



(51) International Patent Classification:

G05D 1/02 (2006.01) G06Q 10/06 (2012.01)
G01C 21/00 (2006.01)

(21) International Application Number:

PCT/US2017/059648

(22) International Filing Date:

02 November 2017 (02.11.2017)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/423,077 16 November 2016 (16.11.2016) US

(71) Applicant: SYMBOL TECHNOLOGIES, LLC
[US/US]; 1 Zebra Plaza, Holtsville, New York 11742 (US).(72) Inventor: KINGSFORD, Thomas D.; 33 Crossovers
Street, Toronto, Ontario M4E 3X2 (CA).(74) Agent: KAPMAR, Dimitry et al.; Zebra Technologies
Corp., 3 Overlook Point, Lincolnshire, Illinois 60069 (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,
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 applicant's entitlement to apply for and be granted a
patent (Rule 4.17(ii))

Published:

— with international search report (Art. 21(3))
— before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments (Rule 48.2(h))

(54) Title: NAVIGATION CONTROL METHOD AND APPARATUS IN A MOBILE AUTOMATION SYSTEM

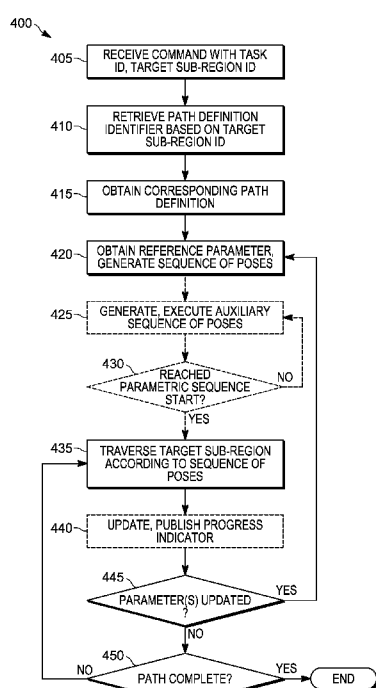


FIG. 4

(57) Abstract: A control method in a mobile automation apparatus includes: - storing a plurality of parametric path definition identifiers and respective corresponding sub-region identifiers identifying sub- regions in a facility; - receiving a task command containing a task identifier, and a target one of the sub-region identifiers identifying a target one of the sub- regions; - retrieving a selected one of the parametric path definition identifiers corresponding to the target sub-region identifier; obtaining a selected one of a plurality of parametric path definitions - including a shape parameter and a reference parameter name - corresponding to the selected parametric path definition identifier; - obtaining a reference parameter value, and generating a sequence of mobile automation apparatus poses according to the parametric path definition - controlling a locomotive assembly of the mobile automation apparatus to traverse the target sub-region according to the sequence of poses.

NAVIGATION CONTROL METHOD AND APPARATUS IN
A MOBILE AUTOMATION SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from U.S. Provisional Patent Application No. 62/423,077, filed November 16, 2016, which is incorporated herein by reference in its entirety.

BACKGROUND

[0001] Environments in which objects are managed, such as retail facilities, may be complex and fluid. For example, a retail facility may include objects such as products for purchase, a distribution environment may include objects such as parcels or pallets, a manufacturing environment may include objects such as components or assemblies, a healthcare environment may include objects such as medications or medical devices.

[0002] A mobile apparatus may be employed to perform tasks within the environment, such as capturing data for use in identifying products that are out of stock, incorrectly located, and the like. The travel of the mobile apparatus through the environment to perform such data capture operations is subject to various requirements, which may conflict with each other. For example, the above-mentioned environments may be complex and contain various obstacles, including customers, requiring the mobile apparatus to repeatedly generate updated paths. Conventional cost-based path generation mechanisms are computationally costly, and may therefore require the mobile apparatus to interrupt the data capture task to await generation of a new path when an obstacle is detected.

[0003] Further, a data capture operation may require the mobile apparatus to implement certain constraints throughout the operation, such as remaining within a certain distance of a structure in the environment. The mobile apparatus' perception of such constraints may also vary during execution of the operation, necessitating updating or regeneration of the path the mobile apparatus will travel. Conventional path generation mechanisms are poorly suited to navigation under such dynamic conditions.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0004] The accompanying figures, where like reference numerals refer to identical or functionally similar elements throughout the separate views, together with the detailed description below, are incorporated in and form part of the specification, and serve to further illustrate embodiments of concepts that include the claimed invention, and explain various principles and advantages of those embodiments.

[0005] FIG. 1 is a schematic of a mobile automation system.

[0006] FIG. 2A depicts a mobile automation apparatus in the system of FIG. 1.

[0007] FIG. 2B is a block diagram of certain internal hardware components of the mobile automation apparatus in the system of FIG. 1.

[0008] FIG. 3 is a block diagram of certain internal components of the mobile automation apparatus of the system of FIG. 1.

[0009] FIG. 4 is a flowchart of a control method at the server of the system of FIG. 1.

[0010] FIG. 5 depicts example parametric paths generated during the performance of the method of FIG. 4.

[0011] FIG. 6 depicts an example performance of the method of FIG. 4.

[0012] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been drawn to scale. For example, the dimensions of some of the elements in the figures may be exaggerated relative to other elements to help to improve understanding of embodiments of the present invention.

[0013] The apparatus and method components have been represented where appropriate by conventional symbols in the drawings, showing only those specific details that are pertinent to understanding the embodiments of the present invention so as not to obscure the disclosure with details that will be readily apparent to those of ordinary skill in the art having the benefit of the description herein.

DETAILED DESCRIPTION

[0014] Examples disclosed herein are directed to a control method in a mobile automation apparatus including: storing a plurality of parametric path definition identifiers and respective corresponding sub-region identifiers identifying sub-regions in a facility; receiving a task command containing (i) a task identifier, and (ii) a target

one of the sub-region identifiers identifying a target one of the sub-regions; retrieving a selected one of the parametric path definition identifiers corresponding to the target sub-region identifier; obtaining a selected one of a plurality of parametric path definitions – including a shape parameter and a reference parameter name – corresponding to the selected parametric path definition identifier; obtaining a reference parameter value, and generating a sequence of mobile automation apparatus poses according to the parametric path definition; each pose corresponding to a location in the facility; and controlling a locomotive assembly of the mobile automation apparatus to traverse the target sub-region according to the sequence of poses.

[0015] Further examples disclosed herein are directed to a computing device for controlling in a mobile automation apparatus, comprising: a memory storing a plurality of parametric path definition identifiers and respective corresponding sub-region identifiers identifying sub-regions in a facility; a task manager configured to receive a task command containing (i) a task identifier, and (ii) a target one of the sub-region identifiers identifying a target one of the sub-regions; a navigation controller configured to retrieve, from the memory, a selected one of the parametric path definition identifiers that corresponds to the target sub-region identifier; a parametric path generator configured to obtain a selected one of a plurality of parametric path definitions that corresponds to the selected parametric path definition identifier; the selected parametric path definition including a shape parameter and a reference parameter name; the parametric path generator further configured to obtain a reference parameter value corresponding to the reference parameter name, and generate a sequence of mobile automation apparatus poses according to the parametric path definition; each pose corresponding to a location in the facility; and a local planner configured to control a locomotive assembly of the mobile automation apparatus to traverse the target sub-region according to the sequence of poses.

[0016] FIG. 1 depicts a mobile automation system 100 in accordance with the teachings of this disclosure. The system 100 includes a control server 101 (also referred to herein simply as the server 101), in communication with at least one mobile automation apparatus 103 (also referred to herein simply as the apparatus 103) and at least one client computing device 105 via communication links 107, illustrated in the present example as including wireless links. In the present example, the links 107 are provided

by a wireless local area network (WLAN) deployed within the retail environment by one or more access points (not shown). In other examples, the server 101, the client device 105, or both, are located outside the retail environment, and the links 107 therefore include wide-area networks (WANs) such as the Internet, mobile networks, and the like. The system 100 also includes a dock 108 for the apparatus 103. The dock 108 is in communication with the server 101 via a link 109 that in the present example is a wired link. In other examples, however, the link 109 is a wireless link. The mobile automation apparatus 103, when coupled to the dock 108, may establish communications with the dock 108 to communicate with the server 101 via the link 109 instead of, or in addition to, the link 107. The dock 108 may also supply power (e.g. electrical power) to the apparatus 103.

[0017] The client computing device 105 is illustrated in FIG. 1 as a mobile computing device, such as a tablet, smart phone or the like. In other examples, the client device 105 includes computing devices such as a desktop computer, a laptop computer, another server, a kiosk, a monitor, or other suitable device. The system 100 can include a plurality of client devices 105, each in communication with the server 101 via respective links 107.

[0018] The system 100 is deployed, in the illustrated example, in a retail environment including a plurality of shelf modules 110-1, 110-2, 110-3 and so on (collectively referred to as shelves 110, and generically referred to as a shelf 110 – this nomenclature is also employed for other elements discussed herein). Each shelf module 110 supports a plurality of products 112. Each shelf module 110 includes a shelf back 116-1, 116-2, 116-3 and a support surface (e.g. support surface 117-3 as illustrated in FIG. 1) extending from the shelf back 116 to a shelf edge 118-1, 118-2, 118-3. The shelf modules 110 may also, in some examples, include other support structures, such as pegs, hangers and the like. The shelf modules 110 are typically arranged in a plurality of aisles, each of which includes a plurality of modules aligned end-to-end. In such arrangements, the shelf edges 118 face into the aisles, through which customers in the retail environment, as well as the apparatus 103, may travel. The aisles occupy regions of the retail environment, and the shelf modules 110 occupy sub-regions within the above-mentioned regions. The aisles and shelf modules 110 are therefore also referred to herein as regions and sub-regions of the facility (e.g. the above-mentioned retail

environment), respectively. Other structures within the environment in which the system 100 is deployed may also occupy such regions and sub-regions instead of the aisles of shelf modules 110.

[0019] The apparatus 103 is deployed within the retail environment, and communicates with the server 101 (e.g., via the link 107) to navigate, autonomously or partially autonomously, the length 119 of at least a portion of the shelves 110. The apparatus 103 is equipped with a plurality of navigation and data capture sensors 104, such as image sensors (e.g. one or more digital cameras) and depth sensors (e.g. one or more Light Detection and Ranging (LIDAR) sensors, one or more depth cameras employing structured light patterns, such as infrared light, and the like), and is further configured to employ the sensors to capture shelf data. In the present example, the apparatus 103 is configured to receive instructions from the server 101 to perform one or more tasks, such as a data capture task, respective to one or more shelf modules 110. The apparatus 103 is configured, responsive to such instructions, to navigate among the relevant shelf modules 110 and perform the specified task, for example by capturing images of the shelf modules 110 for delivery to the server 101.

[0020] The server 101 includes a special purpose controller, such as a processor 120, specifically designed to control the mobile automation apparatus 103 to perform tasks respective to the shelf modules 110 (such as the above-mentioned capture of images). To that end, the server 101 stores, in a memory 122 interconnected with the processor 120, identifiers of regions and sub-regions (e.g. identifiers of shelf modules 110), and may also store operational constraints associated with the shelf modules 110. The processor 120 is configured to provide instructions to the apparatus 103, via a communications interface 124, to perform tasks such as the above-mentioned data capture, and to receive the captured data or other results of the tasks for storage in a repository 132 in the memory 122. The server 101 is also, in some examples, configured to perform various post-processing activities on captured data, for example to determine product status data (e.g. out of stock or low stock products) and to transmit status notifications to the mobile device 105 responsive to the determination of product status data.

[0021] The processor 120 and the memory 122 each comprise one or more integrated circuits. In an embodiment, the processor 120 includes one or more central processing units (CPUs) and/or graphics processing units (GPUs). The memory 122 is a non-transitory computer readable storage medium having stored thereon computer readable instructions for executing control of the apparatus 103, as discussed in further detail below. The memory 122 includes a combination of volatile (e.g. Random Access Memory or RAM) and non-volatile memory (e.g. read only memory or ROM, Electrically Erasable Programmable Read Only Memory or EEPROM, flash memory). In an embodiment, a specially designed integrated circuit, such as a Field Programmable Gate Array (FPGA), is designed to perform the functionality discussed herein, either alternatively or in addition to the processor 120 and memory 122. As those of skill in the art will realize, the client device 105 also includes one or more controllers or processors and/or FPGAs, in communication with the processor 120, specifically configured to process (e.g. to display) notifications received from the server 101.

[0022] The server 101 also includes the above-mentioned communications interface 124 interconnected with the processor 120. The communications interface 124 includes suitable hardware (e.g. transmitters, receivers, network interface controllers and the like) allowing the server 101 to communicate with other computing devices – particularly the apparatus 103, the client device 105 and the dock 108 – via the links 107 and 109. The links 107 and 109 may be direct links, or links that traverse one or more networks, including both local and wide-area networks. The specific components of the communications interface 124 are selected based on the type of network or other links that the server 101 is required to communicate over. In the present example, as noted earlier, a wireless local-area network is implemented within the retail environment via the deployment of one or more wireless access points. The links 107 therefore include either or both wireless links between the apparatus 103 and the mobile device 105 and the above-mentioned access points, and a wired link (e.g. an Ethernet-based link) between the server 101 and the access point.

[0023] The memory 122 stores a plurality of applications, each including a plurality of computer readable instructions executable by the processor 120. The applications stored in the memory 122 include a control application 128, which may also be implemented

as a suite of logically distinct applications, and whose execution by the processor 120 configures the processor 120 to implement various functionality. As will now be apparent, some or all of the functionality implemented by the processor 120 may also be performed by preconfigured hardware elements (e.g. one or more Application-Specific Integrated Circuits (ASICs)) rather than by execution of the control application 128 by the processor 120. In the present example, in particular, the server 101 is configured via the execution of the control application 128 by the processor 120, to issue instructions (also referred to herein as commands) to the apparatus 103 to cause the apparatus 103 to perform one or more specified tasks in the retail environment.

[0024] Turning now to FIGS. 2A and 2B, the apparatus 103 will be described in greater detail. The apparatus 103 includes a chassis 200 containing a locomotive assembly 202 (e.g. one or more electric motors driving wheels, tracks or the like). In one embodiment, a special-purpose controller, such as a processor 220 (which may also be, in some examples, an ASIC and/or FPGA) configured to control navigational and/or data capture aspects of the apparatus 103 is also housed within the chassis 200. The controller 220 may control the locomotive assembly 202 to move the apparatus 103, for example via the execution of computer-readable instructions such as a task execution application 228 stored in a memory 222 interconnected with the processor 220. The memory 222 includes a combination of volatile (e.g. Random Access Memory or RAM) and non-volatile memory (e.g. read only memory or ROM, Electrically Erasable Programmable Read Only Memory or EEPROM, flash memory). The processor 220 and the memory 222 each comprise one or more integrated circuits. The application 228 may also be implemented as a suite of distinct applications, and such applications may be implemented as a combination of computer-readable instructions and/or preconfigured hardware elements such as ASICs and FPGAs.

[0025] As will be described below, the execution of the application 228 by the controller 220 configures the apparatus 103 to receive instructions from the server 101 and, responsive to the instructions, perform tasks (such as the above-mentioned data capture) respective to shelf modules 110 identified in the instructions. In particular, to perform the above-mentioned tasks, the processor 220 is configured to generate and update navigational data, such as a path along which the apparatus 103 travels to perform the required task.

[0026] The memory 222 may also store a repository 232 containing, for example, a map of the environment in which the apparatus 103 operates, for use during the execution of the application 228. The repository 232 may also contain one or more of path definition identifiers and path definitions corresponding to the path definition identifiers, for use during the execution of the application 228 to generate path data. The apparatus 103 may communicate with the server 101, for example to receive the above-mentioned instructions, via a communications interface 224 over the link 107 shown in FIG. 1. The communications interface 224 also enables the apparatus 103 to communicate with the server 101 via the dock 108 and the link 109.

[0027] The apparatus 103 further includes a sensor mast 204 supported on the chassis 200 and, in the present example, extending upwards from the chassis 200. The mast 204 supports the sensors 104 mentioned earlier. In particular, the sensors 104 include at least one imaging sensor 208 such as a digital camera, as well as at least one depth-sensing sensor 212, such as a 3D digital camera. The apparatus 103 also includes additional depth sensors, such as LIDAR sensors 216, in the present example. The sensors 208, 212, 216 are oriented on the mast 204 such that the fields of view of each sensor face a shelf 110 along the length 119 of which the apparatus 103 is travelling. In other examples, the apparatus 103 includes additional sensors, such as one or more RFID readers, temperature sensors, and the like.

[0028] In the present example, the mast 204 supports seven digital cameras 208-1 through 208-7, and two LIDAR sensors 216-1 and 216-2. The mast 204 also supports a plurality of illumination assemblies 218, configured to illuminate the fields of view of the respective cameras 208. That is, the illumination assembly 218-1 illuminates the field of view of the camera 208-1, and so on. The sensors 208 and 216 are oriented on the mast 204 such that the fields of view of each sensor face a shelf 110 along the length 119 of which the apparatus 103 is travelling. The apparatus 103 is configured to track a location of the apparatus 103 (e.g. a location of the center of the chassis 200) in a frame of reference previously established in the retail facility.

[0029] Turning now to FIG. 3, before describing the operation of the application 228 to execute (e.g., at the apparatus 103) task instructions, and particularly to generate navigation data for the execution of task instructions, certain components of the

application 228 will be described in greater detail. As will be apparent to those skilled in the art, in other examples the components of the application 228 may be separated into distinct applications, or combined into other sets of components. Some or all of the components illustrated in FIG. 3 may also be implemented as dedicated hardware components, such as one or more ASICs or FPGAs.

[0030] The control application 228 includes a task manager 300 configured to receive instructions from the server 101 to perform tasks in the retail facility. The instructions received from the server typically include a task identifier (e.g., identifying the task as a data capture task), an identifier of a target sub-region in the retail facility (e.g., one or more aisle identifiers, or one or more module identifiers). The task manager 300 is configured to control the other components of the application 228 in executing the task. For example, the task manager 300 is configured in an embodiment to pass the sub-region identifier from the instruction to a navigator 304 of the application 228.

[0031] The navigator 304 is configured to generate paths through the retail facility traversing the shelf modules 110 identified in the instructions received from the server 101 (via the task manager 300), and to control the locomotive assembly 202 to carry the apparatus 103 along the above-mentioned paths. As will be discussed in greater detail below, under certain conditions the navigator 304 is configured to generate the above paths independently of the previously-mentioned facility map stored in the repository 232, through the use of parametric path definitions.

[0032] Specifically, the navigator 304 includes a navigation controller 308, a parametric path generator 312, an auxiliary path generator 316, and a local planner 320. The local planner 320 is configured to receive a sequence of poses from the navigation controller 308, and to control the locomotive assembly 202 to carry the apparatus 103 along the sequence of poses, based in part on the above-mentioned map stored in the repository 232 (e.g. to verify that the sequence of poses is not intersected by any obstacles). The sequence of poses, in other words, define a path along which the apparatus 103 travels. Each pose is represented as at a set of coordinates in the frame of reference defined for the facility, according to which the apparatus 103 tracks its location and defines the map in the repository 232.

[0033] The parametric path generator 312 and the auxiliary path generator 316 are each configured to generate sequences of poses responsive to instructions from the navigation controller 308. The navigation controller 308 selects which sequence of poses (i.e. that from the parametric path generator 312 or that from the auxiliary path generator 316) to transmit to the local planner 320 for execution. Of particular note, the parametric path generator 312 is configured to generate each sequence of poses not by implementing conventional cost-based path generation techniques based on the environmental map in the repository 232, but rather by identifying, retrieving and processing a parametric path definition, as will be described below in greater detail.

[0034] The other components of the application 228 also include a data capture controller 324 configured to control the sensors 104 (e.g. the image sensors 208 and the depth sensors 212) to capture data such as images of the shelves 110 while the apparatus 103 travels along the paths generated by the navigator 304. The application 228 also includes a system monitor 328 configured to monitor the status of the apparatus 103. For example, the system monitor 328 can be configured to monitor a remaining energy capacity (e.g. fuel, battery level or the like) of the apparatus 103, a safety sensor (e.g. a safety lidar sensor configured to detect obstacles in the path of the apparatus 103), or the like. The system monitor 328 is configured to generate and provide alerts to the task manager 300 under certain conditions to interrupt task execution.

[0035] The functionality of the application 228 will now be described in greater detail, with reference to the components illustrated in FIG. 3. Turning first to FIG. 4, a method 400 of generating and executing navigational data for the apparatus 103 is shown. The method 400 will be described in conjunction with its performance by the apparatus 103 as described above.

[0036] At block 405, the apparatus 103 is configured to receive a task command including at least a task identifier and an identifier of at least one target sub-region of the facility. The task command may be received, for example, from the server 101 (which may, in turn, have received a request from the client device 105 to transmit the task command) via the link 107. The task command is received at the processor 220 via the communications interface 124. More specifically, the task manager 300 is

configured to receive the task command via the execution of the application 228 by the processor 220.

[0037] The task identifier is selected at the server 101, and indicates a task to be performed by the apparatus 103 in the facility. In the present example, the task identifier is assumed to identify a data capture task, indicating that the apparatus 103 is to traverse the target sub-regions identified in the task command and to capture data (e.g. images, depth measurements and the like) during the traverse. In other performances of the method 400, the task may be, for example, a guidance task, e.g. to guide a patron in the facility to a particular location. In embodiments in which only a single task is implemented, the task identifier may be omitted from the task command.

[0038] The target sub-region identifier(s) included in the task command identify one or more sub-regions of the facility. In the present example, therefore, each target sub-region identifier identifies a shelf module 110 in the facility. The task command received at block 405 can identify the target sub-region(s) explicitly (that is, by individual sub-region identifiers), or implicitly via a region identifier, such as the identifier of an aisle or department (containing a plurality of aisles) in the facility.

[0039] Following receipt of the task command at block 405, the task manager is configured to provide the target sub-region identifier(s) to the navigator 304 for further processing. In some embodiments, the task command may also include imaging constraints (e.g., a number of frames per second at which to capture images). The task manager 300 may be configured to pass such parameters to the data capture controller 324; however, the handling and execution of data capture constraints is not central to the discussion below.

[0040] At block 410, the navigator 304, and particularly the navigation controller 308, is configured to receive the target sub-region identifier(s) from the task manager 300, and to retrieve one or more parametric path definition identifiers from the memory 222. To that end, the apparatus 103 stores, for example in the repository 232, a plurality of sub-region identifiers corresponding to each of the sub-regions in the facility (i.e. each of the modules 110 in the facility, in the present embodiment). For each sub-region identifier, the apparatus 103 also stores a corresponding parametric path definition

identifier. Table 1 below illustrates the sub-region identifiers and corresponding parametric path definition identifiers stored in the memory 222.

Table 1: Sub-Region and Parametric Path Definition Identifiers

Department	Aisle(s)	Module(s)	Parametric Path Definition ID
Grocery	1	1-A – 1-G	PPD1
	2	2-A – 2-G	PPD2
...	...	...	...

[0041] As seen above, the apparatus 103 may also store a region identifier corresponding to one or more sub-region identifiers, such as the “aisle” identifiers noted above. In some embodiments, the apparatus 103 also stores identifiers corresponding to supersets of regions, such as the “department” identifier above. Thus, as noted above, the task command received at block 405 can include sub-region identifiers, or in other embodiments can include an aisle or department identifier, from which the navigation controller 308 can retrieve the corresponding sub-region identifiers. In further embodiments, the task manager 300, rather than the navigation controller, performs any required translation from region to sub-region identifiers based on records such as those shown above, and the navigation controller 308 therefore receives only target sub-region identifiers.

[0042] As seen in Table 1, each parametric path definition identifier corresponds to a group of sub-regions. Thus, for example, the parametric path definition identifier corresponds to the module identifiers 1-A, 1-B, 1-C, 1-D, 1-E, 1-F and 1-G. In other examples, a different parametric path definition identifier may be stored in association with each individual sub-region identifier. Typically, a given parametric path definition identifier is stored in association with a contiguous set of sub-regions (that is, a set of modules 110 disposed end-to-end, with minimal or no space between the modules 110). In the examples discussed herein, the modules 1-A to 1-G are assumed to be contiguous, and the modules 2-A to 2-F are also assumed to be contiguous, but separate from the modules 1-A to 1-G.

[0043] Although Table 1 includes one parametric path definition identifier per group of sub-regions, in other examples, the apparatus 103 can store a plurality of parametric path definition identifiers for each group of sub-regions. For example, in another

embodiment Table 1 can list distinct parametric path definition identifiers for each of a plurality of tasks for a given sub-region or group of sub-regions.

[0044] At block 410, therefore, the navigation controller 308 is configured to retrieve a selected one (or more, depending on which sub-regions were identified in the task command) of the parametric path definition identifiers from Table 1 corresponding to the target sub-regions identified in the task command. In the present example performance of method 400, it is assumed that the task command identified the modules 1-A to 1-G, and at block 410 the navigation controller 308 is therefore configured to retrieve (e.g., from Table 1) the parametric path definition identifier “PPD1”. Having retrieved the parametric path definition identifier, the navigation controller 308 is configured to pass the selected parametric path definition identifier to the parametric path generator 312.

[0045] At block 415, the parametric path generator 312 is configured to obtain a selected one of a plurality of parametric path definitions that corresponds to the selected parametric path definition identifier. In the present embodiment, the parametric path generator 312 is configured to obtain the parametric path definition by retrieving the parametric path definition from the memory 222 (e.g., from the repository 232). The memory 222 therefore contains a plurality of parametric path definitions, each with a corresponding parametric path definition identifier.

[0046] In other embodiments, the parametric path definition and any or all of the associated data shown above (i.e. the initial values, constraints and progress indicator definition) are stored not in the memory 222, but in a storage location external to the apparatus 103. For example, in some embodiments the parametric path definitions are stored at the server 101 (e.g., in the repository 132). In such embodiments, at block 415 the parametric path generator 312 is configured to obtain the parametric path definition by transmitting a request (e.g., via the navigation controller 304, the task manager 300 and the communications interface 224) to the server 101 including the parametric path definition identifier retrieved at block 410.

[0047] Table 2, below, illustrates an example parametric path definition and associated data, as stored in the memory 222.

Table 2: Parametric Path Definitions

Path Def. ID	Path Definition	Config. Values	Boundary Constraints	Progress Indicator
PPD1	$X(\lambda) = 50 - [\text{Dst}] + [\text{Dadj}]$ $Y(\lambda) = 20\lambda + 10$ $\text{Dadj} = \text{PID}(\text{Dm}, \text{Dst})$	$\lambda_1=0; \lambda_N=1$ $N = 8$ $\text{Dadj} = 0$	$\text{Dm-max} = 70$ $\text{Dm-min} = 30$	$P = \lambda \times 100$
...	...	...	...	...

[0048] As seen in Table 2, for each parametric path definition identifier, the memory 222 stores a parametric path definition. The parametric path definition, in general, defines a path within the facility, independently of the environmental map stored in the repository 132. The parametric path definition typically does not, however, define the path in absolute terms (i.e. as a path at a fixed location within the facility frame of reference). Rather, the parametric path definition includes one or more shape parameters defining the general shape of the path and, optionally, placing certain constraints on the location of the path. The parametric path definition also includes at least one reference parameter name. As will be discussed below, the reference parameter name permits the insertion of a reference parameter value that can be altered between performances of the method 400 or even during a given performance of the method 400, in order to alter the resulting path without requiring a different parametric path definition. The reference parameter(s) and the shape parameters may collectively be referred to as local parameters.

[0049] In the example of Table 2, the parametric path definition defines each of a plurality of poses, or locations, along the path in two dimensions (X and Y) in the facility frame of reference. The number and spacing of the poses along the path are defined by an independent parameter, λ . In the above example, each pose is defined in part by a first reference parameter named “Dst”, and a second reference parameter named “Dadj”. The first reference parameter, in the present example, corresponds to a target distance to the relevant shelf (that is, the shelf of the modules 1-A to 1-G). The second reference parameter named “Dadj”, in the present example, is employed to adjust the position of the path during execution of the path based on an error between the target distance to shelf Dst and a measured distance to shelf Dm (e.g., obtained via

the lidar sensors 216). In other words, the reference parameters define operational constraints relative to an object in the facility (in particular, a shelf module). The assignment of values to the reference parameters will be discussed in greater detail below.

[0050] Each pose is further defined by one or more shape parameters, for example in the form of exponents applied to the X and Y dimensions of the path. In the present example, all exponents in the path definition have values of one (and are therefore not explicitly shown in Table 2), indicating that the path forms a straight line (e.g., rather than a curve). The shape parameters also include, in the present example, parameters placing constraints on the location and length of the path. The parameters of the parametric path definition will be described below in greater detail in connection with an illustration of a resulting path.

[0051] As seen in Table 2, the memory 222 also stores, in conjunction with the parametric path definition, one or more configuration values. The configuration values include an initial value λ_1 for the independent parameter λ mentioned above, as well as a final value λ_N for the independent parameter, and a number N of steps through which the independent parameter is incremented from the initial value to the final value in order to generate the path. In the present example, the configuration value also includes a default value for the reference parameter D_{adj} . In further embodiments (not shown), the configuration values also include a default value for the Dst reference parameter.

[0052] The memory 222 also stores, in conjunction with the parametric path definition, one or more boundary constraints, which may also be referred to as global parameters. The global parameters define outer boundaries for, for example, values of a measured distance to shelf D_m . That is, if the measured distance D_m falls outside the range defined by the global parameters during execution of the path, the navigator 304 may be configured to abort execution of the path. In other embodiments, the boundary constraints may be omitted. Other boundary constraints can also be specified, including maximum speed of travel of the apparatus 103, as well as a maximum permissible deviation in an angle of orientation, θ , of the apparatus 103.

[0053] The memory 222 further stores, in conjunction with the parametric path definition, a progress indicator definition. As will be discussed below, the progress

indicator definition defines a measurement of progress made along the path by the apparatus 103 according to the independent parameter. In other embodiments, the progress indicator definition may be omitted.

[0054] In the present example performance of block 415, therefore, the parametric path generator 312 is configured to retrieve the parametric path definition having the identifier PPD1, as well as the associated configuration values, boundary constraints and progress indicator definition. At block 420, the parametric path generator 312 is configured to obtain values for the reference parameters in the parametric path definition, and to generate a sequence of poses (that is, to generate the path itself) according to the parametric path definition. The generation of the path (i.e., of the sequence of poses) will be discussed below in connection with FIG. 5.

[0055] Turning to FIG. 5, the above-mentioned shelf modules 1-A to 1-G and 2-A to 2-G are illustrated from above. A two-dimensional frame of reference 500 is defined in the facility. The frame of reference 500 may also include a third dimension (i.e., height), however in the present example the apparatus 103 travels along the ground of the facility in two dimensions, and the third dimension is therefore omitted. It is assumed that an edge 504 of the modules 1-A to 1-G facing into the aisle is located at coordinates $X=50$ in the frame of reference 500 (that is, the shelf edge 504 is parallel to the Y axis). It is further assumed that the edge 504 travels from a first end at coordinates of $(X=50, Y=10)$ (near the bottom of the page) to $(X=50, Y=30)$ (near the top of the page). A target distance to shelf D_{st} is illustrated between the shelf edge 504 and a path 508 defined by the parametric path definition PPD1.

[0056] Values for the target distance to shelf D_{st} and the adjustment parameter D_{adj} , may be obtained at block 420 from the memory 222 (e.g. from the default values of Table 2) or from the task command received at block 405. In the present example performance of the method 400, a value for the target distance to shelf is received at block 405 from the server 101 and passed to the parametric path generator 312 by the navigation controller 304. The value of D_{st} is assumed to be 5, in the present example performance of block 415. A value for the adjustment parameter D_{adj} , meanwhile, is retrieved from the configuration values in Table 2 (the value is zero, in the present example).

[0057] Having obtained values for the reference parameters specified by the parametric path definition, the parametric path generator 312 is configured to generate a sequence of poses, each pose corresponding to a value of the independent parameter. In the present example, as seen from Table 2, eight poses 512 are generated, each defined at least by X and Y coordinates. Each pose may also be defined by an angle θ , indicating the angle the apparatus 103 faces at the given pose. Typically, the angle θ is determined as the direction between the current pose and the next pose in the sequence. As will also be apparent from Table 2, the first pose 512 has (X, Y) coordinates of (50, 10), while the last pose 512 has (X, Y) coordinates of (50, 30). The parameter Dadj has no impact on the sequence of poses at this point, as no measured distance to shelf values are yet available.

[0058] As also seen in FIG. 5, other parametric path definitions may define more complex paths, such as a path 520 having a sawtooth shape. The path 520 may be employed, for example, when the corresponding sub-regions (in the present example, the modules 2-A to 2-G) have reflective surfaces such as glass doors that are not suitable for imaging by the apparatus 103 unless the cameras 208 are at non-perpendicular angles to the doors. Such a path definition may include shape parameters imposing the sawtooth shape on the resulting path. The path definition may also include parameters (which may be fixed in the form of shape parameters, or variable in the form of reference parameters) defining the pitch of the sawtooth pattern, the amplitude of the wave, and the distance between the peaks of the wave and the modules 2-A to 2-G.

[0059] Returning to FIG. 4, the output of the parametric path generator at block 420 is shown below in Table 3, with the angle parameter θ omitted for simplicity (as the path 508 is a straight line)

Table 3: Sequence of Poses for Path 508

λ	X	Y
0	45	10
0.143	45	12.857
0.286	45	15.714
0.423	45	18.571
0.571	45	21.429
0.714	45	24.286
0.857	45	27.143
1	45	30

[0060] Responsive to generating the sequence of poses at block 420, the parametric path generator 312 is configured to pass the sequence of poses to the navigation controller 308 for forwarding to the local planner 320. In some embodiments, following the performance of block 420, the navigation controller 308 is also configured, upon receiving the sequence of poses from the parametric path generator 312, to transmit at least a subset of the poses to the auxiliary path generator 316. In the present example, the first pose (i.e. the start of the parametric path 508) is transmitted to the auxiliary path generator 316.

[0061] At block 425, the auxiliary path generator 316 is configured to generate an auxiliary sequence of poses, beginning at a current location of the apparatus 103 and terminating at the first pose of the parametric path 508. In other words, the task of navigating to the start of the parametric path is typically left to the auxiliary path generator 316, which is configured to generate the auxiliary path based on conventional map and cost-based mechanisms.

[0062] At block 430, the navigation controller 308 is configured to execute the auxiliary path by sending the auxiliary poses to the local planner 320, and to determine whether a current location of the apparatus 103 matches the first pose of the parametric path generated at block 420. When the determination is negative, the execution of the auxiliary path continues at block 425. When the determination at block 430 is affirmative, the performance of the method 400 proceeds to block 435.

[0063] At block 435, the navigation controller 308 is configured to transmit the sequence of poses defining the parametric path 508 to the local planner for execution. The local planner 320, in turn, is configured to control the locomotive assembly 202 to traverse the sub-region(s) identified in the task command received at block 405 according to the poses of the parametric path 508. At block 440, the navigation controller 308 can be configured to update and publish a progress indicator according to the progress indicator definition shown in Table 2 (i.e. retrieved previously by the parametric path generator 312). Specifically, the local planner 320 is configured to report the arrival of the apparatus 103 at each pose of the path 508 to the navigation controller 308. The navigation controller 308, in turn, is configured to generate an

updated progress indicator based on the most-recently received report from the local planner 320. In the present example, as will be apparent from Table 2, the progress indicator is a percentage based solely on the independent variable. Thus, for example, following arrival of the apparatus 103 at the third pose of the path 308, the progress indicator is 28.6%. A wide variety of other progress indicator definitions may also be employed.

[0064] The publication of the progress indicator can be achieved, for example, by saving the updated progress indicator to a register in the memory 222 that is accessible by other components (e.g. the task manager 300). In other embodiments, such other components may explicitly subscribe to the progress indicator, and the navigation controller 308 can store identifiers of such components and send the updated progress indicator to each component.

[0065] At block 440, the navigation controller 308 can also be configured to publish the independent parameter itself. For example, the currently active value of the independent parameter may be published (e.g. to a register or by transmission to subscribing components, as mentioned above in connection with the progress indicator).

[0066] At block 445, the navigation controller 308 is configured to monitor for updates to the reference parameters identified in the parametric path definition whose identifier was retrieved at block 415. For example, in the present example, the navigation controller 308 can be configured to determine whether an updated target distance to shelf Dst has been received (e.g. from the server 101). As a further example, the navigation controller 308 can be configured to monitor a measured distance to the shelf during execution of the path 508. The measured distance to the shelf may be obtained from the lidar sensors 216. As will be apparent to those skilled in the art, the measured distance to shelf Dm may not always match the target distance to shelf Dst, for example due to minor localization errors at the apparatus 103, or sensor errors in the lidar sensors 216. As a result of such errors, the trajectory taken by the apparatus 103 may deviate from the path 508. As will be discussed below, the nature of the parametric path definitions may permit the apparatus 103 to dynamically correct for such errors, without requiring computationally expensive cost-based path regeneration.

[0067] Turning to FIG. 6, the path 508 and the modules 1-A to 1-G are shown, along with the trajectory travelled by the apparatus 103 (which is overlaid in solid lines on the path 508, illustrated as a dashed line). In particular, in a first portion 600-1 of the trajectory follows the path 508, as the perceived shelf edge 604 (e.g., by the lidar sensors 216) aligns with the true shelf edge 504. The measured distance to shelf Dm-1 therefore matches the target distance Dst, and the adjustment parameter Dadj therefore remains at zero.

[0068] However, at a later point along the portion 600-1, the perceived shelf edge has shifted to a location 604' (e.g. due to sensor drift). As a result, the measured distance to shelf Dm-2 is smaller than Dst. As seen in Table 2, the navigation controller is configured to generate the adjustment parameter Dadj based on a difference between Dst and Dm. For example, the navigation controller 308 may implement a PID or other suitable digital controller to minimize the difference between Dm and Dst. In the present example, therefore, the difference between Dm-2 and Dst leads to the generation at the navigation controller of an adjustment parameter Dadj-1 at block 445. Following an affirmative determination at block 445, the navigation controller 308 returns to block 420 and causes the parametric path generator 312 to generate an updated sequence of poses based on the updated parameter. That is, in the present example, rather than employing the default value for Dadj at this instance of block 420, the parametric path generator 312 generates a new sequence of poses (beginning at the current value of λ rather than at the initial value of λ) using the value of Dadj generated at block 445.

[0069] The updated path is shown at FIG. 6 as a parametric path 508'. The performance of blocks 435-445 is then repeated for the updated path 508', resulting in travel of the apparatus 103 along a second portion 600-2 of the trajectory, during which a measured distance to shelf Dm-3 matches the target distance to shelf Dst, and therefore during which no further changes are made to the adjustment parameter Dadj. Subsequently, the shelf edge as perceived by the lidar sensors 216 returns to the position 604, leading to an error between the measured distance Dm-4 and the target distance Dst. As a result, via further performances of block 445 and 420, the navigator 304 generates an updated adjustment parameter (which in the present example returns to zero) and a

corresponding parametric path coincident with the original path 508. The remainder of the trajectory 600-3 follows the updated path.

[0070] Returning to FIG. 4, at block 450, following a negative determination at block 445, the navigation controller 308 is configured to determine whether execution of the current path is complete at block 450. When the path is not complete, the performance of method 400 returns to block 435. When the path is complete, however, performance of the method 400 ends. Following termination of the method 400, the navigator 304 may be configured to generate a further path (e.g. via the auxiliary path generator 316) to return to the dock 108.

[0071] As will now be apparent, a wide variety of other path configurations may also be implemented via parametric path definitions. In an embodiment, for example, a parametric path definition may include a further reference parameter, or set of reference parameters, in place of the shape parameter 50 of the definition PPD1 shown in Table 2. As will be apparent from FIGS. 5 and 6, the parameter 50 indicates the position of the shelf edge 504 in the frame of reference 500. Rather than employ a fixed position for the shelf edge 504, the parametric path definition may include a further reference parameter to which a value may be assigned (and therefore altered) that indicates the position of the shelf. Such a value may be received from the server at block 405 (e.g., in the form of a plane or line in the frame of reference 500 that contains the shelf edge 504), or generated at the apparatus 103 (e.g. via a shelf recognition function configured to detect the shelf from images captured by the cameras 208).

[0072] Further variations to the above path generation and execution are contemplated. For example, although the control method 400 is described above as being performed within the apparatus 103, in other embodiments, some or all of the blocks of the method 400 may be performed at the server 101 or the client device 105. That is, the blocks of the method 400, with the exception of the execution of commands by the locomotive assembly 202, may be performed by any other suitable computing device. Such a computing device includes the relevant ones of the components shown in FIG. 3.

[0073] In the foregoing specification, specific embodiments have been described. However, one of ordinary skill in the art appreciates that various modifications and changes can be made without departing from the scope of the invention as set forth in

the claims below. Accordingly, the specification and figures are to be regarded in an illustrative rather than a restrictive sense, and all such modifications are intended to be included within the scope of present teachings.

[0074] The benefits, advantages, solutions to problems, and any element(s) that may cause any benefit, advantage, or solution to occur or become more pronounced are not to be construed as a critical, required, or essential features or elements of any or all the claims. The invention is defined solely by the appended claims including any amendments made during the pendency of this application and all equivalents of those claims as issued.

[0075] Moreover in this document, relational terms such as first and second, top and bottom, and the like may be used solely to distinguish one entity or action from another entity or action without necessarily requiring or implying any actual such relationship or order between such entities or actions. The terms "comprises," "comprising," "has," "having," "includes," "including," "contains," "containing" or any other variation thereof, are intended to cover a non-exclusive inclusion, such that a process, method, article, or apparatus that comprises, has, includes, contains a list of elements does not include only those elements but may include other elements not expressly listed or inherent to such process, method, article, or apparatus. An element preceded by "comprises ...a", "has ...a", "includes ...a", "contains ...a" does not, without more constraints, preclude the existence of additional identical elements in the process, method, article, or apparatus that comprises, has, includes, contains the element. The terms "a" and "an" are defined as one or more unless explicitly stated otherwise herein. The terms "substantially", "essentially", "approximately", "about" or any other version thereof, are defined as being close to as understood by one of ordinary skill in the art, and in one non-limiting embodiment the term is defined to be within 10%, in another embodiment within 5%, in another embodiment within 1% and in another embodiment within 0.5%. The term "coupled" as used herein is defined as connected, although not necessarily directly and not necessarily mechanically. A device or structure that is "configured" in a certain way is configured in at least that way, but may also be configured in ways that are not listed.

[0076] It will be appreciated that some embodiments may be comprised of one or more generic or specialized processors (or “processing devices”) such as microprocessors, digital signal processors, customized processors and field programmable gate arrays (FPGAs) and unique stored program instructions (including both software and firmware) that control the one or more processors to implement, in conjunction with certain non-processor circuits, some, most, or all of the functions of the method and/or apparatus described herein. Alternatively, some or all functions could be implemented by a state machine that has no stored program instructions, or in one or more application specific integrated circuits (ASICs), in which each function or some combinations of certain of the functions are implemented as custom logic. Of course, a combination of the two approaches could be used.

[0077] Moreover, an embodiment can be implemented as a computer-readable storage medium having computer readable code stored thereon for programming a computer (e.g., comprising a processor) to perform a method as described and claimed herein. Examples of such computer-readable storage mediums include, but are not limited to, a hard disk, a CD-ROM, an optical storage device, a magnetic storage device, a ROM (Read Only Memory), a PROM (Programmable Read Only Memory), an EPROM (Erasable Programmable Read Only Memory), an EEPROM (Electrically Erasable Programmable Read Only Memory) and a Flash memory. Further, it is expected that one of ordinary skill, notwithstanding possibly significant effort and many design choices motivated by, for example, available time, current technology, and economic considerations, when guided by the concepts and principles disclosed herein will be readily capable of generating such software instructions and programs and ICs with minimal experimentation.

[0078] The Abstract of the Disclosure is provided to allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, in the foregoing Detailed Description, it can be seen that various features are grouped together in various embodiments for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than

all features of a single disclosed embodiment. Thus the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separately claimed subject matter.

Claims

1. A control method in a mobile automation apparatus, comprising:

storing, in a memory, a plurality of parametric path definition identifiers and respective corresponding sub-region identifiers identifying sub-regions in a facility;

receiving a task command containing (i) a task identifier, and (ii) a target one of the sub-region identifiers identifying a target one of the sub-regions;

retrieving, from the memory, a selected one of the parametric path definition identifiers that corresponds to the target sub-region identifier;

obtaining a selected one of a plurality of parametric path definitions that corresponds to the selected parametric path definition identifier, the selected parametric path definition including a shape parameter and a reference parameter name;

obtaining a reference parameter value corresponding to the reference parameter name, and generating a sequence of mobile automation apparatus poses according to the parametric path definition, each pose corresponding to a location in the facility; and

controlling a locomotive assembly of the mobile automation apparatus to traverse the target sub-region according to the sequence of poses.

2. The control method of claim 1, further comprising:

storing the plurality of parametric path definitions in the memory; wherein obtaining the selected parametric path definition comprises retrieving the selected parametric path definition from the memory.

3. The control method of claim 1, wherein obtaining the selected parametric path definition comprises transmitting a request containing the selected parametric path definition identifier from the mobile automation apparatus to a control server; and receiving the selected parametric path definition responsive to the request.

4. The control method of claim 1, wherein the reference parameter defines an operational constraint relative to an object in the target sub-region.

5. The control method of claim 4, wherein the object includes a shelf module, and wherein the reference parameter includes a distance to the shelf module.
6. The control method of claim 4, wherein obtaining the reference parameter value comprises:
 - receiving sensor data at the mobile automation apparatus; and
 - generating the reference parameter based on the sensor data.
7. The control method of claim 4, wherein obtaining the reference parameter value comprises receiving the reference parameter value in the task command.
8. The control method of claim 1, further comprising:
 - monitoring for an updated reference parameter value;
 - responsive to detecting the updated reference parameter value, generating an updated sequence of mobile automation apparatus poses according to the parametric path definition.
9. The control method of claim 1, further comprising:
 - obtaining a selected one of a plurality of progress indicator definitions that corresponds to the selected parametric path definition identifier; and
 - during the controlling of the locomotive assembly to traverse the target sub-region according to the sequence of poses, generating a progress indicator according to the selected progress indicator definition.
10. The control method of claim 1, further comprising, responsive to generating the sequence of mobile automation apparatus poses:
 - obtaining a current location of the mobile automation apparatus;
 - generating an auxiliary path from the current location to an initial one of the sequence of poses, according to a cost function; and
 - prior to controlling the locomotive assembly to traverse the target sub-region, controlling the locomotion assembly to traverse the auxiliary path.

11. A computing device for controlling navigation of a mobile automation apparatus, comprising:

- a memory storing a plurality of parametric path definition identifiers and respective corresponding sub-region identifiers identifying sub-regions in a facility;

- a task manager configured to receive a task command containing (i) a task identifier, and (ii) a target one of the sub-region identifiers identifying a target one of the sub-regions;

- a navigation controller configured to retrieve, from the memory, a selected one of the parametric path definition identifiers that corresponds to the target sub-region identifier;

- a parametric path generator configured to obtain a selected one of a plurality of parametric path definitions that corresponds to the selected parametric path definition identifier, the selected parametric path definition including a shape parameter and a reference parameter name;

- the parametric path generator further configured to obtain a reference parameter value corresponding to the reference parameter name, and generate a sequence of mobile automation apparatus poses according to the parametric path definition, each pose corresponding to a location in the facility; and

- a local planner configured to control a locomotive assembly of the mobile automation apparatus to traverse the target sub-region according to the sequence of poses.

12. The computing device of claim 11, the memory further storing the plurality of parametric path definitions;

- wherein the parametric path generator is configured to obtain the selected parametric path definition by retrieving the selected parametric path definition from the memory.

13. The computing device of claim 11, wherein the parametric path generator is configured to obtain the selected parametric path definition by:

transmitting a request containing the selected parametric path definition identifier to a control server; and
receiving the selected parametric path definition responsive to the request.

14. The computing device of claim 11, wherein the reference parameter defines an operational constraint relative to an object in the target sub-region.

15. The computing device of claim 14, wherein the object includes a shelf module, and wherein the reference parameter includes a distance to the shelf module.

16. The computing device of claim 14, wherein the parametric path generator is configured to obtain the reference parameter value by:
receiving sensor data at the mobile automation apparatus; and
generating the reference parameter based on the sensor data.

17. The computing device of claim 14, wherein the reference parameter value is included in the task command.

18. The computing device of claim 11, wherein:
the navigation controller is configured to monitor for an updated reference parameter value; and
the parametric path generator is configured, responsive to detection of the updated reference parameter value, to generate an updated sequence of mobile automation apparatus poses according to the parametric path definition.

19. The computing device of claim 11, wherein the navigation controller is further configured to:
obtain a selected one of a plurality of progress indicator definitions that corresponds to the selected parametric path definition identifier; and
during the controlling of the locomotive assembly to traverse the target sub-region according to the sequence of poses, generate a progress indicator according to the selected progress indicator definition.

20. The computing device of claim 11, wherein the navigation controller is further configured, responsive to generation of the sequence of mobile automation apparatus poses, to:

 obtain a current location of the mobile automation apparatus;

 initiate generation of an auxiliary path from the current location to an initial one of the sequence of poses, according to a cost function; and

 prior to controlling the locomotive assembly to traverse the target sub-region, control the locomotion assembly to traverse the auxiliary path.

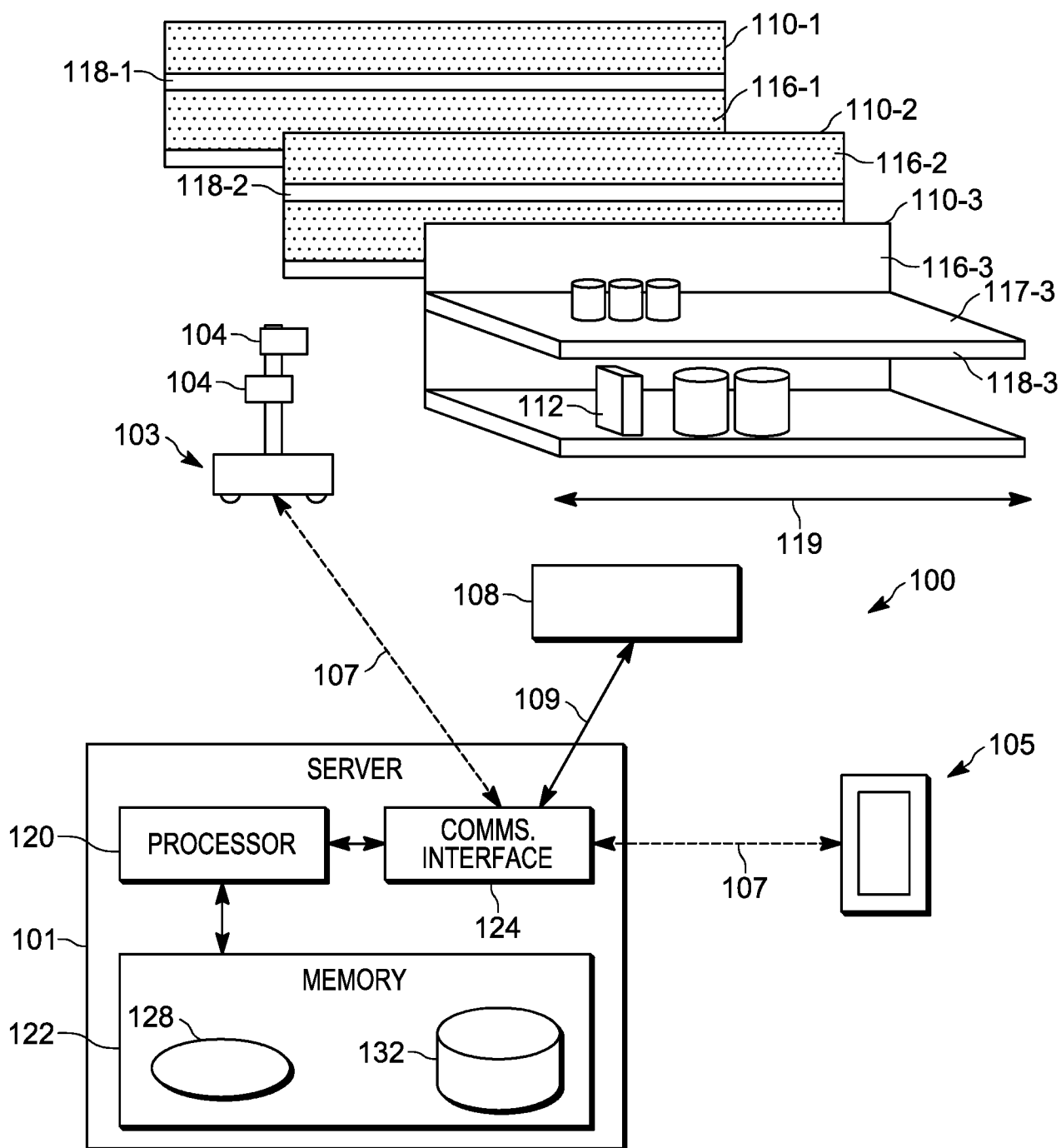


FIG. 1

2/6

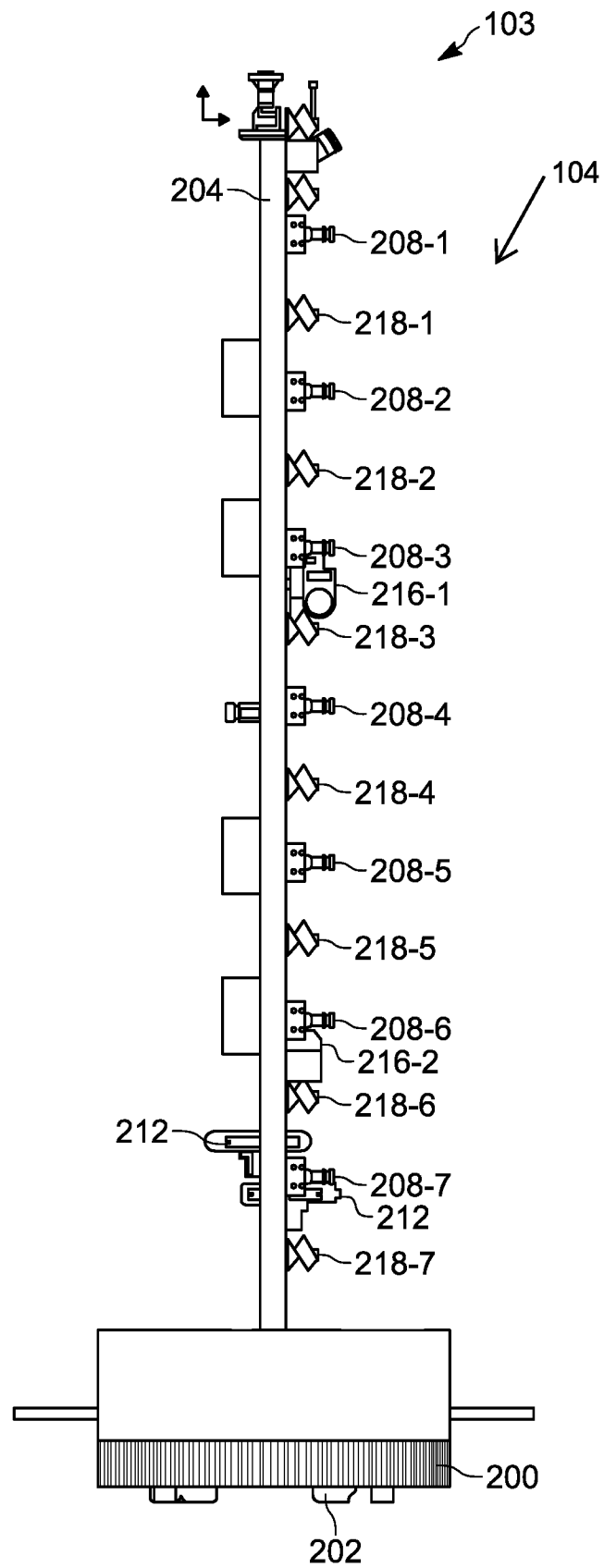


FIG. 2A

3/6

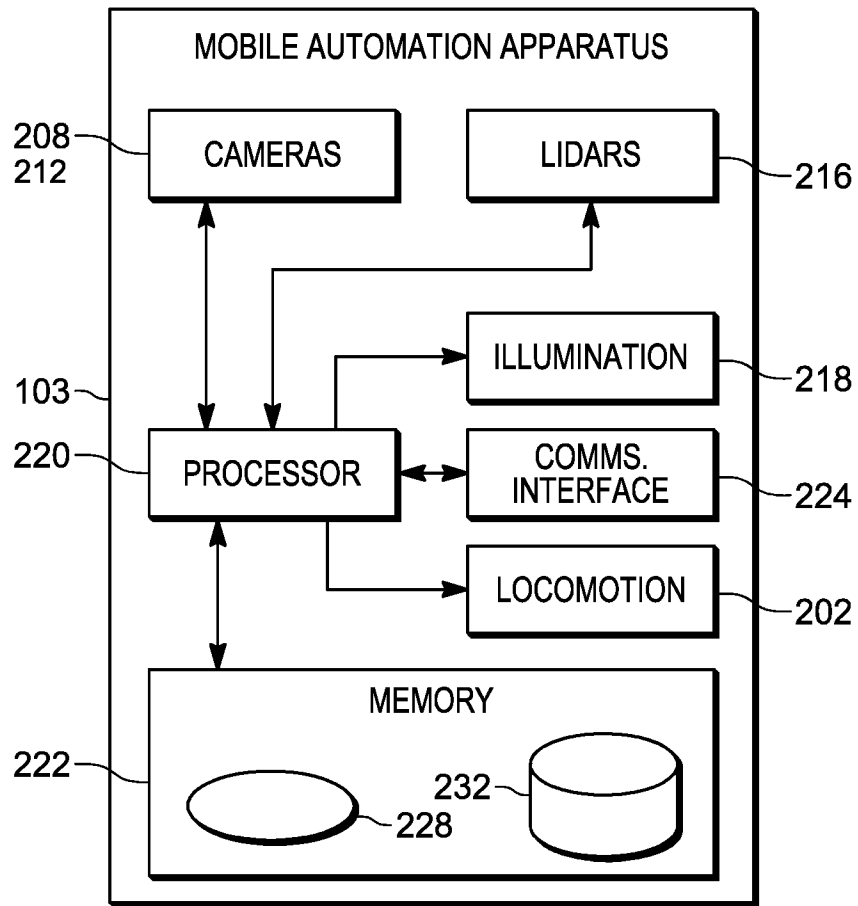


FIG. 2B

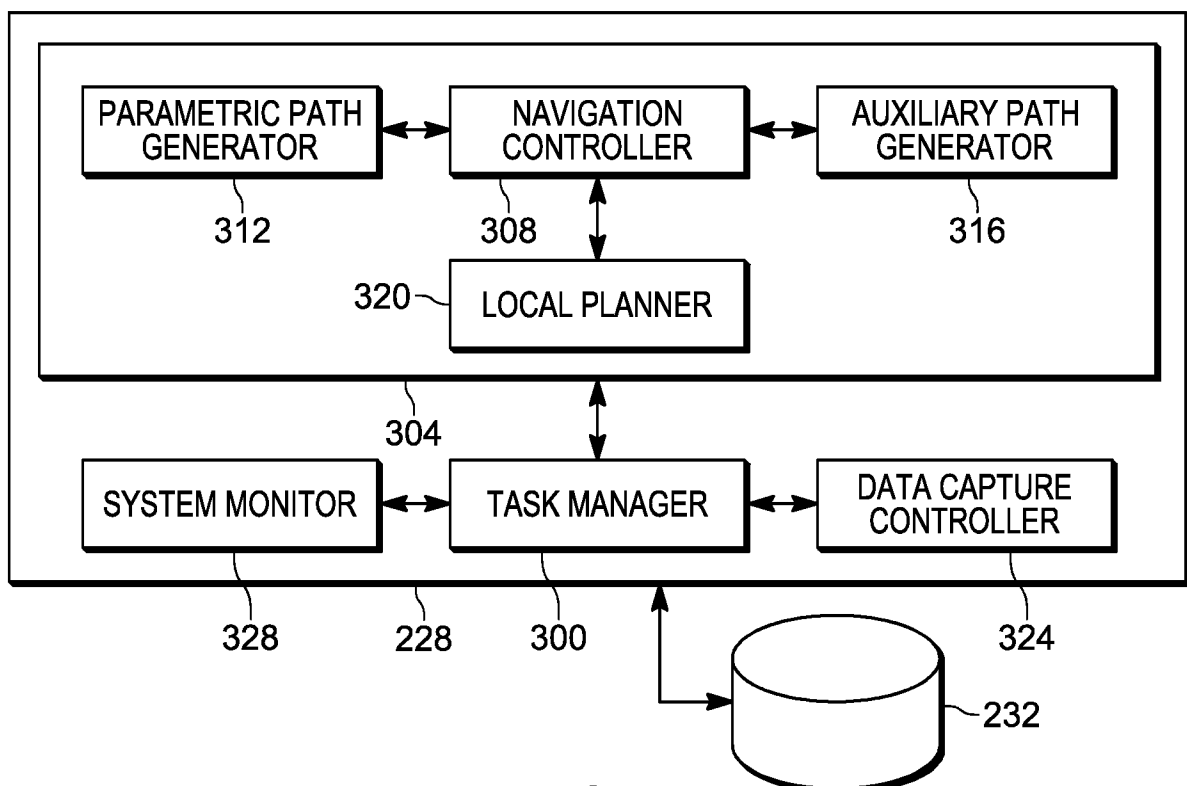


FIG. 3

4/6

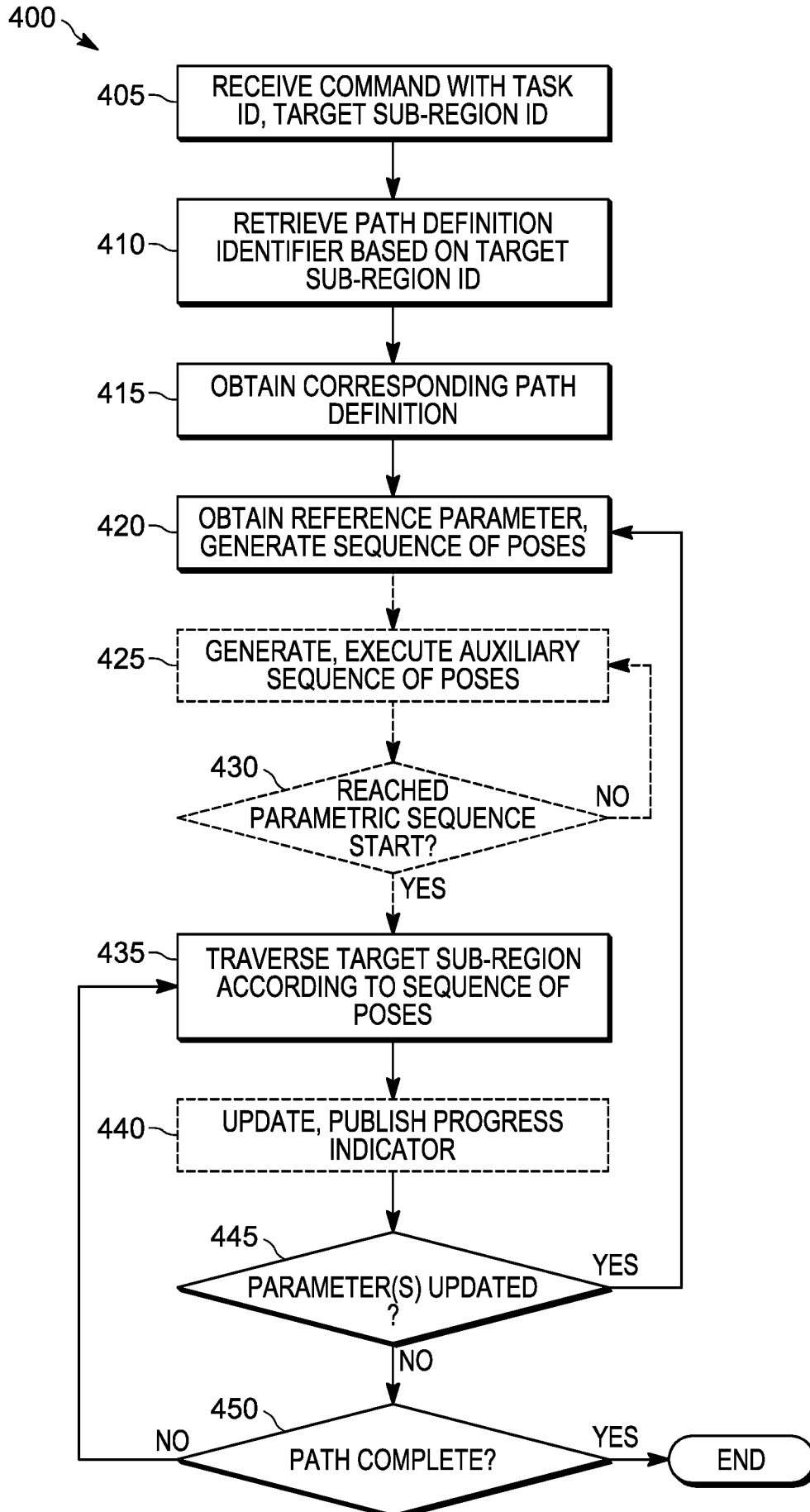


FIG. 4

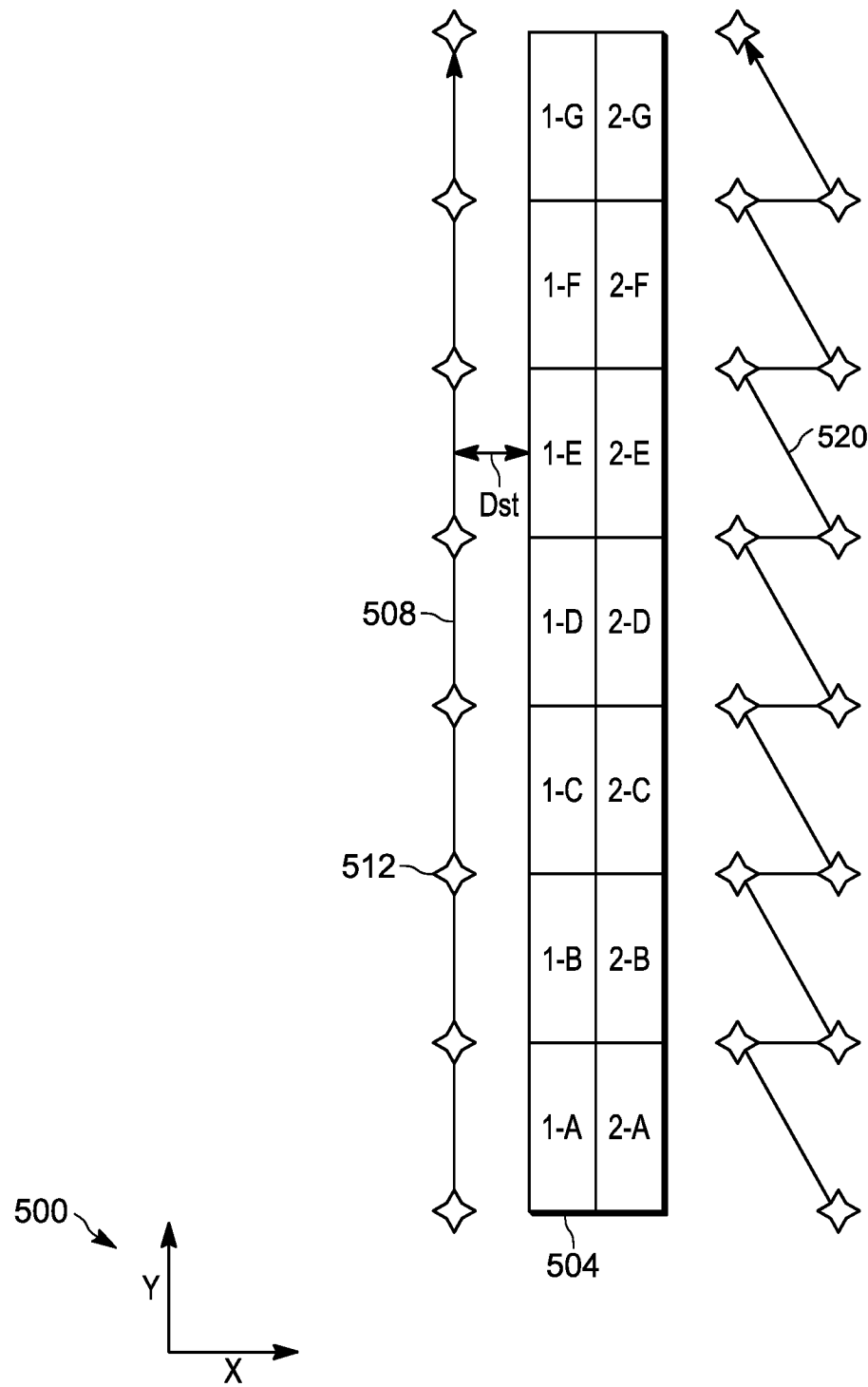
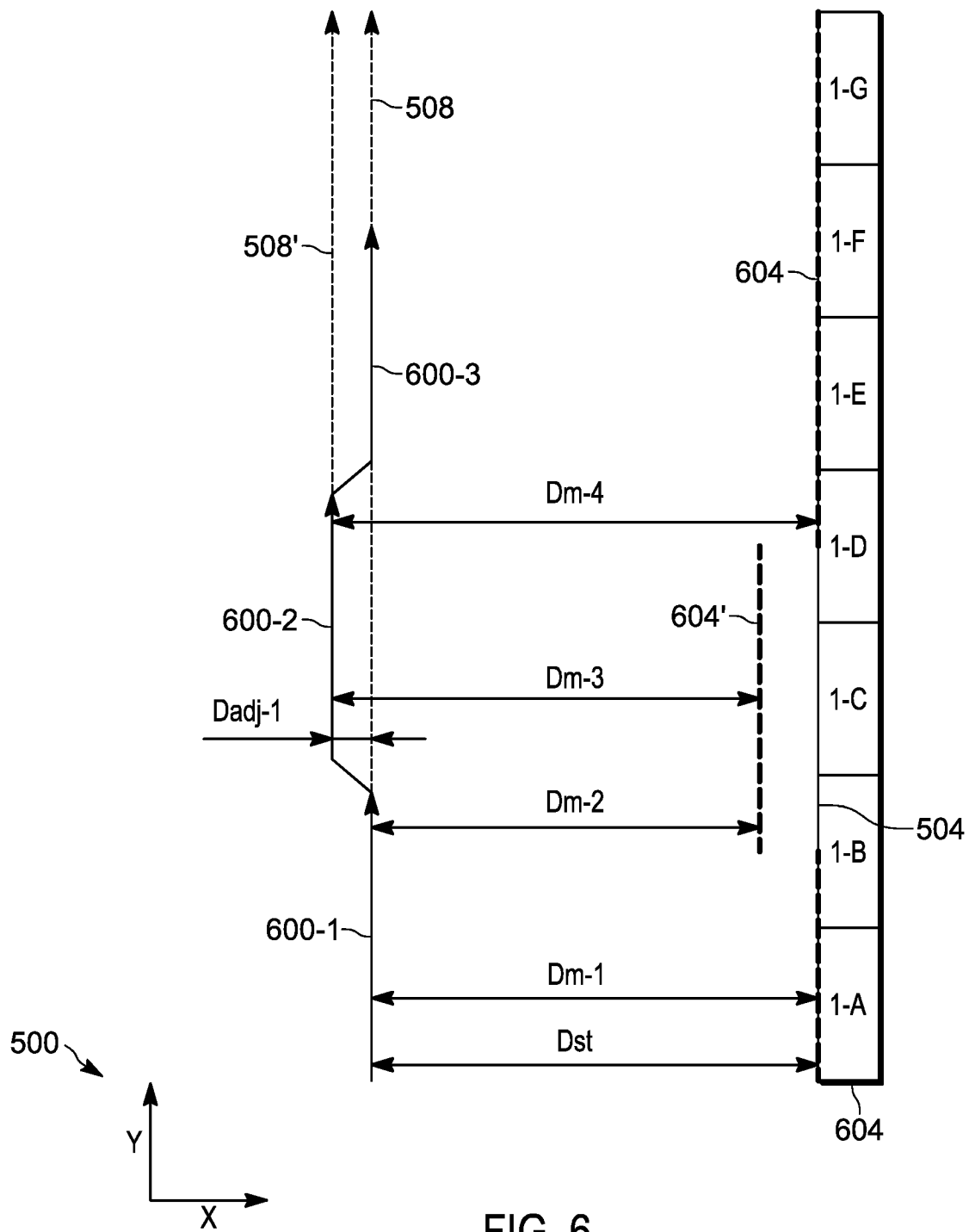


FIG. 5



INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/059648

A. CLASSIFICATION OF SUBJECT MATTER
INV. G05D1/02 G01C21/00 G06Q10/06
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G05D

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2014/277691 A1 (JACOBUS CHARLES J [US] ET AL) 18 September 2014 (2014-09-18) paragraphs [0040] - [0042], [0094] - [0098]; claims 1,3-4; figure 2 -----	1-20
A	ALONZO KELLY ET AL: "Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control", INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH., vol. 22, no. 7-8, 30 July 2003 (2003-07-30), pages 583-601, XP055454160, US ISSN: 0278-3649, DOI: 10.1177/02783649030227008 abstract; figures 1,3,8 pages 586,589 page 596 -----	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

26 February 2018

Date of mailing of the international search report

12/03/2018

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Mallet, Philippe

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2017/059648

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014277691	A1	18-09-2014	NONE
