculating a similarity metric for each pairing of the array T based on the actuation signal timeseries $t(x)$ with one of the arrays R based on one of the sensor response timeseries $r(x)$.

[0196] Still referring to FIG. 9, process 900 is shown to include identifying the sensor response signal with the greatest similarity to the actuation signal and the corresponding sensor (step 910). Step 910 can include comparing the similarity metrics $d(T, R)$ calculated for the actuation device in combination with each of the plurality of sensors. Each similarity metric $d(T, R)$ indicates the similarity (i.e., the closeness) between the actuation signal timeseries $t(x)$ associated with the actuation device and the sensor response timeseries $r(x)$ associated with one of the sensors. For example, the similarity metric $d(T_1, R_1)$ may indicate the similarity between the actuation device and a first sensor of the plurality of sensors (corresponding to array R_1), whereas the similarity metric $d(T_1, R_2)$ may indicate the similarity between the actuation device and a second sensor of the plurality of sensors (corresponding to array R_2).

[0197] Step 910 can include identifying all of the similarity metrics associated with the actuation device (e.g., $d(T_1, R_1), \ldots, d(T_1, R_P)$, where P is the total number of sensors and/or sensor response timeseries $r(x)$). In some embodiments, step 910 includes determining which of the identified similarity metrics is the lowest. The lowest similarity metric may indicate the closest match between the actuation signal timeseries $t(x)$ associated with the actuation device and the sensor response timeseries $r(x)$ associated with one of the sensors. In other embodiments, step 910 can include determining which of the identified similarity metrics is the highest. For example, other techniques for calculating the similarity metric may produce larger similarity metrics when two timeseries match more closely. Regardless of how the similarity metric is calculated, step 910 can include identifying the similarity metric which indicates the closest match between the actuation signal timeseries $t(x)$ and the corresponding sensor response timeseries $r(x)$.

[0198] If the actuation device has the same similarity metric with multiple sensors (e.g., $d(T_1, R_1) = d(T_1, R_2)$), step 910 can include examining the time delay Δw between the actuation signal timeseries $t(x)$ associated with the actuation device and sensor response

timeseries $r(x)$ associated with each of the sensors. The time delay Δw may indicate the delay between the time w_1 at which the actuation signal is applied to the actuation device and the time w_2 at which the effects of the actuation signal are evident in each sensor response (e.g., $\Delta w = w_2 - w_1$). Step 910 can include identifying the sensor and/or sensor response timeseries $r(x)$ with the lowest time delay Δw relative to the actuation signal timeseries $t(x)$.

[0199] Still referring to FIG. 9, process 900 is shown to include establishing a device pairing between the actuation device and the identified sensor (step 912). The device pairing can include the actuation device and the sensor identified in step 910. The device pairing between the actuation device and the sensor indicates that the actuation device is capable of affecting the value measured by the sensor. In some embodiments step 910 includes using the device pairing to automatically generate and store causal relationships between the actuation device and the identified sensor.

[0200] Process 900 is shown to include using the device pairing to generate and provide control signals to the actuation device (step 914). Step 914 can include using the device pairing to create a feedback control loop for HVAC system 600. The feedback control loop can receive a feedback signal from the identified sensor and can provide a control signal to the actuation device. Step 914 can include using the device pairing to define the sensor and actuation device in the control loop. For example, step 914 can include creating a control loop which receives a feedback signal from the sensor in the device pairing and provides a control signal to the actuation device in the device pairing. Step 914 can include mapping the sensor readings from the sensor in the device pairing to the feedback signal in the control loop. Similarly, step 914 can include mapping the actuation signals provided to the actuation device in the device pairing to the control signal in the control loop.

[0201] Step 914 can include using the feedback control loop to generate actuation signals for the actuation device. Step 914 can include using state-based algorithms, extremum seeking control (ESC) algorithms, proportional-integral (PI) control algorithms, proportional-integral-derivative (PID) control algorithms, model predictive control (MPC) algorithms, or any other type of control methodology to generate the actuation signals for the actuation device based on the sensor readings. For example, if the sensor reading from the sensor indicates that the temperature of a particular building zone is below a temperature setpoint for the building zone, step 914 can include providing an actuation signal to the actuation device to increase the amount of heating provided to the building zone.

Advantageously, process 900 can be performed to automatically establish relationships between various actuation devices and sensors based on the device pairings to allow controller 602 to determine which of the actuation devices can be operated to affect a given sensor reading.

[0202] Referring now to FIG. 10, a flowchart of a process 1000 for handling dimensional mismatches between actuation signal timeseries and sensor response timeseries is shown, according to an exemplary embodiment. In some embodiments, process 1000 is performed by one or more components of controller 602, as described with reference to FIG. 6A. Process 1000 can be used in a building and/or a building HVAC system to automatically establish device pairings between various sensors and actuation devices when the timeseries have different sampling rates and/or different numbers of samples. The device pairings can then be used to automatically generate control loops for use in controlling the actuation devices.

[0203] Process 1000 is shown to include performing a discrete cosine transformation (DCT) of actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$ to generate sets of DCT coefficients (step 1002). In some embodiments, step 1002 is performed by discrete cosine transformer 620, as described with reference to FIG. 6A. Step 1002 can include performing a DCT for each actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. A DCT expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. There are eight standard DCT variants, commonly referred to as DCT-I, DCT-II, DCT-III, DCT-IV, DCT-V, DCT-VI, DCT-VII, and DCT-VIII. One of these variants (i.e., DCT-II) is discussed in detail below. However, it should be understood that step 1002 can use any standard or non-standard DCT variant in other embodiments.

[0204] In some embodiments, step 1002 includes performing a DCT for each actuation signal timeseries $t(x)$ using the following equation:

$$T(k) = \sum_{i=0}^{N-1} t_i \cos\left[\frac{\pi}{N}\left(i + \frac{1}{2}\right)k\right] \qquad k = 0, \ldots, N-1$$

where $T(k)$ is the kth coefficient of the DCT of the actuation signal timeseries $t(x)$, t_i is the ith sample of the actuation signal timeseries $t(x)$, and N is the number of samples of the actuation signal timeseries $t(x)$. Step 1002 can include generating an array T of the DCT

coefficients (e.g., $T = [T(0), T(1), T(2), \ldots, T(N-2), T(N-1)]$) where the length of the array T is the same as the number of samples N of the actuation signal timeseries $t(x)$.

[0205] Similarly, step 1002 can include performing a DCT for each sensor response timeseries $r(x)$ using the following equation:

$$R(k) = \sum_{i=0}^{M-1} r_i \cos\left[\frac{\pi}{M}\left(i + \frac{1}{2}\right)k\right] \quad k = 0, \ldots, M-1$$

where $R(k)$ is the kth coefficient of the DCT of the sensor response timeseries $r(x)$, r_i is the ith sample of the sensor response timeseries $r(x)$, and M is the number of samples of the sensor response timeseries $r(x)$. Step 1002 can include generating an array R of the DCT coefficients (e.g., $R = [R(0), R(1), R(2), \ldots, R(M-2), R(M-1)]$) where the length of the array R is the same as the number of samples M of the sensor response timeseries $r(x)$.

[0206] The following example illustrates the result of applying DCT to an input timeseries $X(n)$. The input timeseries $X(n)$ can be an actuation signal timeseries $t(x)$ or a sensor response timeseries $r(x)$ as previously described. The samples of the input timeseries $X(n)$ are shown in the following array:

$$X(n) = [1.00, 1.70, 2.00, 2.00, 4.30, 4.50, 3.00, 3.00, 2.30, 2.20, 2.20, 2.30]$$

where the input timeseries $X(n)$ includes twelve timeseries values $X(1), \ldots, X(12)$. Applying DCT to the input timeseries $X(n)$ results in a set of DCT coefficients, shown in the following array:

$$Y(k) = [8.80, -0.57, -2.65, -1.15, 0.81, 0.52, -0.20, -0.93, -0.63, 0.32, 0.50, -0.06]$$

where the array of DCT coefficients $Y(k)$ includes twelve DCT coefficients $Y(1), \ldots, Y(12)$.

[0207] Still referring to FIG. 10, process 1000 is shown to include identifying a dimensional mismatch between the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ (step 1004). In some embodiment, step 1004 is performed by dimensional mismatch identifier 618, as described with reference to FIG. 6A. A dimensional mismatch may occur when two timeseries have a different number of samples and/or sampling rates. In some embodiments, step 1004 includes determining the size of each timeseries. For example, step 1004 can include determining the number of samples N in the actuation

signal timeseries $t(x)$ and the number of samples M in the sensor response timeseries $r(x)$. The number of samples N and M can be determined by counting the number of samples in the original timeseries $t(x)$ and $r(x)$ or the number of elements in each of the arrays of DCT coefficients T and R. Step 1004 can include identifying a dimensional mismatch in response to a determination that the number of samples N in the actuation signal timeseries $t(x)$ or the array T is different from the number of samples M in the sensor response timeseries $r(x)$ or the array R (i.e., $N \neq M$).

[0208] In some embodiments, step 1004 includes determining the sampling rate of each timeseries. In some embodiments, the sampling rate of a timeseries may be stored as metadata associated with the timeseries in timeseries database 616. Step 1004 can include determining the sampling rate of a timeseries by reading the sampling rate from the metadata in timeseries database 616. In other embodiments, step 1004 includes calculating the sampling rate for one or more timeseries based on the size of the timeseries and the range of time spanned by the timeseries.

[0209] Step 1004 can include identifying a start time and an end time for the timeseries by reading the timestamps associated with the first and last data samples in the timeseries. Step 1004 can include calculating the sampling rate by dividing the size of the timeseries by the difference between the end time and the start time, as shown in the following equation:

$$sampling_rate = \frac{size_of_timeseries}{end_time - start_time}$$

where $size_of_timeseries$ is the number of samples M or N in the timeseries, end_time is the timestamp associated with the last sample in the timeseries, $start_time$ is the timestamp associated with the first sample in the timeseries, and $sampling_rate$ is the sampling rate of the timeseries, expressed as the number of samples per unit time (e.g., 0.8 samples/hour). Step 1004 can include identifying a dimensional mismatch in response to a determination that two timeseries have different sampling rates.

[0210] In some embodiments, step 1004 includes correcting a dimensional mismatch by increasing the number of samples of the timeseries with the fewer number of samples (e.g., by interpolating between samples). In other embodiments, step 1004 includes correcting dimensional mismatch by reducing the number of samples of the timeseries with the greater number of samples (e.g., by discarding extra samples). In other embodiments, step 1004

merely identifies a dimensional mismatch which is corrected by subsequent steps of process 1000.

[0211] Still referring to FIG. 10, process 1000 is shown to include applying a quantization to the sets of DCT coefficients generated in step 1002 to equalize the number of DCT coefficients in each set (step 1006). In some embodiments, step 1006 is performed by DCT quantizer 622, as described with reference to FIG. 6A. Step 1006 can be performed in response to a determination in step 1004 that a dimensional mismatch exists between the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$.

[0212] As described above, the DCT process performed in step 1002 converts an input data timeseries $X(n)$ into a sum of cosine functions which oscillate at different frequencies. The cosine function with the lowest frequency is typically first in the summation, followed by cosine functions with successively higher frequencies. Accordingly, the DCT coefficient which occurs first in the array of DCT coefficients $Y(k)$ represents the magnitude of the lowest frequency cosine function. Each of the following DCT coefficients represents the magnitude of a cosine function with a successively higher oscillation frequency.

[0213] Step 1006 can include applying a quantization process to the sets of DCT coefficients by filling some of the higher frequency DCT coefficients with zeros. This has the effect of removing some of the higher frequency components (i.e., cosine functions) from the summation while retaining the lower frequency components. In some embodiments, step 1006 includes performing the quantization process using a predetermined quantization level. The quantization level may define the number of the DCT coefficients which are retained (i.e., not filled with zeros). For example, a quantization level of six may retain the DCT coefficients applied to the six lowest frequency cosine functions (e.g., the first six DCT coefficients in the array) while the remaining DCT coefficients are filled with zeros.

[0214] The following example illustrates the result of a quantization process which can be performed in step 1006. Step 1006 can include modifying the array of DCT coefficients $Y(k)$ shown above to form the following quantized array $QY(k)$:

$$QY(k) = [8.80, -0.57, -2.65, -1.15, 0.81, 0.52, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]$$

In this example, a quantization level of six is applied, meaning that only the first six DCT coefficients are retained from the original array of DCT coefficients $Y(k)$. The remaining

DCT coefficients are filled with zeros. True compression can be achieved by not storing the zeros. For example, step 1006 can include storing the following compressed array $C(k)$:

$$C(k) = [8.80, -0.57, -2.65, -1.15, 0.81, 0.52]$$

in which the coefficients filled with zeros are discarded to produce a compressed array with a length equal to the quantization level applied. In this example, the compressed array $C(k)$ has a length of six resulting from the use of a quantization level of six. In various embodiments, step 1006 can use a quantization level of six or any other quantization level to produce compressed arrays of various lengths.

[0215] In some embodiments, step 1006 includes automatically determining the quantization level to apply based on the number of samples in each of the original actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. As described above, the number of DCT coefficients produced in step 1002 for a given input timeseries $X(n)$ may be equal to the number of samples in the timeseries $X(n)$ prior to performing DCT. For example, an input timeseries $X_1(n)$ with twelve samples may result in twelve DCT coefficients in the resultant DCT coefficient array $Y_1(k)$, whereas an input timeseries $X_2(n)$ with ten samples may result in ten DCT coefficients in the resultant DCT coefficient array $Y_2(k)$. In some embodiments, step 1006 includes identifying the actuation signal timeseries $t(x)$ or sensor response timeseries $r(x)$ with the fewest samples and applies a quantization level equal to the number of samples in the identified timeseries.

[0216] In some embodiments, step 1006 includes applying the same quantization level to the sets of DCT coefficients corresponding to each of the original actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. Using the same quantization level for each of the original timeseries may result in the same number of compressed DCT coefficients being stored for each of the original actuation signal timeseries $t(x)$ and sensor response timeseries $r(x)$. In some embodiments, the number of stored DCT coefficients is equal to the number of samples in the original timeseries with the fewest samples. Advantageously, this allows for direct comparison of the DCT coefficients in the compressed arrays $C(k)$ generated for each of the original timeseries without requiring decompression, interpolation, synchronization, or other processing steps after the compressed arrays $C(k)$ are generated.

[0217] In some embodiments, step 1006 includes generating a compressed timeseries T_α based on each compressed array of DCT coefficients. Step 1006 can include storing the compressed timeseries T_α using the following data structure:

$$T_\alpha = \langle \alpha, \delta, \rho, \kappa, \langle \psi \rangle, \langle v_1, v_2, \dots v_\rho \rangle \rangle$$

where α is the timeseries ID of the source timeseries (e.g., the actuation signal timeseries $t(x)$ or sensor response timeseries $r(x)$), δ is the dimension of the source timeseries (e.g., the number of samples in the source timeseries), ρ is the quantization level applied by in step 1006, κ is a pointer for metadata, $\langle \psi \rangle$ indicates the start time and end time of samples in the source timeseries, and $\langle v_1, v_2, \dots v_\rho \rangle$ is the array of compressed DCT coefficients. An example of a compressed timeseries stored using this data structure is as follows:

$$T_{203} = \langle 203,12,6, \langle 2016{:}\,10{:}\,05{:}\,12{:}\,00{:}\,00,2016{:}\,10{:}\,05{:}\,13{:}\,00{:}\,00 \rangle, \langle 8.80, -0.57, -2.65, -1.15, 0.81, 0.52 \rangle \rangle$$

where 203 is the timeseries ID of the source timeseries, 12 is size of the source timeseries (e.g., 12 samples in the source timeseries), 6 is the quantization level applied by in step 1006, 2016:10:05:12:00:00 is the start time of the source timeseries (e.g., the timestamp of the earliest sample in the source timeseries), 2016:10:05:13:00:00 is the end time of the source timeseries (e.g., the timestamp of the latest sample in the source timeseries), and the array ⟨8.80, -0.57, -2.65, -1.15, 0.81, 0.52⟩ includes the compressed DCT coefficients generated in step 1006.

[0218] Still referring to FIG. 10, process 1000 is shown to include determining a similarity between the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ based on the quantized sets of DCT coefficients (step 1008). The quantized DCT coefficients generated for the actuation signal timeseries $t(x)$ can be represented by an array T, and the compressed DCT coefficients generated for a given sensor response timeseries $r(x)$ can be represented by an array R. The arrays T and R can be particular instances of the compressed array $C(k)$ generated in step 1006 for the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$, respectively. Each array T and R can include a predetermined number N of DCT coefficients, defined by the quantization level applied in step 1006. Examples of arrays T and R are as follows:

$$T = \langle t_1, t_2, \dots, t_N \rangle$$
$$R = \langle r_1, r_2, \dots, r_N \rangle$$

[0219] In some embodiments, step 1008 includes calculating a similarity metric for the source timeseries $t(x)$ and $r(x)$ based on the corresponding arrays T and R of compressed DCT coefficients. Step 1008 can include calculating the similarity metric using the following equation:

$$d(T,R) = \sum_{i=1}^{i=N} \frac{\sqrt{(t_i - r_i)^2}}{\delta_i}$$

where t_i is the ith DCT coefficient in the array T based on the actuation signal timeseries $t(x)$, r_i is the ith DCT coefficient in the array R based on the sensor response timeseries $r(x)$, δ_i is the standard deviation of the ith DCT coefficients, and N is the number of DCT coefficients in each array T and R. Low values of the similarity metric $d(T,R)$ indicate a greater similarity, whereas high values of the similarity metric $d(T,R)$ indicate a lesser similarity.

[0220] Process 1000 is shown to include establishing a device pairing between an actuation device and a sensor based on the similarity between the corresponding timeseries (step 1010). The actuation device in step 1010 may be the specific actuation device to which the actuation signal $t(x)$ is provided, whereas the sensor in step 1010 may be the specific sensor from which the sensor response signal $r(x)$ is received. In some embodiments, step 1010 includes comparing the similarity metric $d(T,R)$ calculated in step 1008 to a threshold value. Step 1010 can include generating a device pairing between the sensor and the actuation device if the similarity metric $d(T,R)$ is less than the threshold value (e.g., $d(T,R) < threshold$). The threshold value can be a predefined value or a calculated value (e.g., a standard deviation of the DCT coefficients).

[0221] In some embodiments, the threshold value is a similarity metric between the actuation signal timeseries $t(x)$ and a baseline (e.g., average) sensor signal timeseries $a(x)$ over a predetermined time period. The baseline sensor signal timeseries $a(x)$ can indicate the average sensor response from the sensor before the actuation signal is applied to the actuation device (e.g., baseline sensor readings), whereas the sensor response timeseries $r(x)$ can indicate the sensor response from the same sensor after the actuation signal is applied to the actuation device. If the actuation device affects the sensor, the actuation signal timeseries $t(x)$ is expected to be more similar to the sensor response timeseries $r(x)$ than the baseline sensor signal timeseries $a(x)$. Accordingly, the similarity metric $d(T,R)$ between the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ is

expected to be lower (i.e., more similar) than the similarity metric $d(T, A)$ between the actuation signal timeseries $t(x)$ and the baseline sensor signal timeseries $a(x)$.

[0222] In some embodiments, step 1010 includes generating DCT coefficients and compressed DCT coefficients for each baseline sensor signal timeseries $a(x)$, sensor response timeseries $r(x)$, and actuation signal timeseries $t(x)$. Step 1010 can include calculating a similarity metric $d(T, R)$ between the actuation signal timeseries $t(x)$ and the sensor response timeseries $r(x)$ associated with the temperature sensor. Step 1010 can also include calculating a baseline similarity metric $d(T, A)$ between the actuation signal timeseries $t(x)$ and baseline sensor signal timeseries $a(x)$ associated with the same temperature sensor. Step 1010 can include generating a device pairing between the sensor and the actuation device if the similarity metric $d(T, R)$ for the combination of the sensor and the actuation device indicates a greater similarity (e.g., a lower similarity metric) than the baseline similarity metric $d(T, A)$ for the sensor and the actuation device.

[0223] In some embodiments, step 1010 includes comparing the similarity metrics $d(T, R)$ calculated for the actuation device in combination with each of a plurality of sensors. Each similarity metric $d(T, R)$ indicates the similarity (i.e., the closeness) between the actuation signal timeseries $t(x)$ associated with the actuation device and the sensor response timeseries $r(x)$ associated with one of the sensors. For example, the similarity metric $d(T_1, R_1)$ may indicate the similarity between the actuation device and a first sensor of the plurality of sensors (corresponding to array R_1), whereas the similarity metric $d(T_1, R_2)$ may indicate the similarity between the actuation device and a second sensor of the plurality of sensors (corresponding to array R_2).

[0224] Step 1010 can include identifying all of the similarity metrics associated with the actuation device (e.g., $d(T_1, R_1), \ldots, d(T_1, R_P)$, where P is the total number of sensors and/or sensor response timeseries $r(x)$). In some embodiments, step 1010 includes determining which of the identified similarity metrics is the lowest. The lowest similarity metric may indicate the closest match between the actuation signal timeseries $t(x)$ associated with the actuation device and the sensor response timeseries $r(x)$ associated with one of the sensors. In other embodiments, step 1010 can include determining which of the identified similarity metrics is the highest. For example, other techniques for calculating the similarity metric may produce larger similarity metrics when two timeseries match more closely. Regardless of how the similarity metric is calculated, step 1010 can include

identifying the similarity metric which indicates the closest match between the actuation signal timeseries $t(x)$ and the corresponding sensor response timeseries $r(x)$.

[0225] If the actuation device has the same similarity metric with multiple sensors (e.g., $d(T_1, R_1) = d(T_1, R_2)$), step 1010 can include examining the time delay Δw between the actuation signal timeseries $t(x)$ associated with the actuation device and sensor response timeseries $r(x)$ associated with each of the sensors. The time delay Δw may indicate the delay between the time w_1 at which the actuation signal is applied to the actuation device and the time w_2 at which the effects of the actuation signal are evident in each sensor response (e.g., $\Delta w = w_2 - w_1$). Step 1010 can include identifying the sensor and/or sensor response timeseries $r(x)$ with the lowest time delay Δw relative to the actuation signal timeseries $t(x)$.

[0226] Still referring to FIG. 10, process 1000 is shown to include using the device pairing to generate and provide control signals to the actuation device (step 1012). Step 1012 can include using the device pairing to create a feedback control loop for HVAC system 600. The feedback control loop can receive a feedback signal from the identified sensor and can provide a control signal to the actuation device. Step 1012 can include using the device pairing to define the sensor and actuation device in the control loop. For example, step 1012 can include creating a control loop which receives a feedback signal from the sensor in the device pairing and provides a control signal to the actuation device in the device pairing. Step 1012 can include mapping the sensor readings from the sensor in the device pairing to the feedback signal in the control loop. Similarly, step 1012 can include mapping the actuation signals provided to the actuation device in the device pairing to the control signal in the control loop.

[0227] Step 1012 can include using the feedback control loop to generate actuation signals for the actuation device. Step 1012 can include using state-based algorithms, extremum seeking control (ESC) algorithms, proportional-integral (PI) control algorithms, proportional-integral-derivative (PID) control algorithms, model predictive control (MPC) algorithms, or any other type of control methodology to generate the actuation signals for the actuation device based on the sensor readings. For example, if the sensor reading from the sensor indicates that the temperature of a particular building zone is below a temperature setpoint for the building zone, step 1012 can include providing an actuation signal to the actuation device to increase the amount of heating provided to the building zone. Advantageously, process 1000 can be performed to automatically establish relationships

between various actuation devices and sensors based on the device pairings to allow controller 602 to determine which of the actuation devices can be operated to affect a given sensor reading.

Configuration of Exemplary Embodiments

[0228] The construction and arrangement of the systems and methods as shown in the various exemplary embodiments are illustrative only. Although only a few embodiments have been described in detail in this disclosure, many modifications are possible (e.g., variations in sizes, dimensions, structures, shapes and proportions of the various elements, values of parameters, mounting arrangements, use of materials, colors, orientations, etc.). For example, the position of elements can be reversed or otherwise varied and the nature or number of discrete elements or positions can be altered or varied. Accordingly, all such modifications are intended to be included within the scope of the present disclosure. The order or sequence of any process or method steps can be varied or re-sequenced according to alternative embodiments. Other substitutions, modifications, changes, and omissions can be made in the design, operating conditions and arrangement of the exemplary embodiments without departing from the scope of the present disclosure.

[0229] The present disclosure contemplates methods, systems and program products on any machine-readable media for accomplishing various operations. The embodiments of the present disclosure can be implemented using existing computer processors, or by a special purpose computer processor for an appropriate system, incorporated for this or another purpose, or by a hardwired system. Embodiments within the scope of the present disclosure include program products comprising machine-readable media for carrying or having machine-executable instructions or data structures stored thereon. Such machine-readable media can be any available media that can be accessed by a general purpose or special purpose computer or other machine with a processor. By way of example, such machine-readable media can comprise RAM, ROM, EPROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to carry or store desired program code in the form of machine-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer or other machine with a processor. Combinations of the above are also included within the scope of machine-readable media. Machine-executable instructions include, for example, instructions and data which cause a general

purpose computer, special purpose computer, or special purpose processing machines to perform a certain function or group of functions.

[0230] Although the figures show a specific order of method steps, the order of the steps may differ from what is depicted. Also two or more steps can be performed concurrently or with partial concurrence. Such variation will depend on the software and hardware systems chosen and on designer choice. All such variations are within the scope of the disclosure. Likewise, software implementations could be accomplished with standard programming techniques with rule based logic and other logic to accomplish the various connection steps, processing steps, comparison steps and decision steps.

WHAT IS CLAIMED IS:

1. A heating, ventilation, or air conditioning (HVAC) system for a building, the HVAC system comprising:

a plurality of actuation devices operable to affect one or more variables in the building;

a plurality of sensors configured to measure the variables affected by the actuation devices; and

a controller configured to:

operate the actuation devices to affect one or more of the measured variables by providing an actuation signal to the actuation devices;

receive sensor response signals from the sensors, the sensor response signals indicating an effect of the actuation signal on the measured variables;

for each of the sensor response signals, calculate a similarity metric indicating a similarity between the sensor response signal and the actuation signal; and

automatically establish a device pairing comprising one of the actuation devices and one of the sensors based on the similarity metrics.

2. The HVAC system of Claim 1, wherein the device pairing defines a control relationship between the actuation device in the device pairing and the sensor in the device pairing;

wherein the control relationship indicates that the actuation device in the device pairing is operable to control a variable measured by the sensor in the device pairing.

3. The HVAC system of Claim 1, wherein the controller is configured to:

automatically create a feedback control loop comprising the actuation device in the device pairing and the sensor in the device pairing; and

use the feedback control loop to generate and provide control signals to the actuation device in the device pairing based on measurements received from the sensor in the device pairing.

4. The HVAC system of Claim 1, wherein the controller is configured to calculate the similarity metrics based on differences between samples of the actuation signal and corresponding samples of each of the sensor response signals.

5. The HVAC system of Claim 1, wherein the controller is configured to:

determine, for each of the sensor response signals, a delay time of the sensor response signal relative to the actuation signal;

identify a sensor corresponding to the sensor response signal having a minimum of the delay times; and

establish the device pairing such that the identified sensor is included in the device pairing.

6. The HVAC system of Claim 1, wherein the controller is configured to:

generate an actuation signal timeseries comprising a plurality of samples of the actuation signal;

generate a sensor response timeseries for each of the sensor response signals, each sensor response timeseries comprising a plurality of samples of one of the measured variables; and

calculate the similarity metrics by comparing the actuation signal timeseries to each of the sensor response timeseries.

7. The HVAC system of Claim 6, wherein the controller is configured to:

detect a dimensional mismatch between the actuation signal timeseries and one or more of the sensor response timeseries; and

correct the dimensional mismatch by modifying at least one of the actuation signal timeseries and one or more of the sensor response timeseries.

8. The HVAC system of Claim 1, wherein the controller is configured to:

apply a discrete cosine transformation (DCT) to the actuation signal and each of the sensor response signals, each DCT generating a plurality of DCT coefficients; and

calculate the similarity metrics by comparing the DCT coefficients resulting from the DCT of the actuation signal to DCT coefficients resulting from the DCT of each sensor response signal.

9. The HVAC system of Claim 1, wherein the controller is configured to:

receive baseline sensor signals from each of the plurality of sensors, the baseline sensor signals indicating values of the measured variables during a time period before the actuation signal is provided to the actuation devices; and

for each of the baseline sensor signals, calculate a similarity metric indicating a similarity between the baseline sensor signal and the actuation signal.

10. The HVAC system of Claim 9, wherein the controller is configured to:

determine, for each of the plurality of sensors, whether the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal; and

establish the device pairing in response to a determination that the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal.

11. A method for establishing device pairings in a heating, ventilation, or air conditioning (HVAC) system for a building, the method comprising:

operating one or more actuation devices to affect one or more measured variables in the building by providing an actuation signal to the actuation devices;

receiving sensor response signals from a plurality of sensors configured to measure the variables affected by the actuation devices, the sensor response signals indicating an effect of the actuation signal on the measured variables;

calculating a similarity metric for each of the sensor response signals, each similarity metric indicating a similarity between the actuation signal and one of the sensor response signals; and

automatically establishing a device pairing comprising one of the actuation devices and one of the sensors based on the similarity metrics.

12. The method of Claim 11, wherein the device pairing defines a control relationship between the actuation device in the device pairing and the sensor in the device pairing;

wherein the control relationship indicates that the actuation device in the device pairing is operable to control a variable measured by the sensor in the device pairing.

13. The method of Claim 11, further comprising:

automatically creating a feedback control loop comprising the actuation device in the device pairing and the sensor in the device pairing; and

using the feedback control loop to generate and provide control signals to the actuation device in the device pairing based on measurements received from the sensor in the device pairing.

14. The method of Claim 11, wherein the similarity metrics are calculated based on differences between samples of the actuation signal and corresponding samples of each of the sensor response signals.

15. The method of Claim 11, further comprising:

determining, for each of the sensor response signals, a delay time of the sensor response signal relative to the actuation signal;

identifying a sensor corresponding to the sensor response signal having a minimum of the delay times; and

establishing the device pairing such that the identified sensor is included in the device pairing.

16. The method of Claim 11, further comprising:

generating an actuation signal timeseries comprising a plurality of samples of the actuation signal;

generating a sensor response timeseries for each of the sensor response signals, each sensor response timeseries comprising a plurality of samples of one of the measured variables; and

calculating the similarity metrics by comparing the actuation signal timeseries to each of the sensor response timeseries.

17. The method of Claim 16, further comprising:

detecting a dimensional mismatch between the actuation signal timeseries and one or more of the sensor response timeseries; and

correcting the dimensional mismatch by modifying at least one of the actuation signal timeseries and one or more of the sensor response timeseries.

18. The method of Claim 11, further comprising:

applying a discrete cosine transformation (DCT) to the actuation signal and each of the sensor response signals, each DCT generating a plurality of DCT coefficients; and

calculating the similarity metrics by comparing the DCT coefficients resulting from the DCT of the actuation signal to DCT coefficients resulting from the DCT of each sensor response signal.

19. The method of Claim 11, further comprising:

receiving baseline sensor signals from each of the plurality of sensors, the baseline sensor signals indicating values of the measured variables during a time period before the actuation signal is provided to the actuation devices; and

for each of the baseline sensor signals, calculating a similarity metric indicating a similarity between the baseline sensor signal and the actuation signal.

20. The method of Claim 19, further comprising:

determining, for each of the plurality of sensors, whether the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal; and

establishing the device pairing in response to a determination that the similarity metric calculated based on the sensor response signal indicates a greater similarity than the similarity metric calculated based on the baseline sensor signal.

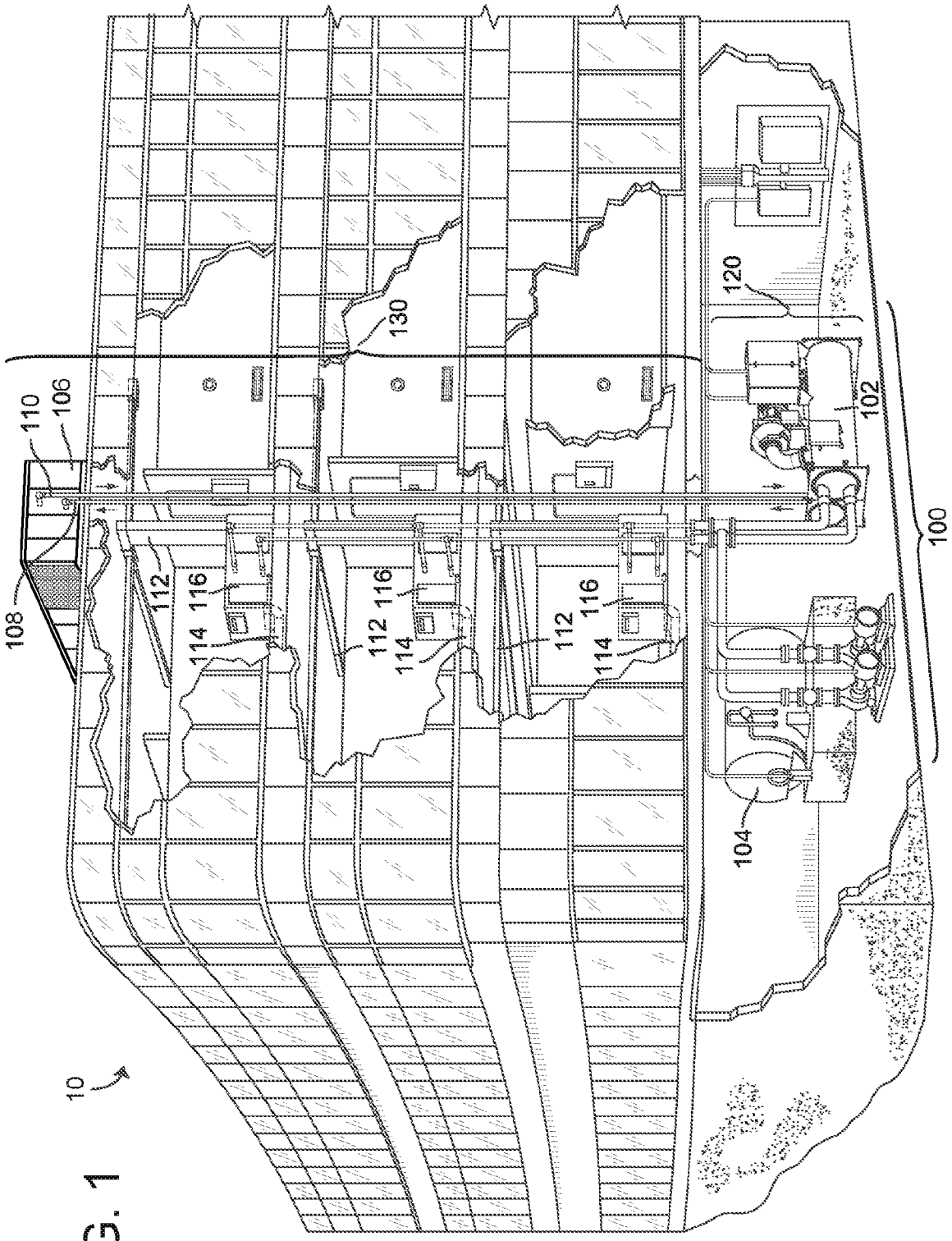


FIG. 1

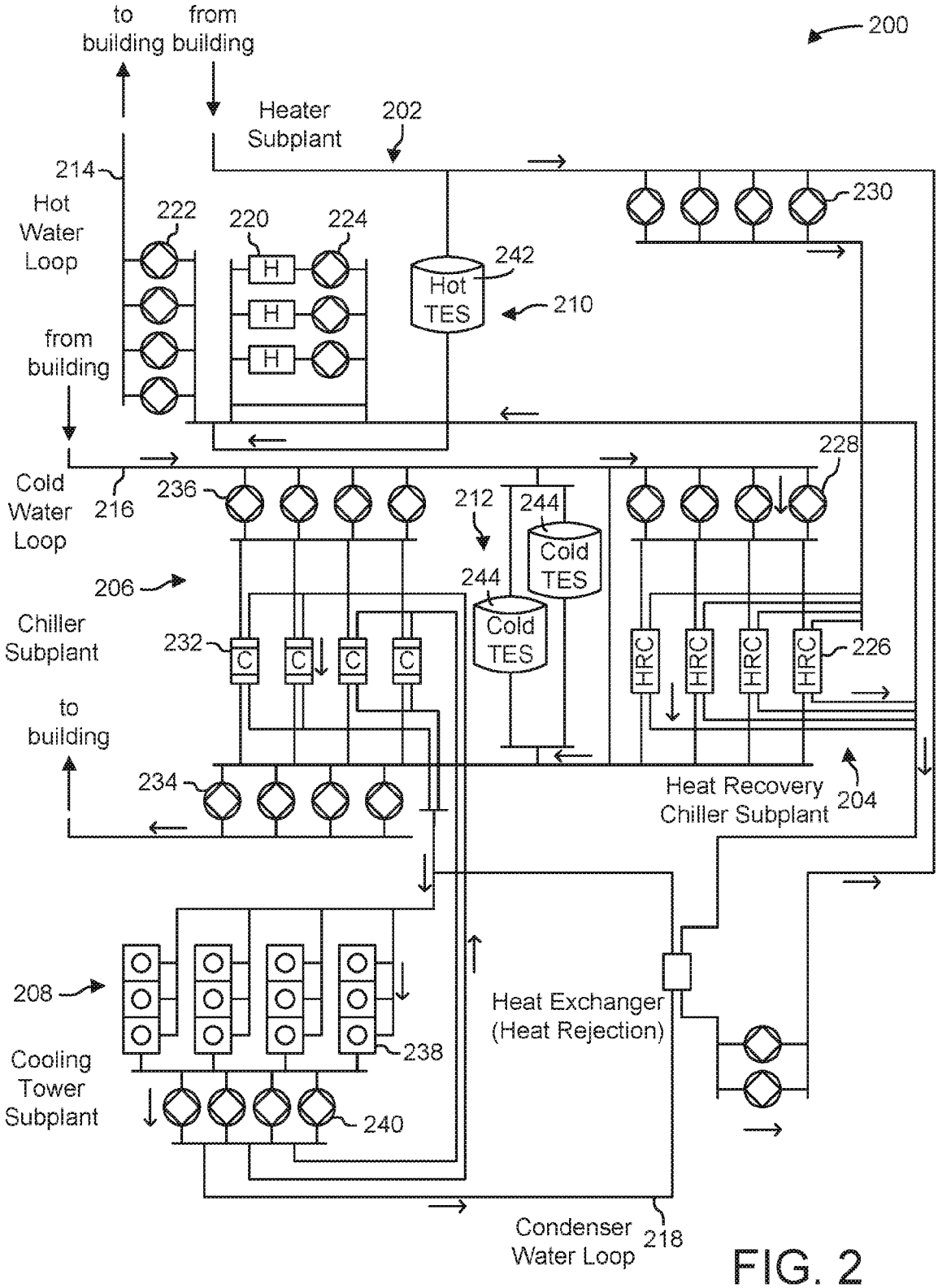

to building
from building
200
Heater Subplant
202
214
Hot Water Loop
222
220
224
H
H
H
242
Hot TES
210
230
from building
228
Cold Water Loop
216
236
212
244
Cold TES
244
Cold TES
206
Chiller Subplant
232
C
C
C
C
HRC
HRC
HRC
HRC
226
to building
234
Heat Recovery Chiller Subplant
204
208
Cooling Tower Subplant
238
240
Heat Exchanger (Heat Rejection)
Condenser Water Loop
218

FIG. 2

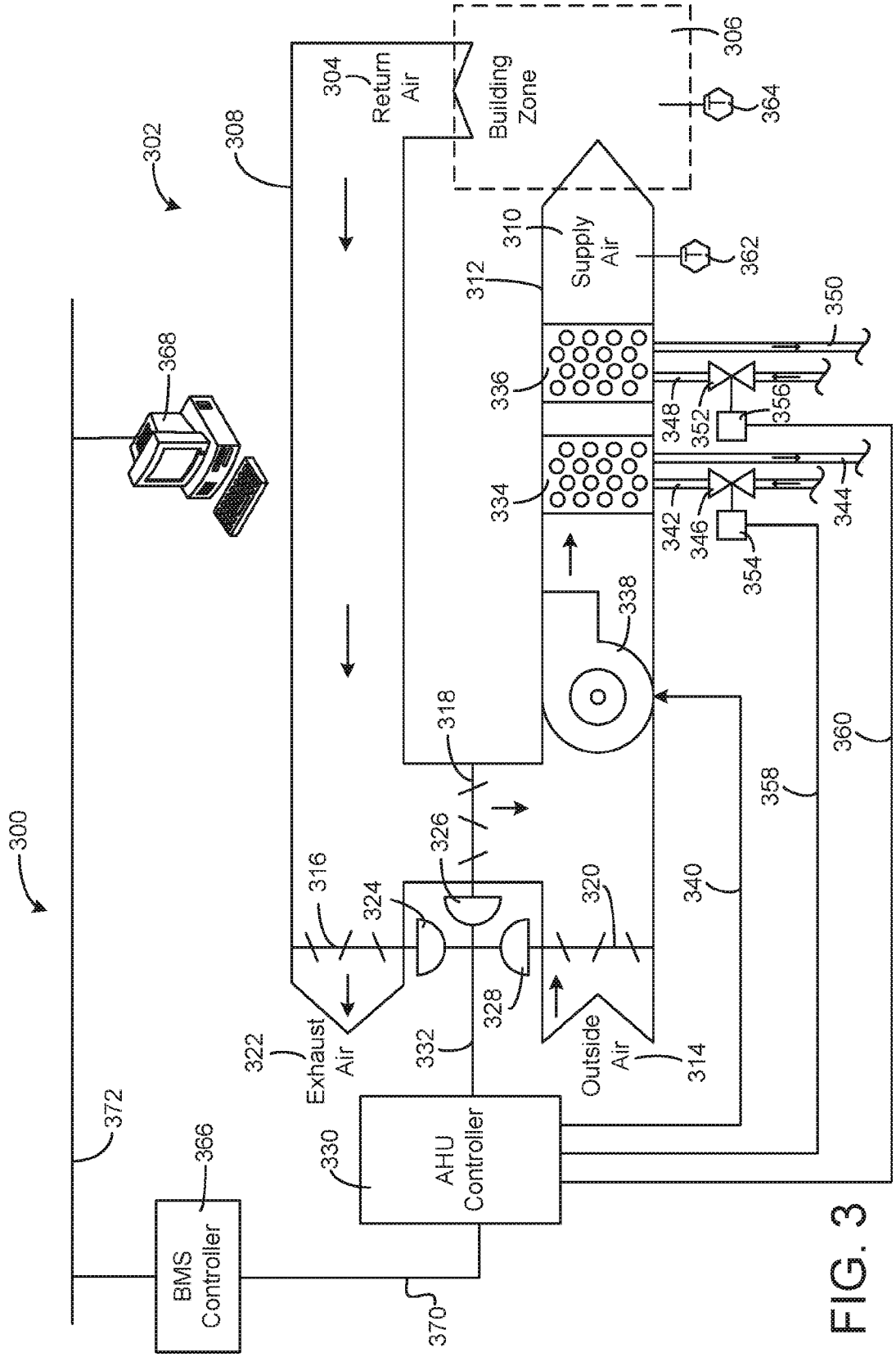

300
302
304
Return Air
306
Building Zone
308
310
Supply Air
312
314
Outside Air
316
318
320
322
Exhaust Air
324
326
328
330
AHU Controller
332
334
336
338
340
342
344
346
348
350
352
354
356
358
360
362
364
366
BMS Controller
368
370
372

FIG. 3

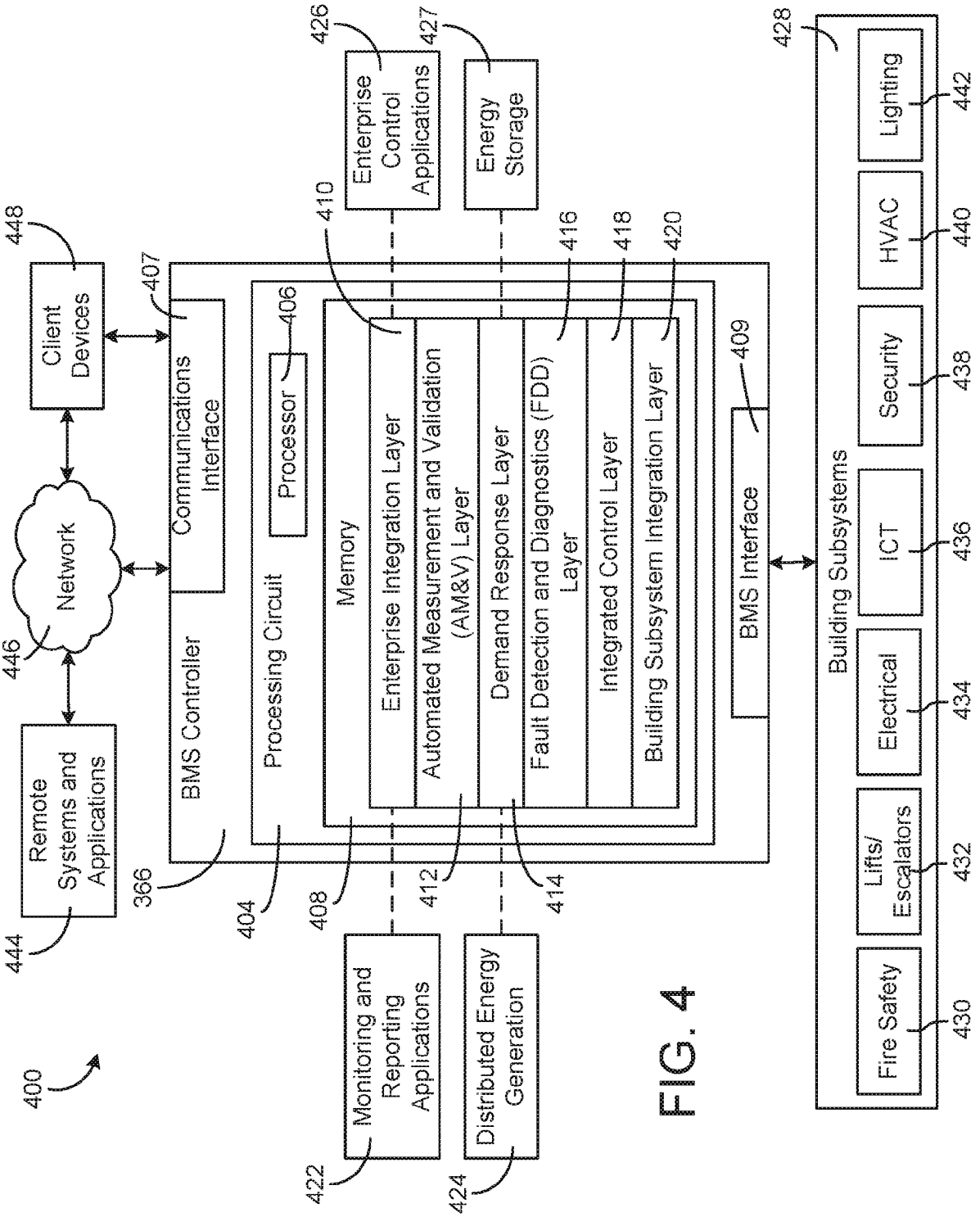

400
Remote Systems and Applications
444
Client Devices
448
Network
446
BMS Controller
366
Communications Interface
407
Processing Circuit
404
Processor
406
Memory
408
Enterprise Integration Layer
410
Automated Measurement and Validation (AM&V) Layer
412
Demand Response Layer
414
Fault Detection and Diagnostics (FDD) Layer
416
Integrated Control Layer
418
Building Subsystem Integration Layer
420
BMS Interface
409
Monitoring and Reporting Applications
422
Distributed Energy Generation
424
Enterprise Control Applications
426
Energy Storage
427
Building Subsystems
428
Fire Safety
430
Lifts/ Escalators
432
Electrical
434
ICT
436
Security
438
HVAC
440
Lighting
442

FIG. 4

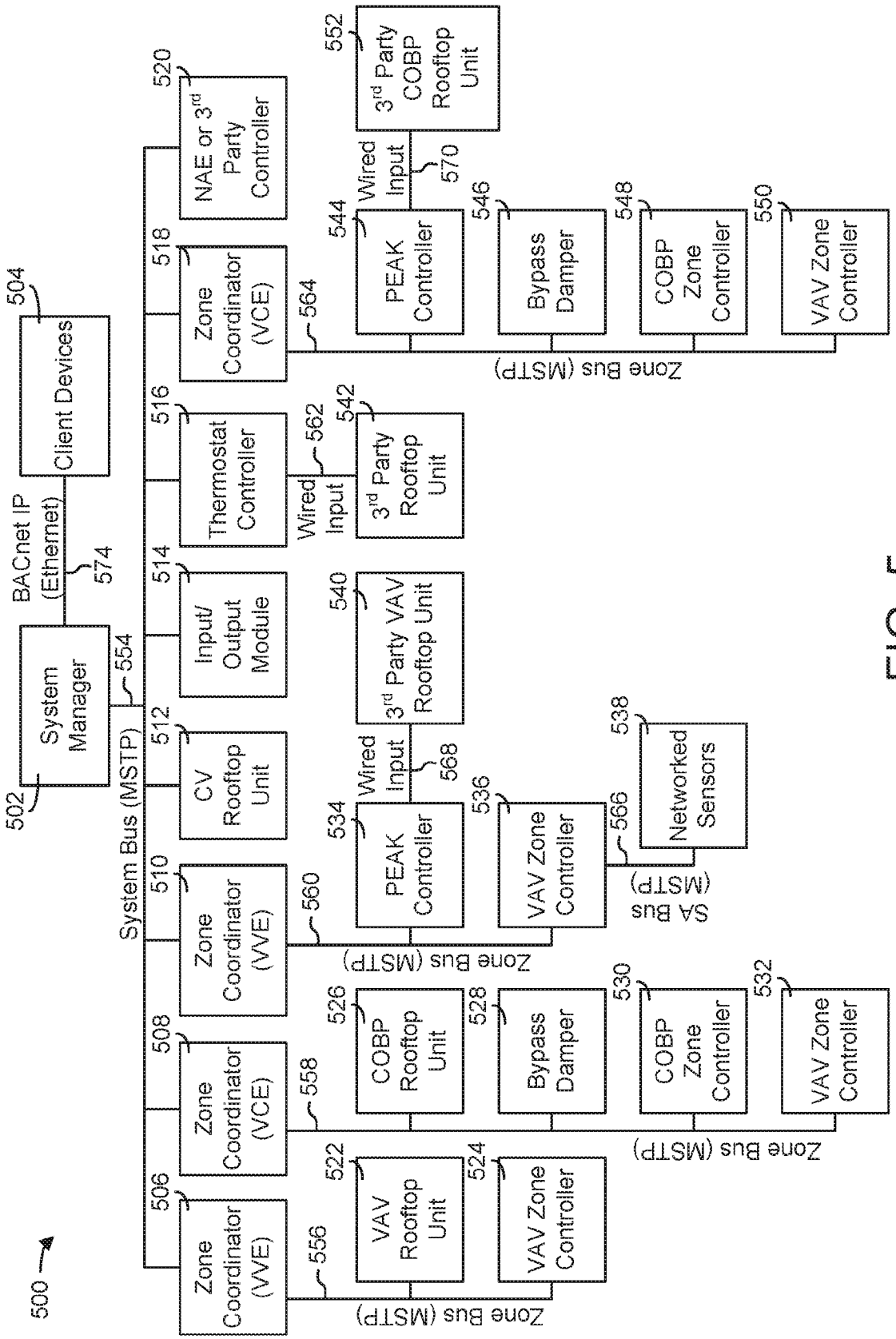

500
502
System Manager
BACnet IP (Ethernet)
574
504
Client Devices
554
System Bus (MSTP)
506
Zone Coordinator (VVE)
508
Zone Coordinator (VCE)
510
Zone Coordinator (VVE)
512
CV Rooftop Unit
514
Input/ Output Module
516
Thermostat Controller
518
Zone Coordinator (VCE)
520
NAE or 3rd Party Controller
522
VAV Rooftop Unit
524
VAV Zone Controller
526
COBP Rooftop Unit
528
Bypass Damper
530
COBP Zone Controller
532
VAV Zone Controller
534
PEAK Controller
536
VAV Zone Controller
538
Networked Sensors
540
3rd Party VAV Rooftop Unit
542
3rd Party Rooftop Unit
544
PEAK Controller
546
Bypass Damper
548
COBP Zone Controller
550
VAV Zone Controller
552
3rd Party COBP Rooftop Unit
556
Zone Bus (MSTP)
558
Zone Bus (MSTP)
560
Zone Bus (MSTP)
562
Wired Input
564
Zone Bus (MSTP)
566
SA Bus (MSTP)
568
Wired Input
570
Wired Input

FIG. 5

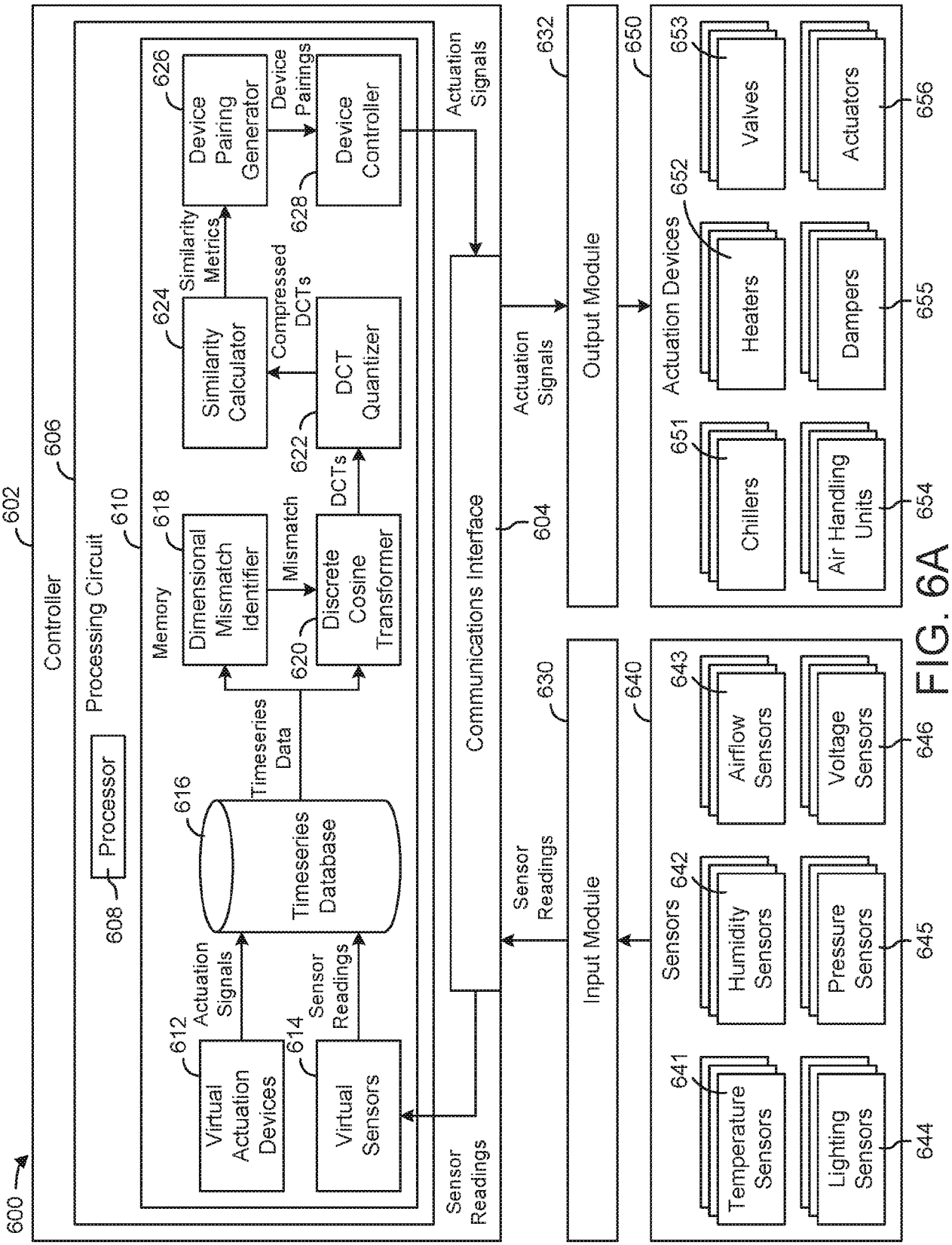

600
602
Controller
606
Processing Circuit
610
608
Processor
Memory
618
Dimensional Mismatch Identifier
Mismatch
620
Discrete Cosine Transformer
DCTs
622
DCT Quantizer
Compressed DCTs
624
Similarity Calculator
Similarity Metrics
626
Device Pairing Generator
Device Pairings
628
Device Controller
Actuation Signals
Timeseries Data
616
Timeseries Database
612
Actuation Signals
Virtual Actuation Devices
614
Sensor Readings
Virtual Sensors
Sensor Readings
Communications Interface
604
630
Input Module
640
Sensors
641
Temperature Sensors
642
Humidity Sensors
643
Airflow Sensors
644
Lighting Sensors
645
Pressure Sensors
646
Voltage Sensors
Actuation Signals
632
Output Module
650
Actuation Devices
651
Chillers
652
Heaters
653
Valves
654
Air Handling Units
655
Dampers
656
Actuators

FIG. 6A

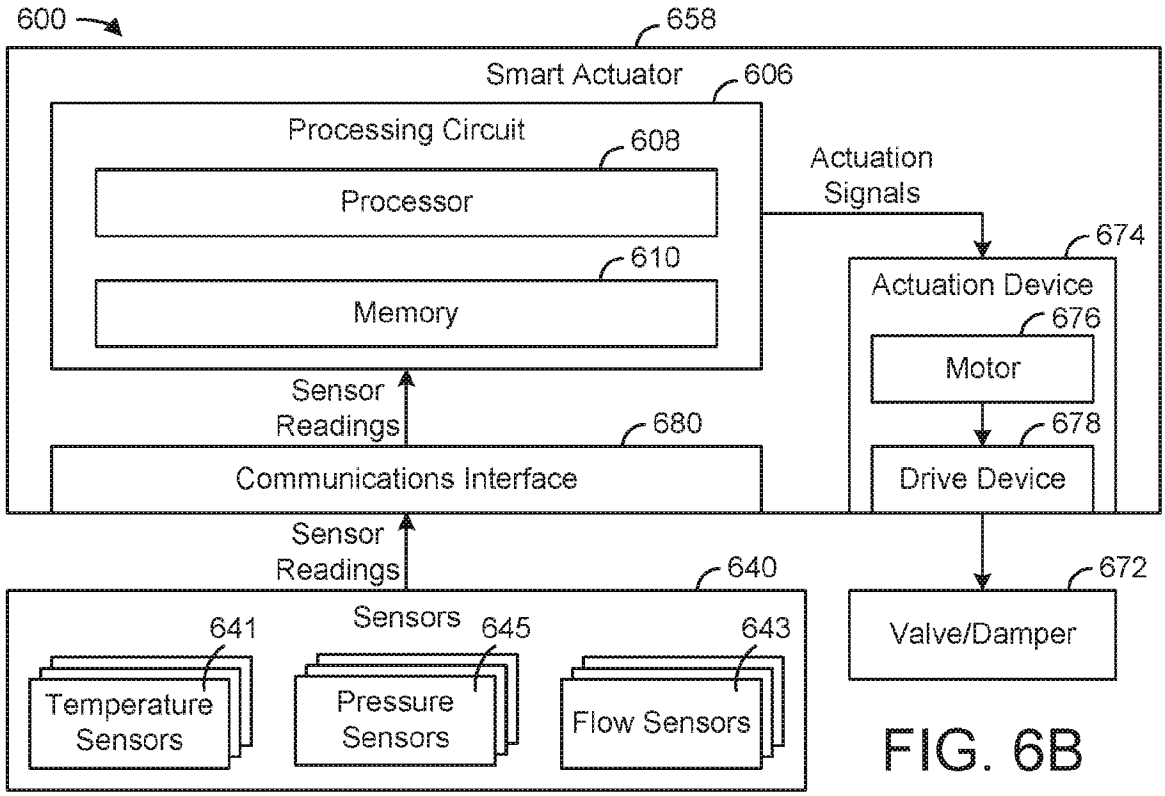

600
658
Smart Actuator
606
Processing Circuit
608
Processor
610
Memory
Actuation Signals
674
Actuation Device
676
Motor
678
Drive Device
Sensor Readings
680
Communications Interface
Sensor Readings
640
672
Valve/Damper
Sensors
641
645
643
Temperature Sensors
Pressure Sensors
Flow Sensors

FIG. 6B

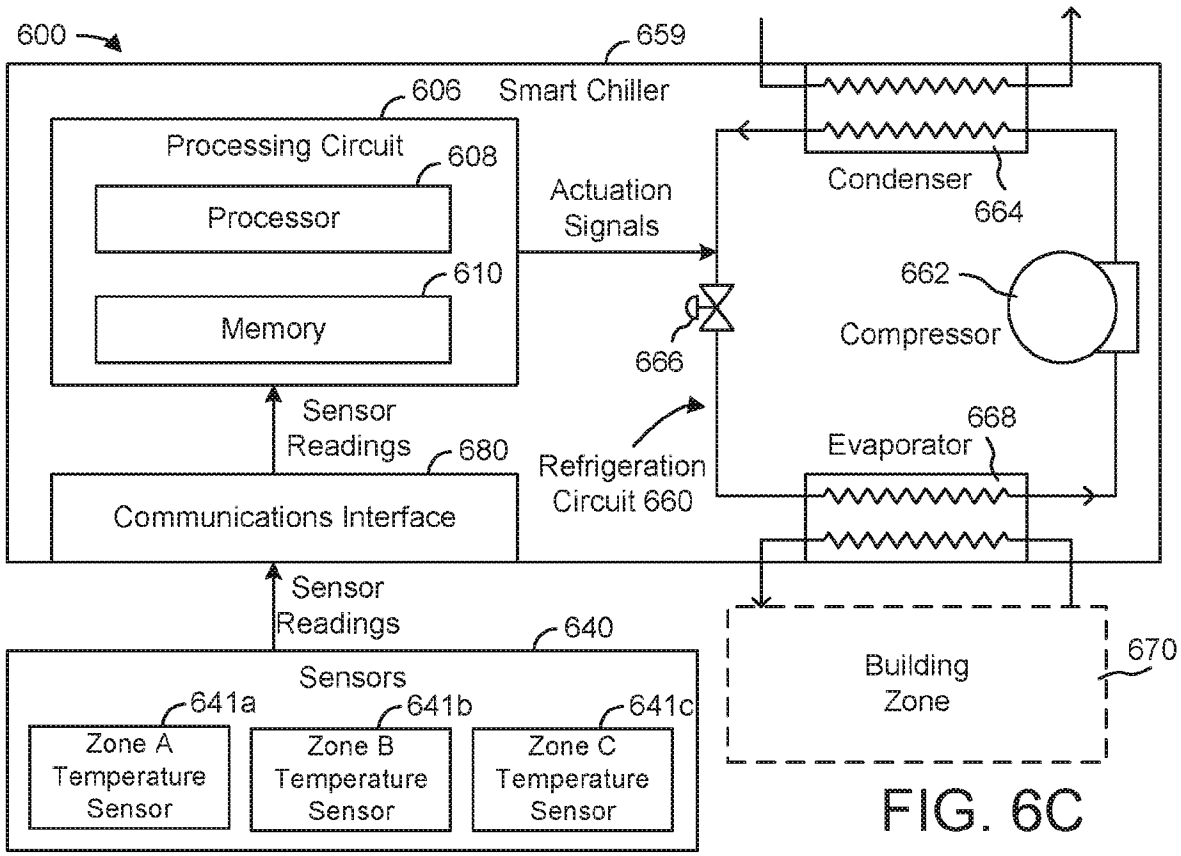

600
659
606
Smart Chiller
Processing Circuit
608
Processor
610
Memory
Actuation Signals
Condenser
664
662
Compressor
666
Sensor Readings
680
Refrigeration Circuit 660
668
Evaporator
Communications Interface
Sensor Readings
640
Sensors
641a
641b
641c
Zone A Temperature Sensor
Zone B Temperature Sensor
Zone C Temperature Sensor
Building Zone
670

FIG. 6C

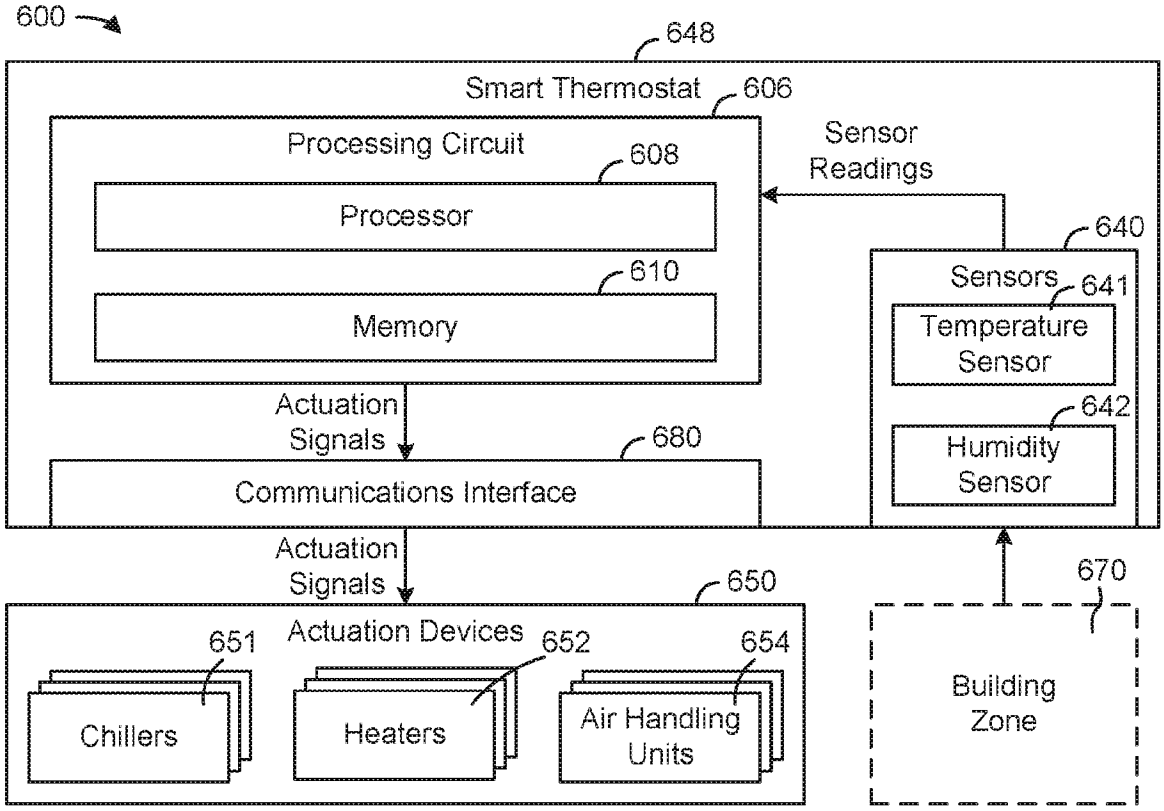

600
648
Smart Thermostat
606
Processing Circuit
608
Processor
610
Memory
Sensor Readings
640
Sensors
641
Temperature Sensor
642
Humidity Sensor
Actuation Signals
680
Communications Interface
Actuation Signals
650
Actuation Devices
651
Chillers
652
Heaters
654
Air Handling Units
670
Building Zone

FIG. 6D

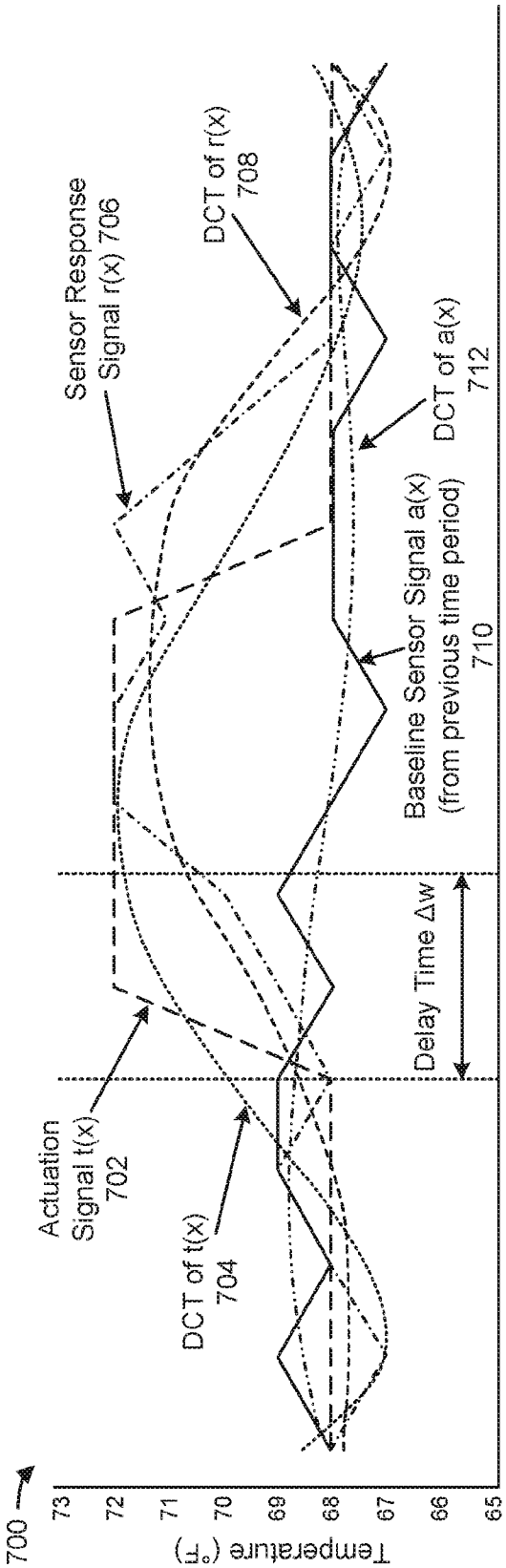

700
Temperature (°F)
73
72
71
70
69
68
67
66
65
Time
Actuation Signal t(x) 702
DCT of t(x) 704
Sensor Response Signal r(x) 706
DCT of r(x) 708
Delay Time Δw
Baseline Sensor Signal a(x) (from previous time period) 710
DCT of a(x) 712

FIG. 7

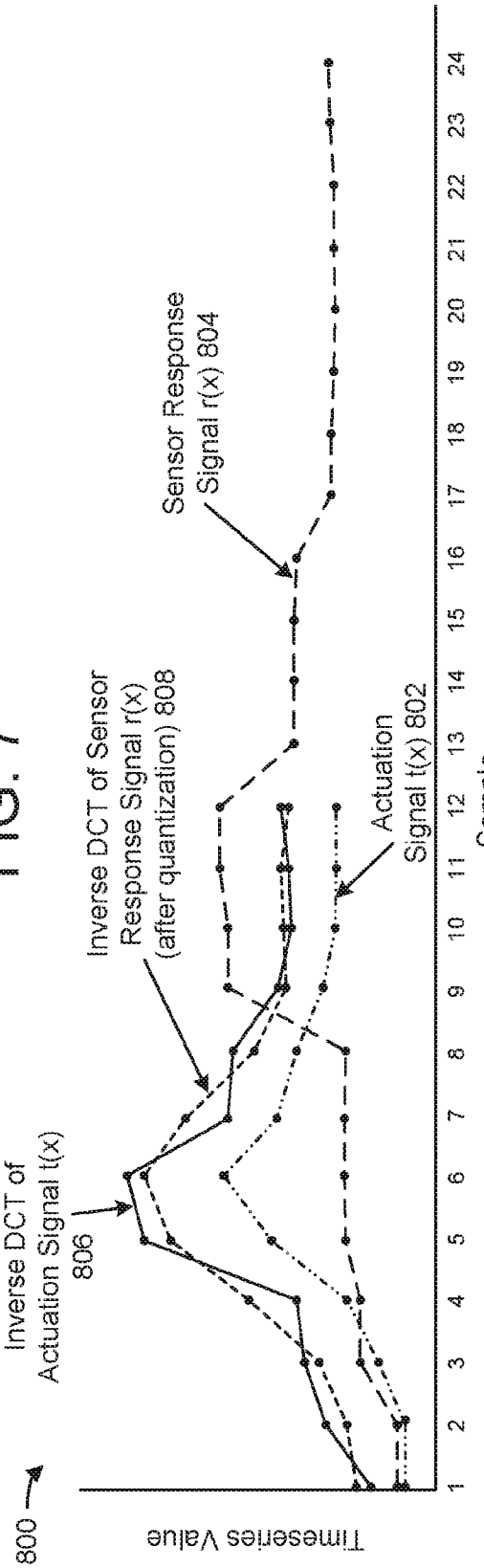

800
Timeseries Value
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Sample
Inverse DCT of Actuation Signal t(x) 806
Inverse DCT of Sensor Response Signal r(x) (after quantization) 808
Sensor Response Signal r(x) 804
Actuation Signal t(x) 802

FIG. 8

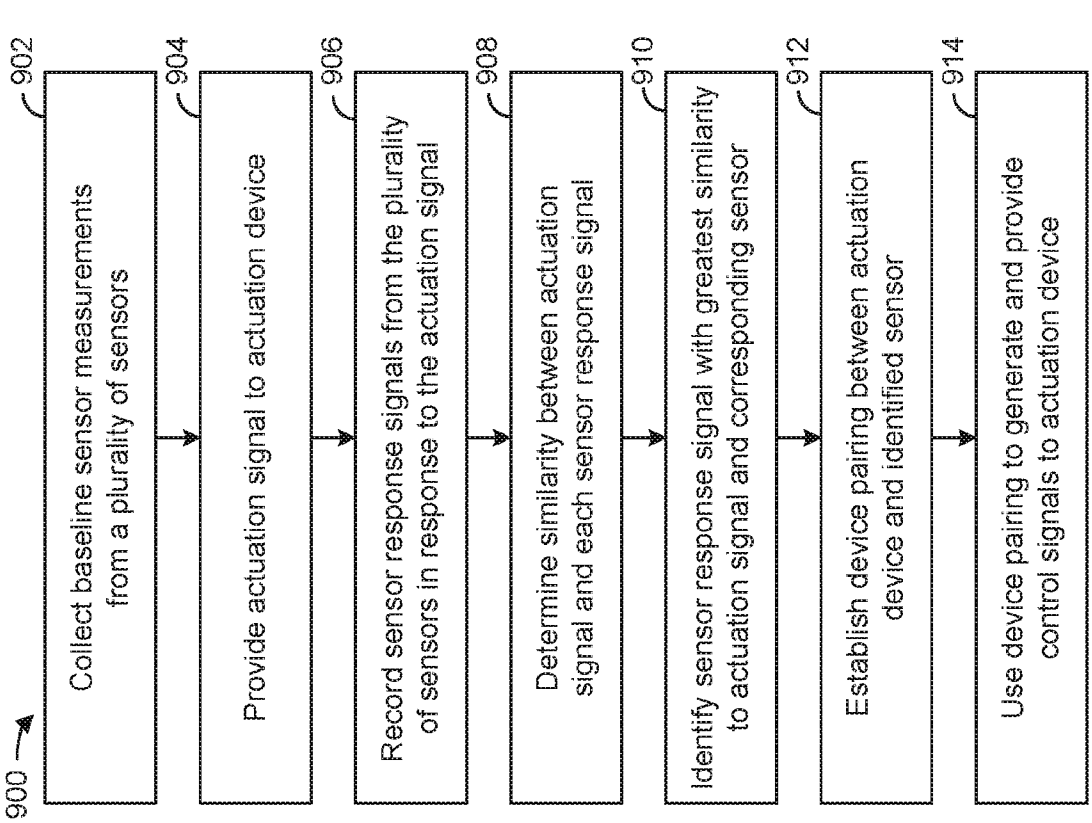

900
902
Collect baseline sensor measurements from a plurality of sensors
904
Provide actuation signal to actuation device
906
Record sensor response signals from the plurality of sensors in response to the actuation signal
908
Determine similarity between actuation signal and each sensor response signal
910
Identify sensor response signal with greatest similarity to actuation signal and corresponding sensor
912
Establish device pairing between actuation device and identified sensor
914
Use device pairing to generate and provide control signals to actuation device

FIG. 9

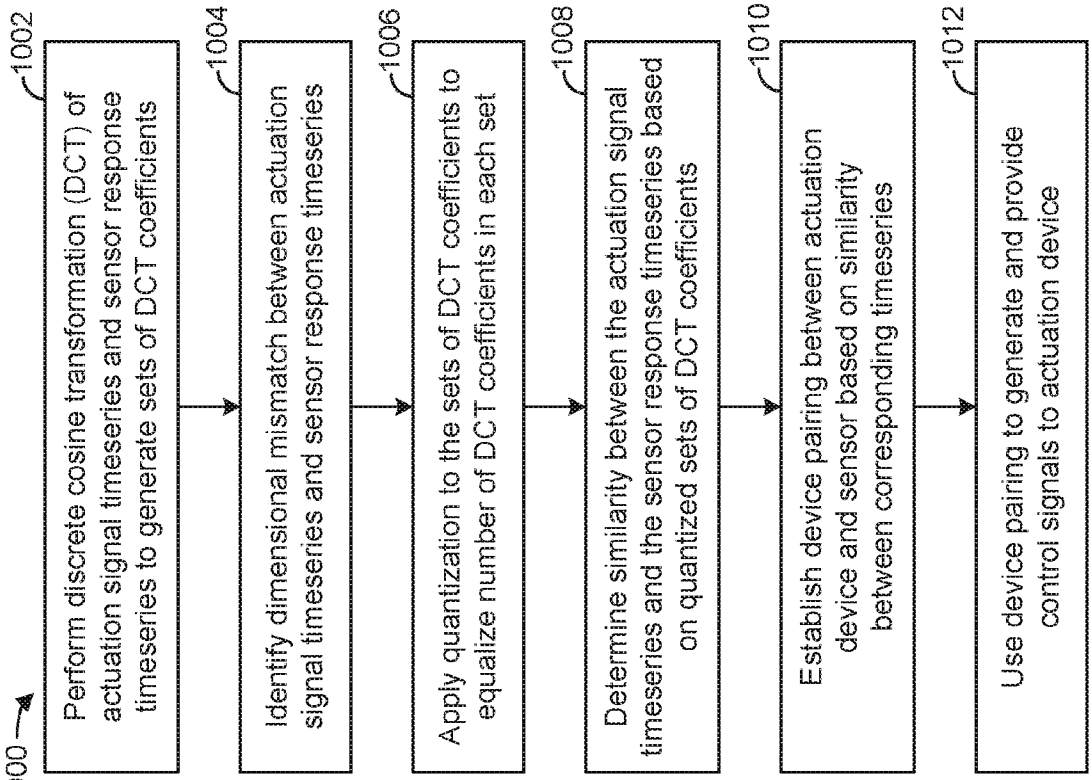

1000
1002
Perform discrete cosine transformation (DCT) of actuation signal timeseries and sensor response timeseries to generate sets of DCT coefficients
1004
Identify dimensional mismatch between actuation signal timeseries and sensor response timeseries
1006
Apply quantization to the sets of DCT coefficients to equalize number of DCT coefficients in each set
1008
Determine similarity between the actuation signal timeseries and the sensor response timeseries based on quantized sets of DCT coefficients
1010
Establish device pairing between actuation device and sensor based on similarity between corresponding timeseries
1012
Use device pairing to generate and provide control signals to actuation device

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/052702

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06Q 10/06; G06N 5/04; G06Q 50/08 (2017.01)
CPC - G06Q 10/06; G06N 5/04; G06Q 50/08 (2017.08)

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC - 700/86; 700/18; 700/87 (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 2008/0231437 A1 (SINGHAL et al) 25 September 2008 (25.09.2008) entire document	1-5, 11-15 --- 6-10, 16-20
Y	US 2014/0058567 A1 (NEST LABS, INC.) 27 February 2014 (27.02.2014) entire document	6, 7, 9, 10, 16, 17, 19, 20
Y	US 2007/0053513 A1 (HOFFBERG) 08 March 2007 (08.03.2007) entire document	8, 18
A	US 2006/0224252 A1 (BRINDAC et al) 05 October 2006 (05.10.2006) entire document	1-20
A	US 2015/0276237 A1 (GOOGLE INC) 01 October 2015 (01.10.2015) entire document	1-20
A	US 2013/0095868 A1 (JOHNSON CONTROLS TECHNOLOGY COMPANY) 18 April 2013 (18.04.2013) entire document	1-20
A	US 2012/0078423 A1 (ZHOU et al) 29 March 2012 (29.03.2012) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
06 November 2017	27 NOV 2017
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)