(19) DANMARK

(10) DK/EP 4043982 T3

(12) Oversættelse af europæisk patentskrift

Patent- og Varemærkestyrelsen

(51) Int.Cl.: *G 05 D 1/00 (2006.01)* *A 61 G 5/06 (2006.01)* *A 61 G 5/10 (2006.01)* *B 60 L 15/10 (2006.01)* *G 05 D 1/02 (2020.01)*

(45) Oversættelsen bekendtgjort den: **2024-01-29**

(80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2023-11-15**

(86) Europæisk ansøgning nr.: **22166186.1**

(86) Europæisk indleveringsdag: **2017-04-13**

(87) Den europæiske ansøgnings publiceringsdag: **2022-08-17**

(30) Prioritet: **2016-04-14 US 201662322522 P**

(62) Stamansøgningsnr: **17734520.4**

(84) Designerede stater: **AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**

(73) Patenthaver: **DEKA Products Limited Partnership, 340 Commercial Street, Manchester, NH 03101-1108, USA**

(72) Opfinder: **PERET, Bob, , Bedford, 03110, USA**
COULTER, Stewart M, , Bedford, 03110, USA
KAMEN, Dean, , Bedford, 03110, USA
KANE, Derek G, , Manchester, 03104, USA

(74) Fuldmægtig i Danmark: **RWS Group, Europa House, Chiltern Park, Chiltern Hill, Chalfont St Peter, Bucks SL9 9FG, Storbritannien**

(54) Benævnelse: **BRUGERKONTROLANORDNING FRA EN TRANSPORTØR**

(56) Fremdragne publikationer:
WO-A2-2007/047510
US-A1- 2012 185 096
US-B2- 6 343 664

DESCRIPTION

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application Serial No. 62/322,522, filed April 14, 2016, entitled USER CONTROL DEVICE FOR A TRANSPORTER.

BACKGROUND

[0002] The present teachings relate generally to personal vehicles, and more specifically to user control devices for vehicles that have heightened requirements for safety and reliability. Currently, personal vehicles can ascend and descend stairs. Such devices can include a plurality of wheels that can rotate about axes that are fixed with respect to a cluster arm. The cluster arm can rotate about an axis so that wheels rest on successive stairs. Currently, a user can board or disembark from an automobile or other enclosed vehicle and can load a personal vehicle into or out of the enclosed vehicle.

[0003] What is needed is a user control device that can automatically determine locations of key features of the environment of the personal vehicle and can automatically cause the personal vehicle to react to the key features.

[0004] US 2012/185096 A1 describes a method of operating a mobile robot to traverse a threshold including detecting a threshold proximate the robot. The robot includes a holonomic drive system having first, second and third drive elements configured to maneuver the robot omni-directionally. The first drive element is moved onto the threshold from a first side and the second drive element is moved onto the threshold to place both the first and second drive elements on the threshold. The first drive element is moved off a second side of the threshold, opposite to the first side of the threshold, and the third drive element is moved onto the threshold, placing both the second and third drive elements on the threshold. Both the second and third drive elements move off the second side of the threshold.

[0005] WO 2007/047510 A2 describes systems, methods and devices for the automated delivery of goods from one to another using a robotic tug and accompanying cart. A computer within the tug or cart stores an electronic map of the building floor plan and intended paths for the tug to take when traversing from one location to the next. During the delivery, a variety of different sensors and scanners gather data that is used to avoid obstacles and/or adjust the movement of the tug in order to more closely follow the intended path. The system preferably includes both wired and wireless networks that allow one or more tugs to communicate with a tug base station, a primary network located at the site of the delivery and a remote host center that monitors the status and data collected by the tugs.

[0006] US 6 343 664 B2 describes a method for operating a device ascending or descending stairs. The device has a plurality of wheels rotatable about axes that are fixed with respect to a cluster arm. The cluster arm itself is rotated about an axis so that wheels rest on successive stairs. The wheels and cluster arms are controlled according to separate control laws by a controller. Whether the device ascends or descends the stairs is governed by the pitch of the device relative to specified front and rear angles.

SUMMARY

[0007] The invention is defined in the attached independent claims to which reference should now be made. Further, optional features may be found in the sub-claims appended thereto.

[0008] The user control device of the present teachings can include, but is not limited to including, a user control processor (UCP) assist that can provide enhanced functionality to a user of a personal vehicle such as the transporter of the present teachings, for example, but not limited to, assisting a user of the transporter in avoiding obstacles, traversing doors, traversing stairs, traveling on elevators, and parking/transporting the transporter. The UCP assist can receive user input and/or input from power base processors (PBPs) that can control the transporter, and can enable the invocation of a processing mode that has been automatically or manually selected. A command processor can enable the invoked mode by generating movement commands based at least on previous movement commands, data from the user, and data from sensors. The command processor can receive user data that can include signals from a joystick that can provide an indication of a desired movement direction and speed of the transporter. User data can also include mode selections into which the transporter could be transitioned. Modes such as door mode, rest room mode, enhanced stair mode, elevator mode, mobile storage mode, and static storage/charging mode can be selected. Any of these modes can include a move-to-position mode, or the user can direct the transporter to move to a certain position. UCP assist can generate commands such as movement commands that can include, but are not limited to including, speed and direction, and the movement commands can be provided to the PBPs which can transmit this information to wheel motor drives and cluster motor drives.

[0009] Sensor data can be collected by sensor-handling processors that can include, but are not limited to including, a transporter geometry processor, a point cloud library (PCL) processor, a simultaneous location and mapping (SLAM) processor, and an obstacle processor. The movement commands can also be provided to the sensor handling processors. The sensors can provide environmental information that can include, for example, but not limited to, obstacles and geometric information about the transporter. The sensors can include at least one time-of-flight sensor that can be mounted anywhere on transporter. There can be multiple sensors mounted on the transporter. The PCL processor can gather and process environmental information, and can produce PCL data that can be processed by a PCL library.

[0010] The transporter geometry processor of the present teachings can receive transporter

geometry information from the sensors, can perform any processing necessary to prepare the transporter geometry information for use by the mode-dependent processors, and can provide the transporter geometry information to mode-dependent processors. The geometry of the transporter can be used for automatically determining whether or not the transporter can fit in and/or through a space such as, for example, a stairway and a door. The SLAM processor can determine navigation information based on, for example, but not limited to, user information, environmental information, and movement commands. The transporter can travel in a path at least in part set out by navigation information. An obstacle processor can locate obstacles and distances to the obstacles. Obstacles can include, but are not limited to including, doors, stairs, automobiles, and miscellaneous features in the vicinity of the path of the transporter.

[0011] The method for obstacle processing of the present teachings can include, but is not limited to including, receiving movement commands and user information, receiving and segmenting PCL data, identifying at least one plane within the segmented PCL data, and identifying at least one obstacle within the at least one plane. The method for obstacle processing can further include determining at least one situation identifier based at least on the obstacles, user information, and movement commands, and determining the distance between the transporter and the obstacles based at least on the situation identifier. The method for obstacle processing can also include accessing at least one allowed command related to the distance, the obstacle, and the situation identifier. The method for obstacle processing can still further include accessing an automatic response to the allowed command, mapping the movement command with one of the allowed commands, and providing the movement command and the automatic response associated with the mapped allowed command to the mode-dependent processors.

[0012] The obstacles can be stationary or moving. The distance can include a fixed amount and/or can be a dynamically-varying amount. The movement command can include, but is not limited to including, a follow command, a pass-the-obstacle command, a travel-beside-the-obstacle command, and a do-not-follow-the-obstacle command. The obstacle data can be stored and retrieved locally and/or in a cloud-based storage area, for example. The method can optionally include storing the obstacle data and allowing access to the stored obstacle data by systems external to the transporter. The method for obstacle processing can optionally include collecting sensor data from a time-of-flight camera mounted on the transporter, analyzing the sensor data using a point cloud library (PCL), tracking the moving object using SLAM based on the location of the transporter, identifying a plane within the obstacle data using, and providing the automatic response associated with the mapped allowed command to the mode-dependent processors. The method for obstacle processing can optionally receive a resume command, and provide, following the resume command, a movement command and the automatic response associated with the mapped allowed command to the mode-dependent processors. The automatic response can include a speed control command.

[0013] The obstacle processor of the present teachings can include, but is not limited to including, a nav/PCL data processor. The nav/PCL processor can receive the movement commands and the user information, and can receive and segment PCL data from a PCL

processor, identify a plane within the segmented PCL data, and identify obstacles within the plane. The obstacle processor can include a distance processor. The distance processor can determine a situation identifier based on user information, the movement command, and the obstacles. The distance processor can determine the distance between the transporter and the obstacles based at least on the situation identifier. The moving object processor and/or the stationary object processor can access the allowed command related to the distance, the obstacles, and the situation identifier. The moving object processor and/or the stationary object processor can access an automatic response from an automatic response list associated with the allowed command. The moving object processor and/or the stationary object processor can access the movement command and map the movement command with one of the allowed commands. The moving object processor and/or stationary object processor can provide movement commands and the automatic response associated with the mapped allowed command to the mode-dependent processors. The movement command can include a follow command, a pass command, a travel-beside command, a move-to-position command, and a do-not-follow command. The nav/PCL processor can store obstacles in local storage and/or on storage cloud, and can allow access to the stored obstacles by systems external to the transporter.

[0014] The method of the present teachings for navigating stairs can include, but is not limited to including, receiving a stair command, and receiving environmental information from sensors mounted on the transporter and/or the obstacle processor. The method for navigating stairs can include locating, based on the environmental information, staircases within environmental information, and receiving a selection of one of the staircases located by the sensors and/or the obstacle processor. The method for navigating stairs can also include measuring the characteristics of the selected staircase, and locating, based on the environmental information, obstacles, if any, on the selected staircase. The method for navigating stairs can also include locating, based on the environmental information, a last stair of the selected staircase, and providing movement commands to move the transporter on the selected staircase based on the measured characteristics, the last stair, and the obstacles, if any. The method for navigating stairs can continue providing movement commands until the last stair is reached. The characteristics can include, but are not limited to including, the height of the stair riser of the selected staircase, the surface texture of the riser, and the surface temperature of the riser. Alerts can be generated if the surface temperature falls outside of a threshold range and the surface texture falls outside of a traction set.

[0015] The method can optionally include locating the at least one staircase based on GPS data, building a map of the selected staircase using SLAM, saving the map, and updating the map while the transporter is moving. The method can optionally include accessing a geometry of the transporter, comparing the geometry to the at least one characteristic of the selected staircase, and modifying the movement of the transporter based on the comparing step. The characteristic can include, but is not limited to including, the height of at least one riser of the selected staircase, the surface texture of the at least one riser, and the surface temperature of the at least one riser. The method can optionally include generating an alert if the surface temperature falls outside of a threshold range and the surface texture falls outside of a traction

set. The threshold range can include, but is not limited to including, temperatures below 33°F. The traction set can include, but is not limited to including, a carpet texture. The method can optionally include determining, based on the sensor data, the topography of an area surrounding the selected staircase, and generating an alert if the topography is not flat. The method can optionally include accessing a set of extreme circumstances.

[0016] The navigating stair processor of the present teachings can include, but is not limited to including, a staircase processor receiving at least one stair command included in user information, and a staircase locator receiving, through, for example, the obstacle processor, environmental information from sensors mounted on the transporter. The staircase locator can locate, based on environmental information, the staircases within the environmental information, and can receive the choice of a selected staircase. The stair characteristics processor can measure the characteristics of the selected staircase, and can locate, based on environmental information, obstacles, if any, on the selected staircase. The stair movement processor can locate, based on environmental information, a last stair of the selected staircase, and can provide to movement processor movement commands to instruct the transporter to move on the selected staircase based on the characteristics, the last stair, and the obstacles, if any. The staircase locator can locate staircases based on GPS data, and can build and save a map of the selected staircase. The map can be saved for use locally and/or by other devices unrelated to the transporter. The staircase processor can access the geometry of the transporter, compare the geometry to the characteristics of the selected staircase, and modify the navigation of the transporter based on the comparison. The staircase processor can optionally generate an alert if the surface temperature of the risers of the selected staircase falls outside of a threshold range and the surface texture of selected staircase falls outside of a traction set. The stair movement processor can determine, based on the environmental information, the topography of an area surrounding the selected staircase, and can generate an alert if the topography is not flat. The stair movement processor can access a set of extreme circumstances that can be used to modify the movement commands generated by the stair movement processor.

[0017] When the transporter traverses the threshold of a door, where the door can include a door swing, a hinge location, and a doorway, the method of the present teachings for navigating a door can include receiving and segmenting environmental information from sensors mounted on the transporter. The environmental information can include the geometry of the transporter. The method can include identifying a plane within the segmented sensor data, and identifying the door within the plane. The method for navigating a door can include measuring the door based on the environmental information. The method for navigating a door can include determining the door swing and providing movement commands to move the transporter for access to a handle of the door. The method for navigating a door can include providing movement commands to move the transporter away from the door as the door opens by a distance based on the door measurements. The method for navigating a door can include providing movement commands to move the transporter forward through the doorway. The transporter can maintain the door in an open position if the door swing is towards the transporter.

[0018] The method of the present teachings for processing sensor data can determine, through information from the sensors, the hinge side of the door, the direction and angle of the door, and the distance to the door. The movement processor of the present teachings can generate commands to PBPs such as start/stop turning left, start/stop turning right, start/stop moving forward, start/stop moving backwards, and can facilitate door mode by stopping the transporter, cancelling the goal that the transporter can be aiming to complete, and centering the joystick. The door processor of the present teachings can determine whether the door is, for example, a push to open, a pull to open, or a slider. The door processor can determine the width of the door based on the current position and orientation of the transporter, and can determine the x/y/z location of the door pivot point. If the door processor determines that the number of valid points in the image of the door derived from the set of obstacles and/or PCL data is greater than a threshold, the door processor can determine the distance from the transporter to the door. The door processor can determine if the door is moving based on successive samples of PCL data from the sensor processor. In some configurations, the door processor can assume that a side of the transporter is even with the handle side of the door, and can use that assumption, along with the position of the door pivot point, to determine the width of the door. The door processor can generate commands to move the transporter through the door based on the swing and the width of the door. The transporter itself can maintain the door in an open state while the transporter traverses the threshold of the door.

[0019] In some configurations, the transporter can automatically negotiate the use of rest room facilities. The doors to the rest room and to the rest room stall can be located as discussed herein, and the transporter can be moved to locations with respect to the doors as discussed herein. Fixtures in the rest room can be located as obstacles as discussed herein, and the transporter can be automatically positioned in the vicinity of the fixtures to provide the user with access to, for example, the toilet, the sink, and the changing table. The transporter can be automatically navigated to exit the rest room stall and the rest room through door and obstacle processing discussed herein. The transporter can automatically traverse the threshold of the door based on the geometry of the transporter.

[0020] The method of the present teachings for automatically storing the transporter in a vehicle, such as, for example, but not limited to, an accessible van, can assist a user in independent use of the vehicle. When the user exits the transporter and enters the vehicle, possibly as the vehicle's driver, the transporter can remain parked outside of the vehicle. If the transporter is to accompany the user in the vehicle for later use, the mobile park mode of the present teachings can provide movement commands to the transporter to cause the transporter to store itself either automatically or upon command, and to be recalled to the door of the vehicle as well. The transporter can be commanded to store itself through commands received from external applications, for example. In some configurations, a computer-driven device such as a cell phone, laptop, and/or tablet can be used to execute one or more external applications and generate information that could ultimately control the transporter. In some configurations, the transporter can automatically proceed to mobile park mode after the user exits the transporter. Movement commands can include commands to locate the door of the

vehicle at which the transporter will enter to be stored, and commands to direct the transporter to the vehicle door. Mobile park mode can determine error conditions such as, for example, but not limited to, if the vehicle door is too small for the transporter to enter, and mobile park mode can alert the user of the error condition through, for example, but not limited to, an audio alert through audio interface and/or a message to one or more external applications. If the vehicle door is wide enough for the transporter to enter, mobile park mode can provide vehicle control commands to command the vehicle to open the vehicle door. Mobile park mode can determine when the vehicle door is open and whether or not there is space for the transporter to be stored. Mobile park mode can invoke the method for obstacle processing to assist in determining the status of the vehicle door and if there is room in the vehicle to store the transporter. If there is enough room for the transporter, mobile park mode can provide movement commands to move the transporter into the storage space in the vehicle. Vehicle control commands can be provided to command the vehicle to lock the transporter into place, and to close the vehicle door. When the transporter is again needed, one or more external applications, for example, can be used to bring the transporter back to the user. The status of the transporter can be recalled, and vehicle control commands can command the vehicle to unlock the transporter and open the door of the vehicle. The vehicle door can be located and the transporter can be moved through the vehicle door and to the passenger door to which it had been summoned by, for example, one or more external applications. In some configurations, the vehicle can be tagged in places such as, for example, the vehicle entry door where the transporter can be stored.

[0021] The method of the present teachings for storing/recharging the transporter can assist the user in storing and possibly recharging the transporter, possibly when the user is sleeping. After the user exits the transporter, commands can be initiated by one or more external applications, to move the perhaps riderless transporter to a storage/docking area. In some configurations, a mode selection by the user while the user occupies the transporter can initiate automatic storage/docking functions after the user has exited the transporter. When the transporter is again needed, commands can be initiated by one or more external applications to recall the transporter to the user. The method for storing/recharging the transporter can include, but is not limited to including, locating at least one storage/charging area, and providing at least one movement command to move the transporter from a first location to the storage/charging area. The method for storing/recharging the transporter can include locating a charging dock in the storage/charging area and providing at least one movement command to couple the transporter with the charging dock. The method for storing/recharging the transporter can optionally include providing at least one movement command to move the transporter to the first location when the transporter receives an invocation command. If there is no storage/charging area, or if there is no charging dock, or if the transporter cannot couple with the charging dock, the method for storing/recharging the transporter can optionally include providing at least one alert to the user, and providing at least one movement command to move the transporter to the first location.

[0022] The method of the present teachings for negotiating an elevator while maneuvering the transporter can assist a user in getting on and off the elevator in the transporter. When the

elevator is, for example, automatically located, and when the user selects the desired elevator direction, and when the elevator arrives and the door opens, movement commands can be provided to move the transporter into the elevator. The geometry of the elevator can be determined and movement commands can be provided to move the transporter into a location that makes it possible for the user to select a desired activity from the elevator selection panel. The location of the transporter can also be appropriate for exiting the elevator. When the elevator door opens, movement commands can be provided to move the transporter to fully exit the elevator.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] The present teachings will be more readily understood by reference to the following description, taken with the accompanying drawings, in which:

FIG. 1 is a schematic representation of a transporter of the present teachings;

FIGs. 2A-2D are schematic block diagrams of the components of the transporter of the present teachings;

FIGs. 3A-3B are schematic block diagrams of mode processing of the present teachings;

FIG. 4 is a schematic block diagram of the electronic components of the transporter of the present teachings;

FIG. 5A is a line-drawing representation of an exemplary visual interface of the present teachings;

FIG. 5B is a line-drawing representation of an exemplary manual interface of the present teachings;

FIG. 6 is a line-drawing representation of exemplary manual interface switches/buttons of the present teachings;

FIG. 6A is a schematic representation of the user control device case of the present teachings;

FIGs. 6B1 and 6B2 are schematic representations of the manual interface cover and UCP assist connection of the present teachings;

FIGs. 6B3 and 6B4 are schematic representations of the UCP assist holder of the present teachings;

FIGs. 6C1 and 6C2 are schematic representations of the UCP assist connection device of the present teachings;

FIGs. 6D1 and 6D2 are schematic representations of the mounting board for the UCP assist connection device of the present teachings;

FIGs. 6D3 and 6D4 are schematic representations of the UCP assist connection device mounted on the mounting board for the UCP assist connection device;

FIGs. 6D5 and 6D6 are schematic representations of another configuration of the UCP assist connection device of the present teachings;

FIG. 6E is a line drawing of a configuration of the positioning of sensors of the transporter of the present teachings;

FIGs. 7A-7E are charts of communications packet information of the present teachings;

FIG. 8 is a graph of a manual interface response template of the present teachings;

FIG. 9 is a control flow diagram of processing of the present teachings;

FIG. 10 is a schematic block diagram of the components of the UCP assist of the present teachings;

FIGs. 11A1-11A2 are schematic block diagrams of a method of obstacle detection of the present teachings;

FIG. 11B is a schematic block diagram of the components of the obstacle detection of the present teachings;

FIGs. 12A-12D are computer-generated representations of the transporter configured with a sensor;

FIG. 13A is schematic block diagrams of a method of enhanced stair climbing of the present teachings;

FIG. 13B is a schematic block diagram of the components of the enhanced stair climbing of the present teachings;

FIGs. 14A1-14A2 are schematic block diagrams of a method of door traversal of the present teachings;

FIG. 14B is a schematic block diagram of the components of the door traversal of the present teachings;

FIG. 15A is a schematic block diagram of a method of rest room navigation of the present teachings;

FIG. 15B is a schematic block diagram of the components of the rest room navigation of the present teachings;

FIGs. 16A1-16A2 are schematic block diagrams of a method of mobile storage of the present teachings;

FIG. 16B is a schematic block diagram of the components of the mobile storage of the present teachings;

FIG. 17A is a schematic block diagrams of a method of storage/charging of the present teachings;

FIG. 17B is a schematic block diagram of the components of the storage/charging of the present teachings;

FIG. 18A is a schematic block diagrams of a method of elevator navigation of the present teachings; and

FIG. 18B is a schematic block diagram of the components of the elevator navigation of the present teachings.

DETAILED DESCRIPTION

[0024] The configuration of a user control device of the present teachings is discussed in detail below in relation to a transporter, for example, but not limited to, a wheelchair. Various types of transporters can interface with the user control device. The user control device can communicate with the transporter via electrical interface(s) that can facilitate communication and data processing among the user interface device and controllers that can control the movement of the transporter. The user control device can perform automated actions based on the environment in which the transporter operates and the user's desired movement of the transporter. External applications can enable monitoring and control of the transporter.

[0025] Referring now to FIG. 1, transporter 120 can include, but is not limited to including, user control device 131, seat 105, chassis 104, power base 160, first wheels 101, second wheels 102, third wheels 103, and cluster 121. UCD 131 can receive user and sensor input and can provide that information to power base 160. UCD 131 can include, but is not limited to including, UCP 130 and UCP assist 145. UCP assist can also be located independently from UCP 130, and can be positioned anywhere on transporter 120 including, but not limited to, on the side and on the back of transporter 120. Power base 160 can control, for example, the movements of wheels 101 and 102, cluster 121, and seat 105 based on inputs from UCD 131 and other factors including, but not limited to, automated enforcement of requirements for, for example, safety and reliability.

[0026] Continuing to refer to FIG. 1, transporter 120 can operate in functional modes such as, for example, but not limited to, standard mode 201 (FIG. 3A) in which transporter 120 can operate on drive wheels 101 and caster wheels 103, and enhanced mode 217 (FIG. 3A) in which transporter 120 can operate on drive wheels 101/102, can be dynamically stabilized through onboard sensors, and can operate having elevated chassis 104, casters 103, and seat 105. Transporter 120 can also operate in balance mode 219 (FIG. 3A) in which transporter 120 can operate on drive wheels 102, can have an elevated height of seat 105, and can be

dynamically stabilized through onboard sensors. Transporter 120 can further operate in stair mode 215 (FIG. 3A) in which transporter 120 can use wheel clusters 121 (FIG. 1) to climb and descend stairs and can be dynamically stabilized. Transporter 120 can still further operate in remote mode 205 (FIG. 3A) in which transporter 120 can operate on drive wheels 101/102 and can be unoccupied. Transporter 120 can optionally operate in docking mode 203 (FIG. 3A) in which transporter 120 can operate on drive wheels 101/102 and caster wheels 103, thereby lowering chassis 104. Some of the modes of transporter 120 are described in United States Patent # 6,343,664, entitled Operating Modes for Stair Climbing in Cluster-wheel Vehicle, issued February 2, 2002.

[0027] Referring now primarily to FIGs. 2A-2D, power base 160 (FIG. 1) can include, but is not limited to including, at least one processor 43A-43D (FIGs. 2C/2D), at least one motor drive 1050, 19, 21, 25, 27, 31, 33, 37 (FIGs. 2C/2D), at least one inertial system 1070, 23, 29, 35 (FIGs. 2C/2D), and at least one power source controller 11A/B (FIG. 2B). Power base 160 (FIG. 1) can be communicatively coupled with, for example, but not limited to, UCD 131 (FIG. 2A) through, for example, but not limited to, electronic communications means 53C and a protocol such as, for example, a controller area network (CAN) bus protocol. UCD 131 (FIG. 2A) can be optionally communicatively coupled with electronic devices 140A (FIG. 2A) such as, for example, but not limited to, computers such as tablets and personal computers, telephones, and lighting systems, and can possibly be executing external applications 140 (FIG. 4). UCD 131 (FIG. 2A) can include, but is not limited to including, at least one manual interface such as, for example, joystick 133 (FIG. 5B) and at least one push button 141A/B/C (FIG. 6), at least one visual interface such as, for example, display (FIG. 5A), and, optionally, at least one UCP assist 145 (FIG. 4). UCD 131 (FIG. 2A) can optionally be communicatively coupled with peripheral control module 1144, sensor aid modules 1141, and autonomous control modules 1142/1143. Communications can be enabled by, for example, but not limited to, a CANbus protocol and an Ethernet protocol. Other protocols can be used.

[0028] Continuing to refer primarily to FIGs. 2A-2D, in some configurations, each at least one processor 43A-43D (FIGs. 2C/2D) can include, but is not limited to including, at least one cluster motor drive 1050, 27 (FIGs. 2C/2D), at least one right wheel motor drive 19, 31 (FIG. 2C), at least one left wheel motor drive 21, 33 (FIGs. 2C/2D), at least one seat motor drive 25, 37 (FIGs. 2C/D), and at least one inertial sensor pack 1070, 23, 29, 35 (FIGs. 2C/2D). Power base 160 can further include at least one cluster brake 57, 69 (FIGs. 2C/2D), at least one cluster motor 83, 89 (FIGs. 2C/2D), at least one right wheel brake 59, 73 (FIG. 2C/2D), at least one left wheel brake 63, 77 (FIGs. 2C/2D), at least one right wheel motor 85, 91 (FIGs. 2C/2D), at least one left wheel motor 87, 93 (FIGs. 2C/2D), at least one seat motor 45, 47 (FIGs. 2C/2D), at least one seat brake 65, 79 (FIGs. 2C/2D), at least one cluster position sensor 55, 71 (FIGs. 2C/2D), and at least one manual brake release 61, 75 (FIGs. 2C/2D).

[0029] Continuing to refer primarily to FIGs. 2A-2D, power base 160 (FIG. 2C) can be used to drive cluster 121 (FIG. 1) of wheels 101/102 (FIG. 1) forming a ground-contacting module. The ground-contacting module can be mounted on cluster 121 (FIG. 1), and each wheel 101/102 (FIG. 1) of the ground-contacting module can be driven by a wheel motor drive such as, for

example, right wheel motor drive A 19 (FIG. 2C), or redundant right wheel motor drive B 31 (FIG. 2D). Cluster 121 (FIG. 1) can rotate about a cluster axis, the rotation being governed by, for example, cluster motor drive A 1050 (FIG. 2C), or redundant cluster motor drive B 27 (FIG. 2D). At least one of the sensors such as, for example, but not limited to, at least one cluster position sensor 55/71 (FIGs. 2C/2D), at least one manual brake release sensor 61/75 (FIGs. 2C/2D), at least one motor current sensor (not shown), and at least one inertial sensor pack 17, 23, 29, 35 (FIGs. 2C/2D) can sense the state of transporter 120 (FIG. 1).

[0030] Continuing to still further refer to FIGs. 2A-2D, processors 43A-43D (FIGs. 2C/2D) can be electronically coupled to UCD 131 (FIG. 2A) for receiving control input, as well as to other controllers for controlling peripheral and extraordinary functions of transporter 120 (FIG. 1). Communications 53A-53C (FIG. 2B) among UCD 131 (FIG. 2A), power source controllers 11A/11B (FIG. 2B), and each of processors 43A-43D (FIGs. 2C/D) can be according to any protocol including, but not limited to, a CANbus protocol. At least one Vbus 95, 97 (FIG. 2B) can connect at least power source controller 11A/B (FIG. 2B) to power base 160 (FIG. 2C) and components external to power base 160 (FIG. 2C) through external Vbus 107 (FIG. 2B). In some configurations, processor A1 43A (FIG. 2C) can be the master of CANbus A 53A (FIG. 2B). Slaves on CANbus A 53A (FIG. 2B) can be processor A2 43B (FIG. 2C), processor B1 43C (FIG. 2D), and processor B2 43D (FIG. 2D). In some configurations, processor B1 43C (FIG. 2D) can be the master of CANbus B 53B (FIG. 2B). Slaves on CANbus B 53B (FIG. 2B) can be processor B2 43C (FIG. 2D), processor A1 43A (FIG. 2C), and processor A2 43B (FIG. 2C). UCD 131 (FIG. 2A) can be the master of CANbus C 53C (FIG. 2B). Slaves on CANbus C 53C (FIG. 2B) can be power source controller 11A/B (FIG. 2B), processor A1 43A (FIG. 2C), processor A2 43B (FIG. 2C), processor B1 43C (FIG. 2D), and processor B2 43D (FIG. 2D). The master node (any of processors 43A-43D (FIG. 2C/D) or UCD 131 (FIG. 2A)) can send data to or request data from the slaves.

[0031] Referring now primarily to FIGs. 2C/2D, in some configurations, power base 160 can include redundant processor sets A/B 39/41 that can control clusters 121 (FIG. 1) and rotating drive wheels 101/102 (FIG. 1). Right/left wheel motor drives A/B 19/21, 31/33 can drive right/left wheel motors A/B 85/87, 91/93 that drive wheels 101/102 (FIG. 1) on the right and left sides of transporter 120 (FIG. 1). Wheels 101/102 (FIG. 1) can be coupled to drive together. Turning can be accomplished by driving left wheel motors A/B 87/93 and right wheel motors A/B 85/91 at different rates. Cluster motor drive A/B 1050/27 can drive cluster motors A/B 83/89 that can rotate the wheel base in the fore/aft direction which can allow transporter 120 (FIG. 1) to remain level while front wheels 101 (FIG. 1) are higher or lower than rear wheels 102 (FIG. 1). Cluster motors A/B 83/89 can keep transporter 120 (FIG. 1) level when climbing up and down curbs, and can rotate the wheel base repeatedly to climb up and down stairs. Seat motor drive A/B 25/37 can drive seat motors A/B 45/47 that can raise and lower seat 105 (FIG. 1).

[0032] Continuing to further refer to FIGs. 2C/2D, cluster position sensors A/B 55/71 can sense the position of cluster 121 (FIG. 1) of wheels 101/102 (FIG. 1). The signals from cluster position sensors A/B 55/71 and seat position sensors A/B 67/81 can be communicated among

processors 43A-43D and can be used by processor set A/B 39/41 to determine signals to be sent to, for example, right wheel motor drive A/B 19/31, cluster motor drive A/B 15/27 and seat motor drive A/B 25/37. The independent control of clusters 121 (FIG. 1) and drive wheels 101/102 (FIG. 1) can allow transporter 120 (FIG. 1) to operate in several modes, thereby allowing processors 43A-43D to switch between modes, for example, in response to the local terrain. The mode switch can occur, for example, automatically and/or at the request of a user.

[0033] Continuing to still further refer to FIGs. 2C/2D, inertial sensor packs 1070, 23, 29, 35 can sense, for example, but not limited to, the orientation of transporter 120 (FIG. 1). Each processor 43A-43D can include, in inertial sensor packs 1070, 23, 29, 35, accelerometers and gyroscopes. In some configurations, each inertial sensor pack 1070, 23, 29, 35 can include, but is not limited to including, four sets of three-axis accelerometers and three-axis gyros. The accelerometer and gyro data can be fused on each of processors 43A-43D. Each processor 43A-43D can produce a gravity vector that can be used to compute the orientation and inertial rotation rates of power base 160 (FIG. 1). The fused data can be shared across processors 43A-43D and can be subjected to threshold criteria. The threshold criteria can be used to improve the accuracy of device orientation and inertial rotation rates. For example, fused data from certain of processors 43A-43D that exceed certain thresholds can be discarded. The fused data from each of processors 43A-43D that are within pre-selected limits can be, for example, but not limited to, averaged or processed in any other form. Inertial sensor packs 1070, 23, 29, 35 can include, but are not limited to including, sensors such as, for example, ST®microelectronics LSM330DLC, or any sensor supplying a 3D digital accelerometer and a 3D digital gyroscope, or further, any sensor that can measure gravity and body rates. Sensor data can be subject to processing, for example, but not limited to, filtering to improve control of transporter 120 (FIG. 1).

[0034] Continuing to still further refer primarily to FIGs. 2C/2D, power base 160 (FIG. 1) can include sensors such as, for example, but not limited to, ALLEGRO™ ACS709 current sensor IC, or any sensor that can sense at least a pre-selected number of motor currents, has bi-directional sensing, has user-selectable overcurrent fault setting, and can handle peak currents above a pre-selected fault limit. Cluster position sensors A/B 55/71, seat position sensors A/B 67/81, and manual brake release sensors A/B 61/75 can include but are not limited to including, Hall sensors.

[0035] Referring now primarily to FIG. 3A, in some configurations, power base processors 100 (FIG. 4) can support at least one operating mode, and active controller 64A can enable navigation among modes. The at least one operating mode can include, but is not limited to including, standard mode 201 (described with respect to FIG. 1), enhanced mode 217 (described with respect to FIG. 1), balance mode 219 (described with respect to FIG. 1), stair mode 215 (described with respect to FIG. 1), docking mode 203 (described with respect to FIG. 1), and remote mode 205 (described with respect to FIG. 1). Service modes can include, but are not limited to including, recovery mode 161, failsafe mode 167 (FIG. 3B), update mode 169 (FIG. 3B), self-test mode 171 (FIG. 3B), calibrate mode 163, power on mode 207 (FIG. 3B), and power off mode 209 (FIG. 3B). With respect to recovery mode 161, if a power off

occurs when transporter 120 (FIG. 1) is not in one of a pre-selected set of modes, such as for example, but not limited to, standard mode 201, docking mode 203, or remote mode 205, transporter 120 (FIG. 1) can enter recovery mode 161 to safely reposition transporter 120 (FIG. 1) into the driving position of standard mode 201. During recovery mode 161, power base processors 100 (FIG. 4) can select certain components to activate such as, for example, seat motor drive A/B 25/37 (FIG. 2C/2D) and cluster motor drive A/B 1050/27 (FIG. 2C/2D). Functionality can be limited to, for example, controlling the position of seat 105 (FIG. 1) and cluster 121 (FIG. 1).

[0036] Referring now primarily to FIG. 3B, transporter 120 (FIG. 1) can be transitioned into failsafe mode 167 when transporter 120 (FIG. 1) can no longer effectively operate. In failsafe mode 167, at least some active operations can be halted to protect against potentially erroneous or uncontrolled motion. Transporter 120 (FIG. 1) can be transitioned from standard mode 201 (FIG. 3A) to update mode 169 to, for example, but not limited to, enable communications with external applications 140 (FIG. 4) that can be executing external to power base 160 (FIG. 1). Transporter 120 (FIG. 1) can be transitioned to self-test mode 171 when transporter 120 (FIG. 1) is first powered. In self-test mode 171, electronics in power base 160 (FIG. 1) can perform self diagnostics and can synchronize with one another. In some configurations, system self-tests can be performed to check the integrity of systems that are not readily testable during normal operation, for example, memory integrity verification tests and disable circuitry tests. While in self-test mode 171, operational functions can be disabled.

[0037] Referring now primarily to FIG. 4, transporter control system 200A can include, but is not limited to including, at least one power base processor 100 and at least one power source controller 11 that can bi-directionally communicate over serial bus 143 with system serial bus messaging system 130F. System serial bus messaging 130F can bi-directionally communicate with I/O interface 130G, external communications 130D, UCP 130, and UCP assist 145. UCP 130 and UCP assist 145 can access peripherals, processors, and controllers through interface modules that can include, but are not limited to including, input/output (I/O) interface 130G, system serial bus (SSB) messaging interface 130F, and external communications interface 130D. In some configurations, I/O interface 130G can transmit/receive messages to/from, for example, but not limited to, at least one of audio interface 150, electronic interface 149, manual interface 153, and visual interface 151. Audio interface 150 can transmit data from, for example, UCP 130 to audio devices such as, for example, speakers that can project, for example, alerts when transporter 120 (FIG. 1) requires attention. Electronic interface 149 can transmit/receive messages to/from, for example, but not limited to, sensors 147. Sensors 147 can include, but are not limited to including, time-of-flight cameras and other sensors. Manual interface 153 can transmit/receive messages to/from, for example, but not limited to, joystick 133 (FIG. 5B) and/or switches/buttons 141/B/C (FIG. 6), and/or information lighting such as LED lights, and/or display 137 (FIG. 5A) having, for example, a touch screen. UCP 130 and UCP assist 145 can transmit/receive information to/from I/O interface 130G, system serial bus messaging 130F, external communications 130D, and each other.

[0038] Continuing to refer primarily to FIG. 4, system serial bus interface 130F can enable

communications among UCP 130, UCP assist 145, power base processors (PBPs) 100 (also shown, for example, as processor A1 43A (FIG. 2C), processor A2 43B (FIG. 2C), processor B1 43C (FIG. 2D), and processor B2 43D (FIG. 2D)), and power source controllers 11 (also shown, for example, as power source controller A 11A (FIG. 2B) and power source controller B 11B (FIG. 2B)). Messages described herein can be exchanged among UCP 130, UCP assist 145, and PBPs 100 using, for example, but not limited to, system serial bus 143. External communications interface 130D can enable communications among, for example, UCP 130, UCP assist 145, and external applications 140 using wireless communications 144 such as, for example, but not limited to, BLUETOOTH® technology. UCP 130 and UCP assist 145 can transmit/receive messages to/from sensors 147 that can be used to enable automatic and/or semi-automatic control of transporter 120 (FIG. 1).

[0039] Referring now primarily to FIGs. 5A, 5B, and 6, switches and buttons 141A/B/C (FIG. 6) associated with transporter 120 (FIG. 1) can generate, upon activation, signals to I/O interface 130G (FIG. 4). The signals can be decoded and debounced by, for example, UCP 130 (FIG. 4) and/or PBPs 100 (FIG. 4). Examples of functions that can be enabled by switches/buttons 141A/B/C (FIG. 6) can include, but are not limited to including, height of seat 105 (FIG. 1), lean of seat 105 (FIG. 1), mode selection, drive setting menu selection, disabling joystick 133 (FIG. 5B), selection confirmation, power off request, alarm status acknowledgement, and horn actuation. An alert, such as a flashing icon, can be provided to bring a condition to the attention of the user. The conditions can include, but are not limited to including, low battery, service required, temperature out of range, parking brake manually overridden that could be inhibiting a user-requested a power off, and a critical fault, warning, or alert. Switches/buttons 141A/B/C (FIG. 6) can have functionality that is context-dependant and can have secondary functionality that occurs if, for example, switches/button 141A/B/C (FIG. 6) are depressed for a certain period of time. Certain switches/buttons 141A/B/C (FIG. 6) can be disabled if, for example, a mode change occurs and/or the battery charger is connected. When joystick 133 (FIG. 5B) is disabled, certain other functions can be disabled such as, for example, but not limited to, mode selection, drive menu selection, and adjustments to seat 105 (FIG. 1). Disabled switches/buttons 141A/B/C (FIG. 6) can be re-enabled under certain conditions such as, for example, when associated switches/buttons 141A/B/C (FIG. 6) are released. In some configurations, button 141C (FIG. 6) can provide a way to indicate power on, and can also provide an indication of device status and/or a means to acknowledge device status. In some configurations, button 141B (FIG. 6) can provide, for example, a hazard flasher and/or a power on flasher. In some configurations, button 141A (FIG. 6) can provide a means to enable a horn and/or a confirmation of a selection.

[0040] Referring now primarily to FIG. 6A, UCD holder 133A can house manual and visual interfaces such as, for example, joystick 133 (FIG. 5B), display 137 (FIG. 5A), and associated electronics. Connector 133C (FIG. 6B2) can allow connection to UCP assist 145 (FIG. 4). In some configurations, UCP assist holder 145A can be attached to visual/manual interface holder 145C tool-lessly. UCP assist holder 145A can house UCP assist 145 that can include sensor 147A (FIG. 12A). Sensor 147A (FIG. 12A) can include, but is not limited to including, an OPT8241 time-of-flight sensor from TEXAS INSTRUMENTS®, or any device that can provide a

three-dimensional location of the data sensed by sensor 147A (FIG. 12A). UCP assist holder 145A and connector 133C (FIG. 6B2) can be located anywhere on transporter 120 (FIG. 1) and may not be limited to being mounted on visual/manual interface holder 145C.

[0041] Referring now primarily to FIGs. 6B1 and 6B2, manual/visual interface holder 145C can include, but is not limited to including, visual interface viewing window 137A (FIG. 6B1) and manual interface mounting cavity 133B (FIG. 6B1) available on first side 133E (FIG. 6B1) of manual/visual interface holder 145C. Connector 133C (FIG. 6B2) can be provided on second side 133D (FIG. 6B2) of manual/visual interface holder 145C. Any of viewing window 137A (FIG. 6B1), manual interface mounting cavity 133B (FIG. 6B1), and connector 133C (FIG. 6B2) can be located on any part of manual/visual interface holder 145C, or can be absent altogether. Manual/visual interface holder 145C, visual interface viewing window 137A (FIG. 6B1), manual interface mounting cavity 133B (FIG. 6B1), and connector 133C (FIG. 6B2) can be any size. Manual/visual interface holder 145C can be constructed of any material suitable for mounting visual interface viewing window 137A (FIG. 6B1), manual interface mounting cavity 133B (FIG. 6B1), and connector 133C (FIG. 6B2). Angle 145M can be associated with various orientations of UCD holder 133A and thus can be various values. UCD holder 133A can have a fixed orientation or can be hinged.

[0042] Referring now primarily to FIGs. 6B3 and 6B4, UCP assist holder 145A can include, but is not limited to including, filter cavity 136G and lens cavity 136F providing visibility to, for example, but not limited to, a time-of-flight sensor optical filter and lens such as, for example, but not limited to, OPT8241 3D time-of-flight sensor by TEXAS INSTRUMENTS®. UCP assist holder 145A can be any shape and size and can be constructed of any material, depending on the mounting position on transporter 120 and the sensors, processors, and power supply, for example, provided within UCP assist holder 145A. Rounded edges on cavities 136G and 136F, on casing 136E, as well as on holder 145A can be replaced by any shape of edge. Cavity 136H can house controlling electronics.

[0043] Referring now primarily to FIGs. 6C1 and 6C2, connector 133C can include, but is not limited to including, connector lead 133G (FIG. 6C1) on connector first side 133H (FIG. 6C1) and connector pins 133F that can protrude from connector second side 133I (FIG. 6C2). Connector lead 133G (FIG. 6C1) and connector pins 133F can be any size and shape, and there can be any number of connector leads 133G (FIG. 6C1) and connector pins 133F. Further, there can be any number of connectors 133C.

[0044] Referring now primarily to FIGs. 6D1 and 6D2, mounting board 134J can include, but is not limited to including, pin holes 134D, mounting holes 134C, and alignment features 134B. Mounting board first side 134A can be identical to mounting board second side 134E, or mounting board first side 134A can have different features from mounting board second side 134E. Mounting holes 134C, pin holes 134D, and alignment features 134B can be any size and/or shape, and there can be any number of mounting holes 134C, pin holes 134D and alignment features 134B. Mounting board 134J can be used to mount connector 133C (FIGs. 6C1/6C2). In some configurations, mounting board 134J can include pin holes 134D that can

accommodate connector pins 133F (FIGs. 6C1/6C2). Mounting board 134J can be provided in multiple pieces and shapes to accommodate connector(s) 133C (FIGs. 6C1/6C2).

[0045] Referring now to FIGs. 6D3 and 6D4, connector pins 133F can be inserted into pin holes 134D to mount connector 133C on mounting board 134J. Connector leads 133G can project from mounting board first side 134A (FIG. 6D3), and connector pins 133F can protrude from mounting board second side 134E (FIG. 6D4). Connector 133C can be positioned anywhere on mounting board 134J, and can cross multiple mounting boards 134J. Multiple of connectors 133C can be mounted on mounting board 134J.

[0046] Referring now primarily to FIGs. 6D5 and 6D6, in some configurations, second configuration connector 139D can be mounted on mounting board 134J (FIG. 6D1) to mount UCP assist holder 145A (FIG. 6A). Arched lead 139A on second configuration connector first side 139E can form cavity 139B into which mated connectors (not shown) from UCP assist holder 145A (FIG. 6A) can be inserted. Second configuration connector second side 139F can include protruding of second configuration connector pins 139C that can be inserted into mounting board 134J (FIG. 6D1).

[0047] Referring now primarily to FIG. 6E, transporter 120 (FIG. 1) can be fitted with any number of sensors 147 (FIG. 4) in any configuration. In some configurations, some of sensors 147 (FIG. 4) can be mounted on transporter rear 122 to accomplish specific goals, for example, backup safety. Stereo color cameras/illumination 122A, ultrasonic beam range finder 122B, time-of-flight cameras 122D/122E, and single point LIDAR sensors 122F can be mounted, for example, but not limited to, to cooperatively sense obstacles behind transporter 120 (FIG. 1). PBPs 100 (FIG. 4) and/or UCP assist 145 (FIG. 4) can receive messages that can include information from the cameras and sensors, and can enable transporter 120 (FIG. 1) to react to what might be happening out of the view of the user. Transporter 120 (FIG. 1) can also include reflectors 122C that can be optionally fitted with further sensors. Stereo color cameras/illumination 122A can also be used as taillights. Other types of cameras and sensors can be mounted on transporter 120 (FIG. 1). Information from the cameras and sensors can be used to enable a smooth transition to balance mode 219 (FIG. 3A) by providing information to UCP assist 145 (FIG. 4) to enable UCP assist 145 (FIG. 4) to locate any obstacles that might impeded the transition to balance mode 219 (FIG. 3A).

[0048] Referring now primarily to FIG. 7A, SSB 143 (FIG. 4) can provide communications through use of, for example, a CAN Bus protocol. Devices connected to SSB 143 (FIG. 4) can be programmed to respond/listen to specific messages received, processed, and transmitted by SSB messaging 130F (FIG. 4). Messages can include packets, which can include, but are not limited to including, eight bytes of data and a CAN Bus device identification that can identify the source of the packet. Devices receiving CAN bus packets can ignore invalid CAN bus packets. When an invalid CAN bus packet is received, the received device can take alternative measures, depending on, for example, the current mode of transporter 120 (FIG. 1), the previous CAN bus messages, and the receiving device. The alternate measures can, for example, maintain stability of transporter 120 (FIG. 1). The bus master of SSB 143 (FIG. 4)

can transmit master sync packet 901 to establish a bus alive sequence on a frame basis and synchronize the time base.

[0049] Referring now primarily to FIG. 7B, user control panel packet #1 903 (FIG. 7A) can include eight bytes and can have, for example, packet format 701. Packet format 701 can include, but is not limited to including, status 701A, error device identification 701B, mode requested 701C, control out byte 701D, commanded velocity 701E, commanded turn rate 701F, seat control byte 701G, and system data 701H. Status 701A can include, but is not limited to including, possibilities such as, for example, self test in progress, device okay, non-fatal device failure (data OK), and fatal device failure in which receiving devices can ignore the data in the packet. If UCP 130, for example, receives a device failure status, UCP 130 can post an error to, for example, a graphical user interface (GUI) on display 137 (FIG. 5A). Error device ID 701B can include the logical ID of the device for which received communications has been determined to be erroneous. Error device ID 701B can be set to zero when no errors are received.

[0050] Referring now primarily to FIG. 7C, mode requested code 701C (FIG. 7B) can be defined such that a single bit error may not indicate another valid mode. For example, mode codes can include, but are not limited to including, self-test, standard, enhanced, stair, balance, docking, remote, calibration, update, power off, power on, fail safe, recovery, flasher, door, mobile storage, static storage/charging, rest room, elevator, and enhanced stair, the meanings of which are discussed herein. Mode requested code 701C can indicate if the mode being requested should be processed to (1) either maintain the current mode or execute an allowed mode change or (2) enable situation-dependent processing. In some configurations, special situations can require automatic control of transporter 120 (FIG. 1). For example, transporter 120 (FIG. 1) can transition from stair mode 215 (FIG. 3A) automatically to enhanced mode 217 (FIG. 3A) when transporter 120 (FIG. 1) has reached a top landing of a staircase. In some configurations, PBPs 100 (FIG. 4) and/or UCP assist 145 (FIG. 4) can, for example, but not limited to, modify the response of PBPs 100 (FIG. 4) to commands from joystick 133 (FIG. 1), for example, by setting transporter 120 (FIG. 1) to a particular mode. In some configurations, transporter 120 (FIG. 1) can automatically be set to a slow driving mode when transporter 120 (FIG. 1) is transitioned out of stair mode 215 (FIG. 3A). In some configurations, when transporter 120 (FIG. 1) transitions from stair mode 215 (FIG. 3A) automatically to enhanced mode 217 (FIG. 3A), joystick 133 (FIG. 1) can be disabled. When a mode is selected by any of UCP assist 145 (FIG. 4), UCP 130 (FIG. 4) through, for example, but not limited to, user entry, and/or PBPs 100 (FIG. 4), mode availability can be determined based at least in part on current operating conditions.

[0051] Continuing to refer primarily to FIG. 7C, in some configurations, if a transition is not allowed to a user-selected mode from the current mode, the user can be alerted. Certain modes and mode transitions can require user notification and possibly user assistance. For example, adjustments to seat 105 (FIG. 1) can be needed when positioning transporter 120 (FIG. 1) for a determination of the center of gravity of transporter 120 (FIG. 1) along with the load on transporter 120 (FIG. 1). The user can be prompted to perform specific operations

based on the current mode and/or the mode to which the transition can occur. In some configurations, transporter 120 (FIG. 1) can be configured for, for example, but not limited to, fast, medium, medium dampened, or slow speed templates. The speed of transporter 120 (FIG. 1) can be modified by using, for example, speed template 700 (FIG. 8) relating output 703 (FIG. 8) (and wheel commands) to joystick displacement 702 (FIG. 8).

[0052] Referring now to FIG. 7D, control out byte 701D (FIG. 7B) can include, but is not limited to including, bit definitions such as, for example, but not limited to, OK to power down 801A, drive selection 801B, emergency power off request 801C, calibration state 801D, mode restriction 801E, user training 801F, and joystick centered 801G. In some configurations, OK to power down 801A can be defined to be zero if power down is not currently allowed, and drive selection 801B can be defined to specify motor drive 1 (bit 6 = 0) or motor drive 2 (bit 6 = 1). In some configurations, emergency power off request 801C can be defined to indicate if an emergency power off request is normal (bit 5 = 0), or an emergency power off request sequence is in process (bit 5 = 1), and calibration state 801D can be defined to indicate a request for user calibration (bit 4 = 1). In some configurations, mode restriction 801E can be defined to indicate whether or not there are restrictions for entering a particular mode. If the mode can be entered without restriction, bit 3 can be zero. If there are restrictions to entering a mode, for example, but not limited to, balance-critical modes can require certain restrictions to maintain the safety of the passenger of transporter 120 (FIG. 1), bit 3 can be one. User training 801F can be defined to indicate if user training is possible (bit 2 = 1), or not (bit 2 = 0), and joystick centered 801G can be defined to indicate if joystick 133 (FIG. 1) is centered (bits 0-1 = 2), or not (bits 0-1 = 1).

[0053] Referring again primarily to FIG. 7B, commanded velocity 701E can be, for example, a value representing forward or reverse speed. Forward velocity can be a positive value and reverse velocity can be a negative value, for example. Commanded turn rate 701F can be a value representing a left or right commanded turn rate. A left turn can be a positive value and a right turn can be a negative value. The value can represent the differential velocity between the left and right of wheels 101/102 (FIG. 1) equivalently scaled to commanded velocity 701E.

[0054] Referring again primarily to FIG. 7D, joystick 133 (FIG. 1) can have multiple redundant hardware inputs. Signals such as, for example, commanded velocity 701E (FIG. 7B), commanded turn rate 701F (FIG. 7B), and joystick-centered 801G can be received and processed. Commanded velocity 701E (FIG. 7B) and commanded turn rate 701F (FIG. 7B) can be determined from a first of the multiple hardware inputs, and joystick-centered 801G can be determined from a second of the hardware inputs. Values of joystick-centered 801G can indicate when a non-zero of commanded velocity 701E (FIG. 7B) and a non-zero of commanded turn rate 701F (FIG. 7B) are valid. Fault conditions for joystick 133 (FIG. 1) in, for example, the X and Y directions can be detected. For example, each axis of joystick 133 (FIG. 1) can be associated with dual sensors. Each sensor pair input (X (commanded velocity 701E (FIG. 7B)) and Y (command turn rate 701F (FIG. 7B)) can be associated with an independent A/D converter, each with a voltage reference channel check input. In some configurations, commanded velocity 701E (FIG. 7B) and commanded turn rate 701F (FIG. 7B) can be held to

zero by the secondary input to avoid mismatch. If joystick-centered 801G is within a minimum deadband, or joystick 133 (FIG. 1) is faulted, joystick 133 (FIG. 1) can be indicated as centered. A deadband can indicate the amount of displacement of joystick 133 (FIG. 1) that can occur before a non-zero output from joystick 133 (FIG. 1) can appear. The deadband range can set the zero reference region to include an electrical center position that can be, for example, but not limited to, 45% to 55% of the defined signal range.

[0055] Referring now primarily to FIG. 7E, seat control byte 701G (FIG. 7B) can convey seat adjustment commands. Frame lean command 921 can include values such as, for example, invalid, lean forward, lean rearward, and idle. Seat height command 923 can include values such as, for example, invalid, lower seat down, raise seat up, and idle.

[0056] Referring again to FIG. 7A, user control packets 905 can include header, message ID, and data for messages traveling primarily to and from external applications 140 (FIG. 4) through, for example, but not limited to, a BLUETOOTH® connection. PBPs packet 907 can include data originated by PBPs 100 (FIG. 4) and destined for PSCs 11 (FIG. 4). PBP A1 43A (FIG. 2C), for example, can be designated the master of SSB 143 (FIG. 4), and PBP B1 43C (FIG. 2D), for example, can be designated as the secondary master of SSB 143 (FIG. 4) if PBP A1 43A (FIG. 2C) is no longer transmitting on the bus. The master of SSB 143 (FIG. 4) can transmit master sync packet 901 at a periodic rate, for example, but not limited to, every 20 ms +/- 1%. Devices communicating using SSB 143 (FIG. 4) can synchronize the transmitting of messages to the beginning of master sync packet 901.

[0057] Referring now primarily to FIG. 8, joystick 133 (FIG. 1) can be configured to have different transfer functions to be used under different conditions according to, for example, the abilities of the user. Speed template (transfer function) 700 shows an exemplary relationship between physical displacement 702 of joystick 133 (FIG. 1) and output 703 of joystick 133 (FIG. 1) after transfer function processing. Forward and reverse travel of joystick 133 (FIG. 1) can be interpreted as forward longitudinal requests and reverse longitudinal requests, respectively, as viewed from a user in seat 105 (FIG. 1), and can be equivalent to commanded velocity 701E (FIG. 7), the X request value. Left and right travel of joystick 133 (FIG. 1) can be interpreted as left turn requests and right turn requests, respectively, as viewed from a user in seat 105 (FIG. 1), and can be equivalent to commanded turn rate 701F, the Y request value. Joystick output 703 can be modified during certain conditions such as, for example, but not limited to, battery voltage conditions, height of seat 105 (FIG. 1), mode, failed conditions of joystick 133 (FIG. 1), and when speed modification is requested by PBPs 100 (FIG. 4). Joystick output 703 can be ignored and joystick 133 (FIG. 1) can be considered as centered, for example, but not limited to, when a mode change occurs, or while in update mode 169 (FIG. 3B), or when the battery charger is connected, or when in stair mode, or when joystick 133 (FIG. 1) is disabled, or under certain fault conditions.

[0058] Continuing to refer primarily to FIG. 8, transporter 120 (FIG. 1) can be configured to suit a particular user. In some configurations, transporter 120 (FIG. 1) can be tailored to user abilities, for example, by setting speed templates and mode restrictions. In some

configurations, transporter 120 (FIG. 1) can receive commands from external applications 140 (FIG. 4) executing on devices such as, for example, but not limited to, a cell phone, a computer tablet, and a personal computer. The commands can provide, for example, default and/or dynamically-determinable settings for configuration parameters. In some configurations, a user and/or an attendant can configure transporter 120 (FIG. 1).

[0059] Referring now to FIG. 9, any of UCP 130, UCP assist 145, and/or PBPs 100 can execute a power up and processing sequence when UCP 130 (FIG. 4) and/or PBPs 100 (FIG. 4) and/or UCP assist 145 (FIG. 4) receive powered on indication 1017. Power on processing 1005 can include, but is not limited to including, integrity checks that can be performed on, for example, stored data and various indicators. Memory tests can be performed, system and configuration parameters can be established in memory, and a readiness to operate can be indicated, for example, but not limited to, by lighted LEDs. Power on processing 1005 can be followed by transitioning to main loop processing 1001 in which sensor data 1003 can be received and processed, input messages 1027 can be received, and output messages 1013 can be generated periodically, for example, but not limited to, once every frame of SSB 143 (FIG. 4). Device data 1009 and communications information 1011 can be accessed. Device data 1009 can include, but is not limited to including, a device status that can, for example, display device diagnostic data. In some configurations, communications with external applications 140 (FIG. 4) can be provided to gather information such as, for example, but not limited to, application code version numbers, application code CRC values, and protocol map compatibility information. UCP 130 (FIG. 4), UCP assist 145 (FIG. 4), and/or PBPs 100 (FIG. 4) can execute a power down sequence upon receiving power off request 1015. Ongoing activities such as, for example, but not limited to, data logging, and receiving data from, for example, switches/ buttons 141A/B/C (FIG. 6), and joysticks 133 (FIG. 1), can be disabled and brought to a consistent close. Configuration, usage, service, security, and other information can be accumulated and stored during power off processing 1007.

[0060] Referring now to FIG. 10, UCP assist 145 can provide enhanced functionality to a user, for example, but not limited to, assisting a user in avoiding obstacles, traversing doors, traversing stairs, traveling on elevators, and parking/transporting transporter 120 (FIG. 1). In general, UCP assist 145 can receive user input (for example UI data 633) and/or input from PBPs 100 (FIG. 4) through, for example, but not limited to, messages from user interface devices and sensors 147. UCP assist 145 can further receive sensor input through, for example, but not limited to sensor processing systems 661. UI data 633 and output from sensor processing systems 661, for example, can inform command processor 601 to invoke the mode that has been automatically or manually selected. Command processor 601 can pass UI data 633 and output from sensor processing systems 661 to a processor that can enable the invoked mode. The processor can generate movement commands 630 at least based on previous movement commands 630, UI data 633, and output from sensor processing systems 661.

[0061] Continuing to refer to FIG. 10, UCP assist 145 can include, but is not limited to including, command processor 601, movement processor 603, simultaneous location and

mapping (SLAM) processor 609, point cloud library (PCL) processor 611, geometry processor 613, and obstacle processor 607. Command processor 601 can receive user interface (UI) data 633 from the message bus. UI data 633 can include, but is not limited to including, signals from, for example, joystick 133 (FIG. 1) providing an indication of a desired movement direction and speed of transporter 120 (FIG. 1). UI data 633 can also include selections such as an alternate mode into which transporter 120 (FIG. 1) could be transitioned. In some configurations, in addition to the modes described with respect to FIGs. 3A/3B, UCP assist 145 can process mode selections such as, but not limited to, door mode 605A, rest room mode 605B, enhanced stair mode 605C, elevator mode 605D, mobile park mode 605E, and static storage/charging mode 605F. Any of these modes can include a move-to-position mode, or the user can direct transporter 120 (FIG. 1) to move to a certain position. Message bus 54 can receive control information in the form of UI data 633 for transporter 120 (FIG. 1), and can receive a result of the processing done by UCP assist 145 in the form of commands such as movement commands 630 that can include, but are not limited to including, speed and direction. Movement commands 630 can be provided, by message bus 54, to PBPs 100 (FIG. 4) which can transmit this information to wheel motor drives 19/21/31/33 (FIGs. 2C/D) and cluster motor drives 1050/27 (FIGs. 2C/D). Movement commands 630 can be determined by movement processor 603 based on information provided by the mode-specific processors. Mode-specific processors can determine mode-dependent data 657, among other things, based on information provided through sensor-handling processors 661.

[0062] Continuing to refer primarily to FIG. 10, sensor-handling processors 661 can include, but are not limited to including, transporter geometry processor 613, PCL processor 611, SLAM processor 609, and obstacle processor 607. Movement processor 603 can provide movement commands 630 to the sensor-handling processors 661 to provide information necessary to determine future movements of transporter 120 (FIG. 1). Sensors 147 can provide environmental information 651 that can include, for example, but not limited to, obstacles 623 and geometric information about transporter 120 (FIG. 1). In some configurations, sensors 147 can include at least one time-of-flight sensor that can be mounted anywhere on transporter 120 (FIG. 1). There can be multiple of sensors 147 mounted on transporter 120 (FIG. 1). PCL processor 611 can gather and process environmental information 651, and can produce PCL data 655. The PCL, a group of code libraries for processing 2D/3D image data, can, for example, assist in processing environmental information 651. Other processing techniques can be used.

[0063] Continuing to refer primarily to FIG. 10, transporter geometry processor 613 can receive transporter geometry information 649 from sensors 147, can perform any processing necessary to prepare transporter geometry information 649 for use by the mode-dependent processors, and can provide the processed of transporter geometry information 649 to the mode-dependent processors. The geometry of transporter 120 (FIG. 1) can be used for, but is not limited to being used for, automatically determining whether or not transporter 120 (FIG. 1) can fit in and/or through a space such as, for example, a stairway and a door. SLAM processor 609 can determine navigation information 653 based on, for example, but not limited to, UI data 633, environmental information 651 and movement commands 630. Transporter 120

(FIG. 1) can travel in a path at least in part set out by navigation information 653. Obstacle processor 607 can locate obstacles 623 and distances 621 to obstacles 623. Obstacles 623 can include, but are not limited to including, doors, stairs, automobiles, and miscellaneous features in the vicinity of the path of transporter 120 (FIG. 1).

[0064] Referring now to FIGs. 11A1 and 11A2, method 650 for processing at least one obstacle 623 (FIG. 11B) while navigating transporter 120 (FIG. 1) can include, but is not limited to including, receiving 1151 (FIG. 11A1) at least one movement command 630 (FIG. 11B), receiving and segmenting 1153 (FIG. 11A1) PCL data 655 (FIG. 11B), identifying 1155 (FIG. 11A1) at least one plane within the segmented PCL data 655 (FIG. 11B), and identifying 1157 (FIG. 11A1) at least one obstacle 623 (FIG. 11B) within the at least one plane. Method 650 can further include determining 1159 (FIG. 11A1) at least one situation identifier 624 (FIG. 11B) based at least on the at least one obstacle, UI data 633 (FIG. 11B), and movement commands 630 (FIG. 11B), and determining 1161 (FIG. 11A1) distance 621 (FIG. 11B) between transporter 120 (FIG. 1) and at least one obstacle 623 (FIG. 11B) based at least on at least one situation identifier 624 (FIG. 11B). Method 650 can also include accessing 1163 (FIG. 11A1) at least one allowed command related to distance 621 (FIG. 11B), at least one obstacle 623 (FIG. 11B), and at least one situation identifier 624 (FIG. 11B). Method 650 can still further include accessing 1163 (FIG. 11A1) at least one automatic response to the at least one allowed command, mapping 1167 (FIG. 11A2) at least one movement command 630 (FIG. 11B) with one of the at least one allowed commands, and providing 1169 (FIG. 11A2) at least one movement command 630 (FIG. 11B) and the at least one automatic response associated with the mapped allowed command to the mode-dependent processors.

[0065] Continuing to refer to FIGs. 11A1 and 11A2, at least one obstacle 623 (FIG. 11B) can optionally include at least one stationary object and/or at least one moving object. Distance 621 (FIG. 11B) can optionally include a fixed amount and/or a dynamically-varying amount. At least one movement command 630 (FIG. 11B) can optionally include a follow command, at least one pass-the-at-least-one-obstacle command, a travel beside-the-at-least-one-obstacle command, and a do-not-follow-the-at-least-one obstacle command. Method 650 can optionally include storing obstacle data 623 (FIG. 11B), and allowing access to stored obstacle data, for example, stored in cloud storage 607G (FIG. 11B) and/or local storage 607H (FIG. 11B), by systems external to transporter 120 (FIG. 1). PCL data 655 (FIG. 11B) can optionally include sensor data 147 (FIG. 10). Method 650 can optionally include collecting sensor data 147 (FIG. 10) from at least one time-of-flight sensor mounted on transporter 120 (FIG. 1), analyzing sensor data 147 (FIG. 10) using a point cloud library (PCL), tracking the at least one moving object using simultaneous location and mapping (SLAM) with detection and tracking of moving objects (DATMO) based on the location of transporter 120 (FIG. 1), identifying the at least one plane within obstacle data 623 (FIG. 11B) using, for example, but not limited to, random sample consensus and a PCL library, and providing the at least one automatic response associated with the mapped allowed command to the mode-dependent processors. Method 650 can also optionally include receiving a resume command, and providing, following the resume command, at least one movement command 630 (FIG. 11B) and the at least one automatic response associated with the mapped allowed command to the mode-dependent

processors. The at least one automatic response can optionally include a speed control command.

[0066] Referring now to FIG. 11B, obstacle processor 607 for processing at least one obstacle 623 while navigating transporter 120 (FIG. 1) can include, but is not limited to including, nav/PCL data processor 607F receiving and segmenting PCL data 655 from PCL processor 611, identifying at least one plane within the segmented PCL data 655, and identifying at least one obstacle 623 within the at least one plane. Obstacle processor 607 can further include distance processor 607E determining at least one situation identifier 624 based at least on UI data 633, at least one movement command 630, and at least one obstacle 623. Distance processor 607E can determine distance 621 between transporter 120 (FIG. 1) and at least one obstacle 623 based at least on at least one situation identifier 624. Moving object processor 607D and/or stationary object processor 607C can access at least one allowed command related to distance 621, at least one obstacle 623, and at least one situation identifier 624. Moving object processor 607D and/or stationary object processor 607C can access at least one automatic response, from automatic response list 627, associated with the at least one allowed command. Moving object processor 607D and/or stationary object processor 607C can access at least one movement command 630 including, for example, speed/signal command and direction command/signal, and map at least one movement command 630 with one of the at least one allowed commands. Moving object processor 607D and/or stationary object processor 607C can provide at least one movement command 630 and the at least one automatic response associated with the mapped allowed command to the mode-dependent processors.

[0067] Continuing to refer to FIG. 11B, stationary object processor 607C can optionally perform any special processing necessary when encountering at least one stationary object, and moving object processor 607D can optionally perform any special processing necessary when encountering at least one moving object. Distance processor 607E can optionally process distance 621 that can be a fixed and/or a dynamically-varying amount. At least one movement command 630 can optionally include a follow command, a pass command, a travel-beside command, a move-to-position command, and a do-not-follow command. Nav/PCL processor 607F can optionally store obstacles 623, for example, but not limited to, in local storage 607H and/or on storage cloud 607G, and can allow access to the stored obstacles 623 by systems external to transporter 120 (FIG. 1) such as, for example, but not limited to, external applications 140 (FIG. 4). PCL processor 611 can optionally collect sensor data 147 (FIG. 10) from at least one time-of-flight camera mounted on transporter 120 (FIG. 1), and can analyze sensor data 147 (FIG. 10) using a point cloud library (PCL) to yield PCL data 655. Moving object processor 607D can optionally track the at least one moving object using navigation information 653 collected by simultaneous location and mapping (SLAM) processor 609 based on the location of transporter 120 (FIG. 1), identify the at least one plane using, for example, but not limited to, random sample consensus and a PCL library, and can provide at least one movement command 630 based on the at least one automatic response associated with the mapped allowed command to the mode-dependent processors. Obstacle processor 607 can optionally receive a resume command, and provide, following the resume command,

at least one movement command 630 based on the at least one automatic response associated with the mapped allowed command to the mode-dependent processors. The at least one automatic response can optionally include a speed control command. For example, if joystick 133 (FIG. 1) indicates a direction that could position transporter 120 (FIG. 1) in a collision course with obstacle 623, such as, for example, a wall, the at least one automatic response can include speed control to protect transporter 120 (FIG. 1) from a collision. The at least one automatic response could be overridden by a contrary user command, for example, joystick 133 (FIG. 1) could be released and movement of transporter 120 (FIG. 1) could be halted. Joystick 133 (FIG. 1) could then be re-engaged to restart movement of transporter 120 (FIG. 1) towards obstacle 623.

[0068] Referring now primarily to FIGs. 12A-12D, environmental information 651 (FIG. 10) can be received from sensors 147 (FIG. 10). Any of PBPs 100 (FIG. 4), UCP 130 (FIG. 4), and/or UCP assist 145 (FIG. 10) can process environmental information 651 (FIG. 10). In some configurations, PCL processor 611 (FIG. 10) can process environmental information 651 (FIG. 10) using, for example, and depending upon sensor 147 (FIG. 10), point cloud library (PCL) functions. As transporter 120 (FIG. 1) moves along travel path 2001 (FIG. 12D) around potential obstacles 2001A, sensors 147 (FIG. 10) can detect a cloud of points from, for example, and depending upon sensor 147 (FIG. 10), box 2005 (FIGs. 12C-12D) that can include data that could take the shape of frustum 2003 (FIGs. 12B-12D). A sample consensus method, for example, but not limited to, the random sample consensus method, from, for example, but not limited to, the PCL, can be used to find a plane among the cloud of points. Any of UCP 130 (FIG. 4), UCP assist 145 (FIG. 10), and PBPs 100 (FIG. 4) can create a projected cloud and can determine point cloud inliers, and from these, determine a centroid of the projected cloud. Central reference point 148 can be used to determine the location of environmental conditions with respect to transporter 120. For example, whether transporter 120 is moving towards or away from an obstacle, or where a door hinge is with respect to transporter 120 can be determined based on the location of central reference point 148. Sensors 147 (FIG. 10) can include, for example, time-of-flight sensor 147A.

[0069] Referring now primarily to FIG. 13A, method 750 for enabling transporter 120 (FIG. 1) to navigate stairs can include, but is not limited to including, receiving 1251 at least one stair command, and receiving 1253 environmental information 651 (FIG. 10) from sensors 147 (FIG. 10) mounted on transporter 120 (FIG. 1) and through obstacle processor 607 (FIG. 10). Method 750 can further include locating 1255, based on environmental information 651 (FIG. 10), at least one of staircases 643 (FIG. 13B) within environmental information 651 (FIG. 10), and receiving 1257 selection of selected staircase 643A (FIG. 13B) from the at least one of staircases 643 (FIG. 13B). Method 750 can still further include measuring 1259 at least one characteristic 645 (FIG. 13B) of selected staircase 643A (FIG. 13B), and locating 1261, based on environmental information 651 (FIG. 13B), obstacles 623 (FIG. 13B), if any, on selected staircase 643A (FIG. 13B). Method 750 can also include locating 1263, based on environmental information 651 (FIG. 13B), a last stair of selected staircase 643A (FIG. 13B), and providing 1265 movement commands 630 (FIG. 13B) to move transporter 120 (FIG. 1) on selected staircase 643A (FIG. 13B) based on the measured at least one characteristic 645

(FIG. 13B), the last stair, and obstacles 623 (FIG. 13B), if any. If 1267 the last stair has not been reached, method 750 can continue providing movement commands 630 (FIG. 13B) to move transporter 120 (FIG. 1). Method 750 can optionally include locating at least one of staircases 643 (FIG. 13B) based on GPS data and building and saving a map of selected staircase 643A (FIG. 13B) using, for example, but not limited to, SLAM. Method 750 can also optionally include accessing geometry 649 (FIG. 13B) of transporter 120 (FIG. 1), comparing geometry 649 (FIG. 13B) to at least one of characteristics 645 (FIG. 13B) of selected staircase 643A (FIG. 13B), and modifying the step of navigating based on the step of comparing. At least one of characteristics 645 (FIG. 13B) can optionally include the height of at least one riser of selected staircase 643A (FIG. 13B), the surface texture of the at least one riser, and the surface temperature of the at least one riser. Method 750 can optionally include generating an alert if the surface temperature falls outside of a threshold range and the surface texture falls outside of a traction set. The threshold range can optionally include temperatures below 33°F. The traction set can optionally include a carpet texture. Method 750 can further include determining, based on environmental information 651 (FIG. 13B), the topography of an area surrounding selected staircase 643A (FIG. 13B), and generating an alert if the topography is not flat. Method 750 can still further optionally include accessing a set of extreme circumstances.

[0070] Referring now primarily to FIG. 13B, automated navigation of stairs can be enabled by stair processor 605C for enabling transporter 120 (FIG. 1) to navigate stairs. Sensors 147 (FIG. 10) on transporter 120 (FIG. 1) can determine if any environmental information 651 (FIG. 10) includes at least one staircase 643. In conjunction with any automatic determination of a location of at least one staircase 643, UI data 633 can include the selection of stair mode 215 (FIG. 3A) which can invoke an automatic, semi-automatic, or semi-manual stair-climbing process. Either automatic location of at least one staircase 643 or reception of UI data 633 can invoke stair processor 605C for enhanced stair navigation functions. Stair processor 605C can receive data from obstacle processor 607 such as, for example, at least one obstacle 623, distance 621 to at least one obstacle 623, situation 624, navigation information 653, and geometry information 649 for transporter 120 (FIG. 1). Navigation information can include, but is not limited to including, a possible path for transporter 120 (FIG. 1) to traverse. At least one obstacle 623 can include, among other obstacles, at least one staircase 643. Stair processor 605C can locate at least one staircase 643, and can either automatically or otherwise determine selected staircase 643A based on, for example, but not limited to, navigation information 653 and/or UI data 633 and/or transporter geometry information 649. Characteristics 645 of selected staircase 643A, such as, for example, riser information, can be used to determine a first stair and distance to next stair 640. Stair processor 605C can determine movement commands 630 of transporter 120 (FIG. 1) based on, for example, but not limited to, characteristics 645, distance 621, and navigation information 647. Movement processor 603 can move transporter 120 (FIG. 1) based on movement commands 630, and distance to next stair 640, and can transfer control to sensor processing 661 after a stair from selected staircase 643A has been traversed. Sensor processing 661 can either proceed with navigating selected staircase 643A or can continue following the path set out by navigation information 653, depending upon whether transporter 120 (FIG. 1) has completed traversing

selected staircase 643A. While transporter 120 (FIG. 1) is traversing selected staircase 643A, obstacle processor 607 can detect obstacles 623 on selected staircase 643A and stair processor 605C can provide movement commands 630 to avoid obstacles 623. Locations of obstacles 623 can be stored for future use locally to transporter 120 (FIG. 1) and/or external to transporter 120 (FIG. 1).

[0071] Continuing to refer primarily to FIG. 13B, stair processor 605C can include, but is not limited to including, staircase processor 641B receiving at least one stair command included in UI data 633, and staircase locator 641A receiving environmental information 651 (FIG. 10) from sensors 147 (FIG. 10) mounted on transporter 120 (FIG. 1) through obstacle processor 607 (FIG. 10). Staircase locator 641A can further locate, based on environmental information 651 (FIG. 10), at least one of staircases 643 within environmental information 651 (FIG. 10), and can receive the choice of selected staircase 643A from at least one of staircases 643. Selected staircase 643A can be stored in storage 643B for possible future use. Stair characteristics processor 641C can measure at least one of characteristics 645 of selected staircase 643A, and can locate, based on environmental information 651, at least one obstacle 623, if any, on selected staircase 643A. Stair movement processor 641D can locate, based on environmental information 651, a last stair of selected staircase 643A, and provide to movement processor 603 movement commands 630 for transporter 120 (FIG. 1) to move on selected staircase 643A based on the measured at least one of characteristics 645, the last stair, and at least one obstacle 623, if any. Staircase locator 641A can optionally locate at least one of staircases 643 based on GPS data, and can build and save a map of selected staircase 643A using SLAM. The map can be saved for use locally to transporter 120 (FIG. 1), and/or for use by other devices. Staircase processor 641B can optionally access geometry 649 of transporter 120 (FIG. 1), compare geometry 649 to at least one of characteristics 645 of selected staircase 643A, and can modify the navigation of transporter 120 (FIG. 1) based on the comparison. Staircase processor 641B can optionally generate an alert if the surface temperature of the risers of selected staircase 643A falls outside of a threshold range and the surface texture of selected staircase 643A falls outside of a traction set. Stair movement processor 641D can optionally determine, based on environmental information 651 (FIG. 10), the topography of an area surrounding selected staircase 643A, and can generate an alert if the topography is not flat. Stair movement processor 641D can optionally access a set of extreme circumstances.

[0072] Referring now primarily to FIGs. 14A1-14A2, method 850 for negotiating door 675 (FIG. 14B) while maneuvering transporter 120 (FIG. 1), where door 675 (FIG. 14B) can include a door swing, a hinge location, and a doorway, can include, but is not limited to including, receiving and segmenting 1351 (FIG. 14A1) environmental information 651 (FIG. 10) from sensors 147 (FIG. 10) mounted on transporter 120 (FIG. 1). Environmental information 651 (FIG. 10) can include geometry of transporter 120 (FIG. 1). Method 850 can include identifying 1353 (FIG. 14A1) at least one plane within the segmented sensor data, and identifying 1355 (FIG. 14A1) door 675 (FIG. 14B) within the at least one plane. Method 850 can further include measuring 1357 (FIG. 14A1) door 675 (FIG. 14B) to provide door measurements. Method 850 can also include determining 1361 (FIG. 14A1) the door swing. Method 850 can further include

providing 1363 (FIG. 14A2) at least one movement command 630 (FIG. 14B) to move transporter 120 (FIG. 1) for access to a handle of door 675 (FIG. 14B), if necessary, and providing 1365 (FIG. 14A2) at least one movement command 630 (FIG. 14B) to move transporter 120 (FIG. 1) away from door 675 (FIG. 14B), as door 675 (FIG. 14B) opens, by a distance based on the door measurements. If door 675 (FIG. 14B) swings in, method 850 can include providing at least one movement command to move transporter 120 (FIG. 1) against door 675 (FIG. 14B), thus positioning door 675 (FIG. 14B) for movement of transporter 120 (FIG. 1) through the doorway. Method 850 can also include providing 1367 (FIG. 14A2) at least one movement command 630 (FIG. 14B) to move transporter 120 (FIG. 1) forward through the doorway, transporter 120 (FIG. 1) maintaining door 675 (FIG. 14B) in an open position, if the door swing is towards transporter 120 (FIG. 1).

[0073] Referring now to FIG. 14B, sensor processing 661 can determine, through information from sensors 147 (FIG. 10), the hinge side of door 675, and the direction, angle, and distance of door. Movement processor 603 can generate commands to PBPs 100 (FIG. 4) such as start/stop turning left, start/stop turning right, start/stop moving forward, start/stop moving backwards, and can facilitate door mode 605A by stopping transporter 120 (FIG. 1), cancelling the goal that transporter 120 (FIG. 1) can be aiming to complete, and centering joystick 133 (FIG. 1). Door processor 671B can determine whether door 675 is, for example, push to open, pull to open, or a slider. Door processor 671B can determine the width of door 675 by determining the current position and orientation of transporter 120 (FIG. 1), and determining the x/y/z location of the door pivot point. If door processor 671B determines that the number of valid points in the image of door 675 derived from obstacles 623 and/or PCL data 655 (FIG. 10) is greater than a threshold, door processor 671B can determine the distance from transporter 120 (FIG. 1) to door 675. Door processor 671B can determine if door 675 is moving based on successive samples of PCL data 655 (FIG. 10) from sensor processor 661. In some configurations, door processor 671B can assume that a side of transporter 120 (FIG.1) is even with the handle side of door 675, and can use that assumption, along with the position of the door pivot point, to determine the width of door 675.

[0074] Continuing to refer primarily to FIG. 14B, if the movement of door 675 is towards transporter 120 (FIG. 1), door movement processor 671D can generate and provide movement commands 630 to movement processor 603 to move transporter 120 (FIG. 1) backward by a pre-determined or dynamically-determined percentage of the amount door 675 is moving. Movement processor 603 can provide movement commands 630 to UCP 130, and UCP 130 can accept GUI data 633A and provide GUI data 633A to movement processor 603. If door 675 is moving away from transporter 120 (FIG. 1), door movement processor 671D can generate movement commands 630 to direct transporter 120 (FIG. 1) to move forward by a pre-determined or dynamically-determined percentage of the amount that door 675 moves. The amount transporter 120 (FIG. 1) moves either forward or backward can be based on the width of door 675. Door processor 671B can locate the side of door 675 that provides the open/close function for door 675 based on the location of the door pivot point. Door processor 671B can determine the distance to the plane in front of sensors 147 (FIG. 4). Door movement processor 671D can generate movement commands 630 to direct transporter 120 (FIG. 1) to

move through door 675. Door movement processor 671D can wait a pre-selected amount of time for the move of transporter 120 (FIG. 1) to complete, and door movement processor 671D can generate movement commands 630 to adjust the location of transporter 120 (FIG. 1) based on the position of door 675. Door processor 671B can determine the door angle and the door pivot point. Door processor 671B can determine if door 675 is stationary, can determine if door 675 is moving, and can determine the direction door 675 is moving. When door mode 605A is complete, door movement processor 671D can generate movement commands 630 that can direct transporter 120 (FIG. 1) to discontinue movement.

[0075] Continuing to still further refer primarily to FIG. 14B, door mode 605A for negotiating door 675 while maneuvering transporter 120 (FIG. 1), where door 675 can include a door swing, a hinge location, and a doorway, can include, but is not limited to including, sensor processing 661 receiving and segmenting environmental information 651 from sensors 147 (FIG. 10) mounted on transporter 120 (FIG. 1), where environmental information 651 can include geometry 649 of transporter 120 (FIG. 1). Door mode 605A can also include door locator 671A identifying at least one plane within the segmented sensor data, and identifying door 675 within the at least one plane. Door processor 671B can include measuring door 675 to provide door measurements 645A. Door movement processor 671D can provide at least one movement command 630 to move transporter 120 (FIG. 1) away from door 675 if door measurements 645A are smaller than geometry 649 of transporter (FIG. 1). Door processor 671B can also include determining the door swing, and door movement processor 671D can provide at least one movement command 630 to move transporter 120 (FIG. 1) forward through the doorway. Transporter 120 (FIG. 1) can open door 675 and maintain door 675 in an open position if the door swing is away from transporter 120 (FIG. 1). Door movement processor 671D can provide at least one movement command 630 to move transporter 120 (FIG. 1) for access to a handle of door 675, and can provide at least one movement command 630 to move transporter 120 (FIG. 1) away from door 675, as door 675 opens, by a distance based on door measurements 645A. Door movement processor 671D can provide at least one movement command 630 to move transporter 120 (FIG. 1) forward through the doorway. Transporter 120 (FIG. 1) can maintain door 675 in an open position if the door swing is towards transporter 120 (FIG. 1).

[0076] Referring now to FIG. 15A, transporter 120 (FIG. 1) can automatically negotiate using rest room facilities. UCP assist 145 (FIG. 4) can automatically locate a door to a rest room, and to a rest room stall, if there are multiple doors, can automatically generate movement commands 630 (FIG. 15B) to move transporter 120 (FIG. 1) through the door(s), and can automatically position transporter 120 (FIG. 1) relative to rest room fixtures. After use of the rest room fixtures is complete, UCP assist 145 (FIG. 4) can automatically locate the door(s) and automatically generate movement commands 630 (FIG. 15B) to move transporter 120 (FIG. 1) through the door(s) to exit the rest room stall and/or rest room. Method 950 for negotiating, in transporter 120 (FIG. 1), a rest room stall in a rest room, where the rest room stall can have door 675 (FIG. 15B), and door 675 (FIG. 15B) can have a door threshold and a door swing, can include, but is not limited to including, providing 1451 at least one movement command 630 (FIG. 15B) to cause transporter 120 (FIG. 1) to traverse the door threshold

entering the rest room. Method 950 can also include providing 1453 at least one movement command 630 (FIG. 15B) to position transporter 120 (FIG. 1) for accessing an egress handle of the door, and providing 1455 at least one movement command 630 (FIG. 15B) to move transporter 120 (FIG. 1) away from door 675 (FIG. 15B), as door 675 (FIG. 15B) closes, if the door swing is towards transporter 120 (FIG. 1). Method 950 can also include providing 1457 at least one movement command 630 (FIG. 15B) to move transporter 120 (FIG. 100) toward door 675 (FIG. 15B), as door 675 (FIG. 15B) closes, if the door swing is away from transporter 120 (FIG. 1), and providing 1459 at least one movement command 630 (FIG. 15B) to position transporter 120 (FIG. 1) alongside a first rest room fixture. Method 950 can include providing 1461 at least one movement command 630 (FIG. 15B) to stop transporter 120 (FIG. 1), and can include providing 1463 at least one movement command 630 (FIG. 15B) to position transporter 120 (FIG. 1) near a second rest room fixture. Method 950 can include providing 1465 at least one movement command 630 (FIG. 15B) to traverse the door threshold to exit the rest room stall.

[0077] Continuing to refer primarily to FIG. 15A, automatically traversing the door threshold can optionally include, but is not limited to including, receiving and segmenting 1351 (FIG. 14A1) environmental information 651 (FIG. 10) from sensors 147 (FIG. 10) mounted on transporter 120 (FIG. 1). Environmental information 651 (FIG. 10) can include geometry of transporter 120 (FIG. 1). Automatically traversing the door threshold can also optionally include identifying 1353 (FIG. 14A1) at least one plane within the segmented sensor data, and identifying 1355 (FIG. 14A1) door 675 (FIG. 14B) within the at least one plane. Automatically traversing the door threshold can further optionally include measuring 1357 (FIG. 14A1) door 675 (FIG. 14B) to provide door measurements, and providing 1359 (FIG. 14A1) at least one movement command 630 (FIG. 15B) to move transporter 120 (FIG. 1) away from door 675 (FIG. 14B) if the door measurements are smaller than geometry 649 (FIG. 14B) of transporter (FIG. 1). Automatically traversing the door threshold can also optionally include determining 1361 (FIG. 14A1) the door swing, and providing 1363 (FIG. 14A1) at least one movement command 630 (FIG. 15B) to move transporter 120 (FIG. 1) forward through the doorway, transporter 120 (FIG. 1) opening door 675 (FIG. 14B) and maintaining door 675 (FIG. 1) in an open position, if the door swing is away from transporter 120 (FIG. 1). Automatically traversing the door threshold can further optionally include providing 1365 (FIG. 14A2) at least one movement command 630 (FIG. 15B) to move the transporter for access to a handle of the door, and providing 1367 (FIG. 14A2) at least one movement command 630 (FIG. 15B) to move transporter 120 (FIG. 1) away from door 675 (FIG. 14B), as door 675 (FIG. 14B) opens, by a distance based on the door measurements. Automatically traversing the door threshold can also optionally include providing 1369 (FIG. 14A2) at least one movement command 630 (FIG. 15B) to move transporter 120 (FIG. 1) forward through the doorway, transporter 120 (FIG. 1) maintaining door 675 (FIG. 14B) in an open position, if the door swing is towards transporter 120 (FIG. 1). Method 950 can optionally include automatically locating the rest room, and automatically driving transporter 120 (FIG. 1) to the rest room. SLAM techniques can optionally be used to locate a destination, for example, a rest room. UCP assist 145 can optionally access a database of frequently-visited locations, can receive a selection one of the frequently-visited locations, and can provide at least one movement command 630 (FIG. 15B)

to move transporter 120 (FIG. 1) to the selected location which can include, for example, but not limited to, a rest room.

[0078] Referring now to FIG. 15B, rest room mode 605B for negotiating, in transporter 120 (FIG. 1), a rest room stall in a rest room, where the rest room stall can have a door, and the door can have a door threshold and a door swing, can include, but is not limited to including, door mode 605A providing at least one movement command 630 to cause transporter 120 (FIG. 1) to traverse the door threshold entering the rest room. The rest room can also include fixtures such as for example, but not limited to, toilets, sinks, and changing tables. Entry/exit processor 681C can provide at least one movement command 630 to position transporter 120 (FIG. 1) for accessing an egress handle of the door, and can providing at least one movement command 630 to move the transporter away from the door, as the door closes, if the door swing is towards transporter 120 (FIG. 1). Entry/exit processor 681C can provide at least one movement command 630 to move transporter 120 (FIG. 1) toward door 675, as door 675 closes, if the swing of door 675 is away from transporter 120 (FIG. 1). Fixture processor 681B can provide at least one movement command 630 to position transporter 120 (FIG. 1) alongside a first rest room fixture, and can provide at least one movement command to stop transporter 120 (FIG. 1). Fixture processor 681B can also provide at least one movement command 630 to position transporter 120 (FIG. 1) near a second rest room fixture. Entry/exit processor 681C can provide at least one movement command 630 to traverse the door threshold to exit the rest room stall.

[0079] Referring now to FIGs. 16A1 and 16A2, method 1051 for automatically storing transporter 120 in a vehicle, such as, for example, but not limited to, an accessible van, can assist a user in independent use of the vehicle. When the user exits transporter 120 (FIG. 1) and enters the vehicle, possibly as the vehicle's driver, transporter 120 (FIG. 1) can remain parked outside of the vehicle. If transporter 120 (FIG. 1) is to accompany the user in the vehicle for later use, mobile park mode 605E (FIG. 16B) can provide movement commands 630 (FIG. 16B) to transporter 120 (FIG. 1) to cause transporter 120 (FIG. 1) to store itself either automatically or upon command, and to be recalled to the door of the vehicle as well. Transporter 120 (FIG. 1) can be commanded to store itself through commands received from external applications 140 (FIG. 4), for example. In some configurations, a computer-driven device such as a cell phone, laptop, and/or tablet can be used to execute external application 140 (FIG. 4) and generate information that could ultimately control transporter 120 (FIG. 1). In some configurations, transporter 120 (FIG. 1) can automatically proceed to mobile park mode 605E after the user exits transporter 120 (FIG. 1) when transporter 120 (FIG. 1) has been placed in park mode by, for example, the user. Movement commands 630 (FIG. 16B) can include commands to locate the door of the vehicle at which transporter 120 (FIG. 1) will enter to be stored, and to direct transporter 120 (FIG. 1) to the door. Mobile park mode 605E (FIG. 16B) can determine error conditions such as, for example, but not limited to, if the door is too small for transporter 120 (FIG. 1) to enter and can alert the user of the error condition through, for example, but not limited to, an audio alert through audio interface 150 (FIG. 4) and/or a message to external application 140 (FIG. 4). If the door is wide enough for transporter 120 (FIG. 1) to enter, mobile park mode 605E (FIG. 16B) can provide vehicle control commands to

command the vehicle to open the door. Mobile park mode 605E (FIG. 16B) can determine when the vehicle door is open and whether or not there is space for transporter 120 (FIG. 1) to be stored. Mobile park mode 605E (FIG. 16B) can invoke obstacle processing 607 (FIG. 14B) to assist in determining the status of the vehicle door and if there is room in the vehicle to store transporter 120 (FIG. 1). If mobile park mode 605E (FIG. 16B) determines that there is enough room for transporter 120 (FIG. 1), mobile park mode 605E (FIG. 16B) can provide movement commands 630 (FIG. 16B) to move transporter 120 (FIG. 1) into the storage space in the vehicle. Mobile park mode 605E (FIG. 16B) can provide vehicle control commands to command the vehicle to lock transporter 120 (FIG. 1) into place, and to close the vehicle door. When transporter 120 (FIG. 1) is again needed, external application 140 (FIG. 1), for example, can be used to invoke mobile park mode 605E. Mobile park mode 605E (FIG. 16B) can recall the status of transporter 120 (FIG. 1) and can begin processing by providing vehicle control commands to command the vehicle to unlock transporter 120 (FIG. 1) and open the door of the vehicle. Mobile park mode 605E (FIG. 16B) can once again locate the door of the vehicle, or can access the location 675A of the door from, for example, local storage 607H (FIG. 14B) and/or cloud storage 607G (FIG. 14B). Mobile park mode 605E (FIG. 16B) can provide movement commands 630 (FIG. 16B) to move transporter 120 (FIG. 1) through the vehicle door and to the passenger door to which it had been summoned by, for example, external application 140 (FIG. 4). In some configurations, the vehicle can be tagged in places such as, for example, the entry door for storage of transporter 120 (FIG. 1). Mobile park mode 605E can recognize the tags, such as, for example, but not limited to, fiducials, bar codes, and/or QR CODES®, and can execute the method described herein as a result of recognizing the tags. Other tags can be included, such as tags within the storage compartment to indicate the proper storage location and tags on vehicle passenger doors. The tags can be RFID enabled, for example, and transporter 120 (FIG. 1) can include an RFID reader.

[0080] Continuing to refer primarily to FIGs. 16A1 and 16A2, method 1051 for automatically storing transporter 120 in a vehicle can include, but is not limited to including, providing 1551 at least one movement command 630 (FIG. 16B) to locate the door of the vehicle at which transporter 120 (FIG. 1) will enter to be stored in a storage space in the vehicle, and providing 1553 at least one movement command 630 (FIG. 16B) to direct transporter 120 (FIG. 1) to the door. If 1555 the vehicle door is wide enough for transporter 120 (FIG. 1) to enter, method 1051 can include providing 1557 at least one vehicle control command to command the vehicle to open the door. If 1559 the door is open and if 1561 there is room in the vehicle to store transporter 120 (FIG. 1), method 1051 can include providing 1563 at least one movement command 630 (FIG. 16B) to move transporter 120 (FIG. 1) into the storage space in the vehicle. Method 1051 can include providing 1565 at least one vehicle control command to command the vehicle to lock transporter 120 (FIG. 1) into place, and to close the door of the vehicle. If 1555 the vehicle door is not wide enough, or if 1559 the vehicle door is not open, or if 1561 there is no space for transporter 120 (FIG. 1), method 1051 can include alerting 1567 the user, and providing 1569 at least one movement command 630 (FIG. 16B) to return transporter 120 (FIG. 1) to the user.

[0081] Continuing to refer primarily to FIGs. 16A1 and 16A2, the at least one movement

command 630 (FIG. 16B) to store transporter 120 (FIG. 100) can be received from external application 140 (FIG. 4) and/or automatically generated. Method 1051 can optionally include alerting the user of error conditions through, for example, but not limited to, an audio alert through audio interface 150 (FIG. 4) and/or a message to external application 140 (FIG. 4). Method 1051 can optionally invoke obstacle processing 607 (FIG. 14B) to assist in locating the door of the vehicle, to determine if there is enough room in the vehicle to store transporter 120 (FIG. 1), and to locate any locking mechanism in the vehicle. When transporter 120 (FIG. 1) is again needed, that is, when the user has arrived at a destination in the vehicle, external application 140 (FIG. 1), for example, can be used to invoke transporter 120 (FIG. 1). Method 1051 can include recalling the status of transporter 120 (FIG. 1) and can include providing vehicle control commands to command the vehicle to unlock transporter 120 (FIG. 1) and open the door of the vehicle. Method 1051 can include locating the door of the vehicle, or can include accessing the location of the vehicle door from, for example, local storage 607H (FIG. 14B) and/or cloud storage 607G (FIG. 14B). Method 1051 can include providing movement commands 630 (FIG. 16B) to move transporter 120 (FIG. 1) through the vehicle door and to the passenger door to which it had been summoned by, for example, but not limited to, external application 140 (FIG. 4).

[0082] Referring now to FIG. 16B, mobile park mode 605E can include, but is not limited to including, vehicle door processor 691D that can provide at least one movement command 630 to locate door 675 of the vehicle at which transporter 120 (FIG. 1) will enter to be stored in a storage space in the vehicle. Vehicle door processor 691D can also provide at least one movement command 630 to direct transporter 120 (FIG. 1) to door 675. If door 675 is wide enough for transporter 120 (FIG. 1) to enter, vehicle command processor 691C can provide at least one vehicle control command to command the vehicle to open door 675. If door 675 is open and if there is room in the vehicle to store transporter 120 (FIG. 1), space processor 691B can provide at least one movement command 630 to move transporter 120 (FIG. 1) into the storage space in the vehicle. Vehicle command processor 691C can provide at least one vehicle control command to command the vehicle to lock transporter 120 (FIG. 1) into place, and to close door 675 of the vehicle. If door 675 is not wide enough, or if door 675 is not open, or if there is no space for transporter 120 (FIG. 1), error processor 691E can alert the user, and can provide at least one movement command 630 to return transporter 120 (FIG. 1) to the user.

[0083] Continuing to refer to FIG. 16B, vehicle door processor 691D can optionally recall the status of transporter 120 (FIG. 1), and vehicle command processor 691C can provide vehicle control commands to command the vehicle to unlock transporter 120 (FIG. 1) and open door 675 of the vehicle. Vehicle door processor 691D can once again locate door 675 of the vehicle, or can access the location of door 675 from, for example, local storage 607H (FIG. 14B) and/or cloud storage 607G (FIG. 14B), and/or door database 673B. Vehicle door processor 691D can provide movement commands 630 to move transporter 120 (FIG. 1) through door 675 and to the passenger door to which it had been summoned by, for example, external application 140 (FIG. 4).

[0084] Referring now primarily to FIG. 17A, method 1150 for storing/recharging transporter 120 (FIG. 1) can assist the user in storing and possibly recharging transporter 120 (FIG. 1). For example, transporter 120 (FIG. 1) could be recharged when the user sleeps. After the user exits transporter 120 (FIG. 1), commands can be initiated at, for example, external application 140 (FIG. 4), to move perhaps riderless transporter 120 (FIG. 1) to a storage/docking area. In some configurations, a mode selection by the user while the user occupies transporter 120 (FIG. 1) can initiate automatic storage/docking functions after the user has exited transporter 120 (FIG. 1). When transporter 120 (FIG. 1) is again needed, commands can be initiated by external application 140 (FIG. 4) to recall transporter 120 (FIG. 1) to the user. Method 1150 can include, but is not limited to including, locating 1651 at least one storage/charging area, and providing 1655 at least one movement command 630 (FIG. 17B) to move transporter 120 (FIG. 1) from a first location to the storage/charging area. Method 1150 can include locating 1657 a charging dock in the storage/charging area and providing 1663 at least one movement command 630 (FIG. 17B) to couple transporter 120 (FIG. 1) with the charging dock. Method 1150 can optionally include providing at least one movement command 630 (FIG. 17B) to move transporter 120 (FIG. 1) to the first location when transporter 120 (FIG. 1) receives an invocation command. If 1653 there is no storage/charging area, or if 1659 there is no charging dock, or if 1666 transporter 120 (FIG. 1) cannot couple with the charging dock, method 1150 can optionally include providing 1665 at least one alert to the user, and providing 1667 at least one movement command 630 (FIG. 17B) to move transporter 120 (FIG. 1) to the first location.

[0085] Referring now to FIG. 17B, static storage/charging mode 605F can include, but is not limited to including, storage/charging area processor 702A that can locate at least one storage/charging area 695, and can provide at least one movement command 630 to move transporter 120 (FIG. 1) from a first location to storage/charging area 695. Coupling processor 702D can locate a charging dock in a storage/charging area, and can provide at least one movement command 630 to couple transporter 120 (FIG. 1) with the charging dock. Return processor 702B can optionally provide at least one movement command 630 to move transporter 120 (FIG. 1) to the first location when transporter 120 (FIG. 1) receives an invocation command. If there is no storage/charging area 695, or if there is no charging dock, or if transporter 120 (FIG. 1) cannot couple with the charging dock, error processor 702E can optionally provide at least one alert to the user, and can providing at least one movement command 630 to move transporter 120 (FIG. 1) to the first location.

[0086] Referring now to FIG. 18A, method 1250 for negotiating an elevator while maneuvering transporter 120 (FIG. 1) can assist a user in getting on and off elevator 685 (FIG. 18B) in transporter 120 (FIG. 1). Sensor processing 661 can be used to locate elevator 685 (FIG. 18B), for example, or elevator location 685A (FIG. 18B) can be determined from local storage 607H (FIG. 14B) and/or storage cloud 607G (FIG. 14B). When elevator 685 (FIG. 18B) is located, and when the user selects the desired elevator direction, and when elevator 685 (FIG. 18B) arrives and the door opens, elevator mode 605D (FIG. 18B) can provide movement commands 630 (FIG. 18B) to move transporter 120 (FIG. 1) into elevator 685 (FIG. 18B). The geometry of elevator 685 (FIG. 18B) can be determined and movement commands 630 (FIG. 18B) can be provided to move transporter 120 (FIG. 1) into a location that makes it possible for

the user to select a desired activity from the elevator selection panel. The location of transporter 120 (FIG. 1) can also be appropriate for exiting elevator 685 (FIG. 18B). When the elevator door opens, movement commands 630 (FIG. 18B) can be provided to move transporter 120 (FIG. 1) to fully exit elevator 685 (FIG. 18B). Method 1250 can include, but is not limited to including, locating 1751 elevator 685 (FIG. 18B), where elevator 685 (FIG. 18B) has an elevator door and an elevator threshold associated with the elevator door. Method 1250 can include providing 1753 at least one movement command 630 (FIG. 18B) to move transporter 120 (FIG. 1) through the elevator door beyond the elevator threshold. Method 1250 can also include determining 1755 the geometry of elevator 685 (FIG. 18B), and providing 1757 at least one movement command 630 (FIG. 18B) to move transporter 120 (FIG. 1) into a floor selection/exit location relative to the elevator threshold. Method 1250 can also include providing 1759 at least one movement command 630 (FIG. 18B) to move transporter 120 (FIG. 1) across and beyond the elevator threshold to exit elevator 685 (FIG. 18B).

[0087] Referring now primarily to FIG. 18B, elevator mode 605D can include, but is not limited to including, elevator locator 711A that can locate elevator 685 having an elevator door and an elevator threshold associated with the elevator door. Elevator locator 711A can save obstacles 623, elevators 685, and elevator locations 685A in elevator database 683B, for example. Elevator database 683B can be located locally or remotely from transporter 120. Entry/exit processor 711B can provide at least one movement command 630 to move transporter 120 (FIG. 1) through the elevator door beyond the elevator threshold to either enter or exit elevator 685. Elevator geometry processor 711D can determine the geometry of elevator 685. Entry/exit processor 711B can provide at least one movement command 630 to move transporter 120 (FIG. 1) into a floor selection/exit location relative to the elevator threshold.

[0088] Configurations of the present teachings are directed to computer systems for accomplishing the methods discussed in the description herein, and to computer readable media containing programs for accomplishing these methods. The raw data and results can be stored for future retrieval and processing, printed, displayed, transferred to another computer, and/or transferred elsewhere. Communications links can be wired or wireless, for example, using cellular communication systems, military communications systems, and satellite communications systems. Parts of system 200A (FIG. 4), for example, can operate on a computer having a variable number of CPUs. Other alternative computer platforms can be used.

[0089] The present configuration is also directed to software and/or firmware and/or hardware for accomplishing the methods discussed herein, and computer readable media storing software for accomplishing these methods. The various modules described herein can be accomplished by the same CPU, or can be accomplished by different CPUs tightly or loosely coupled. The various modules can be accomplished by specially-designed integrated circuits. In compliance with the statute, the present configuration has been described in language more or less specific as to structural and methodical features. It is to be understood, however, that the present configuration is not limited to the specific features shown and described, since the

means herein disclosed comprise various forms of putting the present teachings into effect.

[0090] Methods 650 (FIGs. 11A1-11A2), 750 (FIG. 13A), 850 (FIGs. 14A1-14A2), 950 (FIG. 15A), 1050 (FIGs. 16A1-16A2), 1150 (FIG. 17A), and 1250 (FIG. 18A), can be, in whole or in part, implemented electronically. Signals representing actions taken by elements of system 200A (FIG. 4) and other disclosed configurations can travel over at least one live communications network 143/144 (FIG. 4). Control and data information can be electronically executed and stored on at least one computer-readable medium. The system can be implemented to execute on at least one computer node in at least one live communications network. Common forms of at least one computer-readable medium can include, for example, but not be limited to, a floppy disk, a flexible disk, a hard disk, magnetic tape, or any other magnetic medium, a compact disk read only memory or any other optical medium, punched cards, paper tape, or any other physical medium with patterns of holes, a random access memory, a programmable read only memory, and erasable programmable read only memory (EPROM), a Flash EPROM, or any other memory chip or cartridge, or any other medium from which a computer can read.

[0091] While the present teachings have been described above in terms of specific configurations, it is to be understood that they are not limited to these disclosed configurations. Many modifications and other configurations will come to mind to those skilled in the art to which this pertains, and which are intended to be and are covered by both this disclosure and the appended claims. It is intended that the scope of the present teachings should be determined by proper interpretation and construction of the appended claims and their legal equivalents, as understood by those of skill in the art relying upon the disclosure in this specification and the attached drawings.

REFERENCES CITED IN THE DESCRIPTION

Cited references

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US62322522 [0001]

- US2012185096A1 **[0004]**
- WO2007047510A2 **[0005]**
- US6343664B2 **[0006]**
- US6343664B **[0026]**

Patentkrav

1. Fremgangsmåde (850) til kommunikation af en transportør (120) med en dør (675), hvor døren (675) omfatter mindst ét håndtag, en dørbevægelsesretning og en døråbning, idet fremgangsmåden (850) omfatter:
modtagelse (1351) og segmentering af miljøinformationer (651) fra sensorer (147), der er monteret på transportøren (120), idet miljøinformationerne (651) omfatter transportørens (120) geometri
identifikation (1353) af mindst ét plan inden for de segmenterede miljøinformationer (651)
identifikation (1355) af døren (675) i det mindst ene plan
måling (1357) af døren (675) på baggrund af de segmenterede miljøinformationer (651) for at tilvejebringe dørmål
tilvejebringelse (1363) af en første bevægelseskommando (630) til at bevæge transportøren (120) for at få adgang til det mindst ene håndtag
tilvejebringelse (1365) af en anden bevægelseskommando (630), hvor den anden bevægelseskommando (630), hvis dørbevægelsesretningen er mod transportøren (120), er en kommando til at bevæge transportøren (120) væk fra døren (675), når døren åbner, med en afstand baseret på dørmålene
tilvejebringelse (1367) af en tredje bevægelseskommando (630) til at bevæge transportøren (120) fremad gennem døråbningen, og
tilvejebringelse af en fjerde bevægelseskommando (630) til at bevæge transportøren (120) væk fra døren (675), hvis dørmålene er mindre end transportørens (120) geometri.

2. Fremgangsmåde (850) ifølge krav 1, hvorved den anden bevægelseskommando (630), hvis dørbevægelsesretningen er mod transportøren (120), er en kommando til at bevæge transportøren (120) bagud med en forud fastlagt eller dynamisk bestemt afstand på baggrund af mindst en størrelse af den målte dør (675).

3. Fremgangsmåde (850) ifølge krav 2, der ydermere omfatter indsamling og behandling af punkt"cloud"data (655) og bestemmelse af den størrelse, som døren (675) bevæger sig med,

på baggrund af successive måleværdier af punkt"cloud"dataene (655).

4. Fremgangsmåde (850) ifølge et af de foregående krav, hvorved den anden bevægelseskommando (630), hvis dørbevægelsesretningen er væk fra transportøren (120), er en kommando til at bevæge transportøren (120) fremad med en forud fastlagt eller dynamisk bestemt afstand på baggrund af mindst en størrelse af den målte dør (675).

5. Fremgangsmåde (850) ifølge et af de foregående krav, hvorved den anden bevægelseskommando (630), hvis dørbevægelsesretningen er væk fra transportøren (120), omfatter en kommando til at bevæge transportøren (120) mod døren (675) for at placere døren (675) til bevægelse af transportøren (120) gennem døråbningen.

6. Fremgangsmåde (850) ifølge et af de foregående krav, hvorved målingen (1357) af døren (675) omfatter måling af en bredde af døren (675), og hvorved den størrelse, som transportøren (120) bliver styret til at bevæge sig bagud eller fremad med, er baseret på dørens bredde (675).

7. Fremgangsmåde (850) ifølge krav 6, hvorved målingen af dørens (675) bredde omfatter bestemmelse af transportørens (120) position og orientering og bestemmelse af placeringen af et dørdrejepunkt.

8. Fremgangsmåde (850) ifølge et af de foregående krav, der ydermere omfatter bestemmelse (1361) af dørbevægelsesretningen, hvor dørbevægelsesretningen bestemmes på baggrund af en bestemmelse af en hængselside af døren (675), en bestemmelse af dørens (675) retning, en bestemmelse af dørens (675) vinkel og en bestemmelse af afstanden fra transportøren (120) til døren (675).

9. Fremgangsmåde (850) ifølge krav 8, der ydermere omfatter indsamling og behandling af punkt"cloud"data (655) og, efter at

have fastslået, at et antal gyldige punkter i et billede af døren (675), der er afledt af punkt"cloud"dataene (655), er større end en tærskel, bestemmelse af afstanden fra transportøren (120) til døren (675) på baggrund af punkt"cloud"dataene (655).

10. Fremgangsmåde (850) ifølge krav 8 eller 9, hvorved bestemmelsen af dørbevægelsesretningen omfatter bestemmelse af, om døren (675) er en dør, der åbnes ved at skubbe, en dør, der åbnes ved at trække, eller en skydedør.

11. Fremgangsmåde (850) ifølge et af de foregående krav, hvorved fremgangsmåden (850) ydermere omfatter tilvejebringelse (1459) af en femte bevægelseskommando (630) til at placere transportøren (120) ved siden af et første fast hvilerumstilbehør og/eller tilvejebringelse (1463) af en sjette bevægelseskommando (630) til at placere transportøren (120) nær et andet fast hvilerumstilbehør.

12. Fremgangsmåde (850) ifølge det foregående krav, hvorved fremgangsmåden (850) ydermere omfatter tilvejebringelse af en yderligere bevægelseskommando (630) til at bevæge transportøren (120) tilbage gennem den døråbning, gennem hvilken transportøren (120) har bevæget sig.

13. Fremgangsmåde (850) ifølge et af de foregående krav, hvorved sensorerne (147) omfatter flyvetidssensorer.

14. Transportør (120) til udførelse af fremgangsmåden ifølge krav 1, hvor transportøren (120) omfatter:
en sensorprocessor (661), der er konfigureret til at modtage (1351) og segmentere miljøinformationer (651) fra sensorer (147), der er monteret på transportøren (120), og ud fra sensordata at bestemme en hængselside af døren (675) samt dørens (675) retning og vinkel, hvor miljøinformationerne (651) omfatter transportørens (120) geometri
en dørlokalisator (671A), der er konfigureret til at identificere (1353) mindst ét plan i de segmenterede

miljøinformationer (651) og identificere (1355) døren (675) i det mindst ene plan

en dørprocessor (671B), der er konfigureret til at bestemme karakteristiske egenskaber ved døren (675), idet dørprocessoren (671B) er konfigureret til at bestemme afstanden fra transportøren (120) til døren (675), og idet dørprocessoren (671B) er konfigureret til på baggrund af de segmenterede miljøinformationer (641) at måle (1357) døren (675) for at tilvejebringe dørmål

en dørbevægelsesprocessor (671D), der er konfigureret til at tilvejebringe en første bevægelseskommando (630), en anden bevægelseskommando (630), en tredje bevægelseskommando (630) og en fjerde bevægelseskommando (630), hvor den første bevægelseskommando (630) er en kommando til at bevæge transportøren (120) for at få adgang til det mindst ene håndtag, hvor den anden bevægelseskommando (630) er en kommando til, hvis dørbevægelsesretningen er mod transportøren (120), at bevæge transportøren (120) væk fra døren (675), når døren åbner, med en afstand baseret på dørmålene, hvor den tredje bevægelseskommando (630) er en kommando til at bevæge transportøren (120) fremad gennem døren (675) på baggrund af dørbevægelsesretningen og dørens (675) bredde, hvor den fjerde bevægelseskommando (630) er en kommando til at bevæge transportøren (120) væk fra døren (675), hvis dørmålene er mindre end transportørens (120) geometri.

15. Transportør (120) ifølge krav 14, hvor bevægelsesprocessoren (603) er konfigureret til at modtage navigationsinformationer (653) fra sensordataene og generere den mindst ene bevægelseskommando (630),

hvor transportøren (120) ydermere omfatter:

en geometriprocessor, der er konfigureret til at tilvejebringe geometriske informationer om transportøren (120)

en processor (611) for et punkt"cloud"bibliotek (PCL), der er konfigureret til at indsamle miljøinformationer (651), punkt"cloud"data (655) og de geometriske informationer

en processor for samtidig lokalisering og afbildning (SLAM), der er konfigureret til at behandle miljøinformationerne (651) og den mindst ene bevægelseskommando (630)

en forhindringsprocessor (607), der er konfigureret til at få adgang til navigationsinformationerne (653), den mindst ene bevægelseskommando (630) og punkt"cloud"dataene (655) fra PCL-processoren (611) og SLAM-processoren (609), idet forhindringsprocessoren (607) er konfigureret til at bestemme forhindringsinformationer, der er knyttet til transportøren (120), i det mindste på baggrund af navigationsinformationerne (653), den mindst ene bevægelseskommando (630), de geometriske informationer, brugerinputtet og punkt"cloud"dataene (655), og

en kommandoprocessor, der er konfigureret til at modtage forhindringsinformationer fra forhindringsprocessoren (607), idet kommandoprocessoren er konfigureret til at sende den mindst ene bevægelseskommando (630) til bevægelsesprocessoren (603) i det mindste på baggrund af forhindringsinformationerne.

DRAWINGS

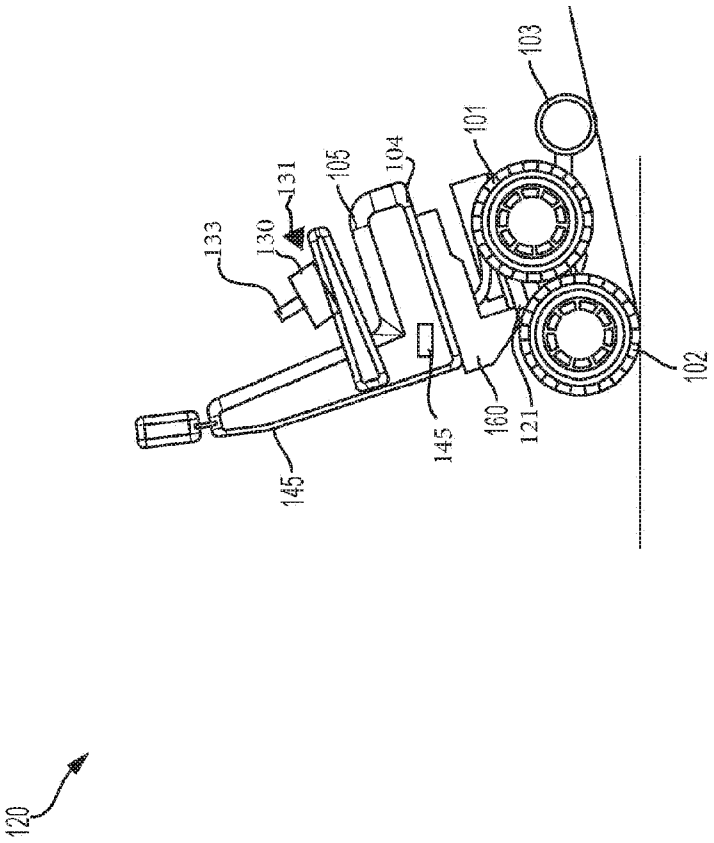


FIG. 1

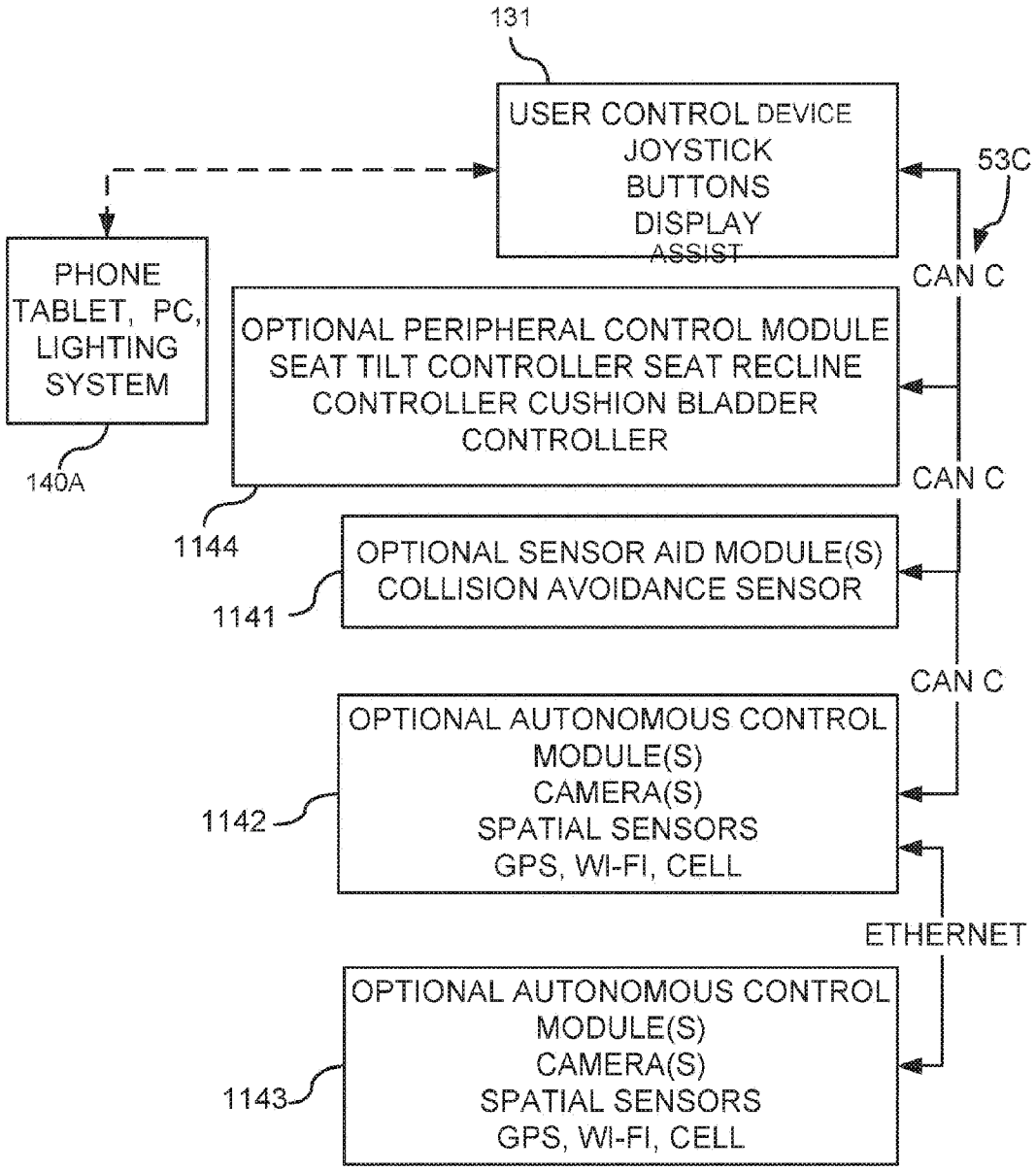

131
USER CONTROL DEVICE
JOYSTICK
BUTTONS
DISPLAY
ASSIST
53C
CAN C
PHONE
TABLET, PC,
LIGHTING
SYSTEM
140A
OPTIONAL PERIPHERAL CONTROL MODULE
SEAT TILT CONTROLLER SEAT RECLINE
CONTROLLER CUSHION BLADDER
CONTROLLER
1144
CAN C
OPTIONAL SENSOR AID MODULE(S)
COLLISION AVOIDANCE SENSOR
1141
CAN C
OPTIONAL AUTONOMOUS CONTROL
MODULE(S)
CAMERA(S)
SPATIAL SENSORS
GPS, WI-FI, CELL
1142
ETHERNET
OPTIONAL AUTONOMOUS CONTROL
MODULE(S)
CAMERA(S)
SPATIAL SENSORS
GPS, WI-FI, CELL
1143

FIG. 2A

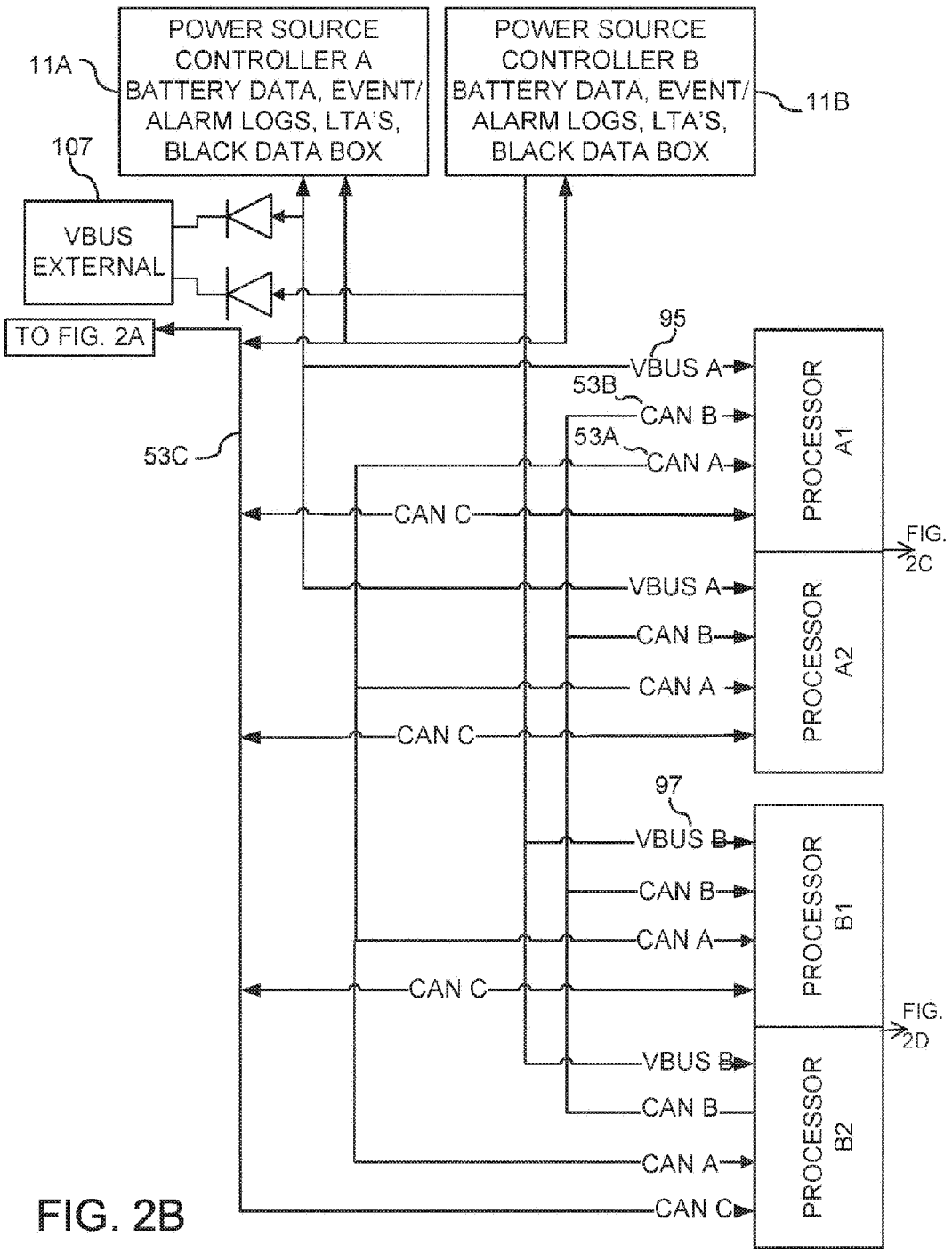

POWER SOURCE CONTROLLER A BATTERY DATA, EVENT/ ALARM LOGS, LTA'S, BLACK DATA BOX
11A
POWER SOURCE CONTROLLER B BATTERY DATA, EVENT/ ALARM LOGS, LTA'S, BLACK DATA BOX
11B
107
VBUS EXTERNAL
TO FIG. 2A
95
VBUS A
53B
CAN B
53A
CAN A
53C
CAN C
PROCESSOR A1
FIG. 2C
VBUS A
CAN B
CAN A
CAN C
PROCESSOR A2
97
VBUS B
CAN B
CAN A
CAN C
PROCESSOR B1
FIG. 2D
VBUS B
CAN B
CAN A
CAN C
PROCESSOR B2

FIG. 2B

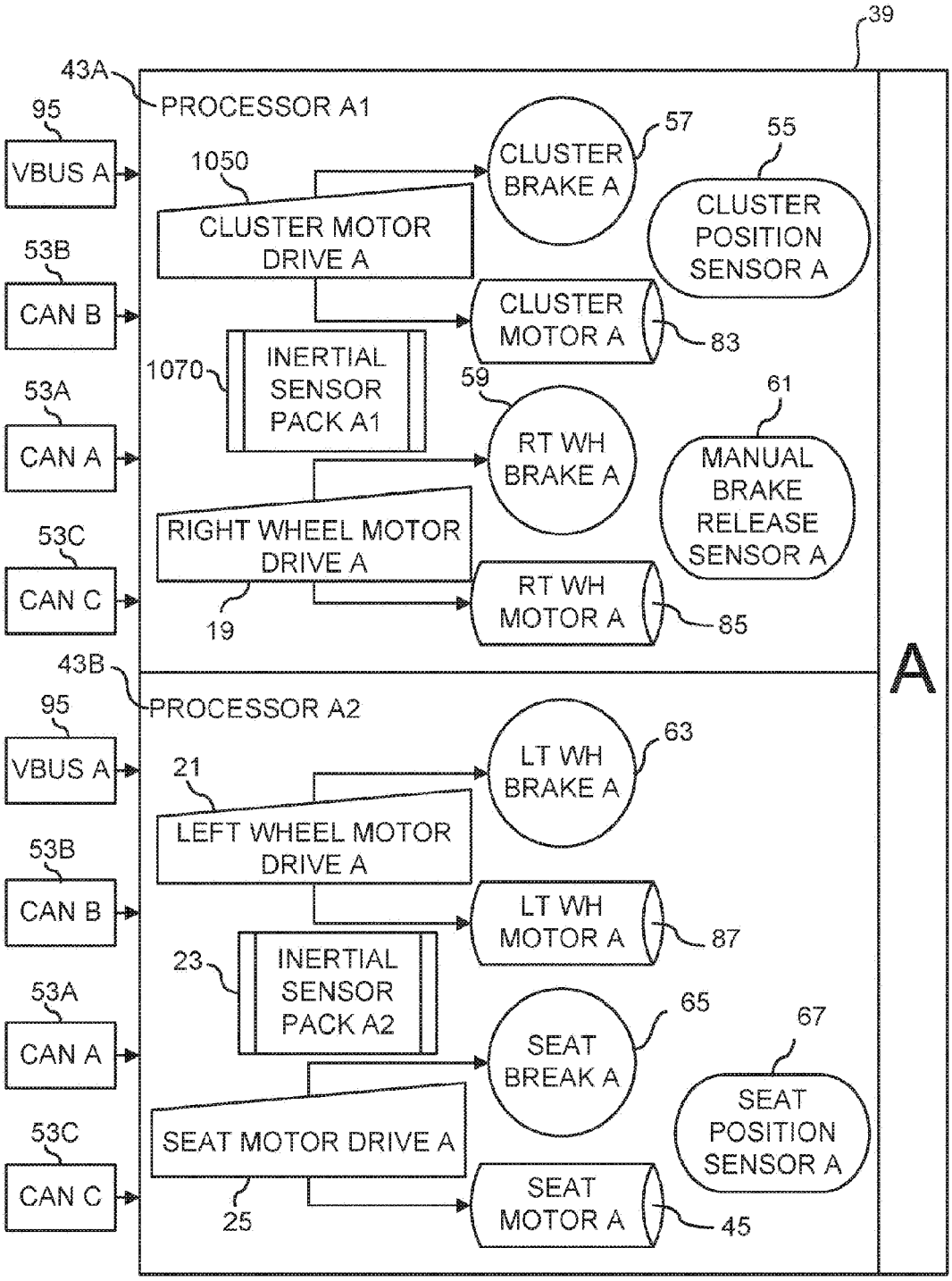

39
43A
PROCESSOR A1
95
VBUS A
1050
CLUSTER MOTOR DRIVE A
CLUSTER BRAKE A
57
55
CLUSTER POSITION SENSOR A
53B
CAN B
CLUSTER MOTOR A
83
1070
INERTIAL SENSOR PACK A1
59
RT WH BRAKE A
61
MANUAL BRAKE RELEASE SENSOR A
53A
CAN A
RIGHT WHEEL MOTOR DRIVE A
53C
CAN C
RT WH MOTOR A
85
19
A
43B
PROCESSOR A2
95
VBUS A
21
LEFT WHEEL MOTOR DRIVE A
LT WH BRAKE A
63
53B
CAN B
LT WH MOTOR A
87
23
INERTIAL SENSOR PACK A2
53A
CAN A
SEAT BREAK A
65
67
SEAT POSITION SENSOR A
SEAT MOTOR DRIVE A
53C
CAN C
SEAT MOTOR A
45
25

FIG. 2C

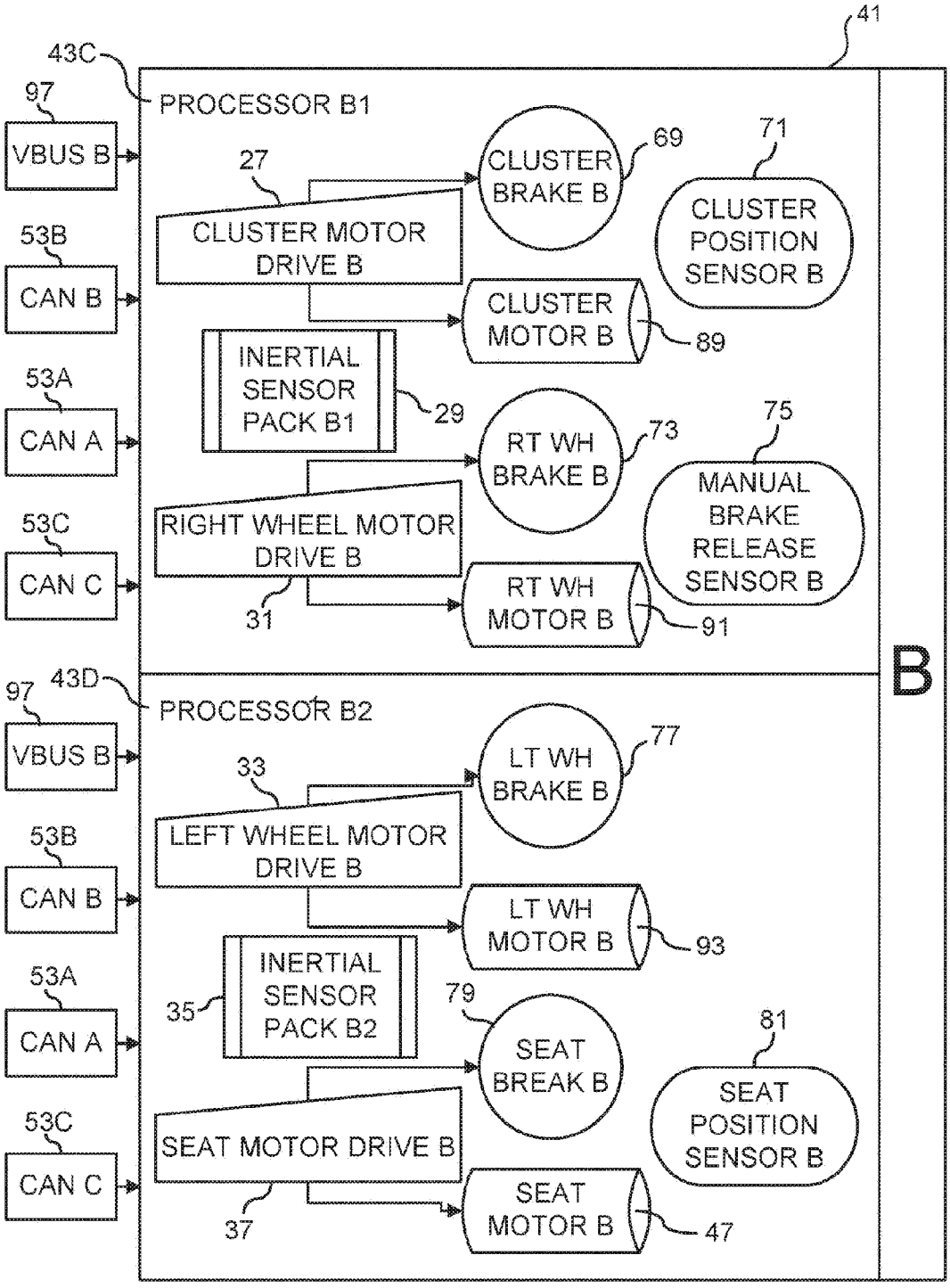

41
43C
PROCESSOR B1
97
VBUS B
53B
CAN B
53A
CAN A
53C
CAN C
27
CLUSTER MOTOR DRIVE B
CLUSTER BRAKE B
69
71
CLUSTER POSITION SENSOR B
CLUSTER MOTOR B
89
INERTIAL SENSOR PACK B1
29
RT WH BRAKE B
73
75
MANUAL BRAKE RELEASE SENSOR B
RIGHT WHEEL MOTOR DRIVE B
31
RT WH MOTOR B
91
B
43D
PROCESSOR B2
97
VBUS B
53B
CAN B
53A
CAN A
53C
CAN C
33
LEFT WHEEL MOTOR DRIVE B
LT WH BRAKE B
77
LT WH MOTOR B
93
35
INERTIAL SENSOR PACK B2
79
SEAT BREAK B
81
SEAT POSITION SENSOR B
SEAT MOTOR DRIVE B
37
SEAT MOTOR B
47

FIG. 2D

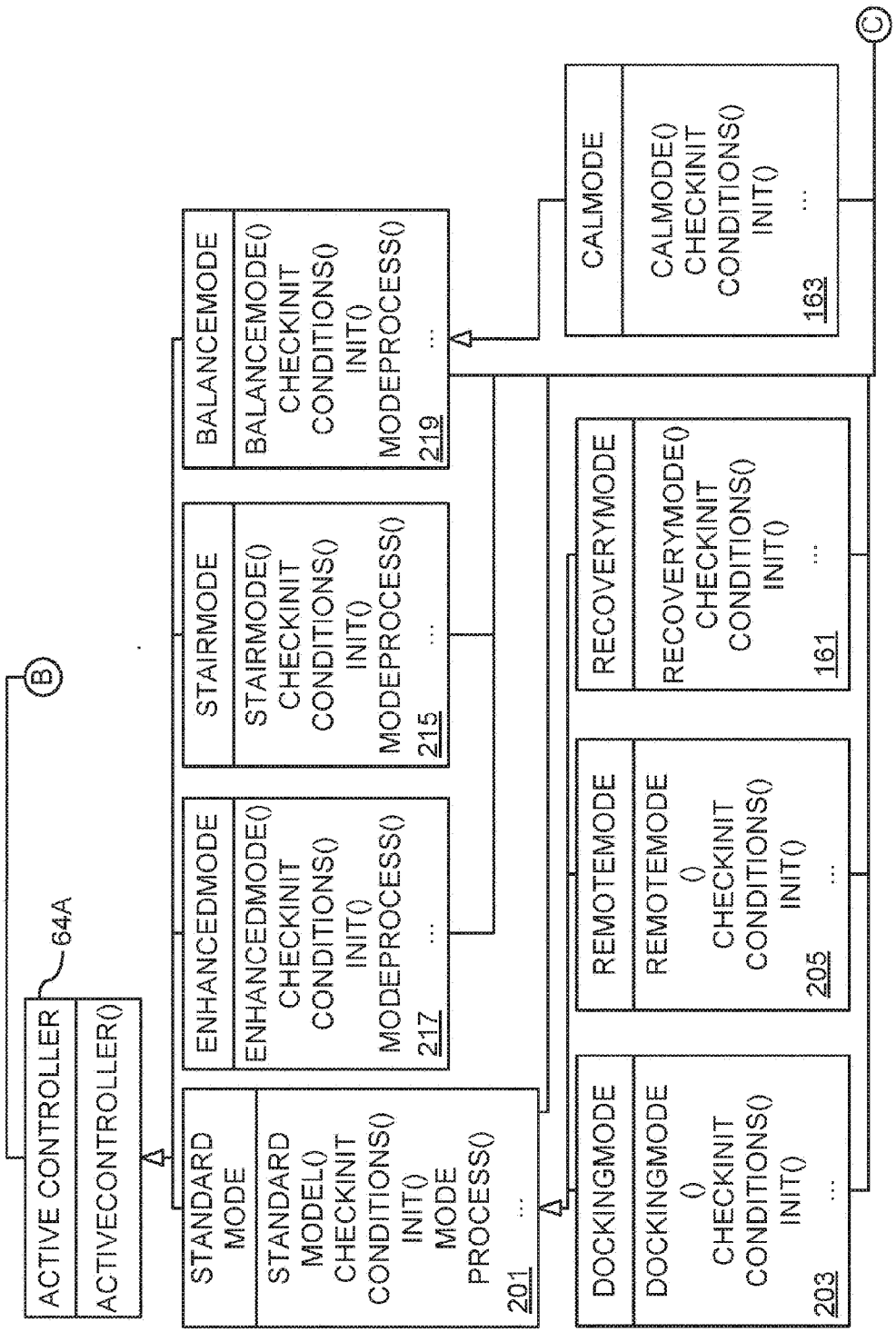

ACTIVE CONTROLLER
ACTIVECONTROLLER()
64A
B
STANDARD MODE
STANDARD MODEL() CHECKINIT CONDITIONS() INIT() MODE PROCESS()
201
ENHANCEDMODE
ENHANCEDMODE() CHECKINIT CONDITIONS() INIT() MODEPROCESS()
217
STAIRMODE
STAIRMODE() CHECKINIT CONDITIONS() INIT() MODEPROCESS()
215
BALANCEMODE
BALANCEMODE() CHECKINIT CONDITIONS() INIT() MODEPROCESS()
219
DOCKINGMODE
DOCKINGMODE() CHECKINIT CONDITIONS() INIT()
203
REMOTEMODE
REMOTEMODE() CHECKINIT CONDITIONS() INIT()
205
RECOVERYMODE
RECOVERYMODE() CHECKINIT CONDITIONS() INIT()
161
CALMODE
CALMODE() CHECKINIT CONDITIONS() INIT()
163
C

FIG. 3A

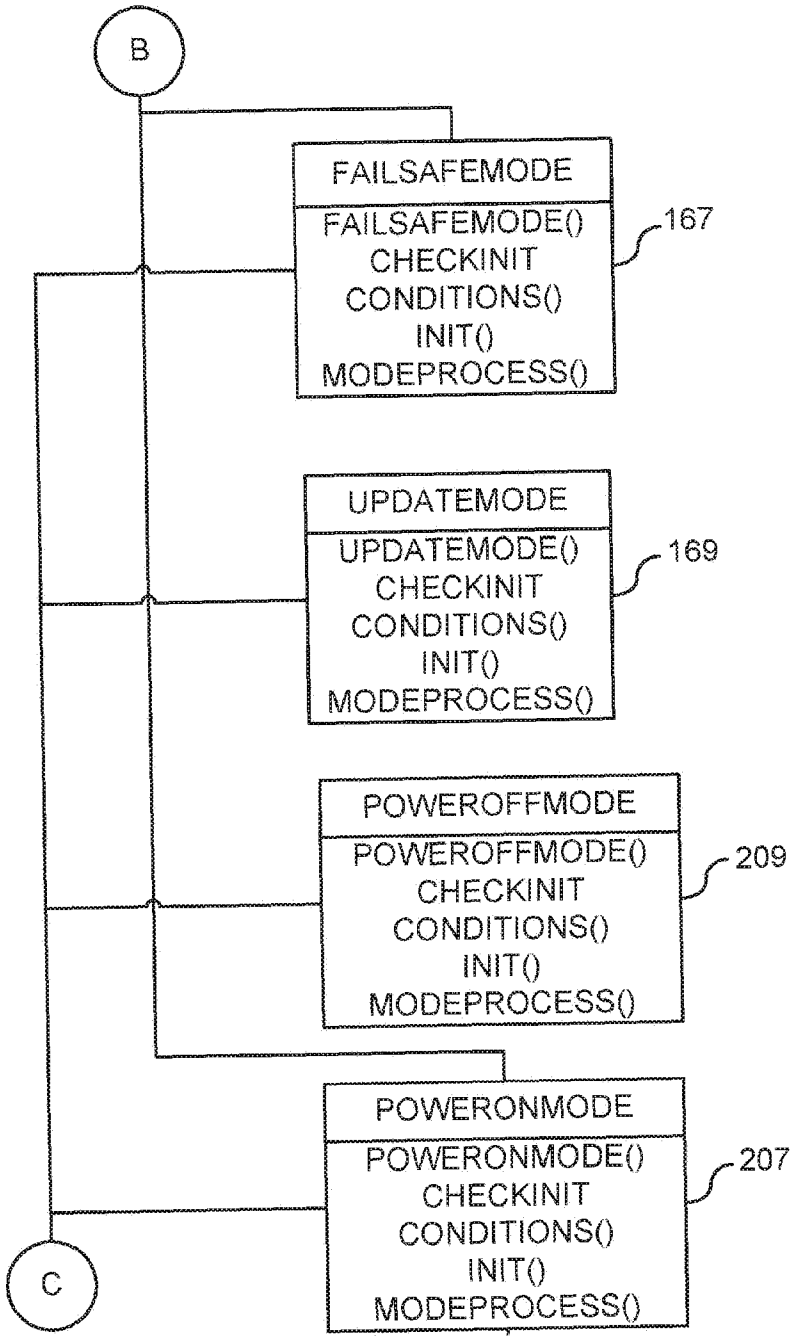

B
FAILSAFEMODE
FAILSAFEMODE()
CHECKINIT
CONDITIONS()
INIT()
MODEPROCESS()
167
UPDATEMODE
UPDATEMODE()
CHECKINIT
CONDITIONS()
INIT()
MODEPROCESS()
169
POWEROFFMODE
POWEROFFMODE()
CHECKINIT
CONDITIONS()
INIT()
MODEPROCESS()
209
POWERONMODE
POWERONMODE()
CHECKINIT
CONDITIONS()
INIT()
MODEPROCESS()
207
C

FIG. 3B

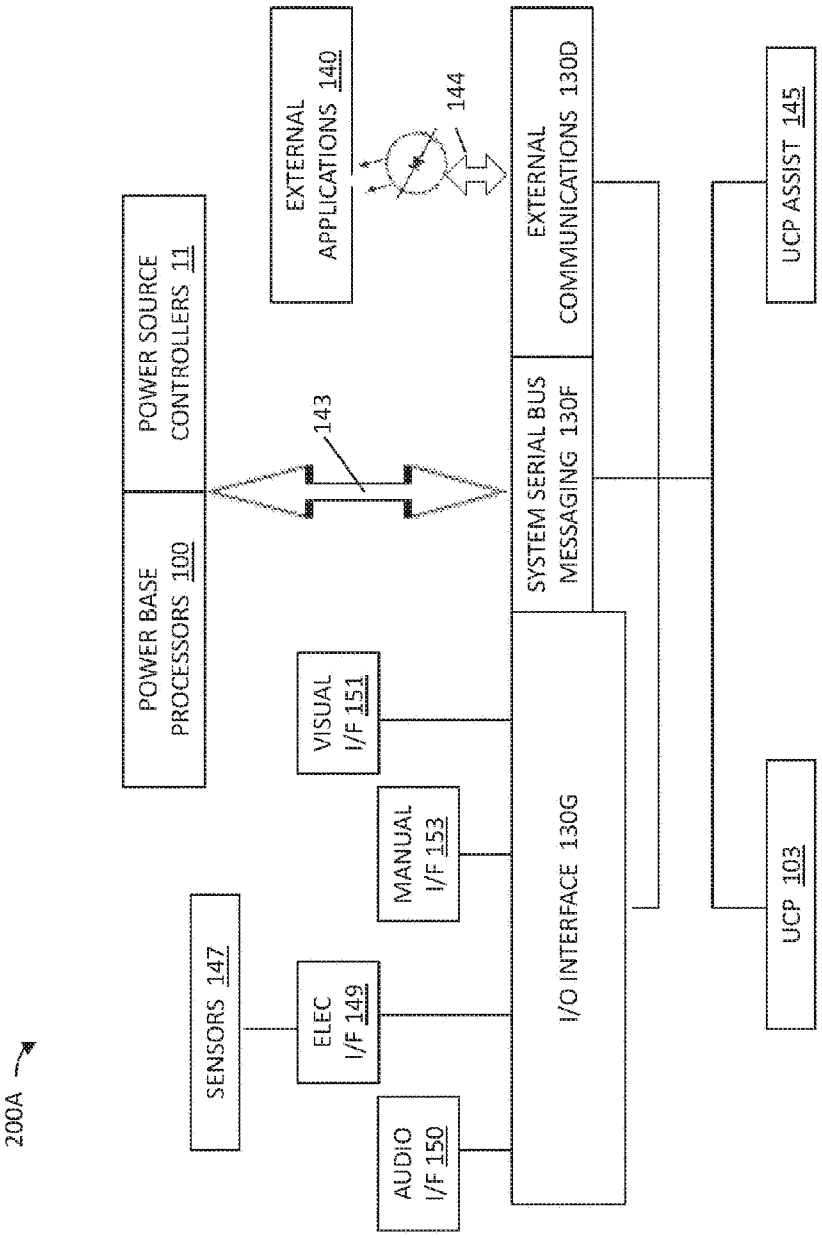

200A
POWER BASE PROCESSORS 100
POWER SOURCE CONTROLLERS 11
143
EXTERNAL APPLICATIONS 140
144
EXTERNAL COMMUNICATIONS 130D
SYSTEM SERIAL BUS MESSAGING 130F
I/O INTERFACE 130G
VISUAL I/F 151
MANUAL I/F 153
ELEC I/F 149
AUDIO I/F 150
SENSORS 147
UCP ASSIST 145
UCP 103

FIG. 4

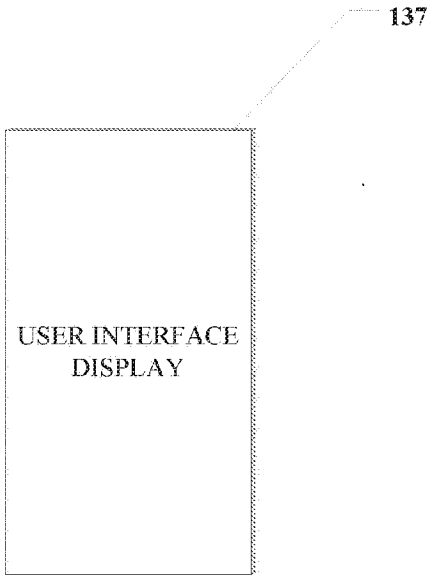


FIG. 5A

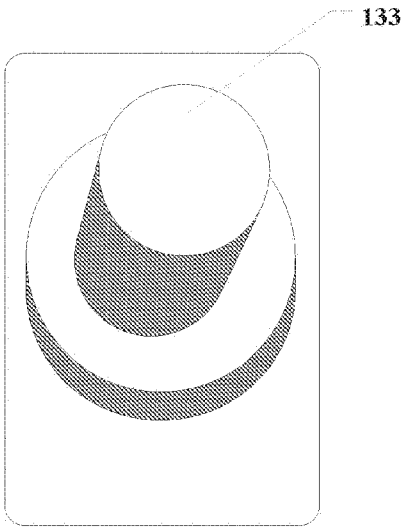


FIG. 5B

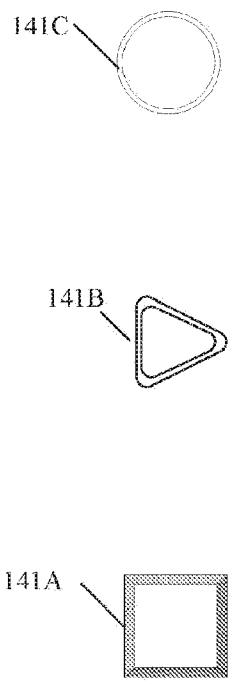


FIG. 6

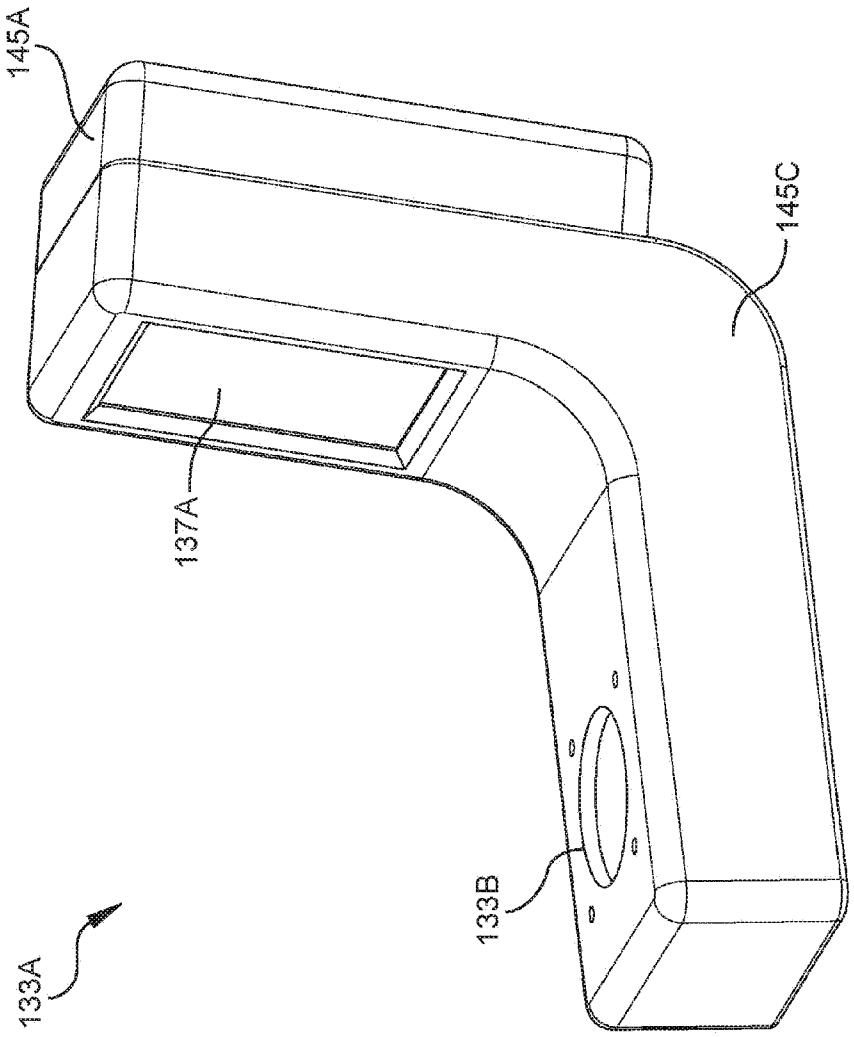


FIG. 6A

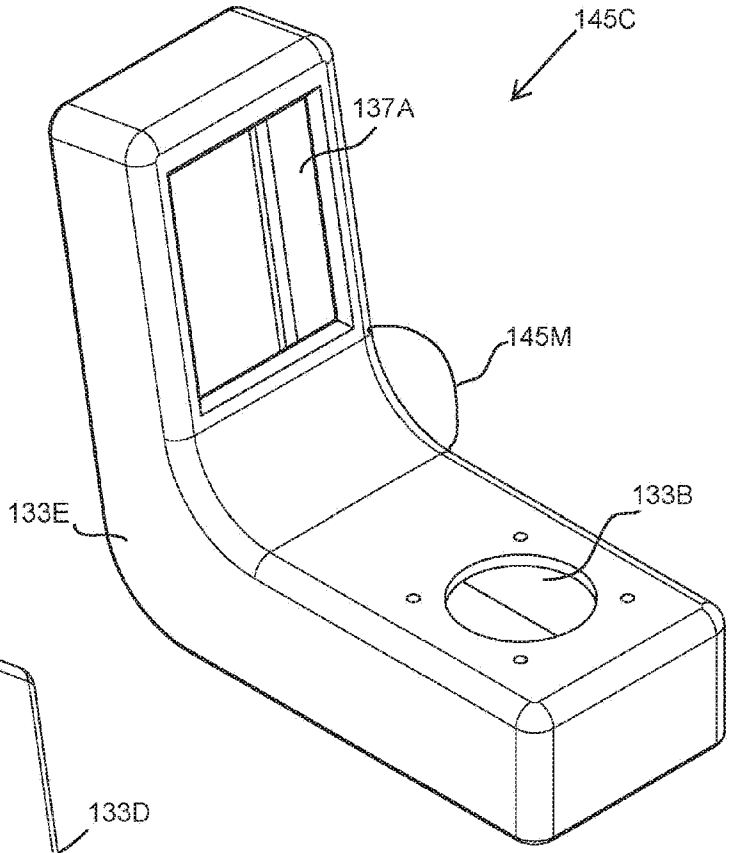


FIG. 6B1

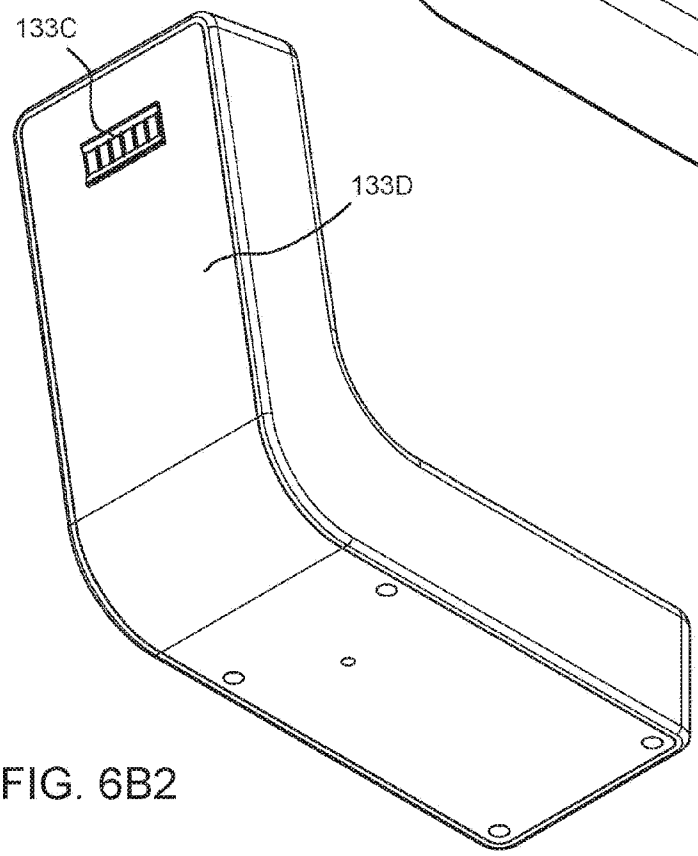


FIG. 6B2

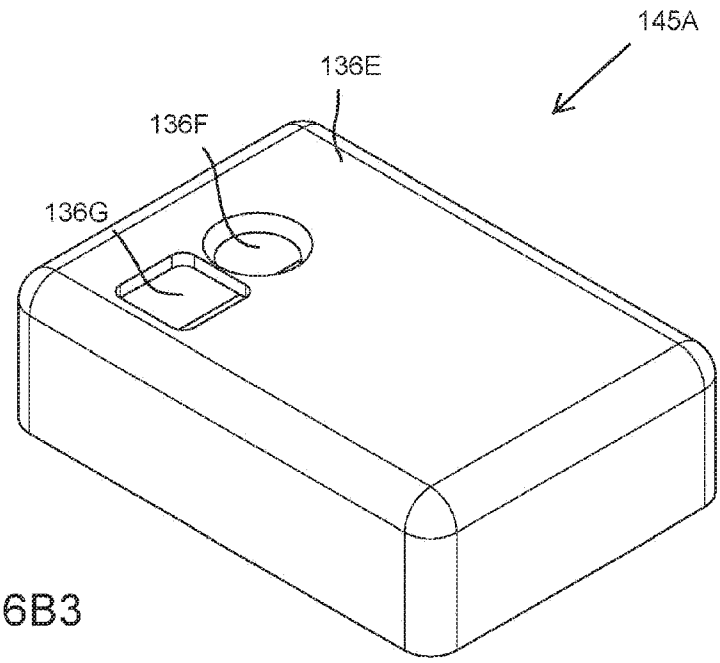


FIG. 6B3

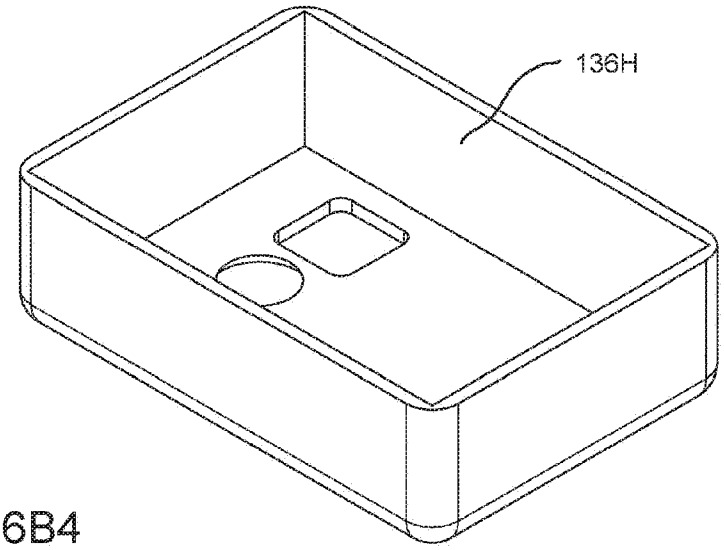


FIG. 6B4

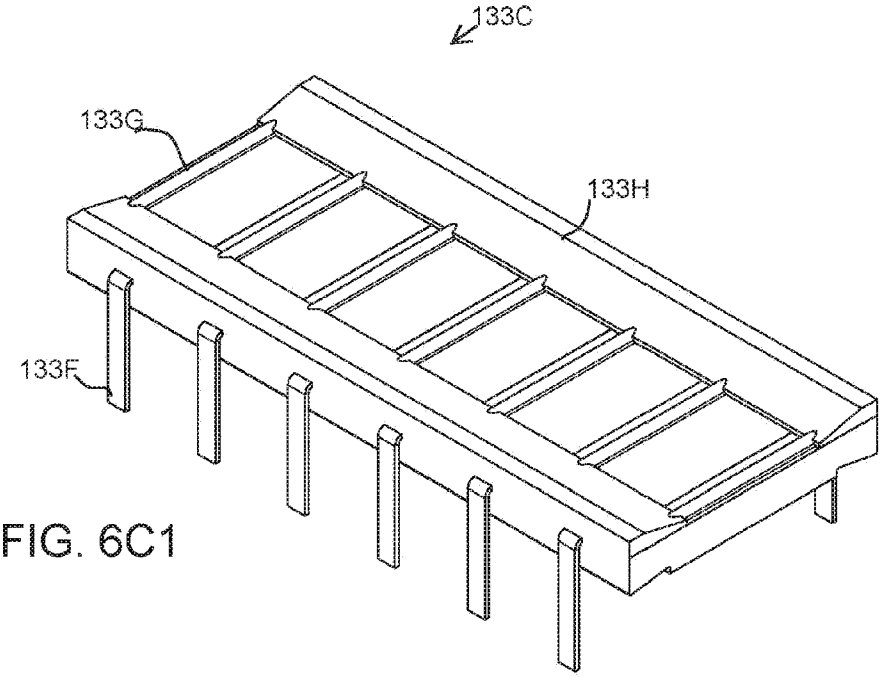


FIG. 6C1

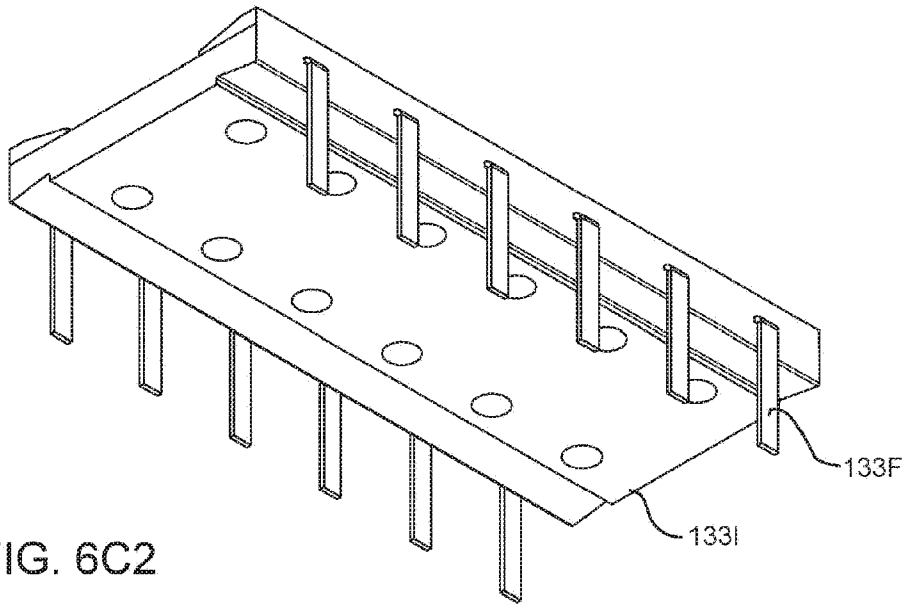


FIG. 6C2

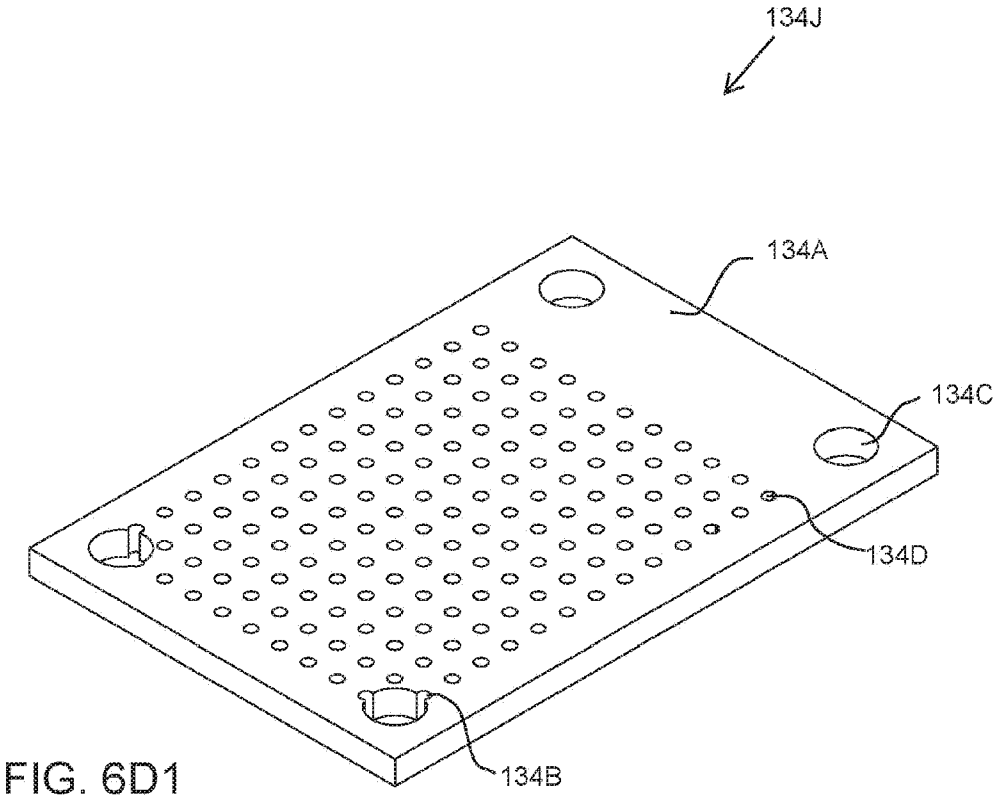


FIG. 6D1

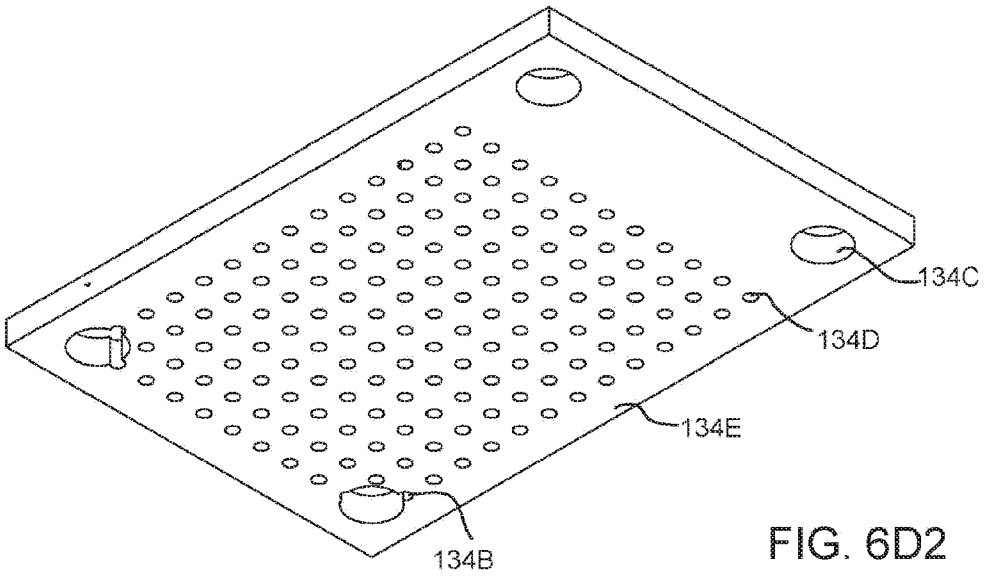


FIG. 6D2

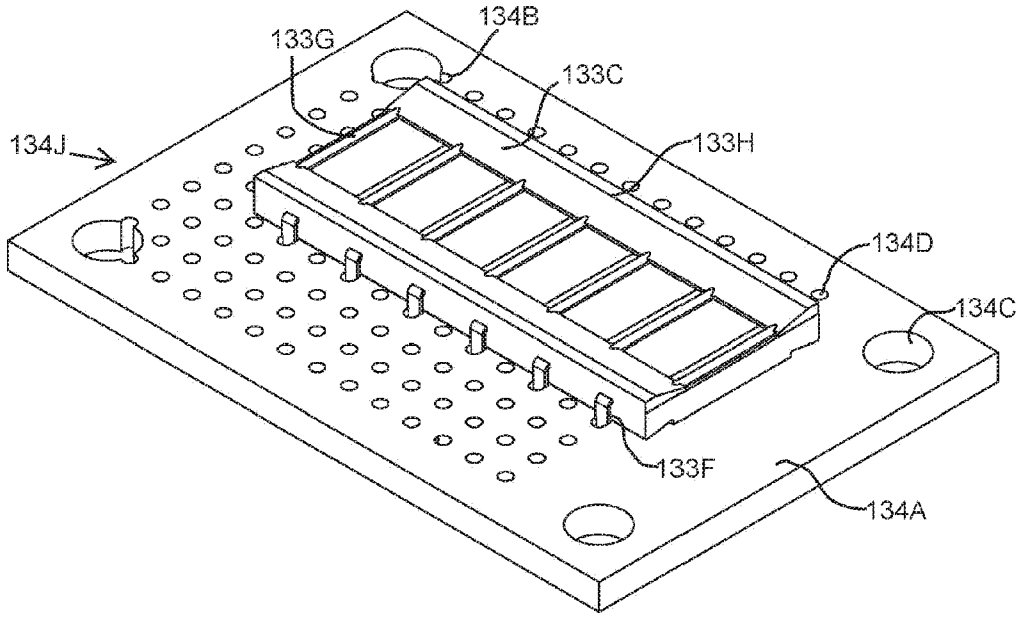


FIG. 6D3

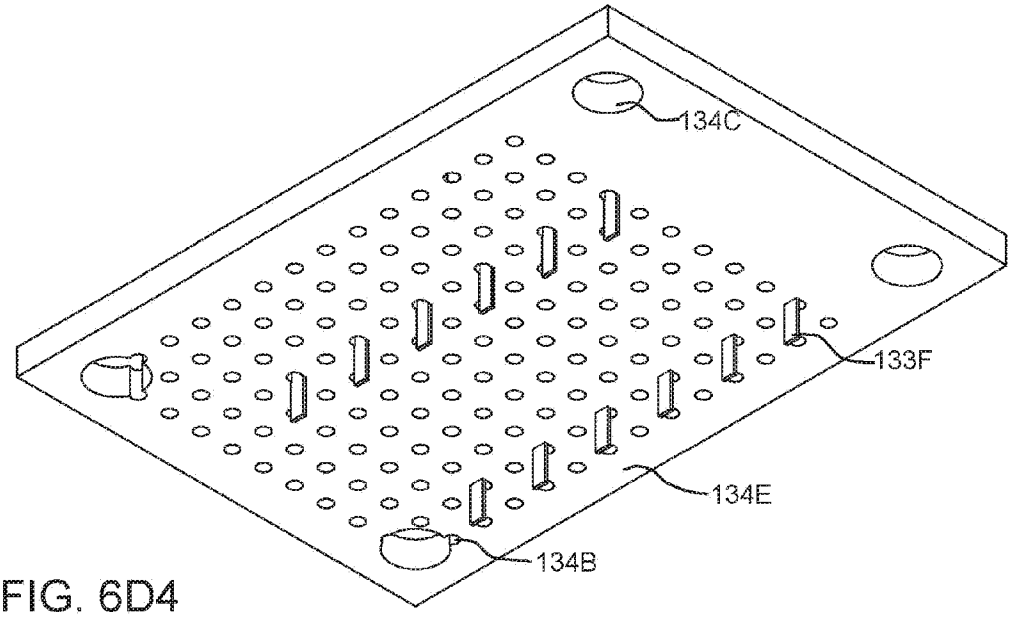


FIG. 6D4

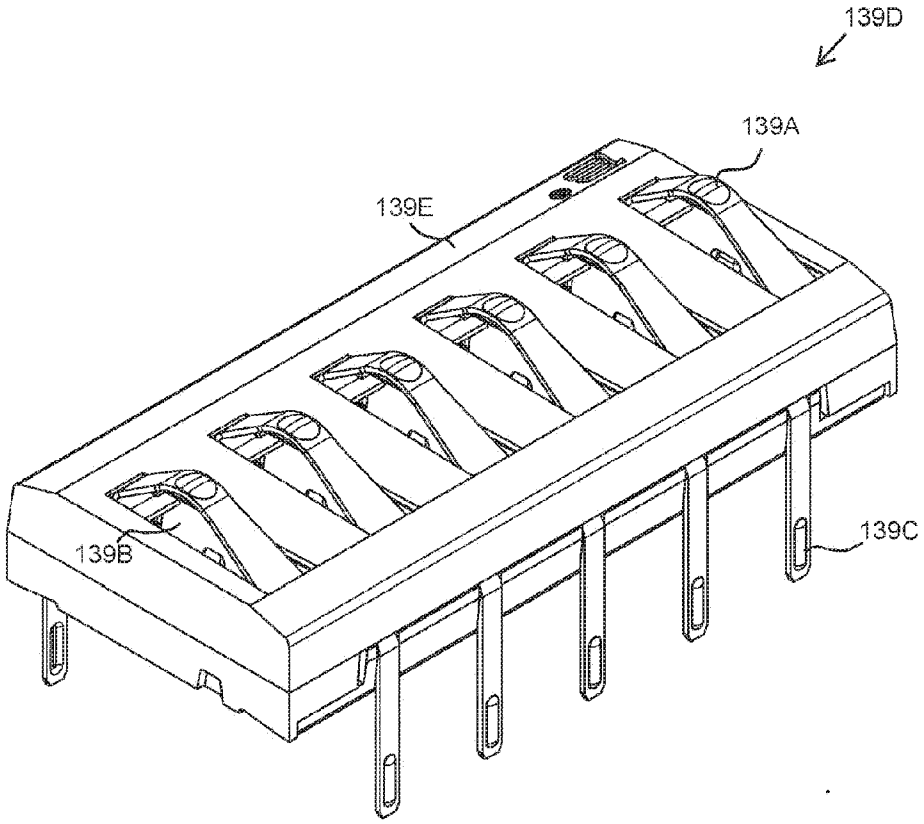


FIG. 6D5

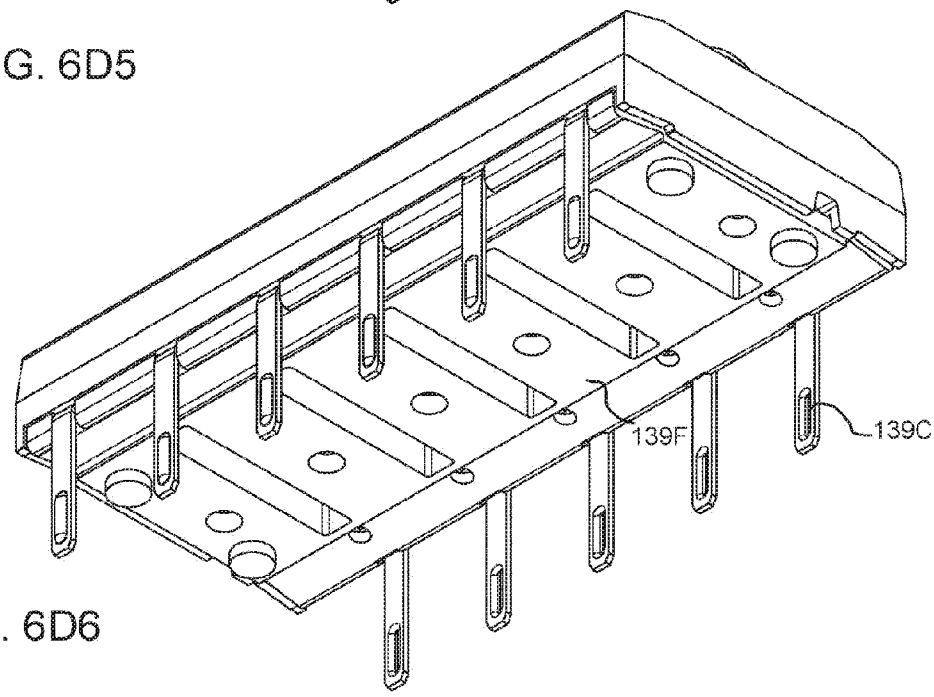


FIG. 6D6

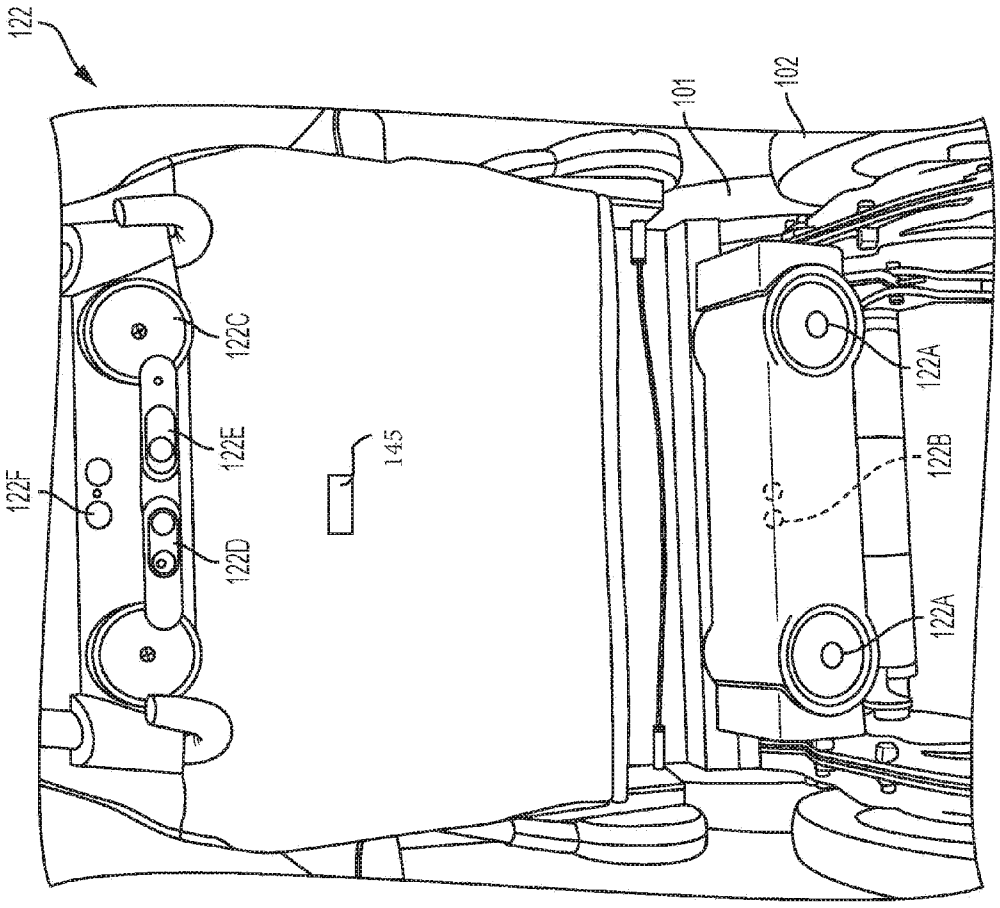


FIG. 6E

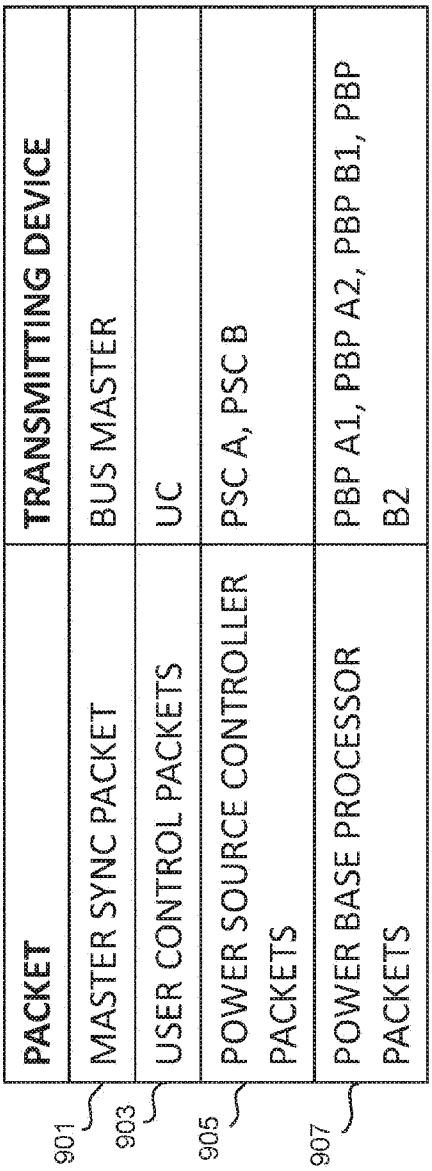
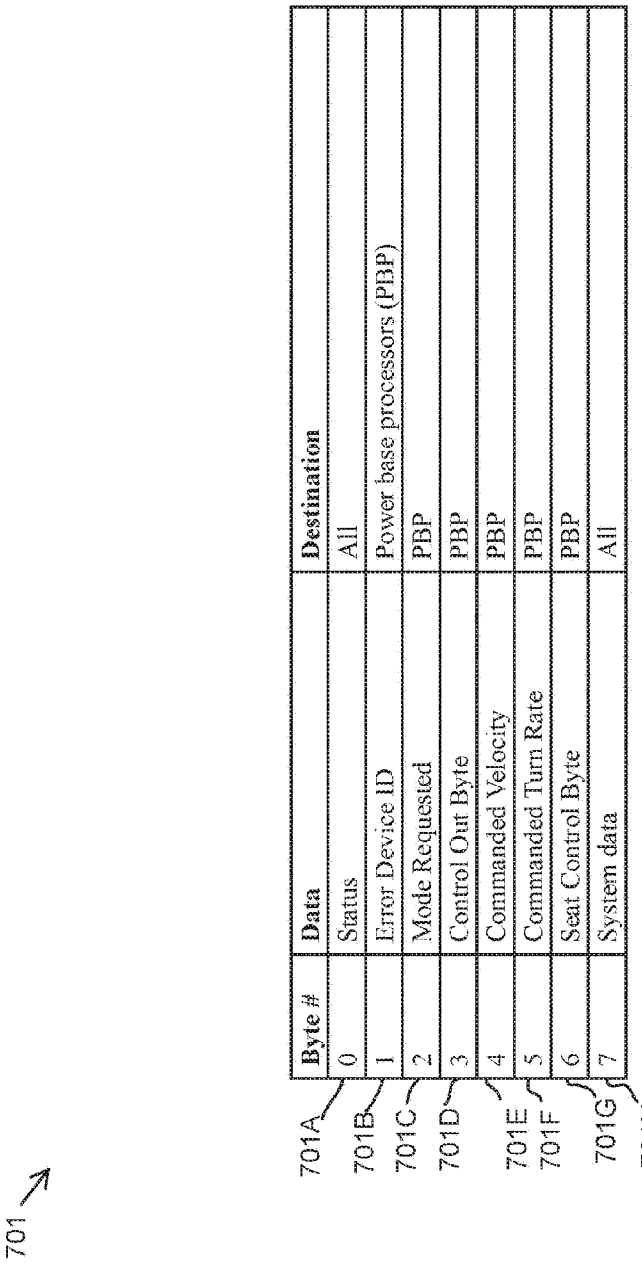

PACKET
TRANSMITTING DEVICE
MASTER SYNC PACKET
BUS MASTER
USER CONTROL PACKETS
UC
POWER SOURCE CONTROLLER PACKETS
PSC A, PSC B
POWER BASE PROCESSOR PACKETS
PBP A1, PBP A2, PBP B1, PBP B2
901
903
905
907

FIG. 7A



Byte #	Data	Destination
0	Status	All
1	Error Device ID	Power base processors (PBP)
2	Mode Requested	PBP
3	Control Out Byte	PBP
4	Commanded Velocity	PBP
5	Commanded Turn Rate	PBP
6	Seat Control Byte	PBP
7	System data	All

FIG. 7B

Data	Bit	Meaning
Request Type	7	0 = Normal Request Mode being requested should be processed by the PBP as a maintenance of current mode or a normal mode change request. 1 = Special Request This bit indicates a special mode request that requires situation-dependant consideration.
Requested Mode	6:0	mode byte codes set in association with possible modes

FIG. 7C

DK/EP 4043982 T3

	Data	Bit	Meaning
801A	OK to Power Down	7	0= Not OK
801B	Drive Selection	6	0= Drive 1 1= Drive 2
801C	Emergency Power Off Request	5	0= Normal 1= EPO request sequence is in process (clears when aborted/completed)
801D	Calibration State	4	1= Request for User Calibration state Note: Transition is only valid in Utility mode.
801E	Mode Restriction	3	0 = No restriction 1 = Balance critical functions should be restricted
801F	User Training	2	0= Disabled 1= Enabled
801G	Joystick centered	1:0	00= Invalid 10= Centered 01= Not Centered 11= Invalid

FIG. 7D

DK/EP 4043982 T3

	Data	Bit	Meaning
921	Frame Lean Command	7:6	00 = Invalid 10 = Lean Forward 01 = Lean Rearward 11 = Idle
	Reserved	5:2	-
923	Seat Height Command	1:0	00 = Invalid 10 = Lower Seat Down 01 = Raise Seat Up 11 = Idle

FIG. 7E

DK/EP 4043982 T3

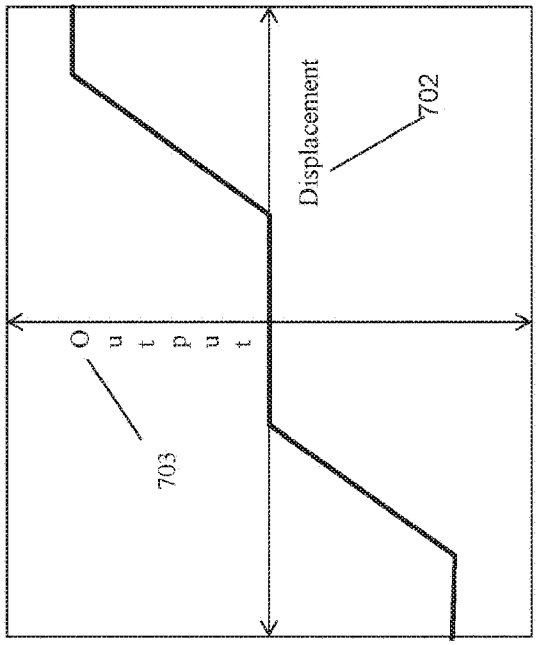


FIG. 8

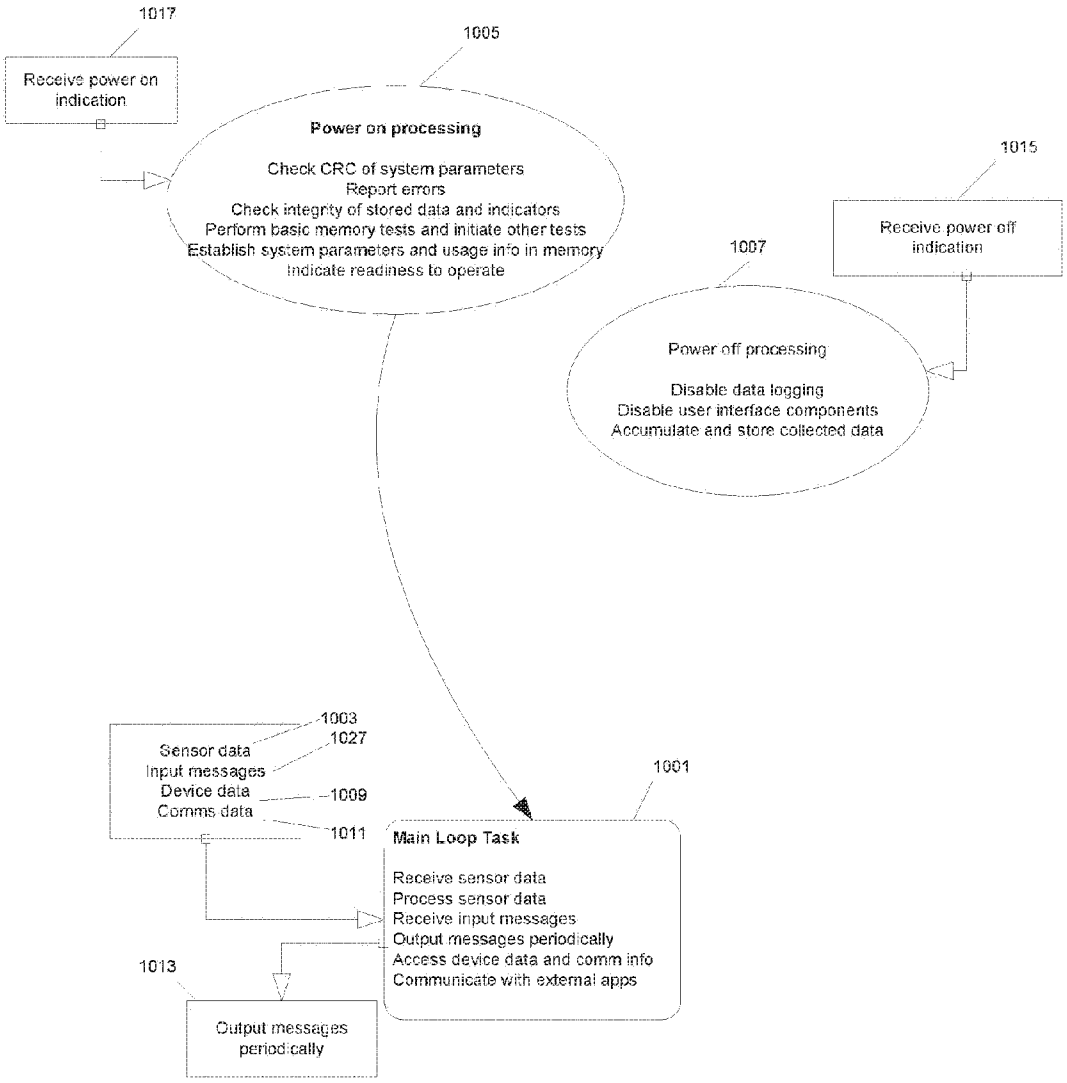

1017
Receive power on indication
1005
Power on processing
Check CRC of system parameters
Report errors
Check integrity of stored data and indicators
Perform basic memory tests and initiate other tests
Establish system parameters and usage info in memory
Indicate readiness to operate
1015
Receive power off indication
1007
Power off processing
Disable data logging
Disable user interface components
Accumulate and store collected data
1003
Sensor data
1027
Input messages
Device data
1009
Comms data
1011
1001
Main Loop Task
Receive sensor data
Process sensor data
Receive input messages
Output messages periodically
Access device data and comm info
Communicate with external apps
1013
Output messages periodically

FIG. 9

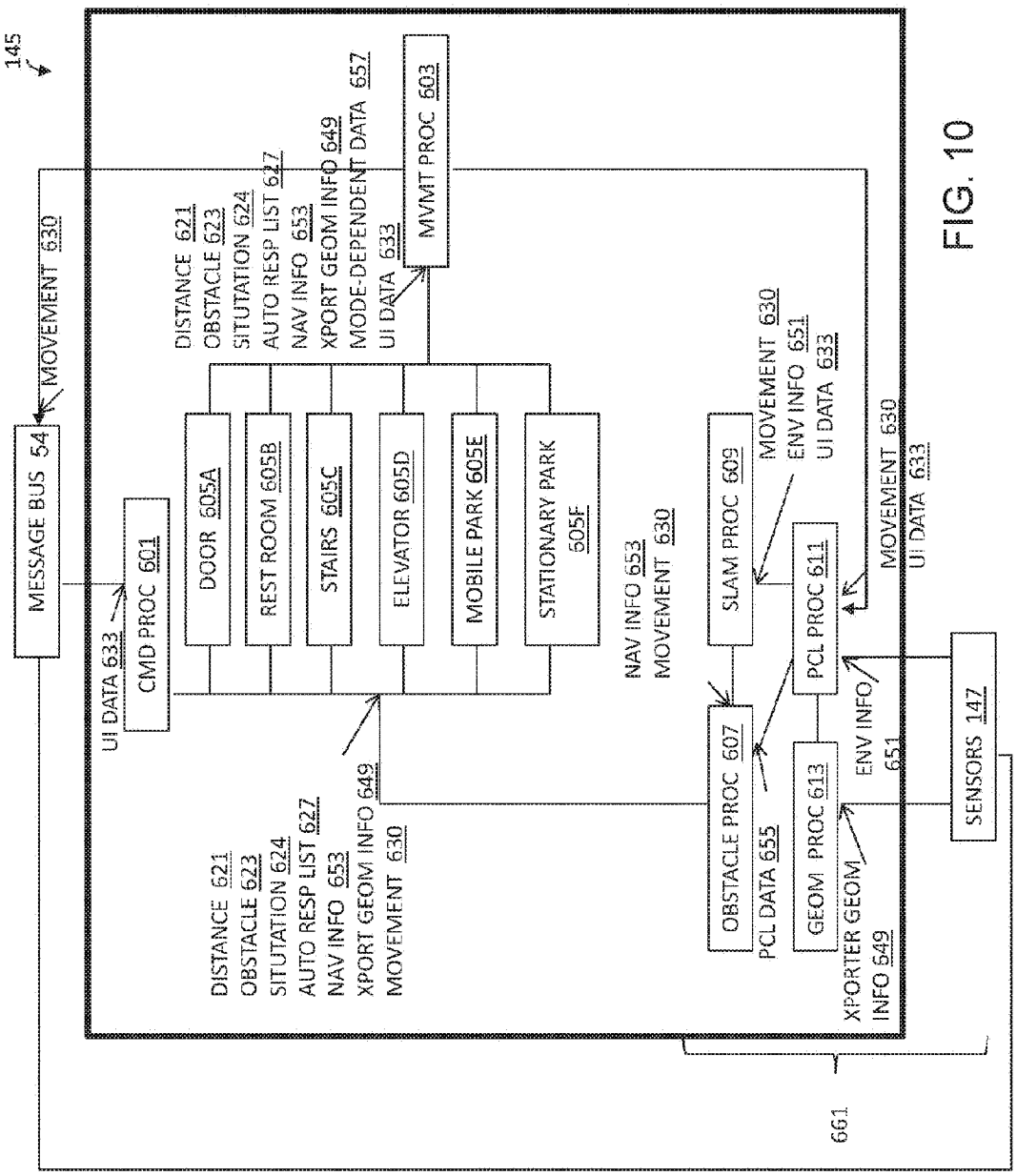

145
MESSAGE BUS 54
MOVEMENT 630
UI DATA 633
CMD PROC 601
DOOR 605A
REST ROOM 605B
STAIRS 605C
ELEVATOR 605D
MOBILE PARK 605E
STATIONARY PARK 605F
DISTANCE 621
OBSTACLE 623
SITUTATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
MODE-DEPENDENT DATA 657
UI DATA 633
MVMT PROC 603
NAV INFO 653
MOVEMENT 630
SLAM PROC 609
MOVEMENT 630
ENV INFO 651
UI DATA 633
PCL PROC 611
MOVEMENT 630
UI DATA 633
DISTANCE 621
OBSTACLE 623
SITUTATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
MOVEMENT 630
OBSTACLE PROC 607
PCL DATA 655
GEOM PROC 613
ENV INFO 651
XPORTER GEOM INFO 649
SENSORS 147
661

FIG. 10

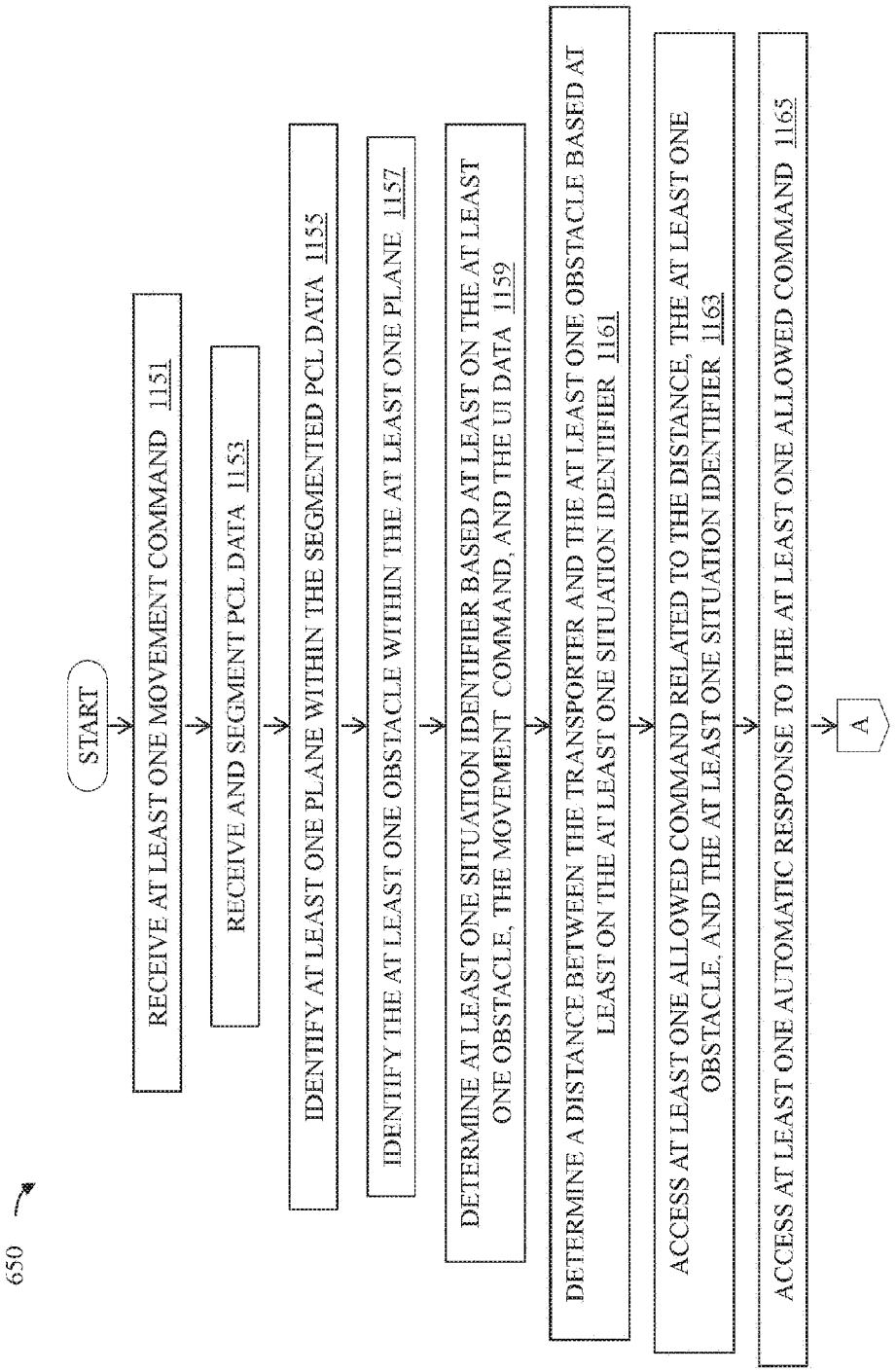

650
START
RECEIVE AT LEAST ONE MOVEMENT COMMAND 1151
RECEIVE AND SEGMENT PCL DATA 1153
IDENTIFY AT LEAST ONE PLANE WITHIN THE SEGMENTED PCL DATA 1155
IDENTIFY THE AT LEAST ONE OBSTACLE WITHIN THE AT LEAST ONE PLANE 1157
DETERMINE AT LEAST ONE SITUATION IDENTIFIER BASED AT LEAST ON THE AT LEAST ONE OBSTACLE, THE MOVEMENT COMMAND, AND THE UI DATA 1159
DETERMINE A DISTANCE BETWEEN THE TRANSPORTER AND THE AT LEAST ONE OBSTACLE BASED AT LEAST ON THE AT LEAST ONE SITUATION IDENTIFIER 1161
ACCESS AT LEAST ONE ALLOWED COMMAND RELATED TO THE DISTANCE, THE AT LEAST ONE OBSTACLE, AND THE AT LEAST ONE SITUATION IDENTIFIER 1163
ACCESS AT LEAST ONE AUTOMATIC RESPONSE TO THE AT LEAST ONE ALLOWED COMMAND 1165
A

FIG. 11A1

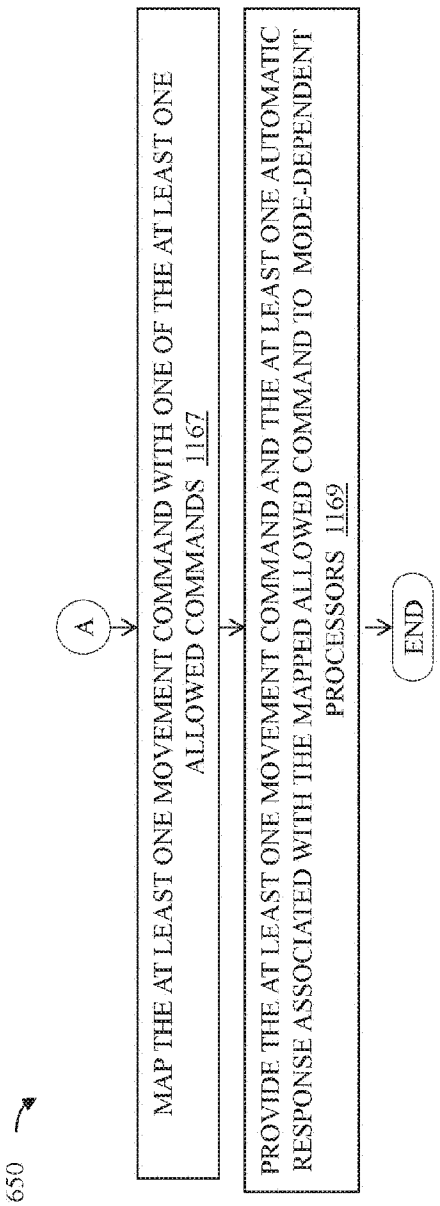

650
A
MAP THE AT LEAST ONE MOVEMENT COMMAND WITH ONE OF THE AT LEAST ONE ALLOWED COMMANDS 1167
PROVIDE THE AT LEAST ONE MOVEMENT COMMAND AND THE AT LEAST ONE AUTOMATIC RESPONSE ASSOCIATED WITH THE MAPPED ALLOWED COMMAND TO MODE-DEPENDENT PROCESSORS 1169
END

FIG. 11A2

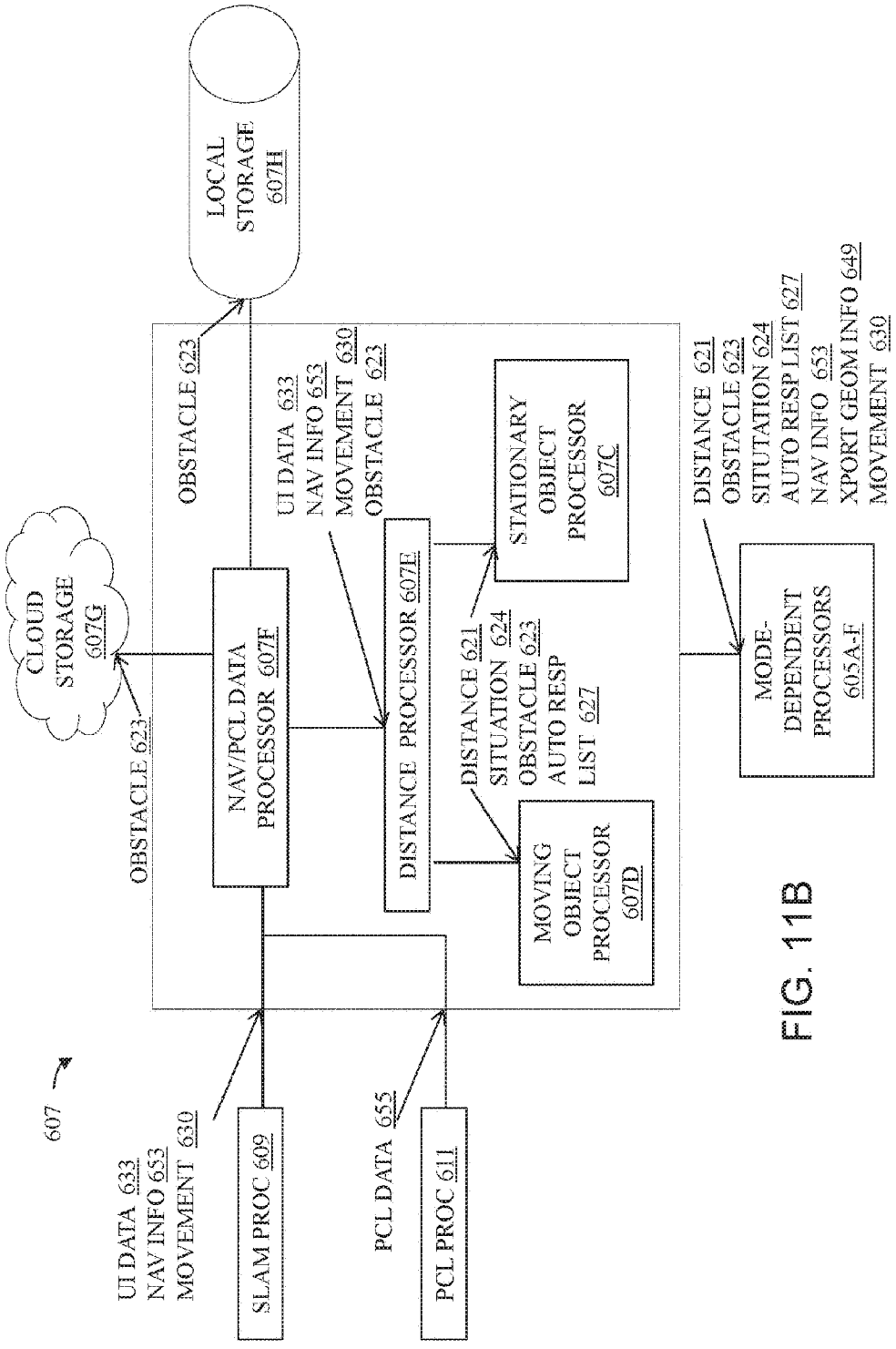

607
UI DATA 633
NAV INFO 653
MOVEMENT 630
SLAM PROC 609
PCL DATA 655
PCL PROC 611
OBSTACLE 623
NAV/PCL DATA PROCESSOR 607F
CLOUD STORAGE 607G
LOCAL STORAGE 607H
OBSTACLE 623
UI DATA 633
NAV INFO 653
MOVEMENT 630
OBSTACLE 623
DISTANCE PROCESSOR 607E
DISTANCE 621
SITUATION 624
OBSTACLE 623
AUTO RESP LIST 627
MOVING OBJECT PROCESSOR 607D
STATIONARY OBJECT PROCESSOR 607C
DISTANCE 621
OBSTACLE 623
SITUTATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
MOVEMENT 630
MODE-DEPENDENT PROCESSORS 605A-F

FIG. 11B

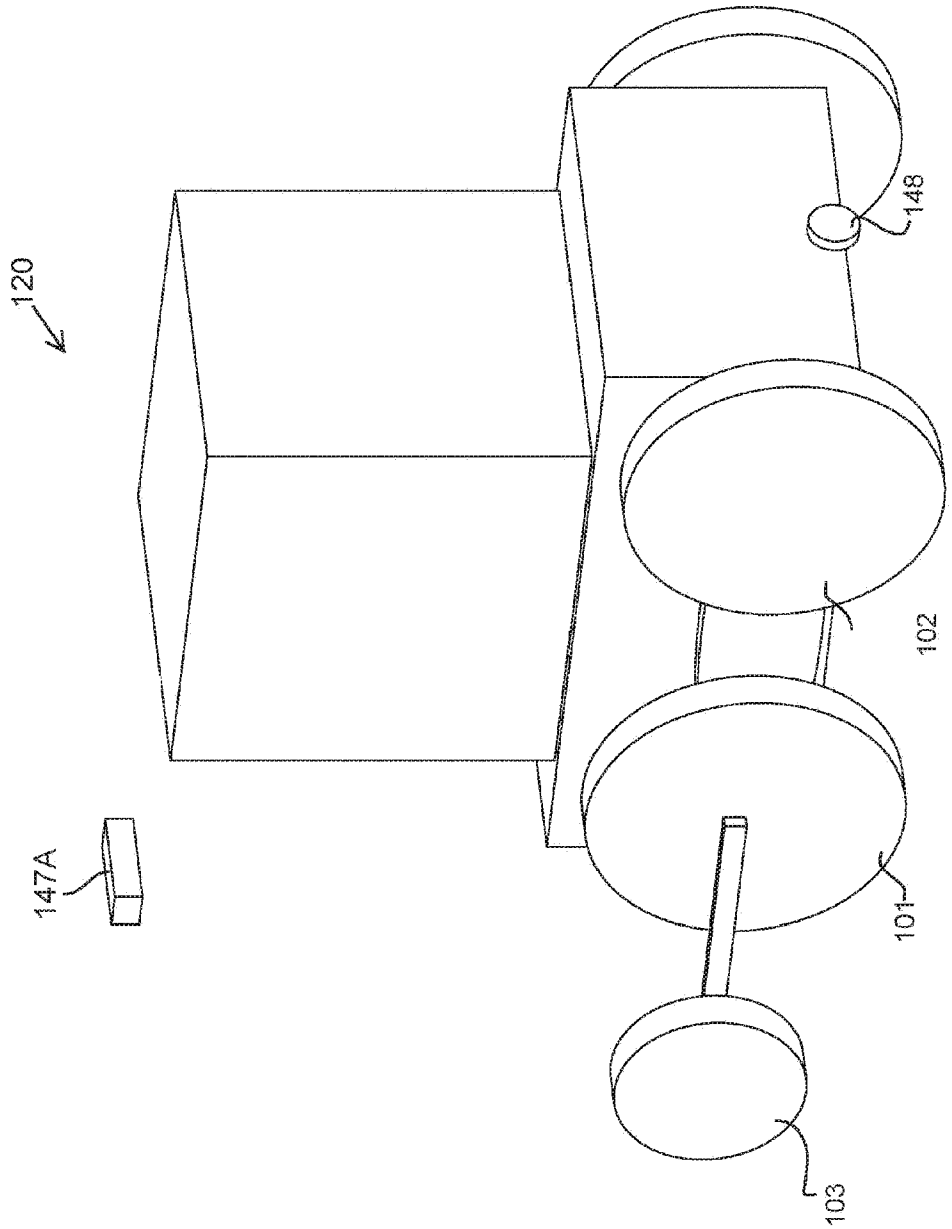


FIG. 12A

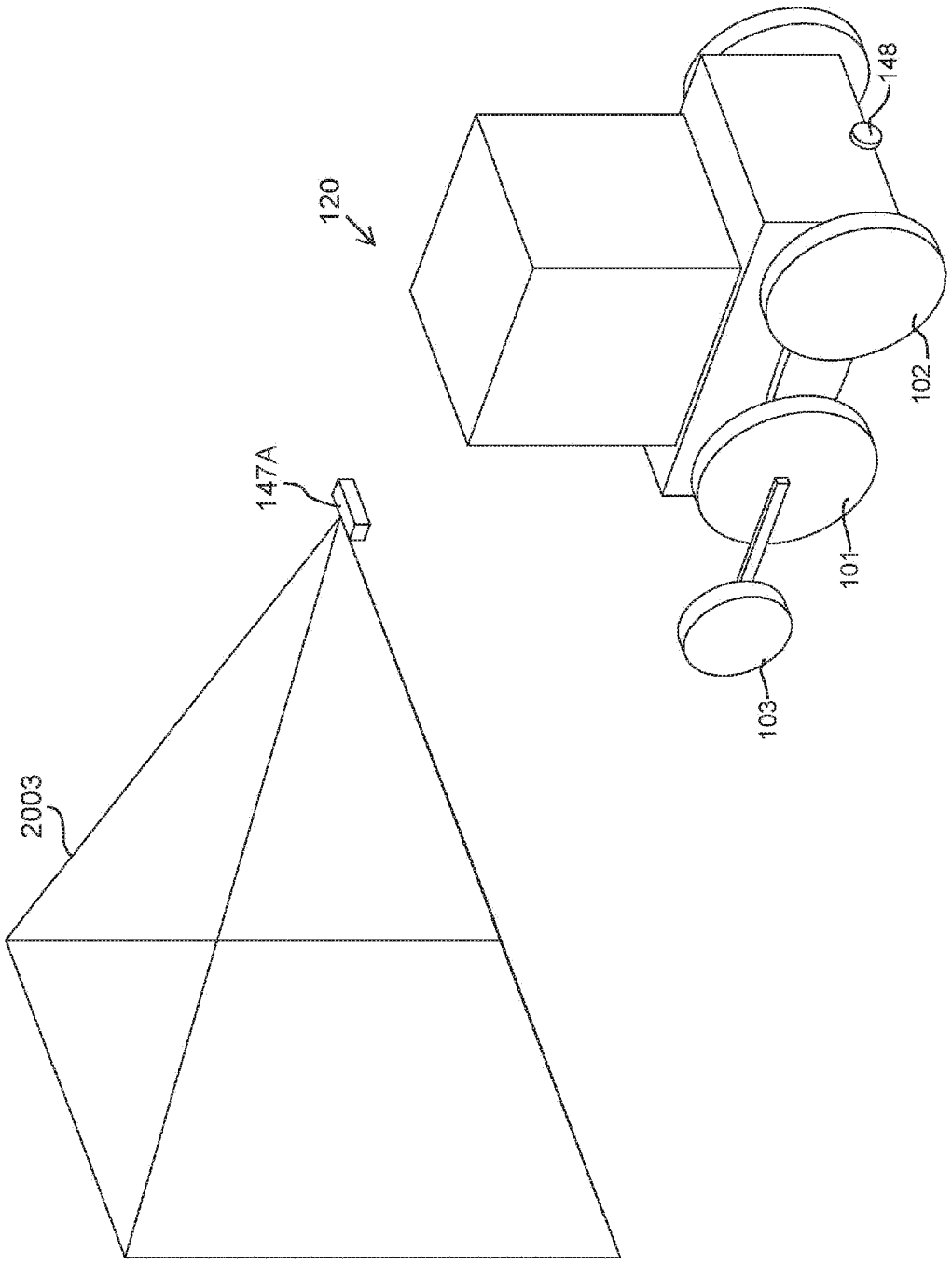


FIG. 12B

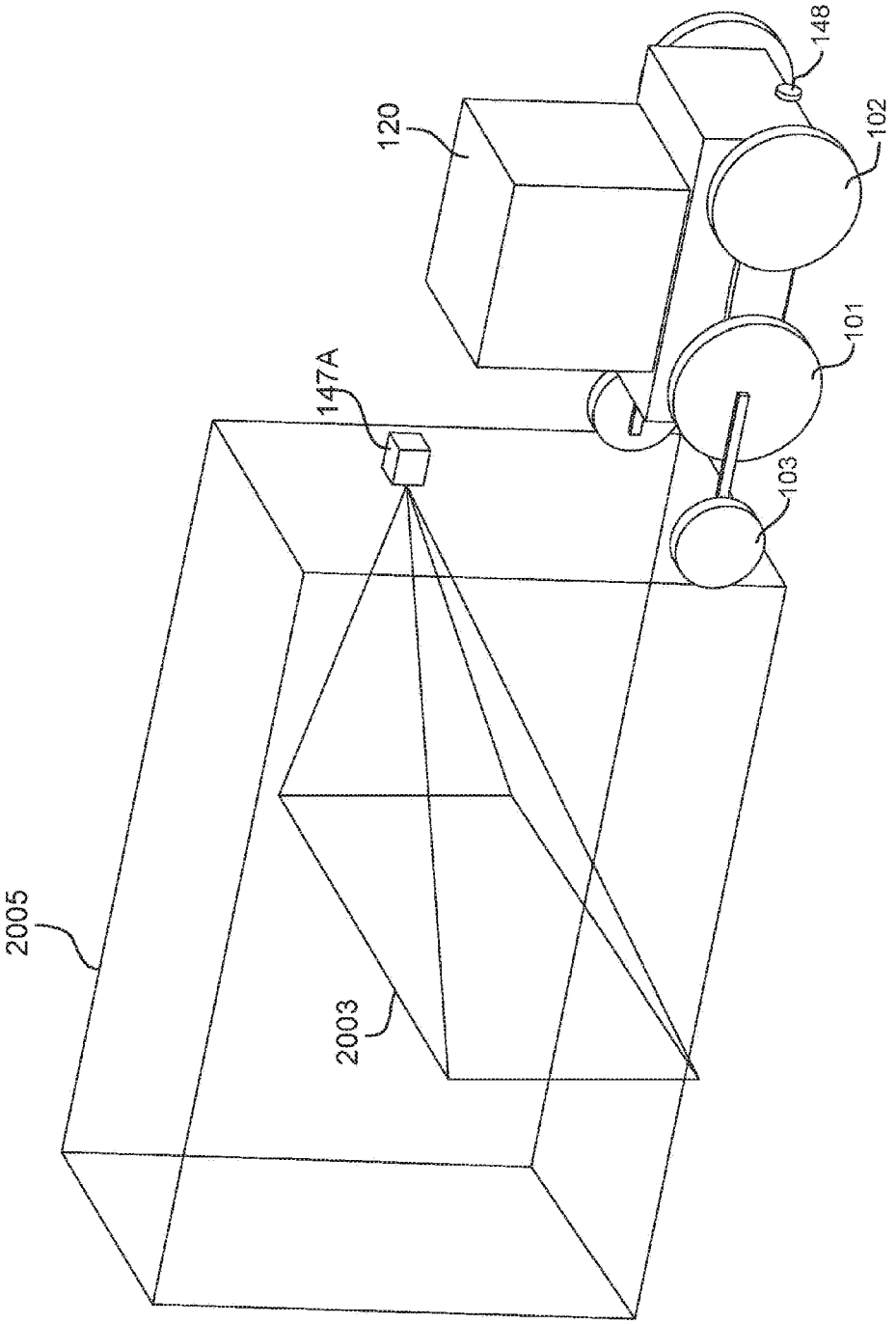


FIG. 12C

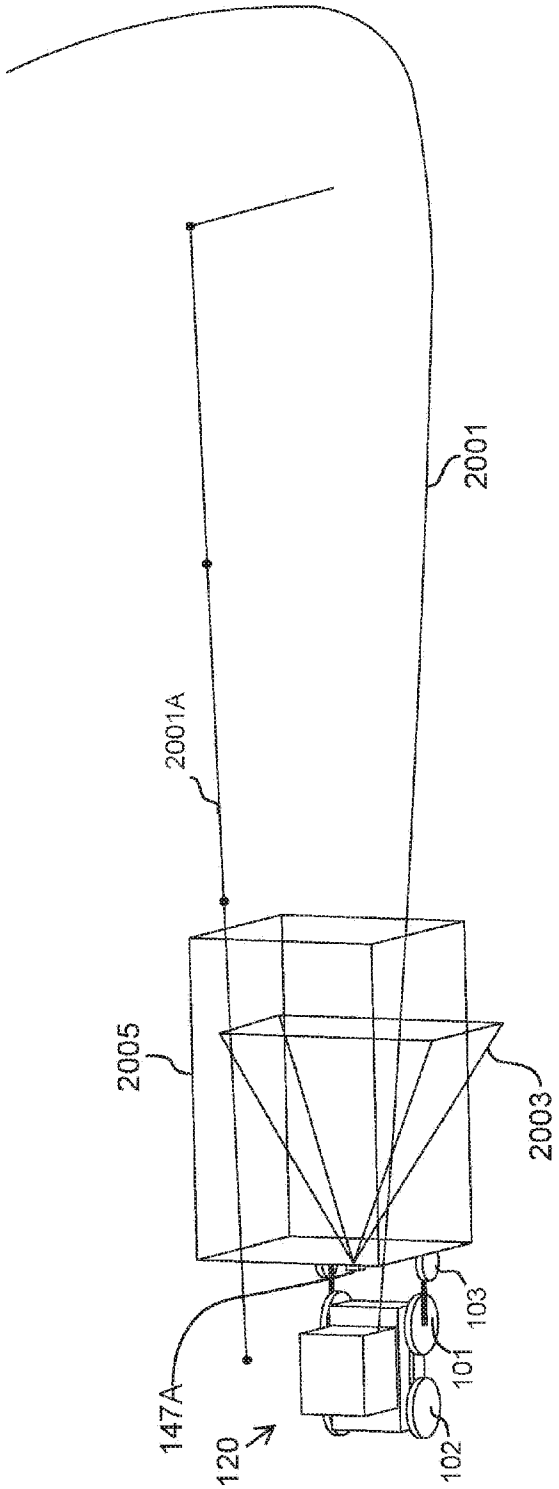


FIG. 12D

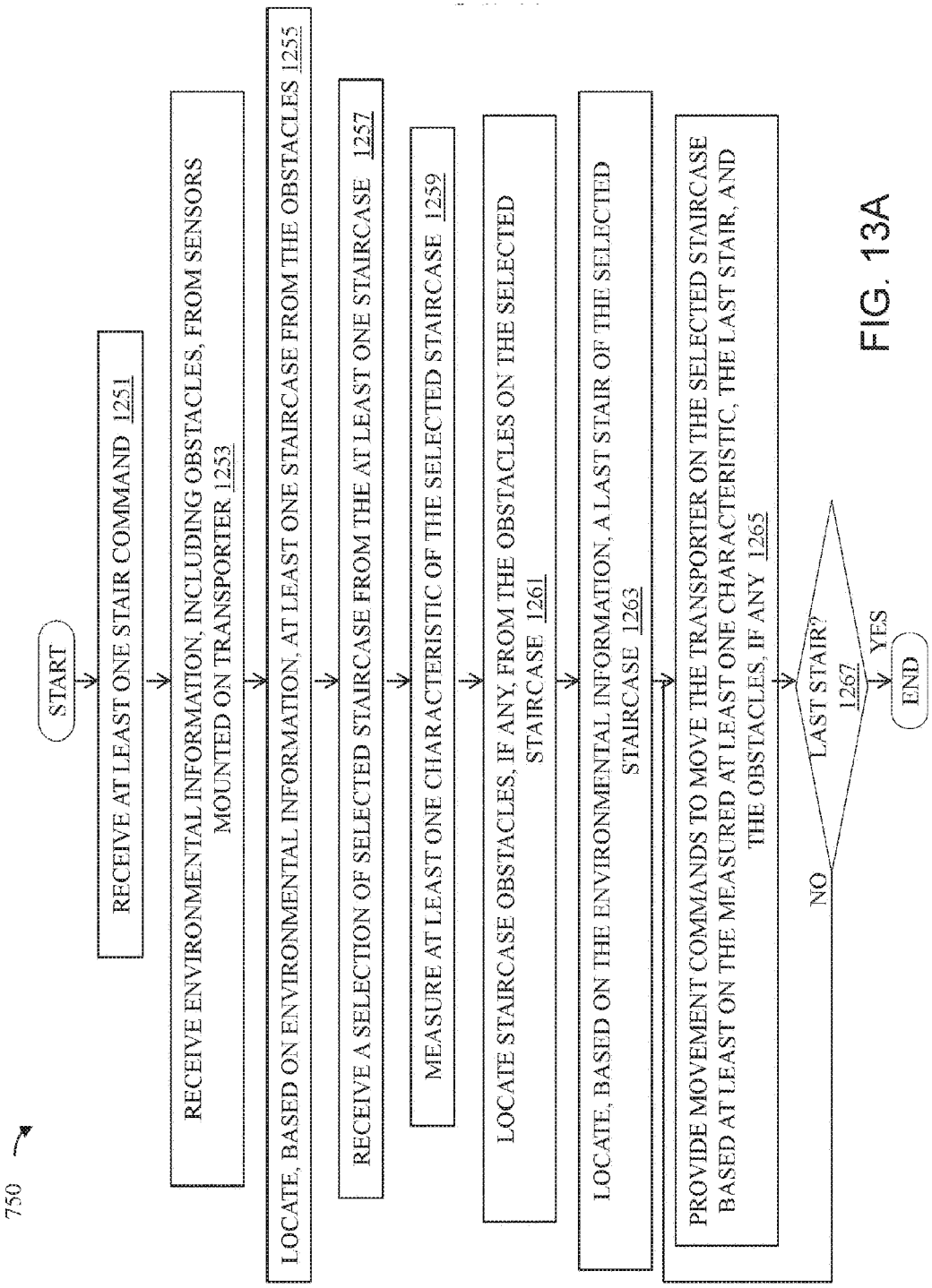

750
START
RECEIVE AT LEAST ONE STAIR COMMAND 1251
RECEIVE ENVIRONMENTAL INFORMATION, INCLUDING OBSTACLES, FROM SENSORS MOUNTED ON TRANSPORTER 1253
LOCATE, BASED ON ENVIRONMENTAL INFORMATION, AT LEAST ONE STAIRCASE FROM THE OBSTACLES 1255
RECEIVE A SELECTION OF SELECTED STAIRCASE FROM THE AT LEAST ONE STAIRCASE 1257
MEASURE AT LEAST ONE CHARACTERISTIC OF THE SELECTED STAIRCASE 1259
LOCATE STAIRCASE OBSTACLES, IF ANY, FROM THE OBSTACLES ON THE SELECTED STAIRCASE 1261
LOCATE, BASED ON THE ENVIRONMENTAL INFORMATION, A LAST STAIR OF THE SELECTED STAIRCASE 1263
PROVIDE MOVEMENT COMMANDS TO MOVE THE TRANSPORTER ON THE SELECTED STAIRCASE BASED AT LEAST ON THE MEASURED AT LEAST ONE CHARACTERISTIC, THE LAST STAIR, AND THE OBSTACLES, IF ANY 1265
LAST STAIR? 1267
NO
YES
END

FIG. 13A

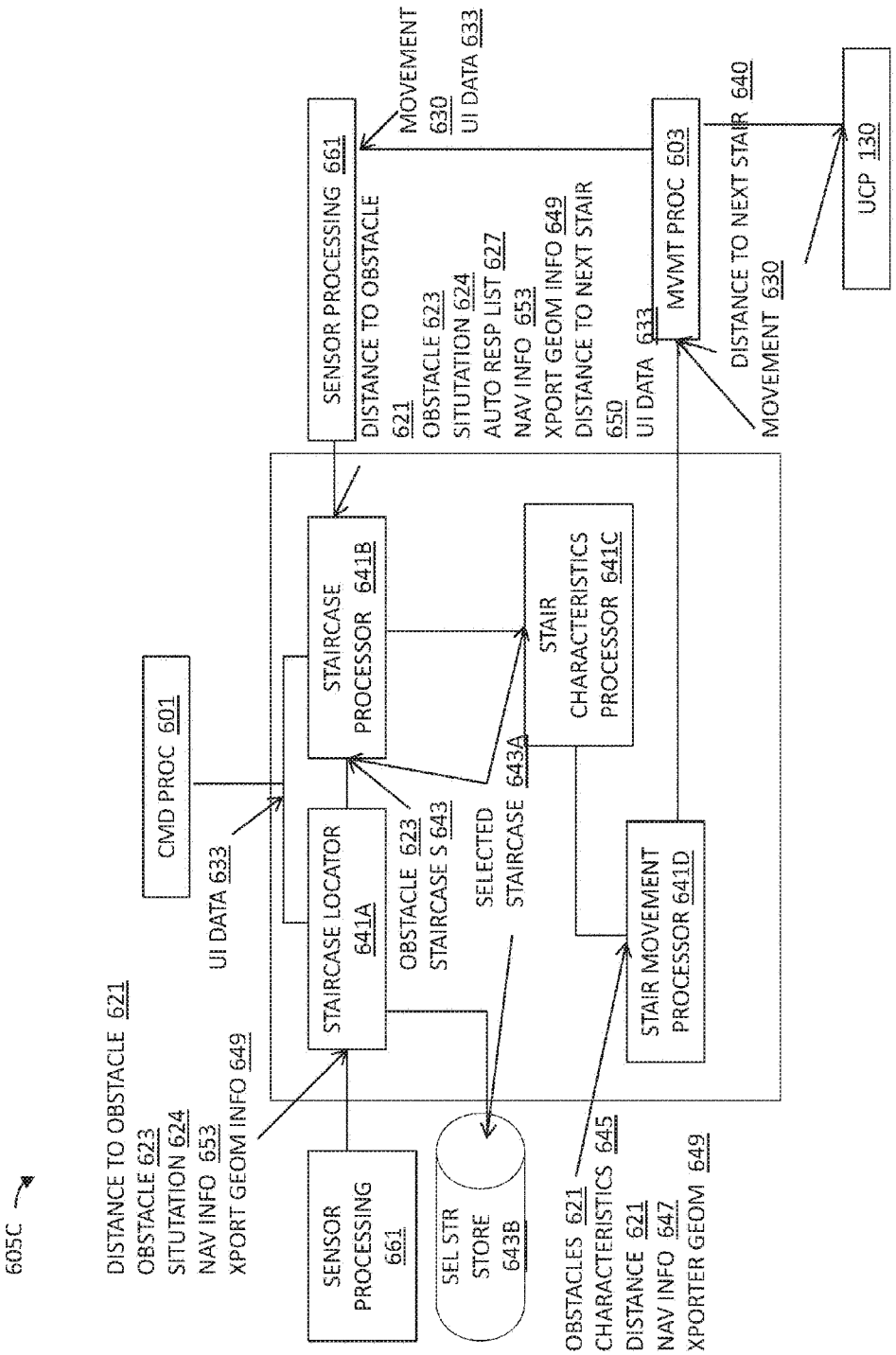

605C
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUTATION 624
NAV INFO 653
XPORT GEOM INFO 649
CMD PROC 601
UI DATA 633
SENSOR PROCESSING 661
SEL STR STORE 643B
STAIRCASE LOCATOR 641A
OBSTACLE 623
STAIRCASE S 643
SELECTED STAIRCASE 643A
STAIRCASE PROCESSOR 641B
STAIR CHARACTERISTICS PROCESSOR 641C
STAIR MOVEMENT PROCESSOR 641D
OBSTACLES 621
CHARACTERISTICS 645
DISTANCE 621
NAV INFO 647
XPORTER GEOM 649
SENSOR PROCESSING 661
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUTATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
DISTANCE TO NEXT STAIR 650
UI DATA 633
MOVEMENT 630
UI DATA 633
MVMT PROC 603
DISTANCE TO NEXT STAIR 640
MOVEMENT 630
UCP 130

FIG. 13B

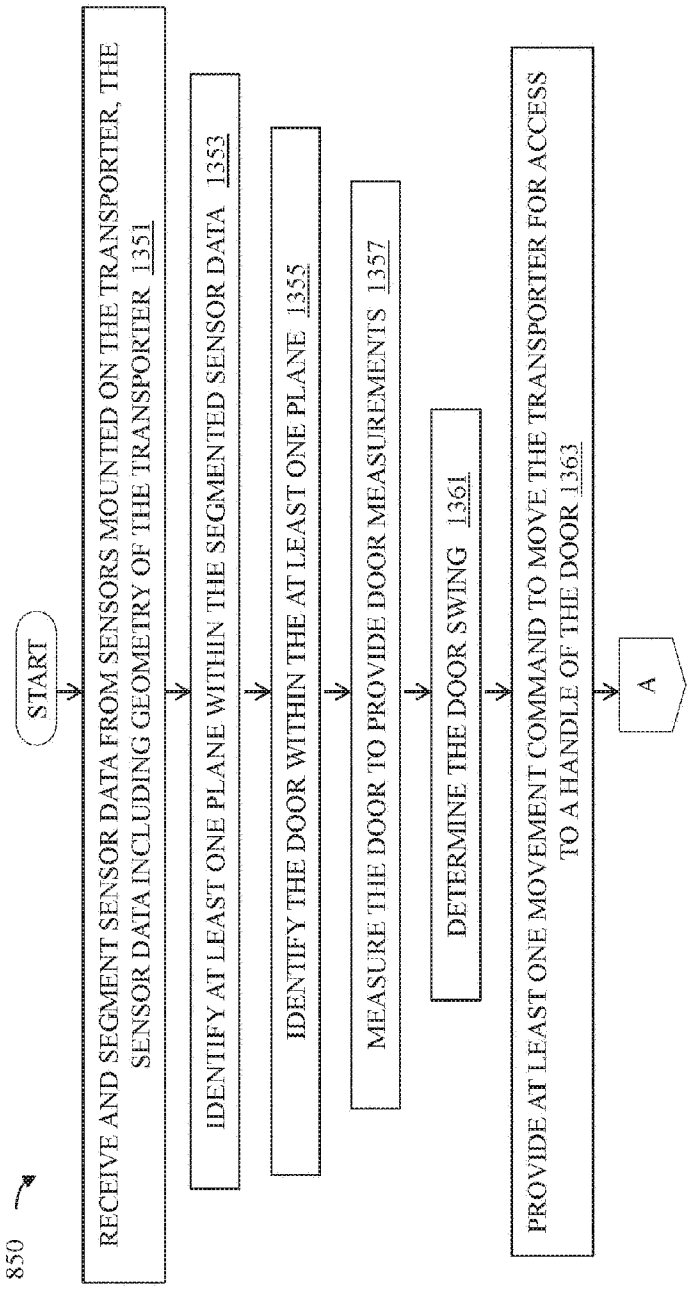

850
START
RECEIVE AND SEGMENT SENSOR DATA FROM SENSORS MOUNTED ON THE TRANSPORTER, THE SENSOR DATA INCLUDING GEOMETRY OF THE TRANSPORTER 1351
IDENTIFY AT LEAST ONE PLANE WITHIN THE SEGMENTED SENSOR DATA 1353
IDENTIFY THE DOOR WITHIN THE AT LEAST ONE PLANE 1355
MEASURE THE DOOR TO PROVIDE DOOR MEASUREMENTS 1357
DETERMINE THE DOOR SWING 1361
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER FOR ACCESS TO A HANDLE OF THE DOOR 1363
A

FIG. 14A1

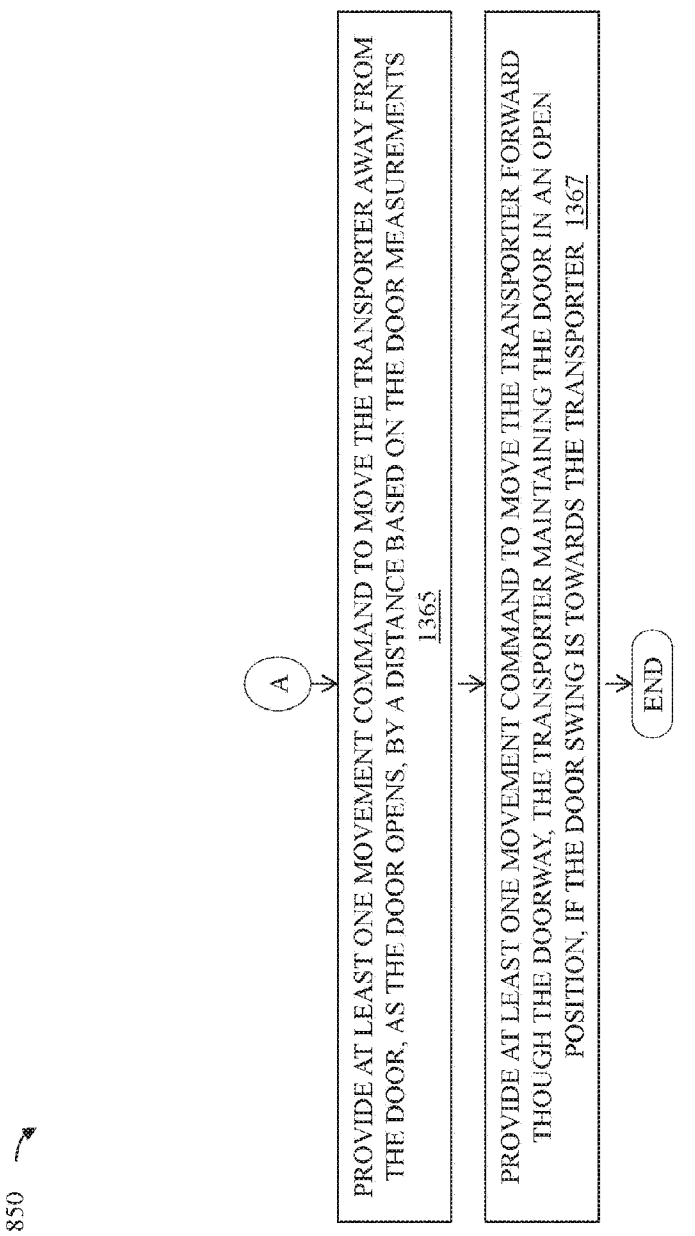

850
A
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER AWAY FROM THE DOOR, AS THE DOOR OPENS, BY A DISTANCE BASED ON THE DOOR MEASUREMENTS 1365
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER FORWARD THOUGH THE DOORWAY, THE TRANSPORTER MAINTAINING THE DOOR IN AN OPEN POSITION, IF THE DOOR SWING IS TOWARDS THE TRANSPORTER 1367
END

FIG. 14A2

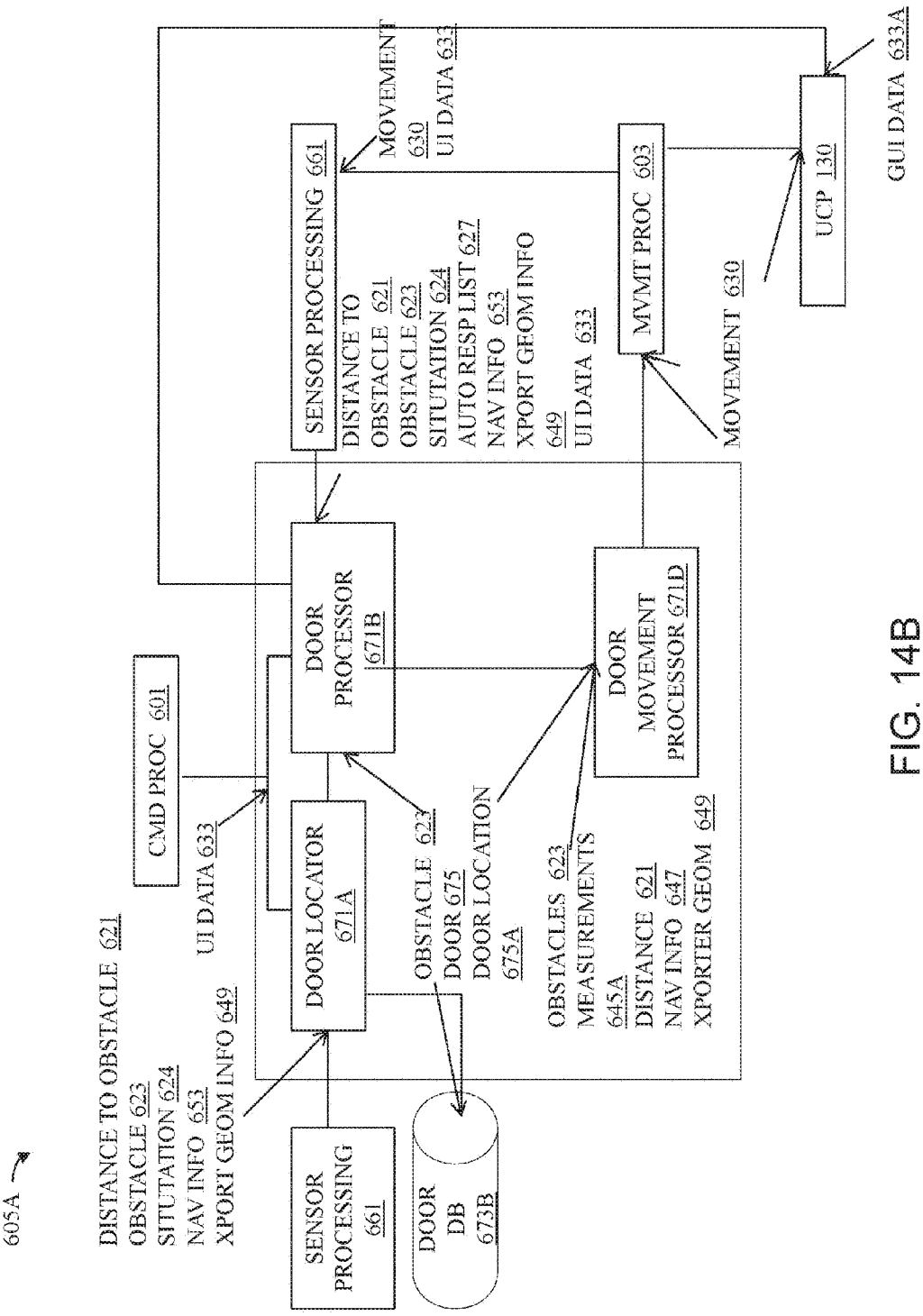

605A
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUTATION 624
NAV INFO 653
XPORT GEOM INFO 649
UI DATA 633
CMD PROC 601
SENSOR PROCESSING 661
DOOR LOCATOR 671A
DOOR PROCESSOR 671B
DOOR MOVEMENT PROCESSOR 671D
DOOR DB 673B
OBSTACLE 623
DOOR 675
DOOR LOCATION 675A
OBSTACLES 623 MEASUREMENTS 645A
DISTANCE 621
NAV INFO 647
XPORTER GEOM 649
SENSOR PROCESSING 661
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUTATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
UI DATA 633
MOVEMENT 630
UI DATA 633
MVMT PROC 603
MOVEMENT 630
UCP 130
GUI DATA 633A

FIG. 14B

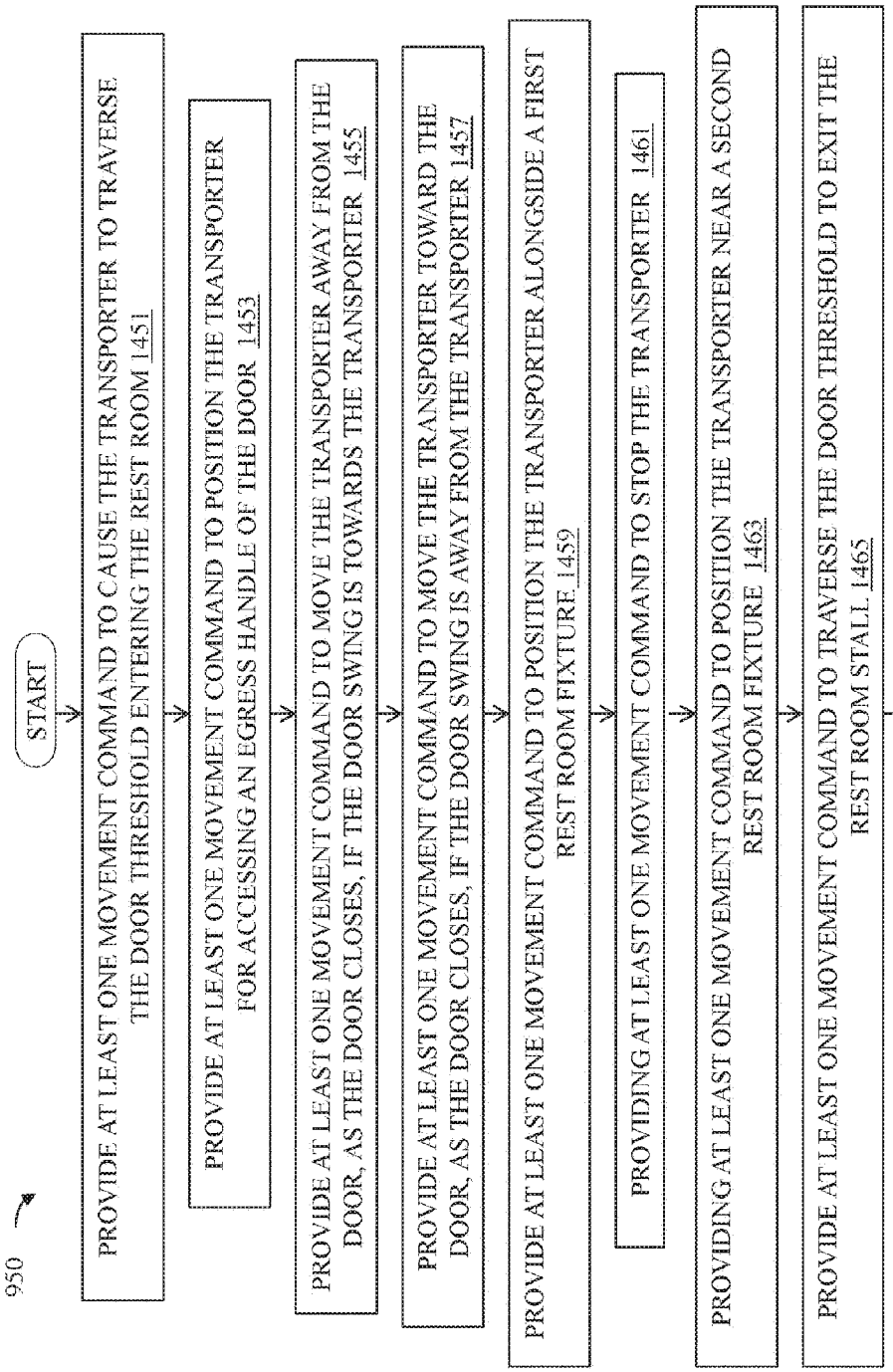

950
START
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO CAUSE THE TRANSPORTER TO TRAVERSE THE DOOR THRESHOLD ENTERING THE REST ROOM 1451
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO POSITION THE TRANSPORTER FOR ACCESSING AN EGRESS HANDLE OF THE DOOR 1453
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER AWAY FROM THE DOOR, AS THE DOOR CLOSES, IF THE DOOR SWING IS TOWARDS THE TRANSPORTER 1455
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER TOWARD THE DOOR, AS THE DOOR CLOSES, IF THE DOOR SWING IS AWAY FROM THE TRANSPORTER 1457
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO POSITION THE TRANSPORTER ALONGSIDE A FIRST REST ROOM FIXTURE 1459
PROVIDING AT LEAST ONE MOVEMENT COMMAND TO STOP THE TRANSPORTER 1461
PROVIDING AT LEAST ONE MOVEMENT COMMAND TO POSITION THE TRANSPORTER NEAR A SECOND REST ROOM FIXTURE 1463
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO TRAVERSE THE DOOR THRESHOLD TO EXIT THE REST ROOM STALL 1465
END

FIG. 15A

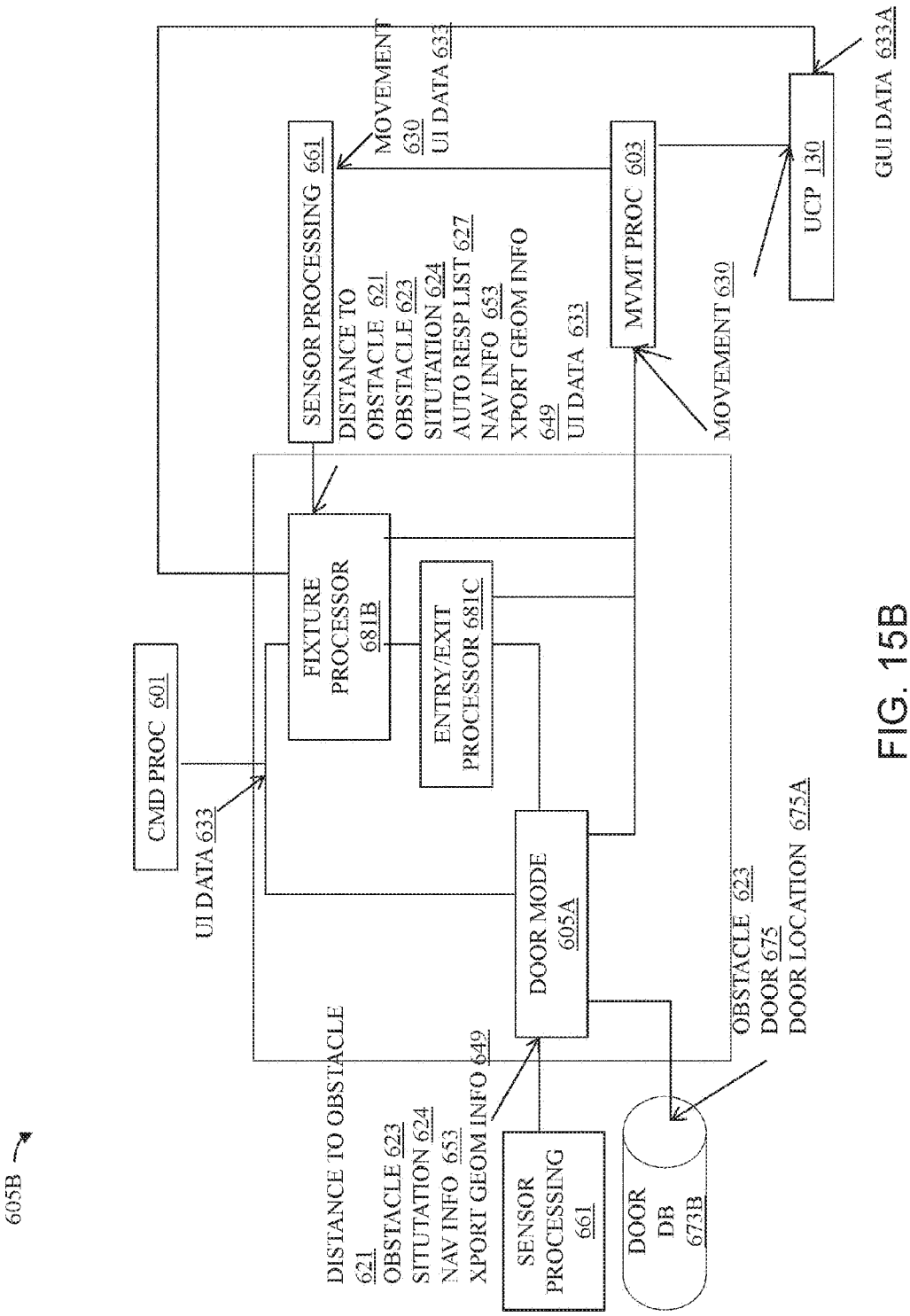

605B
CMD PROC 601
UI DATA 633
FIXTURE PROCESSOR 681B
ENTRY/EXIT PROCESSOR 681C
SENSOR PROCESSING 661
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUTATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
UI DATA 633
MOVEMENT 630
UI DATA 633
MVMT PROC 603
MOVEMENT 630
UCP 130
GUI DATA 633A
DOOR MODE 605A
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUTATION 624
NAV INFO 653
XPORT GEOM INFO 649
SENSOR PROCESSING 661
DOOR DB 673B
OBSTACLE 623
DOOR 675
DOOR LOCATION 675A

FIG. 15B

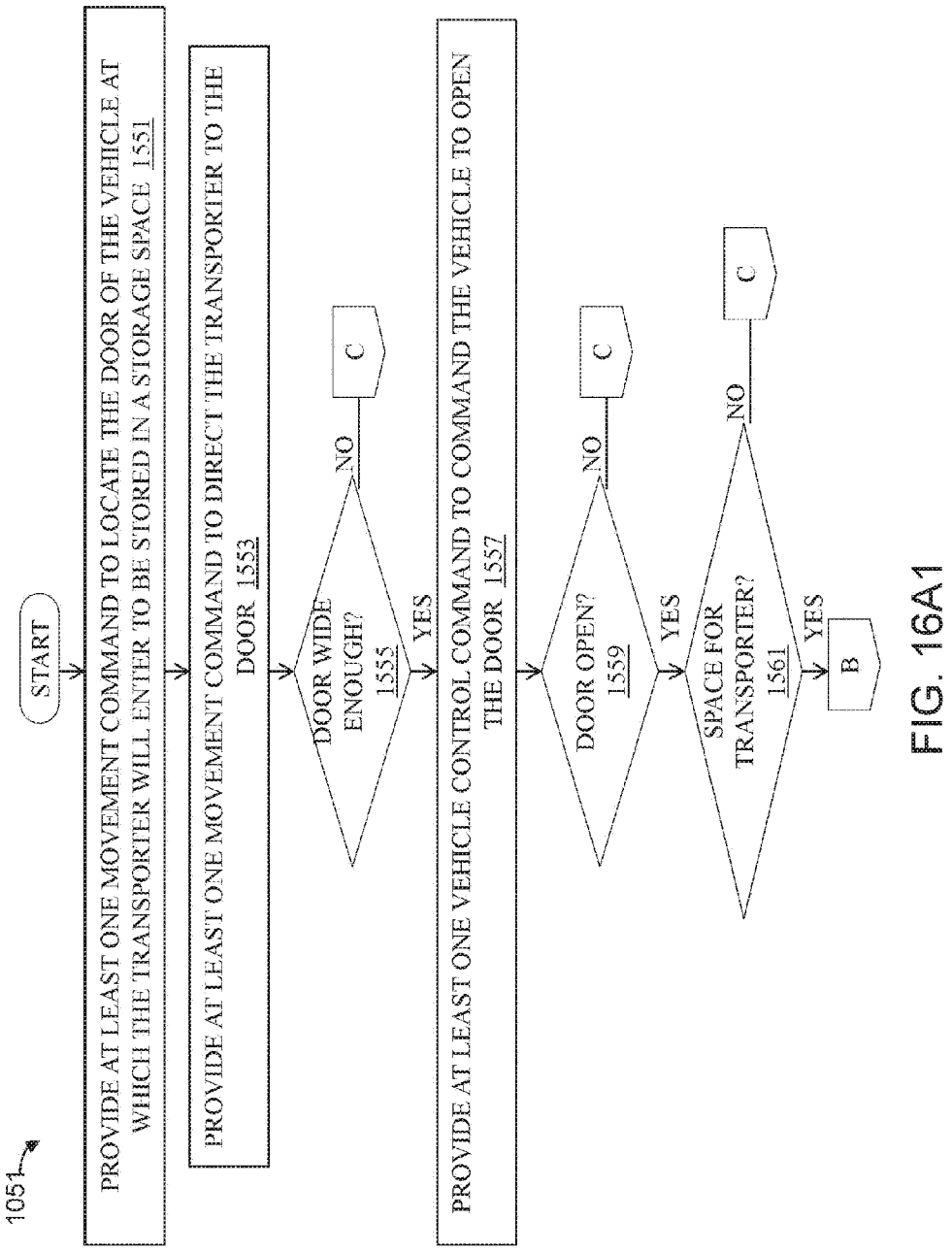

1051
START
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO LOCATE THE DOOR OF THE VEHICLE AT WHICH THE TRANSPORTER WILL ENTER TO BE STORED IN A STORAGE SPACE 1551
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO DIRECT THE TRANSPORTER TO THE DOOR 1553
DOOR WIDE ENOUGH? 1555
NO
C
YES
PROVIDE AT LEAST ONE VEHICLE CONTROL COMMAND TO COMMAND THE VEHICLE TO OPEN THE DOOR 1557
DOOR OPEN? 1559
NO
C
YES
SPACE FOR TRANSPORTER? 1561
NO
C
YES
B

FIG. 16A1

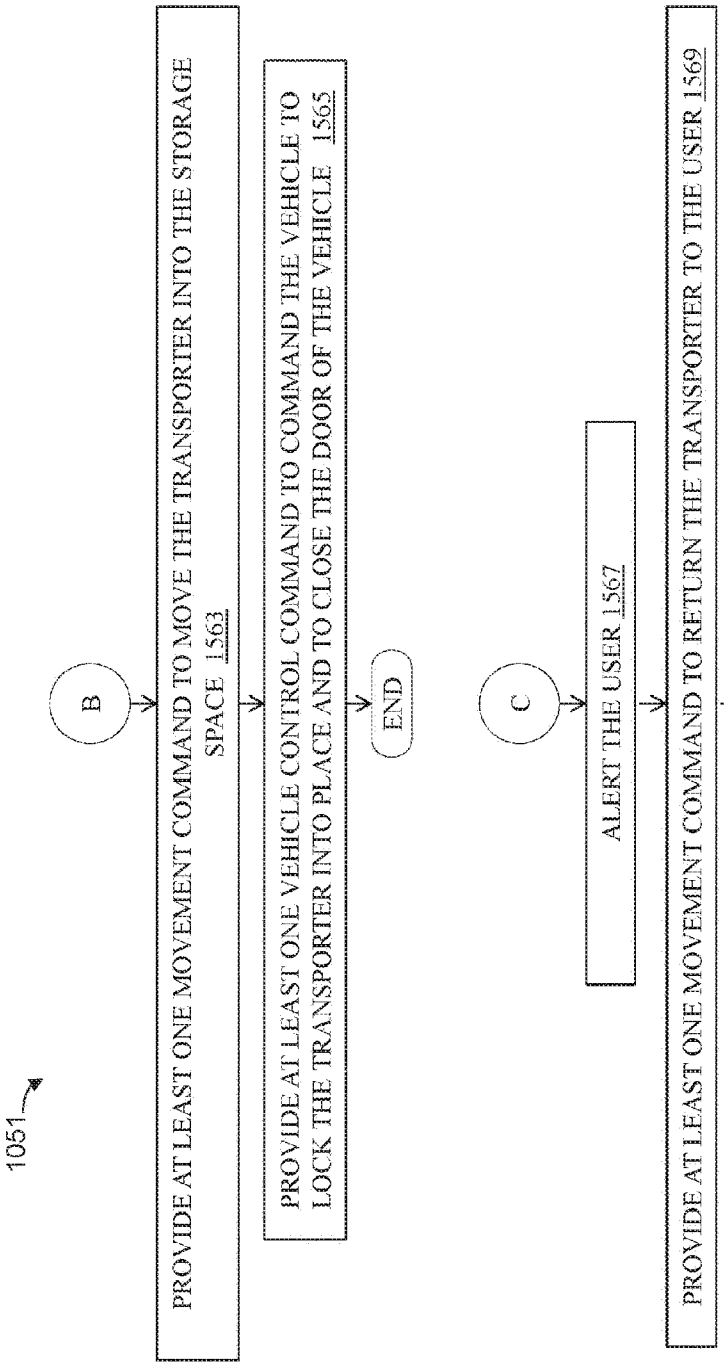

1051
B
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER INTO THE STORAGE SPACE 1563
PROVIDE AT LEAST ONE VEHICLE CONTROL COMMAND TO COMMAND THE VEHICLE TO LOCK THE TRANSPORTER INTO PLACE AND TO CLOSE THE DOOR OF THE VEHICLE 1565
END
C
ALERT THE USER 1567
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO RETURN THE TRANSPORTER TO THE USER 1569
END

FIG. 16A2

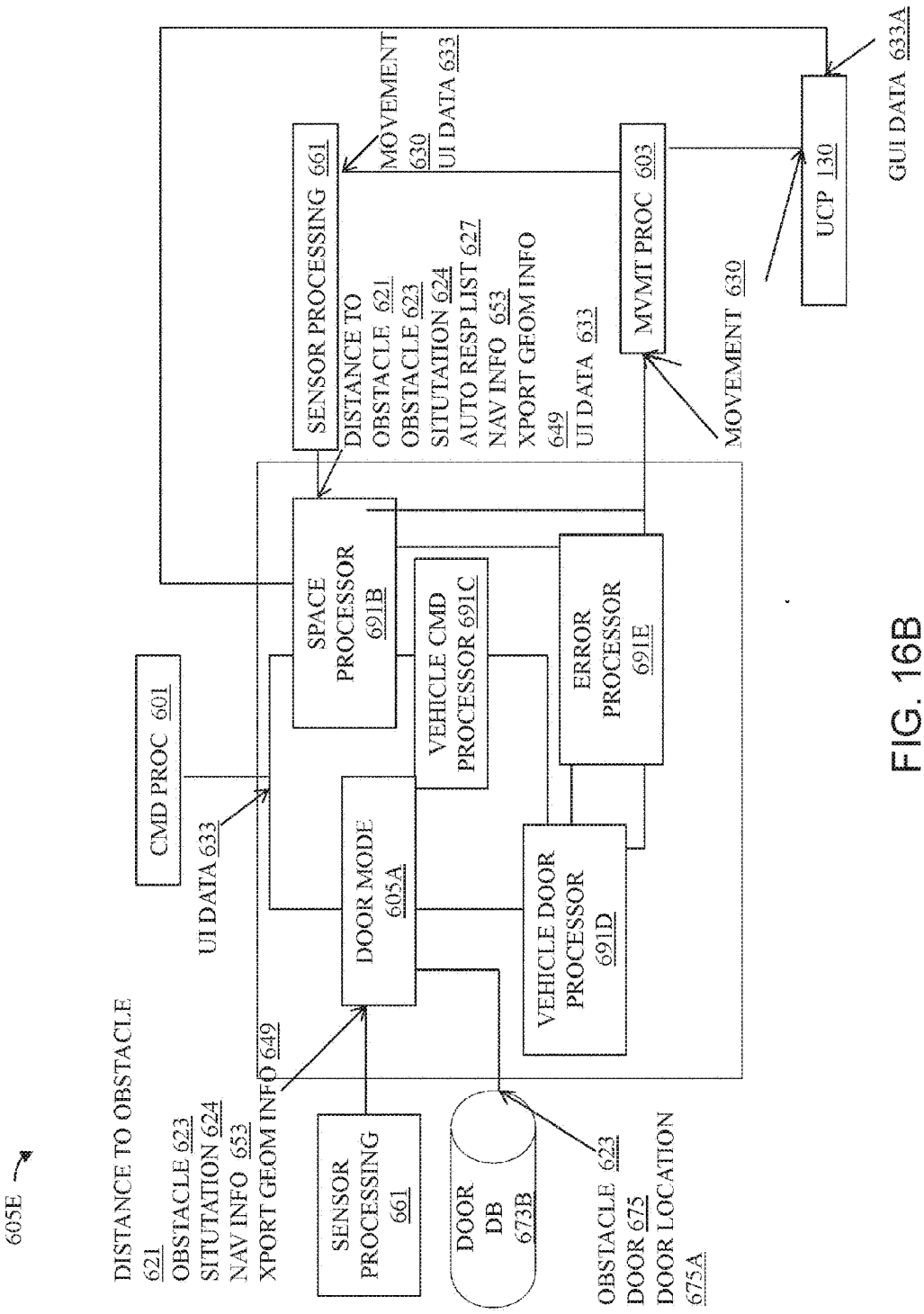

605E
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUATION 624
NAV INFO 653
XPORT GEOM INFO 649
CMD PROC 601
UI DATA 633
SENSOR PROCESSING 661
DOOR DB 673B
OBSTACLE 623
DOOR 675
DOOR LOCATION 675A
DOOR MODE 605A
SPACE PROCESSOR 691B
VEHICLE CMD PROCESSOR 691C
VEHICLE DOOR PROCESSOR 691D
ERROR PROCESSOR 691E
SENSOR PROCESSING 661
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
UI DATA 633
MOVEMENT 630
UI DATA 633
MVMT PROC 603
MOVEMENT 630
UCP 130
GUI DATA 633A

FIG. 16B

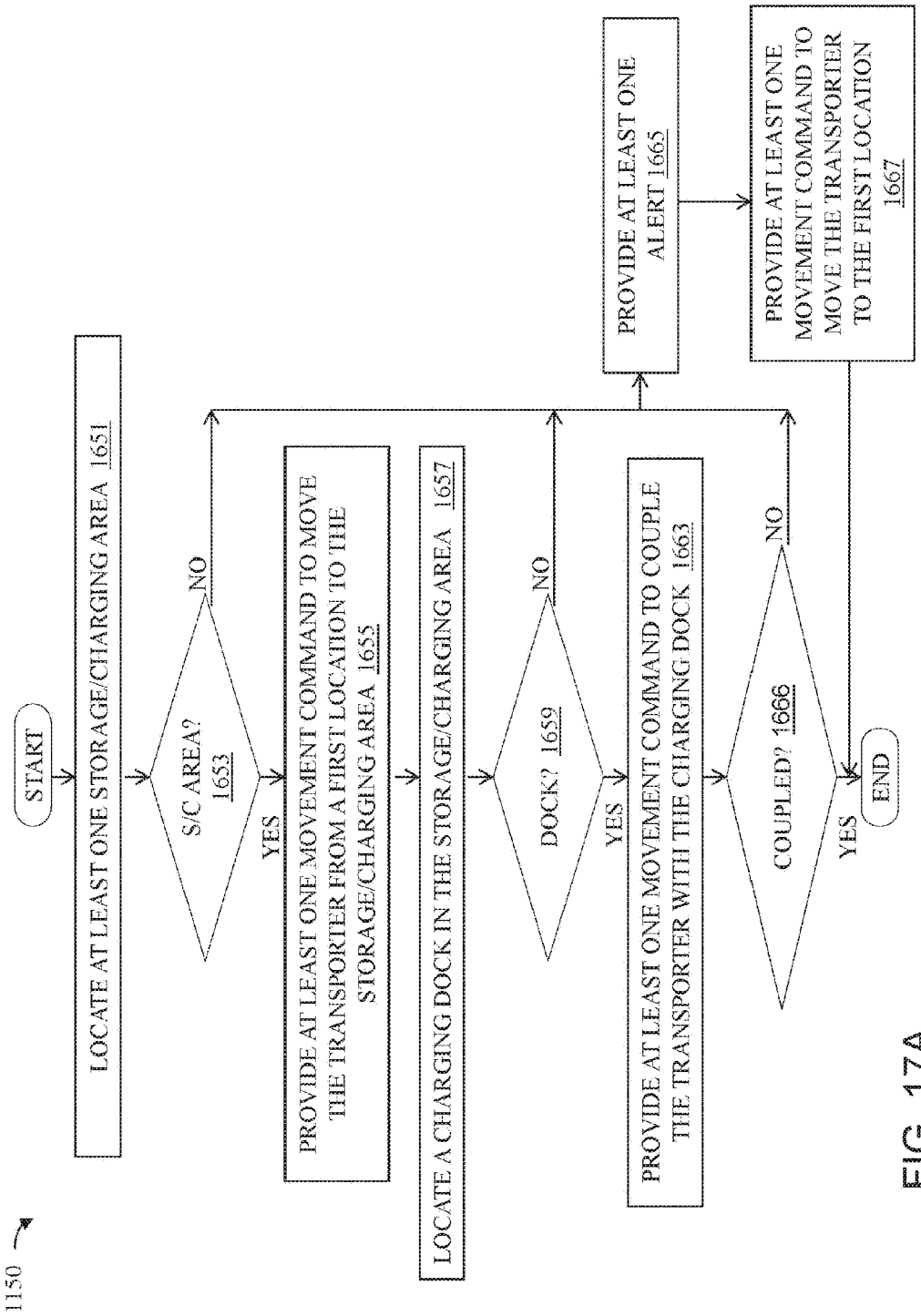

1150
START
LOCATE AT LEAST ONE STORAGE/CHARGING AREA 1651
S/C AREA? 1653
NO
YES
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER FROM A FIRST LOCATION TO THE STORAGE/CHARGING AREA 1655
LOCATE A CHARGING DOCK IN THE STORAGE/CHARGING AREA 1657
DOCK? 1659
NO
YES
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO COUPLE THE TRANSPORTER WITH THE CHARGING DOCK 1663
COUPLED? 1666
NO
YES
PROVIDE AT LEAST ONE ALERT 1665
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER TO THE FIRST LOCATION 1667
END

FIG. 17A

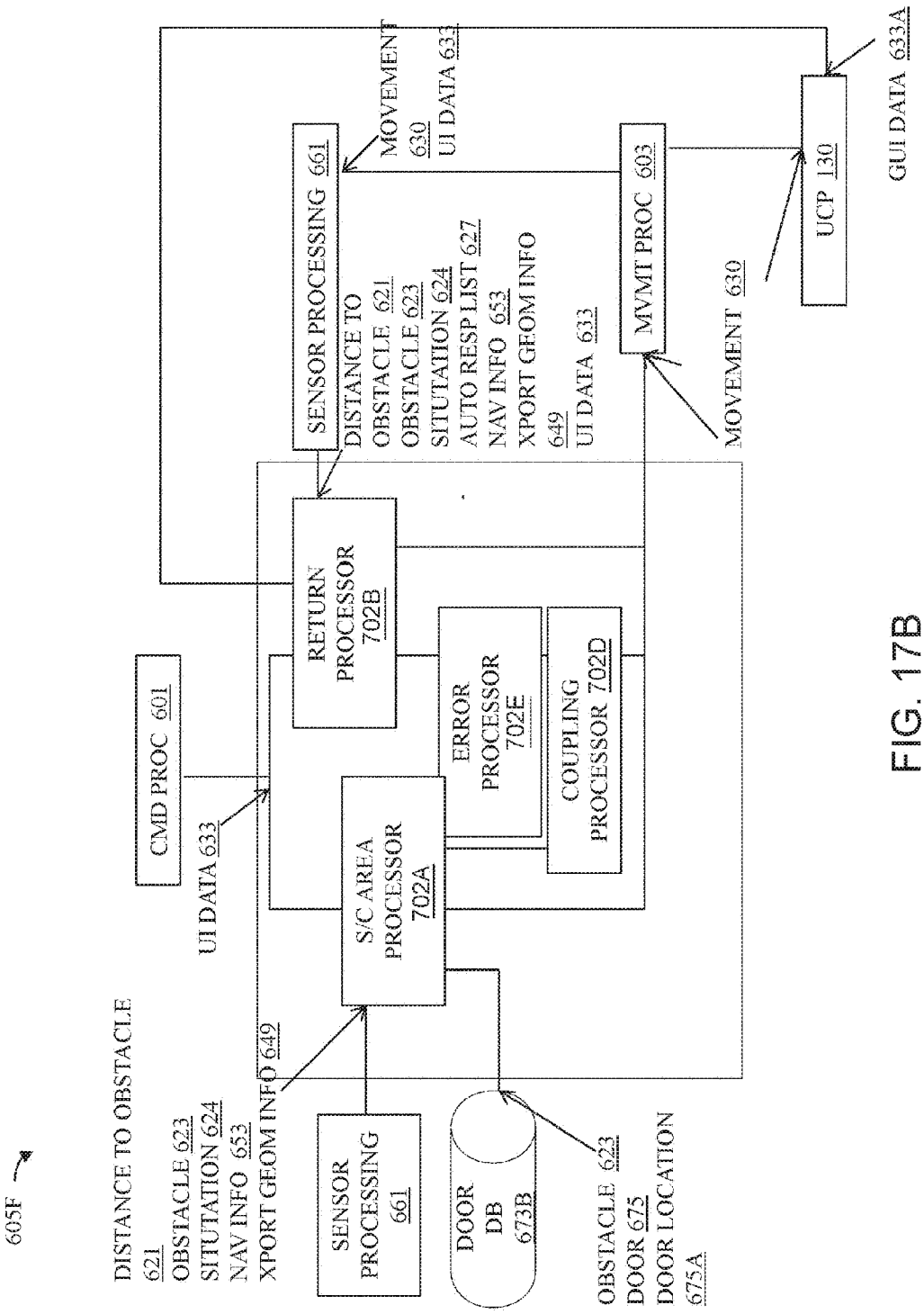

605F
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUATION 624
NAV INFO 653
XPORT GEOM INFO 649
SENSOR PROCESSING 661
DOOR DB 673B
OBSTACLE 623
DOOR 675
DOOR LOCATION 675A
S/C AREA PROCESSOR 702A
CMD PROC 601
UI DATA 633
RETURN PROCESSOR 702B
ERROR PROCESSOR 702E
COUPLING PROCESSOR 702D
SENSOR PROCESSING 661
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
UI DATA 633
MOVEMENT 630
UI DATA 633
MVMT PROC 603
MOVEMENT 630
UCP 130
GUI DATA 633A

FIG. 17B

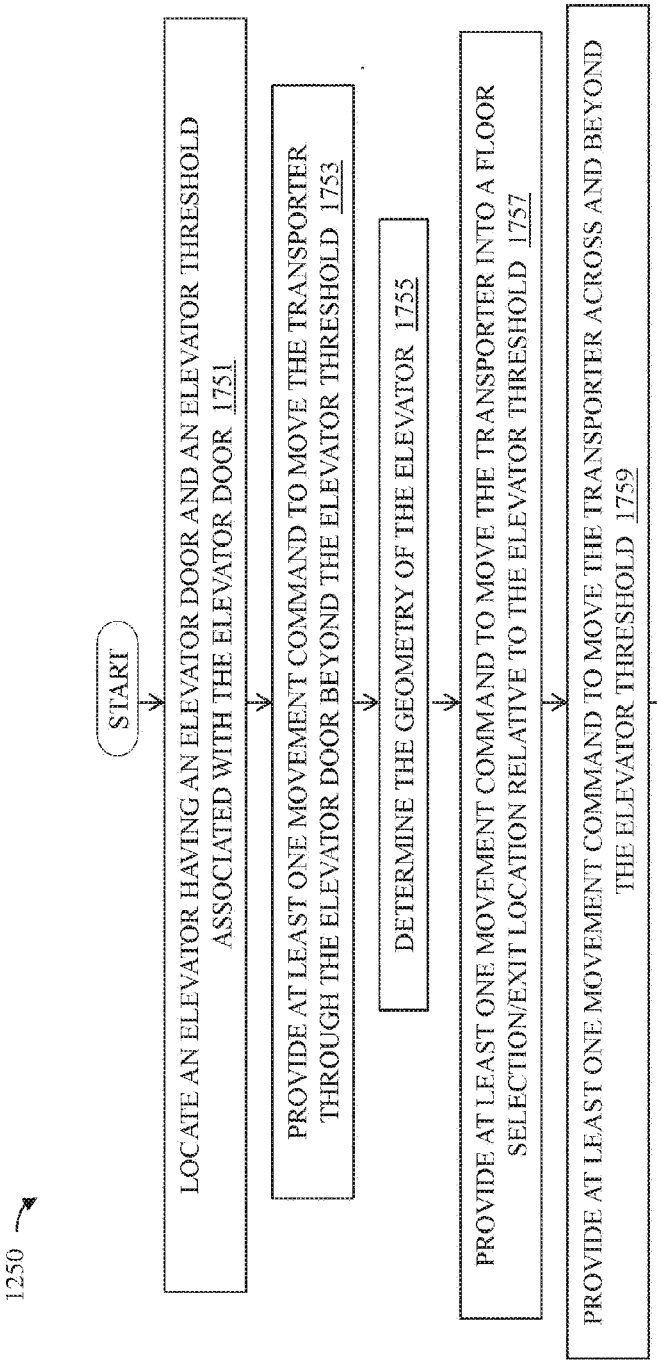

1250
START
LOCATE AN ELEVATOR HAVING AN ELEVATOR DOOR AND AN ELEVATOR THRESHOLD ASSOCIATED WITH THE ELEVATOR DOOR 1751
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER THROUGH THE ELEVATOR DOOR BEYOND THE ELEVATOR THRESHOLD 1753
DETERMINE THE GEOMETRY OF THE ELEVATOR 1755
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER INTO A FLOOR SELECTION/EXIT LOCATION RELATIVE TO THE ELEVATOR THRESHOLD 1757
PROVIDE AT LEAST ONE MOVEMENT COMMAND TO MOVE THE TRANSPORTER ACROSS AND BEYOND THE ELEVATOR THRESHOLD 1759
END

FIG. 18A

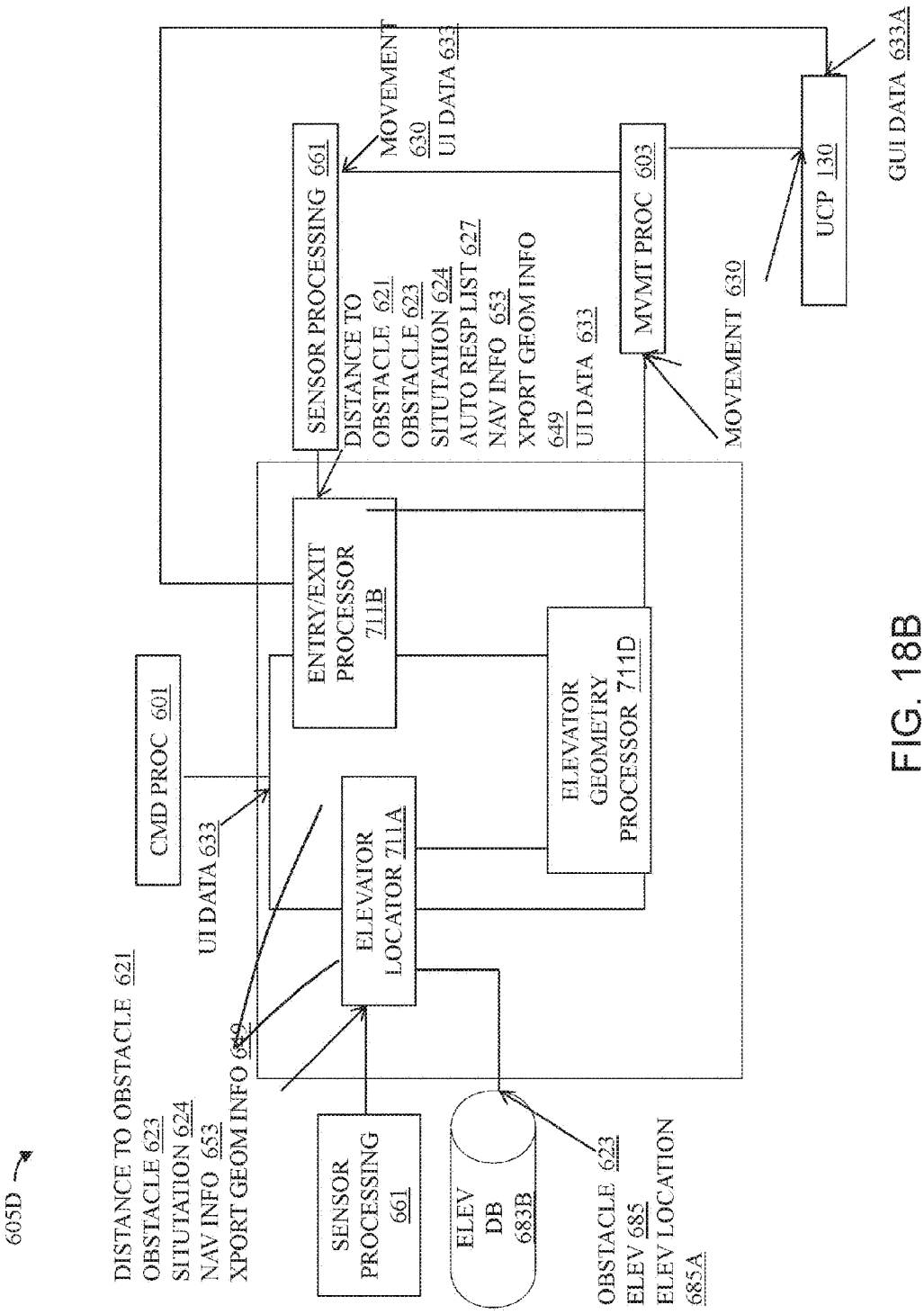

605D
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUATION 624
NAV INFO 653
XPORT GEOM INFO 649
SENSOR PROCESSING 661
OBSTACLE 623
ELEV 685
ELEV LOCATION 685A
ELEV DB 683B
CMD PROC 601
UI DATA 633
ENTRY/EXIT PROCESSOR 711B
ELEVATOR LOCATOR 711A
ELEVATOR GEOMETRY PROCESSOR 711D
SENSOR PROCESSING 661
DISTANCE TO OBSTACLE 621
OBSTACLE 623
SITUATION 624
AUTO RESP LIST 627
NAV INFO 653
XPORT GEOM INFO 649
UI DATA 633
MOVEMENT 630
UI DATA 633
MVMT PROC 603
MOVEMENT 630
UCP 130
GUI DATA 633A

FIG. 18B