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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 October 2021 (28.10.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/216254 A1

(51) International Patent Classification:
G05D 1/02 (2020.01)

(21) International Application Number:
PCT/US2021/024785

(22) International Filing Date:
30 March 2021 (30.03.2021)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

63/012,500	20 April 2020 (20.04.2020)	US
16/884,669	27 May 2020 (27.05.2020)	US

(71) Applicant: **BOSTON DYNAMICS, INC.** [US/US]; 200 Smith Street, Suite 4100, Waltham, MA 02451 (US).

(72) Inventors: **CHESTNUTT, Joel**; 200 Smith Street, Suite 4100, Waltham, MA 02451 (US). **FAY, Gina**; 78 Fourth Avenue, Waltham, MA 02451 (US).

(74) Agent: **KRUEGER, Brett, A.**; Honigman LLP, 650 Trade Centre Way, Suite 200, Kalamazoo, MI 49002-0402 (US).

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

(54) Title: CONSTRUCTION CONSTRAINED MOTION PRIMITIVES FROM ROBOT MAPS

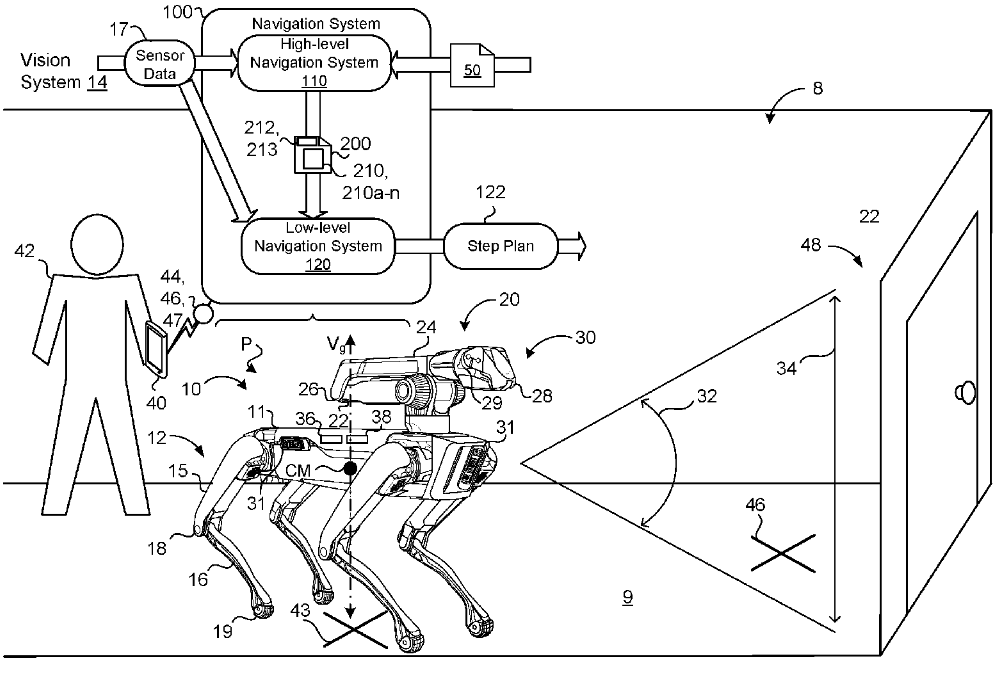


FIG. 1

(57) Abstract: A method (800) includes receiving sensor data (17) of an environment (7, 800) and generating a plurality of waypoints (510) and a plurality of edges (520) each connecting a pair of the waypoints. The method includes receiving a target destination (46) to navigate to and determining a route specification (200) based on the waypoints and corresponding edges to follow for navigating to the target destination selected from waypoints and edges previously generated. For each waypoint, the method includes generating a goal region (220) encompassing the corresponding waypoint and generating at least one constraint region (230) encompassing a goal region. The at least one constraint region establishes boundaries to remain within while traversing toward the target destination. The method includes navigating to the target destination by traversing through each goal region while maintaining within the at least one constraint region.

WO 2021/216254 A1

[Continued on next page]

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

Published:

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

Construction Constrained Motion Primitives from Robot Maps

TECHNICAL FIELD

[0001] This disclosure relates to construction constrained motion primitives from robot maps.

BACKGROUND

[0002] Robotic devices are increasingly being used to navigate constrained environments to perform a variety of tasks or functions. These robotic devices often need to navigate through these constrained environments without contacting the obstacles or becoming stuck or trapped. As these robotic devices become more prevalent, there is a need for real-time navigation and route planning that avoids contact with dynamic obstacles while successfully navigating to the destination.

SUMMARY

[0003] One aspect of the disclosure provides a method for constructing constrained motion primitives from robot maps. The method includes receiving, at data processing hardware, from at least one sensor in communication with the data processing hardware, sensor data of an environment about a robot while the robot traverses through the environment. The method also includes generating, by the data processing hardware, using the sensor data received while the robot traverses through the environment, a plurality of waypoints and a plurality of edges. Each edge connects a respective pair of the plurality of waypoints. After the robot traverses through the environment, the method includes receiving, at the data processing hardware, a target destination within the environment for the robot to navigate to. The method also includes determining, by the data processing hardware, a route specification based on a series of waypoints and corresponding edges for the robot to follow for navigating the robot to the target destination. The series of waypoints and corresponding edges are selected from the plurality of waypoints and the plurality of edges generated while the robot traversed through the environment. For each waypoint in the series of waypoints, the method includes generating, by the data processing hardware, a goal region encompassing the corresponding waypoint in the route specification. The method also includes generating, by the data processing hardware, at least one constraint region encompassing at least one goal region. The at least one constraint region establishes boundaries for the robot to remain within while traversing toward the target destination. The robot also includes navigating, by the data processing hardware, the robot to the target destination by traversing the robot through each goal region while maintaining the robot within the at least one constraint region.

[0004] Implementations of the disclosure may include one or more of the following optional features. In some implementations, receiving the target destination includes receiving the target destination from a user device in communication with the data processing hardware. Optionally, traversing the robot through each goal region includes traversing a center point of the robot through each goal region with a valid yaw configuration. At least one goal region in the route specification may include a yaw constraint configured to constrain a yaw of the robot within the at least one goal region.

[0005] In some examples, the method further includes, when generating the goal region encompassing the corresponding waypoint in the route specification, generating a target region encompassing the target destination. The target region includes a smaller area than at least one of the generated goal regions. In some implementations, the target region includes an area smaller than each of the generated goal regions. The at least one constraint region may encompass two goal regions.

[0006] Optionally, generating the at least one constraint region includes aligning the at least one constraint region with the corresponding edge connecting the pair of waypoints each encompassed by a respective one of the two goal regions encompassed by the at least one constraint region. At least one of the goal regions may be configured to cause the robot to align with a terrain feature prior to the robot traversing the terrain feature. In some examples, at least one of the goal regions is configured to ensure localization information from previously captured sensor data is valid.

[0007] In some implementations, the at least one constraint region includes route cost information. In some examples, the method includes receiving input from a user directing the robot to move about the environment and, as the robot moves about the environment, capturing the sensor data of the environment. The at least one sensor may include an image sensor comprising one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

[0008] Another aspect of the disclosure provides a system for constructing constrained motion primitives from robot maps. The system includes data processing hardware and memory hardware in communication with the data processing hardware. The memory hardware stores instructions that when executed on the data processing hardware cause the data processing hardware to perform operations. The operations include receiving, from at least one sensor in communication with the data processing hardware, sensor data of an environment about a robot while the robot traverses through the environment. The operations also include generating, using the sensor data received while the robot traverses through the environment, a plurality of waypoints and a plurality of edges. Each edge connects a respective pair of the plurality of waypoints. After the robot traverses through the environment, the operations also include receiving a target destination within the environment for the robot to navigate to. The operations also include determining a route specification based on a series of waypoints and corresponding edges for the robot to follow for navigating the robot to the target destination. The series of waypoints and corresponding edges are selected from the plurality of waypoints and the plurality of edges generated while the robot traversed through the environment. For each waypoint in the series of waypoints, the operations include generating a goal region encompassing the corresponding waypoint in the route specification. The operations also include generating at least one constraint region encompassing at least one goal region. The at least one constraint region establishes boundaries for the robot to remain within while traversing toward the target destination. The robot also includes navigating the robot to the target destination by traversing the robot through each goal region while maintaining the robot within the at least one constraint region.

[0009] This aspect may include one or more of the following optional features. In some implementations, receiving the target destination includes receiving the target destination from a user device in communication with the data processing hardware. Optionally, traversing the robot through each goal region includes traversing a center point of the robot through each goal region with a valid yaw configuration. At least one goal region in the route specification may include a yaw constraint configured to constrain a yaw of the robot within the at least one goal region.

[0010] In some examples, the operations further include, when generating the goal region encompassing the corresponding waypoint in the route specification, generating a target region encompassing the target destination. The target region includes a smaller area than at least one of the generated goal regions. In some implementations, the target region includes an area smaller than each of the generated goal regions. The at least one constraint region may encompass two goal regions.

[0011] Optionally, generating the at least one constraint region includes aligning the at least one constraint region with the corresponding edge connecting the pair of waypoints each encompassed by a respective one of the two goal regions encompassed by the at least one constraint region. At least one of the goal regions may be configured to cause the robot to align with a terrain feature prior to the robot traversing the terrain feature. In some examples, at least one of the goal regions is configured to ensure localization information from previously captured sensor data is valid.

[0012] In some implementations, the at least one constraint region includes route cost information. In some examples, the operations include receiving input from a user directing the robot to move about the environment and, as the robot moves about the environment, capturing the sensor data of the environment. The at least one sensor may include an image sensor comprising one or more of a stereo camera, a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

[0013] The details of one or more implementations of the disclosure are set forth in the accompanying drawings and the description below. Other aspects, features, and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0014] FIG. 1 is a schematic view of an example system that provides constrained robot autonomy language.

[0015] FIG. 2 is a schematic view of exemplary components of a route specification.

[0016] FIG. 3 is a schematic view of a route specification including an initial path.

[0017] FIG. 4 is a schematic view of a route specification including a plurality of sequential goal regions.

[0018] FIGS. 5A–5C are schematic views of a route specification generated from a previously generated map.

[0019] FIG. 6 is a schematic view of a constraint box applying orientation constraints to a robot prior to traversing a staircase.

[0020] FIG. 7 is a schematic view of a plurality of goal regions helping a robot to navigate a staircase.

[0021] FIG. 8 is a flowchart of an example arrangement of operations for a method of providing constrained robot autonomy language.

[0022] FIG. 9 is a schematic view of an example computing device that may be used to implement the systems and methods described herein.

[0023] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0024] As robotic devices (also referred to as "robots") become more prevalent, there is an increasing need for the robots to autonomously navigate environments that are constrained in a number of ways. Often, the robots rely on high-level map data that stores information relating to large and/or static objects (e.g., walls, doors, etc.). When given a destination goal, the robot will often first plot an initial path or course from this high-level map using a high-level navigation system to navigate the static obstacles and then rely on a local navigation system that gathers local navigation data to navigate around small and dynamic objects encountered while travelling.

[0025] While navigating, the robot requires a way for the high-level components of the navigation system, which have access to large-scale navigation information, to communicate goals, context, and intentions to lower-level navigation and obstacle avoidance systems so that the lower-level systems (i.e., local systems) may make more informed decisions about how to locally navigate the robot in a safe and efficient manner. Conventional techniques of representing goals as single poses lose the required context needed by the lower levels to make these optimizations and decisions. By including this context, the lower-level navigation systems can navigate the robot in a safe manner while still achieving the goals set by the high-level navigation system.

[0026] Typically, the high-level navigation system, when providing the initial path to the local navigation system (i.e., the navigation system that operates on a small, local map of the robot's environment versus a building-scale map of the high-level navigation system), most, if not all, context related to the path and destination is lost. That is, conventional techniques may specify the goals of a route, but the execution of the route is limited in how the local navigation system can respond to disturbances or obstacles along its path. The local navigation system lacks the context or information to determine whether a particular deviation from the goal is acceptable or even preferred. As a result, conventional techniques output local trajectories directly based on the map and route details, as well as the state of the robot. This over-constrains the systems that follow this trajectory (e.g., the local navigation system), because the systems are not aware of the amount of freedom available to deviate from the trajectory.

[0027] Implementations herein are directed toward systems and methods for a navigation system that utilizes a route specification language that includes additional context in a compact representation that allows a local navigation system (e.g., obstacle avoidance systems) to make appropriate decisions and optimizations. For example, the route specification language may represent constraints of staircases, including yaw and position. The route specification language may represent how near the robot must get to previously recorded sensor data in order to localize to the sensor data. In yet another example, the language may specify that the robot should not change its yaw or orientations in certain situations or locations.

[0028] By including relevant context and navigational information, the robot may perform tasks in a safer manner. For example, the robot provides larger safety margins and smoother paths, because the local navigation system has the necessary information to know acceptable ways to steer while still achieving the goals commanded by the higher-level navigation system. Additionally, the language enables new applications to be built by allowing new systems to specify goals and requirements in a more precise manner than was previously possible, while still allowing the local navigation system the freedom to adjust the robot in ways that do not affect the goals and requirements.

[0029] Referring to FIG. 1, a robot or robotic device 10 includes a body 11 with two or more legs 12 and executes a navigation system 100 for enabling the robot 10 to navigate a constrained environment 8. Each leg 12 is coupled to the body 11 and may have an upper portion 15 and a lower portion 16 separated by a leg joint 18. The lower portion 16 of each leg 12 ends in a foot 19. The foot 19 of each leg is optional and the terminal end of the lower portion of one or more of the leg 12 may be coupled to a wheel or the distal end of each leg 12 may directly contact a ground surface 9. The robot 10 has a vertical gravitational axis Vg along a direction of gravity, and a center of mass CM, which is a point where the weighted relative position of the distributed mass of the robot 10 sums to zero. The robot 10 further has a pose P based on the CM relative to the vertical gravitational axis Vg (i.e., the fixed reference frame with respect to gravity) to define a particular attitude or stance assumed by the robot 10. The attitude of the robot 10 can be defined by an orientation or an angular position of the robot 10 in space. Movement by the legs 12 relative to the body 11 alters the pose P of the robot 10 (i.e., the combination of the position of the CM of the robot and the attitude or orientation of the robot 10).

[0030] In some implementations, the robot 10 further includes one or more appendages, such as an articulated arm 20 disposed on the body 11 and configured to move relative to the body 11. The articulated arm 20 may have five-degrees or more of freedom. Moreover, the articulated arm 20 may be interchangeably referred to as a manipulator arm or simply an appendage. In the example shown, the articulated arm 20 includes two portions 22, 24 rotatable relative to one another and also the body 11; however, the articulated arm 20 may include more or less portions without departing from the scope of the present disclosure. The first portion 22 may be separated from second portion 24 by an articulated arm joint 26. An end effector 28, which may be interchangeably referred to as a manipulator head 28, may be coupled to a distal end of the second portion 24 of the articulated arm 20 and may include one or more actuators 29 for gripping/grasping objects.

[0031] The robot 10 also includes a vision system 30 with at least one imaging sensor or camera 31, each sensor or camera 31 capturing image data or sensor data 17 of the environment 8 surrounding the robot 10 with an angle of view 32 and within a field of view 34. The vision system 30 may be configured to move the field of view 34 by adjusting the angle of view 32 or by panning and/or tilting (either independently or via the robot 10) the camera 31 to move the field of view 34 in any direction. Alternatively, the vision system 30 may include multiple sensors or cameras 31 such that the vision system 30 captures a generally 360-degree field of view around the robot 10.

[0032] The camera(s) 31 of the vision system 30, in some implementations, include one or more stereo cameras (e.g., one or more RGBD stereo cameras). In other examples, the vision system 30 includes one or more radar sensors such as a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor, a light scanner, a time-of-flight sensor, or any other three-dimensional (3D) volumetric image sensor (or any such combination of sensors).

[0033] The vision system 30 provides image data or sensor data 17 captured by the cameras or sensors 31 to data processing hardware 36 of the robot 10. The data processing hardware 36 is in digital communication with memory hardware 38 that stores instructions executable by the data processing hardware 36 for performing operations. While the example shown depicts the data processing hardware 36 and the memory hardware 38 residing on the robot 10, other configurations may include the data processing hardware 36 and the memory hardware 38 implemented on a remote system in communication with the robot 10. The remote system may be a single computer, multiple computers, or a distributed system (e.g., a cloud environment) having scalable / elastic computing resources and/or storage resources. A navigation system 100 of the robot 10 executes on the data processing hardware 36. The navigation system 100 may include a high-level navigation system 110 and a local navigation system 120.

[0034] In some implementations, the navigation system 100 receives a navigation command 44 to navigate the robot 10 to a mission destination 46 (also referred herein as a target destination 46) within the environment 8 of the robot 10. The navigation command 44 may specify mission parameters for performing one or more actions/behaviors by the robot 10 at the mission destination 46. The navigation system 100 may receive the navigation command 44 from a user device 40 associated with a user 42. The user 42 may use the user device 40 to control/navigate the robot 10 around the environment 8 and may be any appropriate device such as a tablet, a mobile phone, a laptop or other computing system, or a dedicated controller. The mission destination 46 indicates a target location within the environment 8 that the navigation system 100 must navigate the robot 10 to. In the example shown, the mission destination 46 indicates a location in front of a door 48 in preparation for the robot 10 to open the door 48.

[0035] In some implementations, the navigation system 100 includes the high-level navigation system 110 that receives map data 50 (i.e., high-level navigation data representative of locations of static obstacles in an area the robot 10 is to navigate). The high-level navigation system 110 may use the map data 50 and the navigation command 44 to generate a route specification 200 for navigating the robot 10 from a current location 43 to the mission destination 46. The route specification 200 generated by the high-level navigation system 110 includes a series of route segments 210, 210a–n (FIG. 2), with each route segment 210 including an initial path 310 (FIG. 3) for the robot 10 to follow while traversing the corresponding route segment 210. Each segment 210, as discussed in more detail below, may include additional constraints 212 for constraining the robot 10 while traversing the environment 8.

[0036] With continued reference to FIG. 1, the high-level navigation system 110 sends the route specification 200 to the local navigation system 120. The local navigation system 120, based on the route specification 200 and the sensor data 17, generates a step plan 122 that plots each individual step of the robot 10 to navigate from the current location 43 of the robot 10 to the mission destination 46. Using the step plan 122, the robot 10 maneuvers through the environment 8 (e.g., around local and dynamic obstacles) by placing the feet 19 or distal ends of the leg 12 on the ground surface 9 at locations specified by the step plan 122.

[0037] In some examples, at least a portion of the navigation system 100 executes on a remote device in communication with the robot 10. For instance, the high-level navigation system 110 may execute on a remote device to generate the route specification 200 and the local navigation system 120 may execute on the robot 10 to receive the route specification 200 from the remote device. Optionally, the entire navigation system 100 may execute on a remote device and the remote device may control/instruct the robot 10 to maneuver the environment 8 based the step plan 122. In additional examples, the remote device includes a remote computing device (e.g., a cloud-based server) in communication with the user device 40 and the robot 10.

[0038] Referring now to FIG. 2, an exemplary route specification 200 provides the local navigation system 120 with context to aid in navigating the robot 10 from a current location 43 in the environment 8 to the mission destination 46 in the environment. Here, the route specification 200 includes a series of four route segments 210, 210a–d. The route segments 210 represent sequential portions of a path the robot 10 must traverse in order to successfully navigate to the mission destination 46. In some examples, each route segment 210 in the series of route segments 210 includes a corresponding goal region 220, 220a–d. Each goal region 220 represents an area that at least a portion of the robot 10 enters while traversing the corresponding respective route segment 210. That is, in order to successfully "pass" or "succeed" the goal region 220 and move on to the next segment 210, at least some portion of the robot 10 must enter the area designated by the goal region 220. In some examples, the goal region 220 represents an area that a center point 202 of the robot 10 must enter while traversing the respective route segment 210. Accordingly, the robot 10 is to sequentially traverse each route segment 210 in the series of route segments 210a–d by passing through each respective goal region 220 while navigating to the mission destination 46 in the environment 8. In some examples, each route segment 210 in the series of route segments overlaps at least one other route segment 210 in the series of route segments 210. In additional examples, each goal region 220 includes a convex shape in SE2 space. SE2 is a coordinate system that includes yaw with standard Cartesian coordinates (i.e., an x coordinate, a y coordinate, and yaw).

[0039] Optionally, each segment 210 includes a constraint region 230, 230a–d that encompasses the respective goal region 220 of the segment 210. The constraint region 230 establishes boundaries for the robot 10 to remain within while traversing toward the goal region 220 of the corresponding route segment 210. That is, the constraint region 230 provides boundaries that allows the local navigation system 120 to freely navigate the robot 10 within while traversing the segment 210. For example, in FIG. 2, an obstacle 250 in the first route segment 210a impedes a direct path from the current location 43 of the robot 10 to the goal region 220a for the corresponding first route segment 210a. However, the constraint region 230a provides the local navigation system 120 with the context to understand that the robot 10 may safely navigate around the obstacle 250 using path 260 without failing the mission. In some examples, the constraint region 230 encompassing the goal region 220 for the corresponding route segment 210 also encompasses the goal region 220 associated with another one of the route segments 210 in the series of route segments 210 that overlaps the corresponding route segment 210. For instance, in the example shown in FIG. 2, the constraint region 230a encompassing the goal region 220a for the route segment 210a also encompasses the goal region 220b for the route segment 210b that overlaps the route segment 210a.

[0040] As another advantage, the local navigation system 120 may dynamically adjust an amount of margin 270 of distance between the robot 10 and obstacles 250 in response to the constraint region 230. For example, when the constraint region 230 is large, the local navigation system 120 may provide additional margin 270 between the robot 10 and the obstacle 250 to provide for safer and smoother avoidance of the obstacle 250. In other examples, when the constraint region 230 is small, the high-level navigation system 110 communicates to the local navigation system 120 that the mission requires a smaller margin 270 between the robot 10 and the obstacle 250 and therefore the local navigation system 120 will navigate the robot 10 closer to the obstacle 250 to remain within the constraint region 230.

[0041] With continued reference to FIG. 2, the exemplary route specification 200 includes four route segments 210a–d that the robot 10 must sequentially traverse when navigating to the mission destination 46. For instance, the robot 10 traverses the first route segment 210a via the constraint region 230a to the corresponding goal region 220a adjacent to the bottom of a staircase 280 in the environment 8. The route segment 210b provides the goal region 220b and the constraint region 230b to align the robot 10 with the staircase 280. The route segment 210c and the constraint region 230c bound the robot 10 within the staircase 280 to allow the robot 10 to traverse up the staircase 280 to the corresponding goal region 220c adjacent to the top of the staircase 280. Once the robot 10 reaches the top of the staircase 280 by entering the goal region 220c, the robot 10 traverses the fourth route segment 210d via the constrain region 230d to the goal region 220d (which, in this case, is also the mission destination 46). Thus, in some examples, the goal region 220 for the last route segment 210 in the series of route segments encompasses the mission destination 46.

[0042] Referring now to FIG. 3, a route segment 210 specified by a route specification 200 may provide the local navigation system 120 with an initial path 310 for the robot 10 to follow while traversing the corresponding route segment 210. The initial path 310 may be a “best guess” by the high-level navigation system 110 that does not take into account dynamic objects or obstacles that require local navigation to avoid. The local navigation system 120 may use the initial path 310 as a starting point for navigation, but deviate from the initial path 310 while remaining within the constraint region 230 as the environment 8 demands. In some examples, both goal regions 220 and constraint regions 230 are convex shapes to simplify problem solving for the navigation system 100. A convex shape does not self-intersect and is a shape in which no line segment between two points on the boundary ever go outside of the shape.

[0043] Each route segment 210 may include any number of segment-specific constraints 212. For example, each route segment 210 includes one or more of goal costs, velocity bounds, position constraints, position costs, velocity costs, yaw constraints/bounds, and/or mobility parameters. A cost may be associated with a “soft” constraint. That is, the robot 10 may violate a constraint with an associated cost under certain circumstances. The higher the cost, the harder the robot 10 (i.e., the navigation system 100) will try to avoid violating the constraint. For example, a velocity cost may deter the navigation system 100 from exceeding a minimum velocity threshold and/or a maximum velocity threshold while traversing the corresponding route segment 210. The velocity bounds may include angular velocity bounds, lateral velocity bounds, and longitudinal bounds. Likewise, position costs may encourage the robot 10 to maintain certain positions. Yaw constraints may impose yaw limitations on the robot 10. For example, in narrow corridors or near ledges the robot 10 may be restricted from turning. Mobility parameters may categorize one or more surfaces covered by the corresponding route segment 210. For example, the categorization includes stairs or flat ground. That is, a route segment 210 (e.g., the route segment 210c of FIG. 2) that causes the robot 10 to traverse stairs may include a mobility parameter that categorizes the route segment 210 as stairs to allow the local navigation system 120 to react accordingly.

[0044] As the robot 10 navigates from one route segment 210 to a next route segment 210 specified by the route specification 200, only the segment-specific constraints 212 associated with the route segment 210 that the robot 10 is currently traversing apply. This is opposed to global constraints 213 which the robot 10 must abide by no matter which route segment 210 the robot 10 is traversing. In some examples, the navigation system 100 includes only segment-specific constraints 212, while in other examples, the navigation system 100 includes a hybrid of segment-specific constraints 212 and global constraints 213. That is, the high-level navigation system 110 may impose both segment-specific constraints 212 and at least one global constraint 213 that constrains the robot 10 while traversing each segment 210. For instance, world-based constraints (e.g., stairs) may be better modeled with global constraints 213.

[0045] Referring now to FIG. 4, another exemplary route specification 200 illustrates that the navigation system 100 does not use sequential composition. That is, the navigation system 100 may require the robot 10 to pass through each goal region 220 without skipping any goal region 220. Here, the robot 10 must first pass through goal region 220e during route segment 210e, then while traversing route segment 210f, pass through goal region 220f, then while traversing route segment 210g, pass through goal region 220g, and so on and so forth until the robot reaches the mission destination 46 (not shown).

[0046] Referring now to FIGS. 5A–5C, in some implementations, the navigation system 100 generates the route specification 200 from a previously recorded map 50 (FIG. 1). The previously recorded map 50 includes a plurality of waypoints 510, 510a–n and a plurality of edges 520, 520a–n generated by the robot 10 using sensor data 17 while previously traversing through the environment 8. Each edge 520 connects a respective pair of the plurality of waypoints 510. For example, the user 42 may command the robot 10 (via the user device 40) to traverse the environment 8, and as the robot 10 moves about the environment 8, the robot 10 captures the sensor data 17 of the environment 8 for initial map generation and localization (e.g., simultaneous localization and mapping (SLAM)).

[0047] After the robot 10 traverses the environment 8 and generates the map 50, the navigation system 100 receives a target destination (interchangeably referred to as "mission destination") 46 within the environment 8 for the robot 10 to navigate to. In some examples, the robot 10 receives a navigation request 47 (FIG. 1) to navigate from the current location 43 in the environment 8 to the mission destination 46 in the environment 8. The high-level navigation system 110 determines the route specification 200 which may include a series of waypoints 510 and corresponding edges 520 for the robot 10 to follow for navigating the robot 10 to the target destination 46. Here, the series of waypoints 510 and corresponding edges 520 may be selected from the plurality of waypoints 510 and the plurality of edges 520 generated while the robot 10 previously traversed through the environment 8 during map generation. That is, the high-level navigation system 110 selects a subset of the plurality of waypoints 510 and associated edges 520 from the previously generated map 50 that best navigate the robot 10 to the mission destination 46. In the example 500a of FIG. 5A, the high-level navigation system 110 selects waypoints 510a–c connected by edges 520a, 520b to navigate the robot to or near the mission destination 46.

[0048] For each waypoint 510 in the series of waypoints 510 in the route specification 200, the high-level navigation system 110 generates a goal region 220 encompassing the corresponding waypoint 510 in the route specification 200. In the example 500b of FIG. 5B, the high-level navigation system 110 generates goal regions 220k–m encompassing the waypoints 510a–c. Specifically, the goal region 220k encompasses the waypoint 510a, the goal region 220l encompasses the waypoint 510b, and the goal region 220m. In some implementations, the high-level navigation system 110 passes additional context to the local navigation system 120 via a size of each goal region 220. That is, when generating the goal region 220 encompassing the corresponding waypoint 510 in the route specification 200, the high-level navigation system 110 generates a goal region 220 (i.e., a target region) encompassing the target destination 46 that includes a smaller area than at least one of the generated goal regions 220. In some implementations, an area of the goal region 220 that encompasses the target destination 46 has an area that is smaller than each of the other goal regions.

[0049] For example, waypoints 510a, 510b are intermediate waypoints 510 on the way to the mission destination 46 and because there is no need for the robot 10 to precisely arrive at each waypoint 510, the high-level navigation system 110 enlarges the size of the goal regions 220k, 220l to allow the local navigation system 120 more flexibility in passing each waypoint 510a, 510b. On the other hand, the waypoint 510c coincides at or near the mission destination 46 and here the high-level navigation system 110 has decreased the size of the goal region 220m to ensure that the local navigation system 120 navigates the robot 10 to the mission destination 46. Because the local navigation system 120 must navigate a portion of the robot 10 (e.g., the center point of the robot) to a space in the environment 8 that is represented by the interior of the goal region 220, the size of the goal region 220 establishes how near the waypoint 510 the robot 10 will get.

[0050] Each goal region 220 may include a yaw bound or yaw constraint that the robot must conform to before successfully satisfy or complete the respective goal region 220 and move on to the next segment 210. That is, the yaw bound may force the robot 10 to enter a specific yaw configuration prior to leaving the goal region 220 before satisfying the goal region 220 and/or segment 210 while additionally or alternatively, the yaw bound forces the robot 10 to enter the specific yaw configuration before entering the goal region 220. For example, when a goal region 220 is located at the bottom of a staircase 280 (e.g., the goal region 220b of FIG. 2), a yaw bound associated with the goal region 220 ensures that the robot 10 is properly aligned with the staircase 280 prior to entering the goal region 220 and/or prior to leaving the goal region 220 and climbing the stairs. Thus, in some examples, one or more of the goal regions 220 are configured to cause the robot 10 to align with an obstacle 250, 280 prior to the robot 10 traversing the obstacle 250, 280.

[0051] In some implementations, the high-level navigation system 110 generates at least one constraint region 230 encompassing at least one goal region 220. The constraint region may encompass a plurality of goal regions 220 (e.g., two goal regions 220). The at least one constraint region 230 establishes the boundaries for the robot 10 to remain within while traversing toward the target destination or mission destination 46. In example 500c of FIG. 5C, the constraint region 230l encompasses the goal regions 220k, 200l while the constraint region 230m encompasses the goal regions 220l, 220m. The constraint regions 230l, 230m bound the areas the robot 10 may enter while the local navigation system 120 navigates the robot around obstacles to the mission destination 46.

Thus, the local navigation system 120 may navigate the robot 10 to the target destination or mission destination 46 by traversing the robot 10 through each goal region 220 (e.g., a center point 202 of the robot 10) while maintaining the robot 10 within the constraint regions 230. The constraint region 230 may be any shape. Multiple goal regions 220 may overlap. In some examples, each segment 210 includes only a single constrained region 230 and a single goal region 220.

[0052] Optionally, the high-level navigation system 110, when generating the constraint regions 230, generates each constraint region 230 such that the respective constraint region 230 is aligned with the corresponding edge 520 connecting the pair of waypoints 510 each encompassed by a goal region 220 and the constraint region 230. In the example shown, the constraint region 230l is aligned with the edge 520a while the constraint region 230m is aligned with the edge 520b.

[0053] In some implementations, one or more of the goal regions 220 provide localization information to the robot 10. For example, the underlying waypoint 510 encompassed by the goal region 220 from the previously generated map 50 allows the navigation system 100 to localize the robot 10 within the map 50. In some examples, the high-level navigation system 110 sizes one or more constraint regions 230 to ensure that the robot 10 stays localized with the previously generated map 50. That is, when the robot 10 strays too far away from waypoints 510 and edges 520 from the previously generated map 50, the navigation system 100 may lose track of the location of the robot 10 within the environment 8. To prevent this, the high-level navigation system 110 sizes the constraint region 230, based on, for example, the capabilities of the vision system 14 and the fidelity of the map 50, to keep the robot 10 localized within the map 50. The constraint regions 230 may include additional contextual information for the local navigation system 120, such as route cost information.

[0054] Referring now to FIG. 6, in some examples, the high-level navigation system 110 solves for orientation constraints when generating the route specification 200. In the example shown, an exemplary environment 600 includes a narrow corridor 610 that precedes a staircase 620. In some examples, it is beneficial for the robot 10 to traverse down stairs backwards to allow for additional space for the shins (e.g., lower portion 16 of each leg 12) of the robot 10. However, in this scenario, the narrow corridor 610 prohibits the robot 10 from turning around at the top of the staircase 620. In this situation, the robot 10 must turn prior to entering the corridor 610 and traverse the corridor 610 backwards. Accordingly, at location 630, the robot 10 executes a turning maneuver to re-orient itself backwards before entering the narrow corridor 61 so that the robot 10 may beneficially traverse down the staircase 620 backwards. By extending a constraint region 230 the entire length of the corridor 610, the constraint region 230 may enforce an orientation constraint on the robot 10, thus commanding the robot 10 to turn around prior (e.g., at or near location 630) to entering the corridor 610.

[0055] Referring now to FIG. 7, another exemplary environment 700 includes a staircase 710. Here, an initial goal region 220n ensures that the robot 10 is aligned with the staircase 710 prior to traversing the stairs. Note that in this example, there is not a constraint region 230 constraining the path of the robot to the initial goal region 220n as the high-level navigation system 110 determined that a constraint region 230 was not necessary in this situation. A second goal region 220o along the stairs allows the local navigation system 120 to notify the high-level navigation system 110 when the robot 10 is passing positions of good localization in the environment 700. That is, in this example, the goal region 220o along the staircase 710 represents an area that allows the robot 10 to localize its location to the previously generated map 50. A final goal region 220p ensures that the robot 10 is properly off the stairs prior to allowing the robot 10 to turn to proceed to the next segment 210 (not shown). The constraint regions 230 provide the robot 10 with position bounds and costs to keep the robot 10 centered on the stairs and to keep the yaw of the robot 10 aligned with the direction of the stairs.

[0056] FIG. 8 is a flowchart of an exemplary arrangement of operations for a method 800 of generating a route specification 200 from previous map 50 data for navigating a robot 10. At operation 802, the method 800 includes receiving, at data processing hardware 36, from at least one sensor 31 in communication with the data processing hardware 36, sensor data 17 of an environment 8 about a robot 10 while the robot 10 traverses through the environment 8. At operation 804, the method 800 includes generating, by the data processing hardware 36, using the sensor data 17 received while the robot 10 traverses through the environment 8, a plurality of waypoints 510 and a plurality of edges 520. Each edge 520 connects a respective pair of the plurality of waypoints 510.

[0057] After the robot 10 traverses through the environment 8, the method 800, at operation 806, includes receiving, at the data processing hardware 36, a target destination 46 within the environment 8 for the robot 10 to navigate to. At operation 808, the method 800 includes determining, by the data processing hardware 36, a route specification 200 includes a series of waypoints 510 and corresponding edges 520 for the robot 10 to follow for navigating the robot 10 to the target destination 46. The series of waypoints 510 and corresponding edges 520 are selected from the plurality of waypoints 510 and the plurality of edges 520 generated while the robot 10 traversed through the environment 8.

[0058] For each waypoint 510 in the series of waypoints 510 in the route specification 200, the method 800, at operation 810 includes, generating, by the data processing hardware 36, a goal region 220 encompassing the corresponding waypoint 510 in the route specification 200. At operation 812, the method includes generating, by the data processing hardware 36, at least one constraint box 230 encompassing at least one goal region 220. The at least one constraint box 230 establishes boundaries for the robot 10 to remain within while traversing toward the target destination 46. At operation 814, the method 800 includes navigating, by the data processing hardware 36, the robot 10 to the target destination 46 by traversing the robot 10 through each goal region 220 while maintaining the robot 10 within the at least one constraint box 230.

[0059] FIG. 9 is schematic view of an example computing device 900 that may be used to implement the systems and methods described in this document. The computing device 900 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. The components shown here, their connections and relationships, and their functions, are meant to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

[0060] The computing device 900 includes a processor 910, memory 920, a storage device 930, a high-speed interface/controller 940 connecting to the memory 920 and high-speed expansion ports 950, and a low speed interface/controller 960 connecting to a low speed bus 970 and a storage device 930. Each of the components 910, 920, 930, 940, 950, and 960, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 910 can process instructions for execution within the computing device 900, including instructions stored in the memory 920 or on the storage device 930 to display graphical information for a graphical user interface (GUI) on an external input/output device, such as display 980 coupled to high speed interface 940. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple computing devices 900 may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

[0061] The memory 920 stores information non-transitorily within the computing device 900. The memory 920 may be a computer-readable medium, a volatile memory unit(s), or non-volatile memory unit(s). The non-transitory memory 920 may be physical devices used to store programs (e.g., sequences of instructions) or data (e.g., program state information) on a temporary or permanent basis for use by the computing device 900. Examples of non-volatile memory include, but are not limited to, flash memory and read-only memory (ROM) / programmable read-only memory (PROM) / erasable programmable read-only memory (EPROM) / electronically erasable programmable read-only memory (EEPROM) (e.g., typically used for firmware, such as boot programs). Examples of volatile memory include, but are not limited to, random access memory (RAM), dynamic random access memory (DRAM), static random access memory (SRAM), phase change memory (PCM) as well as disks or tapes.

[0062] The storage device 930 is capable of providing mass storage for the computing device 900. In some implementations, the storage device 930 is a computer-readable medium. In various different implementations, the storage device 930 may be a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. In additional implementations, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 920, the storage device 930, or memory on processor 910.

[0063] The high speed controller 940 manages bandwidth-intensive operations for the computing device 900, while the low speed controller 960 manages lower bandwidth-intensive operations. Such allocation of duties is exemplary only. In some implementations, the high-speed controller 940 is coupled to the memory 920, the display 980 (e.g., through a graphics processor or accelerator), and to the high-speed expansion ports 950, which may accept various expansion cards (not shown). In some implementations, the low-speed controller 960 is coupled to the storage device 930 and a low-speed expansion port 990. The low-speed expansion port 990, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet), may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device such as a switch or router, e.g., through a network adapter.

[0064] The computing device 900 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server 900a or multiple times in a group of such servers 900a, as a laptop computer 900b, or as part of a rack server system 900c.

[0065] Various implementations of the systems and techniques described herein can be realized in digital electronic and/or optical circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0066] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms “machine-readable medium” and “computer-readable medium” refer to any computer program product, non-transitory computer readable medium, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term “machine-readable signal” refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0067] The processes and logic flows described in this specification can be performed by one or more programmable processors, also referred to as data processing hardware, executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit). Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0068] To provide for interaction with a user, one or more aspects of the disclosure can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube), LCD (liquid crystal display) monitor, or touch screen for displaying information to the user and optionally a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's client device in response to requests received from the web browser.

[0069] A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the disclosure. Accordingly, other implementations are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method (800) comprising:

receiving, at data processing hardware (36), from at least one sensor (31) in communication with the data processing hardware (36), sensor data (17) of an environment (8, 700) about a robot (10) while the robot (10) traverses through the environment (8, 700);

generating, by the data processing hardware (36), using the sensor data (17) received while the robot (10) traverses through the environment (8, 700), a plurality of waypoints (510) and a plurality of edges (520), each edge (520) connecting a respective pair of the plurality of waypoints (510); and

after the robot (10) traverses through the environment (8, 700):

receiving, at the data processing hardware (36), a target destination (46) within the environment (8, 700) for the robot (10) to navigate to;

determining, by the data processing hardware (36), a route specification (200) based on a series of waypoints (510) and corresponding edges (520) for the robot (10) to follow for navigating the robot (10) to the target destination (46), the series of waypoints (510) and corresponding edges (520) selected from the plurality of waypoints (510) and the plurality of edges (520) generated while the robot (10) traversed through the environment (8, 700);

for each waypoint (510) in the series of waypoints (510), generating, by the data processing hardware (36), a goal region (220) encompassing the corresponding waypoint (510) in the route specification (200);

generating, by the data processing hardware (36), at least one constraint region (230) encompassing at least one goal region (220), the at least one constraint region (230) establishing boundaries for the robot (10) to remain within while traversing toward the target destination (46); and

navigating, by the data processing hardware (36), the robot (10) to the target destination (46) by traversing the robot (10) through each goal region (220) while maintaining the robot (10) within the at least one constraint region (230).

2. The method (800) of claim 1, wherein receiving the target destination (46) comprises receiving the target destination (46) from a user device (40) in communication with the data processing hardware (36).

3. The method (800) of claim 1 or 2, wherein traversing the robot (10) through each goal region (220) comprises traversing a center point (202) of the robot (10) through each goal region (220) with a valid yaw configuration.

4. The method (800) of any of claims 1–3, wherein at least one goal region (220) in the route specification (200) comprises a yaw constraint configured to constrain a yaw of the robot (10) within the at least one goal region (220).

5. The method (800) of any of claims 1–4, further comprising, when generating the goal region (220) encompassing the corresponding waypoint (510) in the route specification (200), generating a target region (220m) encompassing the target destination (46), the target region comprising a smaller area than at least one of the generated goal regions (220).

6. The method (800) of claim 5, wherein the target region (220m) comprises an area smaller than each of the generated goal regions (220).

7. The method (800) of any of claims 1–6, wherein the at least one constraint region (230) encompasses two goal regions (220).

8. The method (800) of claim 7, wherein generating the at least one constraint region (230) comprises aligning the at least one constraint region (230) with the corresponding edge (520) connecting the pair of waypoints (510) each encompassed by a respective one of the two goal regions (220) encompassed by the at least one constraint region (230).

9. The method (800) of any of claims 1–8, wherein at least one of the goal regions (220) is configured to cause the robot (10) to align with a terrain feature (280) prior to the robot (10) traversing the terrain feature (280).

10. The method (800) of any of claims 1–9, wherein at least one of the goal regions (220) is configured to provide localization information based on previously captured sensor data (17).

11. The method (800) of any of claims 1–10, wherein the at least one constraint region (230) comprises route cost information (212).

12. The method (800) of any of claims 1–11, further comprising:

receiving input from a user (42) directing the robot (10) to move about the environment (8, 700); and

as the robot (10) moves about the environment (8, 700), capturing the sensor data (17) of the environment (8, 700).

13. The method (800) of any of claims 1–12, wherein the at least one sensor (31) comprises an image sensor comprising one or more of a stereo camera (31), a scanning light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

14. A method (800) comprising:

receiving, at data processing hardware (36), a navigation request (47) requesting a robot (10) to navigate from a current location (43) in a robot (10) environment (8, 700) to a target destination (46) in the robot environment (8, 700);

determining, by the data processing hardware (36), a route specification (200) for navigating the robot (10) from the current location (43) to the target destination (46) within the robot environment (8, 700), the route specification (200) comprising a series of waypoints (510) and corresponding edges (520) each interconnecting a respective pair of the series of waypoints (510);

for each waypoint (510) in the series of waypoints (510) in the route specification (200), generating, by the data processing hardware (36), a goal region (220) encompassing the corresponding waypoint (510) in the route specification (200);

generating, by the data processing hardware (36), at least one constraint region (230) encompassing at least one goal region (220), the at least one constraint region (230) establishing boundaries for the robot (10) to remain within while traversing toward the target destination (46); and

navigating, by the data processing hardware (36), the robot (10) from the current location (43) to the target destination (46) by traversing the robot (10) through each goal region (220) while maintaining the robot (10) within the at least one constraint region (230).

15. A system comprising:

data processing hardware (36); and

memory hardware (38) in communication with the data processing hardware (36), the memory hardware (38) storing instructions that when executed on the data processing hardware (36) cause the data processing hardware (36) to perform operations comprising:

receiving from at least one sensor (31) in communication with the data processing hardware (36), sensor data (17) of an environment (8, 700) about a robot (10) while the robot (10) traverses through the environment (8, 700);

generating using the sensor data (17) received while the robot (10) traverses through the environment (8, 700), a plurality of waypoints (510) and a plurality of edges (520), each edge (520) connecting a respective pair of the plurality of waypoints (510); and

after the robot (10) traverses through the environment (8, 700):

receiving a target destination (46) within the environment (8, 700) for the robot (10) to navigate to;

determining a route specification (200) based on a series of waypoints (510) and corresponding edges (520) for the robot (10) to follow for navigating the robot (10) to the target destination (46), the series of waypoints (510) and corresponding edges (520) selected from the plurality of waypoints (510) and the plurality of edges (520) generated while the robot (10) traversed through the environment (8, 700);

for each waypoint (510) in the series of waypoints (510), generating a goal region (220) encompassing the corresponding waypoint (510) in the route specification (200);

generating at least one constraint region (230) encompassing at least one goal region (220), the at least one constraint region (230) establishing boundaries for the robot (10) to remain within while traversing toward the target destination (46); and

navigating the robot (10) to the target destination (46) by traversing the robot (10) through each goal region (220) while maintaining the robot (10) within the at least one constraint region (230).

16. The system of claim 15, wherein receiving the target destination (46) comprises receiving the target destination (46) from a user device (40) in communication with the data processing hardware (36).

17. The system of claim 15 or 16, wherein traversing the robot (10) through each goal region (220) comprises traversing a center point (202) of the robot (10) through each goal region (220) with a valid yaw configuration.

18. The system of any of claims 15–17, wherein at least one goal region (220) in the route specification (200) comprises a yaw constraint configured to constrain a yaw of the robot (10) within the at least one goal region (220).

19. The system of any of claims 15–18, further comprising, when generating the goal region (220) encompassing the corresponding waypoint (510) in the route specification (200), generating a target region encompassing the target destination (46), the target region comprising a smaller area than at least one of the generated goal regions (220).

20. The system of claim 19, wherein the target region comprises an area smaller than each of the generated goal regions (220).

21. The system of any of claims 15–120, wherein the at least one constraint region (230) encompasses two goal regions (220).

22. The system of claim 21, wherein generating the at least one constraint region (230) comprises aligning the at least one constraint region (230) with the corresponding edge (520) connecting the pair of waypoints (510) each encompassed by a respective one of the two goal regions (220) encompassed by the at least one constraint region (230).

23. The system of any of claims 15–22, wherein at least one of the goal regions (220) is configured to cause the robot (10) to align with a terrain feature (280) prior to the robot (10) traversing the terrain feature (280).

24. The system of any of claims 15–23, wherein at least one of the goal regions (220) is configured to provide localization information based on previously captured sensor data (17).

25. The system of any of claims 15–24, wherein the at least one constraint region (230) comprises route cost information (212).

26. The system of any of claims 15–25, further comprising:

receiving input from a user (42) directing the robot (10) to move about the environment (8, 700); and

as the robot (10) moves about the environment (8, 700), capturing the sensor data (17) of the environment (8, 700).

27. The system of any of claims 15–26, wherein the at least one sensor (31) comprises an image sensor comprising one or more of a stereo camera (31), a scanning

light-detection and ranging (LIDAR) sensor, or a scanning laser-detection and ranging (LADAR) sensor.

28. A system comprising:

data processing hardware (36); and

memory hardware (38) in communication with the data processing hardware (36), the memory hardware (38) storing instructions that when executed on the data processing hardware (36) cause the data processing hardware (36) to perform operations comprising:

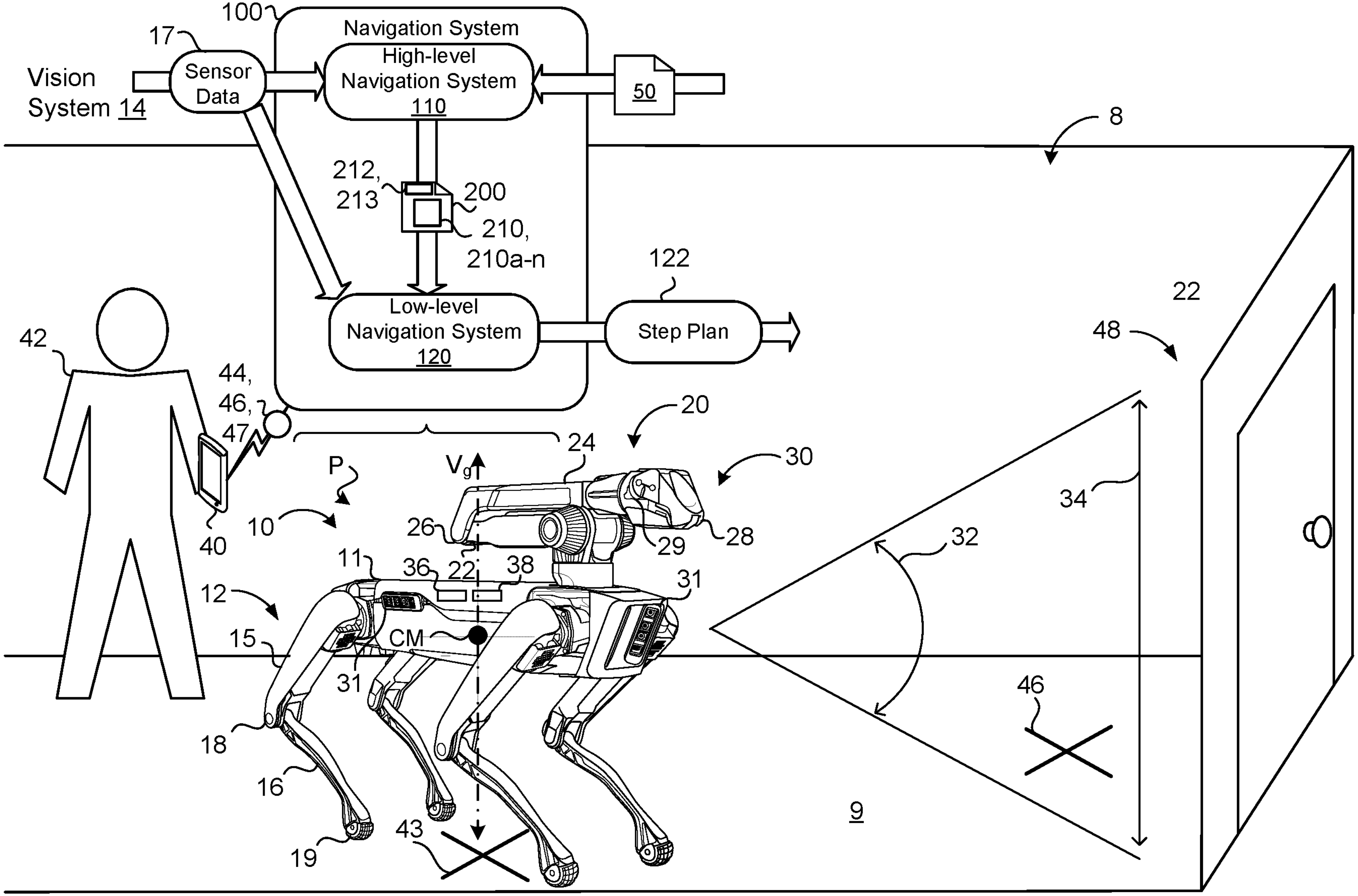
receiving a navigation request (47) requesting a robot (10) to navigate from a current location (43) in a robot environment (8, 700) to a target destination (46) in the robot environment (8, 700);

determining a route specification (200) for navigating the robot (10) from the current location (43) to the target destination (46) within the robot environment (8, 700), the route specification (200) comprising a series of waypoints (510) and corresponding edges (520) each interconnecting a respective pair of the series of waypoints (510);

for each waypoint (510) in the series of waypoints (510) in the route specification (200), generating a goal region (220) encompassing the corresponding waypoint (510) in the route specification (200);

generating at least one constraint region (230) encompassing at least one goal region (220), the at least one constraint region (230) establishing boundaries for the robot (10) to remain within while traversing toward the target destination (46); and

navigating the robot (10) from the current location (43) to the target destination (46) by traversing the robot (10) through each goal region (220) while maintaining the robot (10) within the at least one constraint region (230).



100
Navigation System
17
Vision System 14
Sensor Data
High-level Navigation System 110
50
8
212, 213
200
210, 210a-n
122
Low-level Navigation System 120
Step Plan
22
48
42
44, 46, 47
20
24
30
P
Vg
10
26
29
28
34
32
40
11
36
22
38
31
12
CM
15
31
46
18
16
9
43
19

FIG. 1

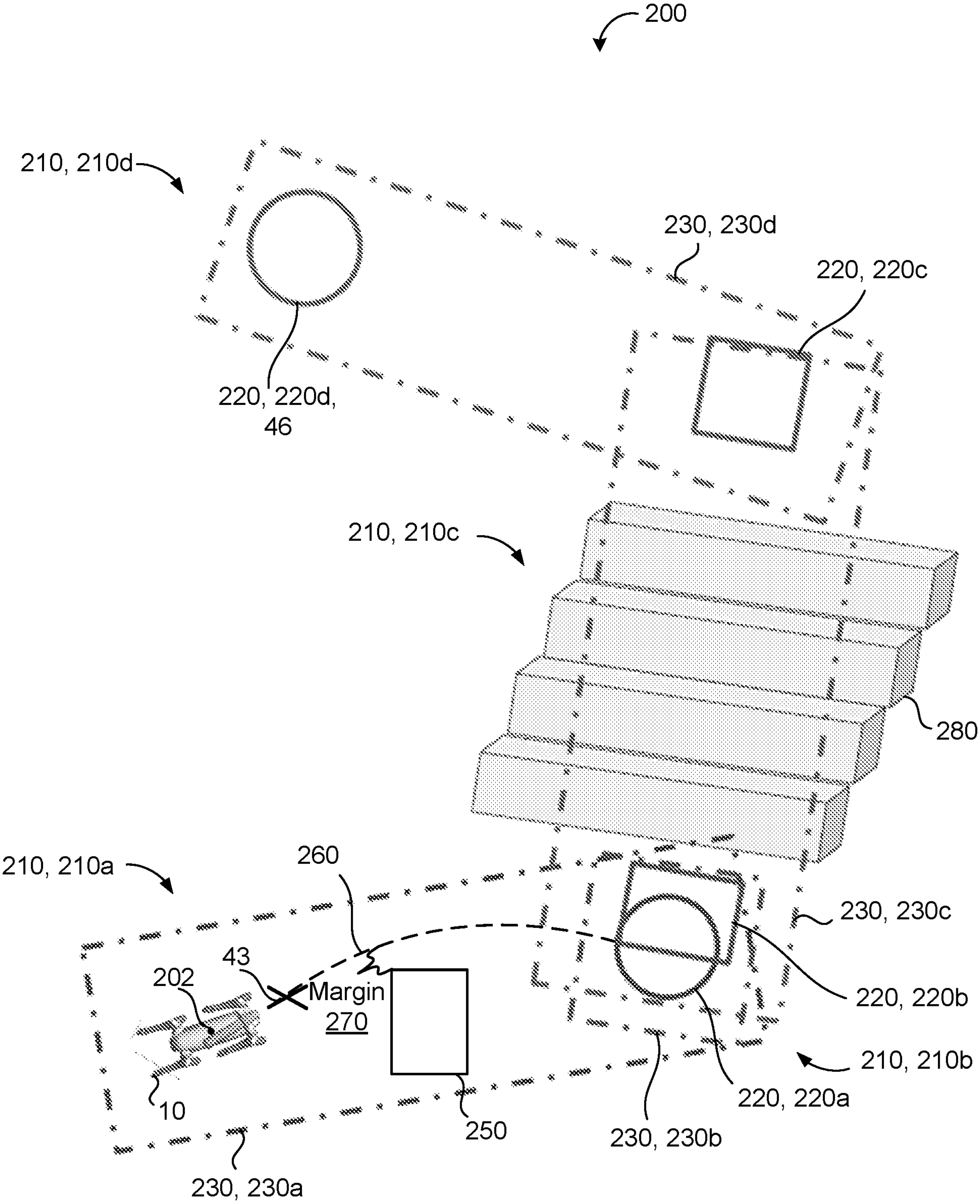

200
210, 210d
230, 230d
220, 220c
220, 220d,
46
210, 210c
280
210, 210a
260
230, 230c
43
202
Margin
270
220, 220b
210, 210b
10
250
230, 230b
220, 220a
230, 230a

FIG. 2

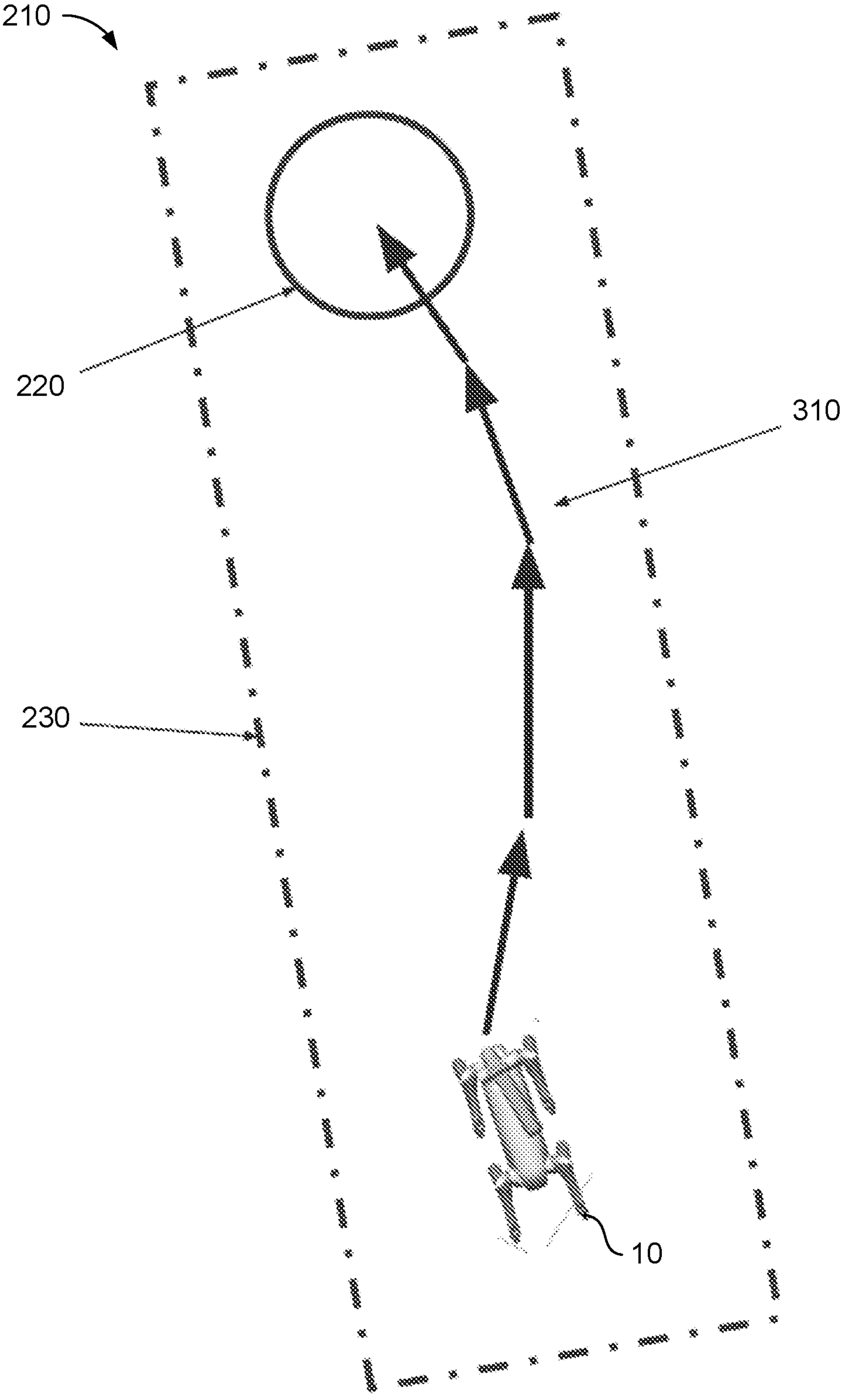


FIG. 3

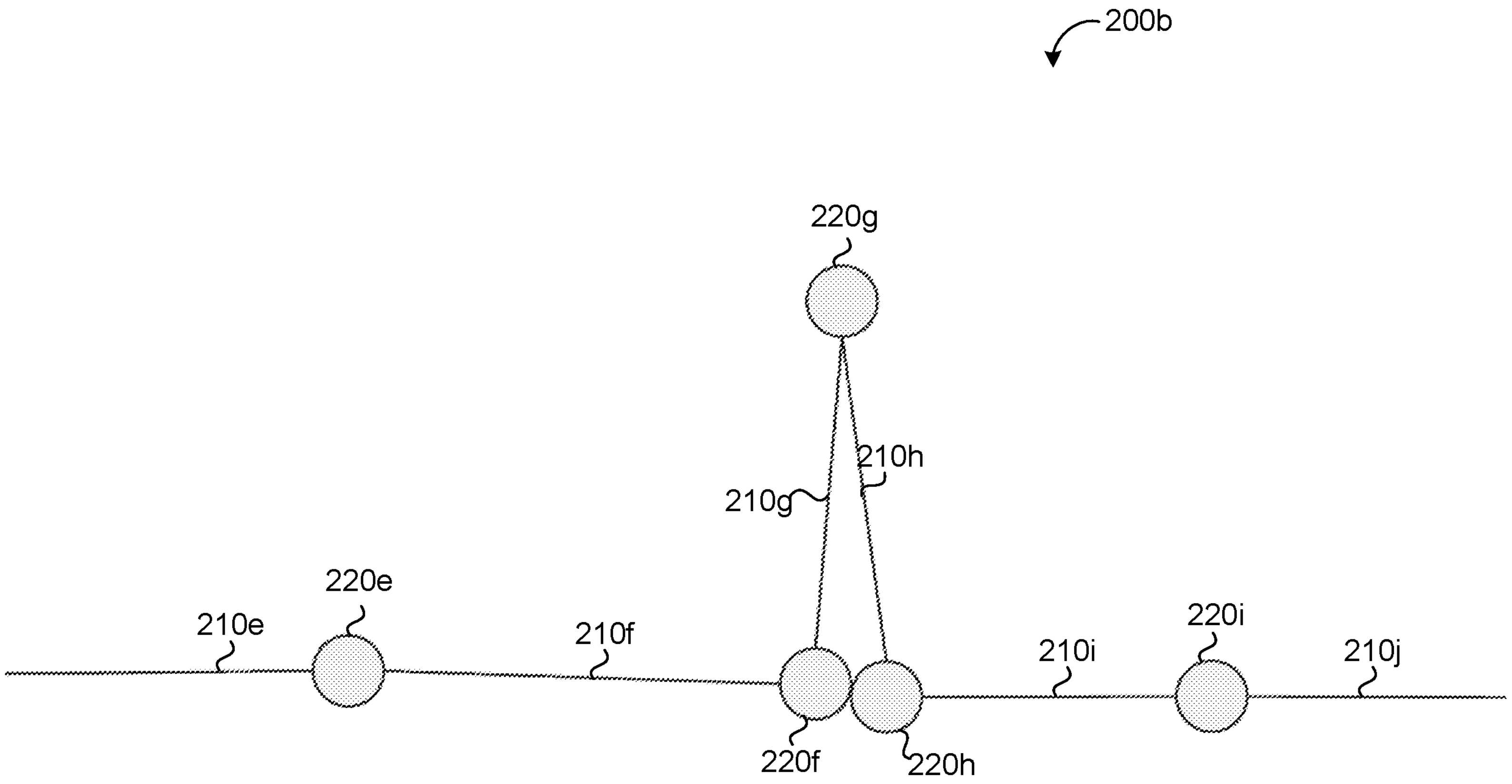


FIG. 4

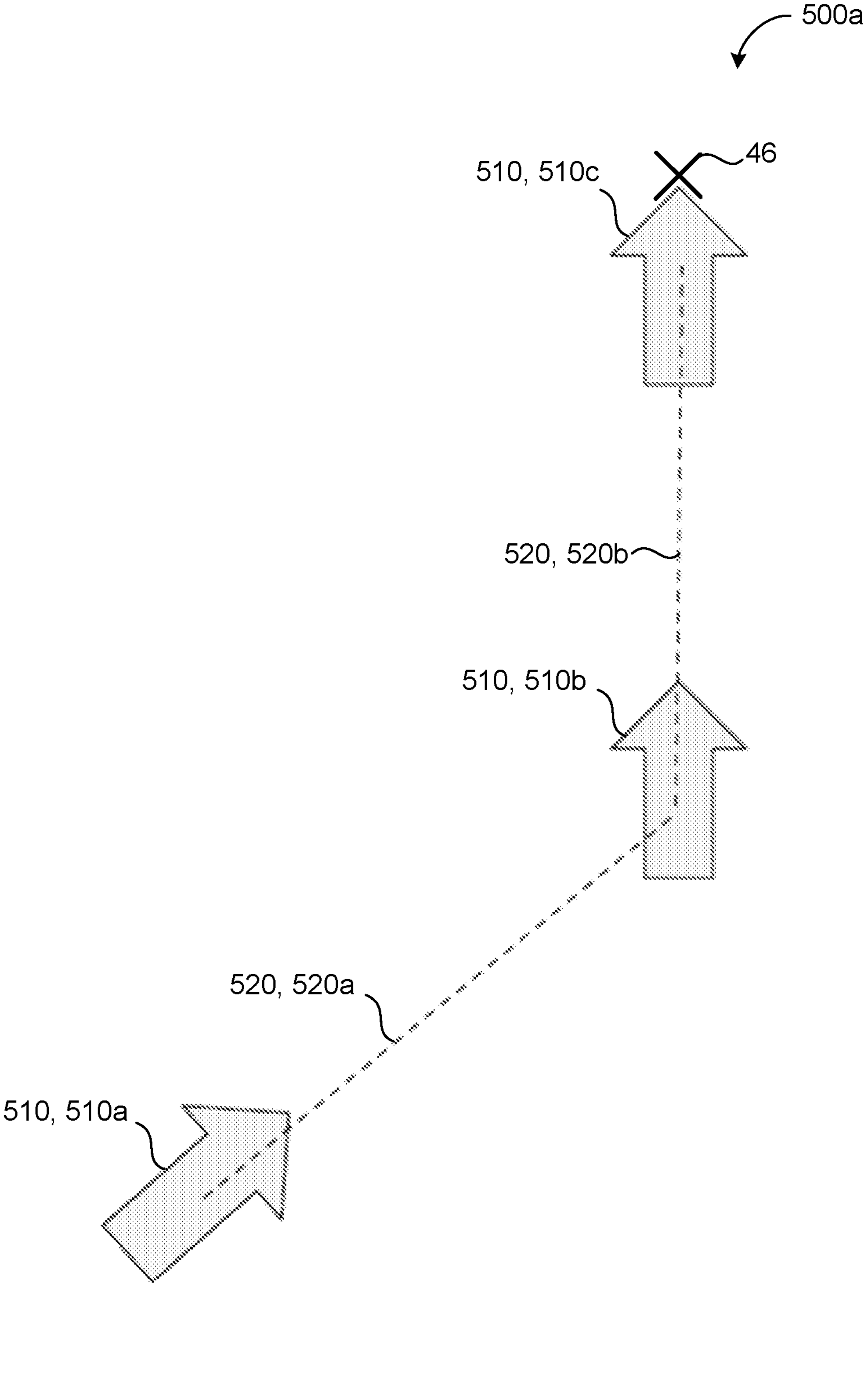


FIG. 5A

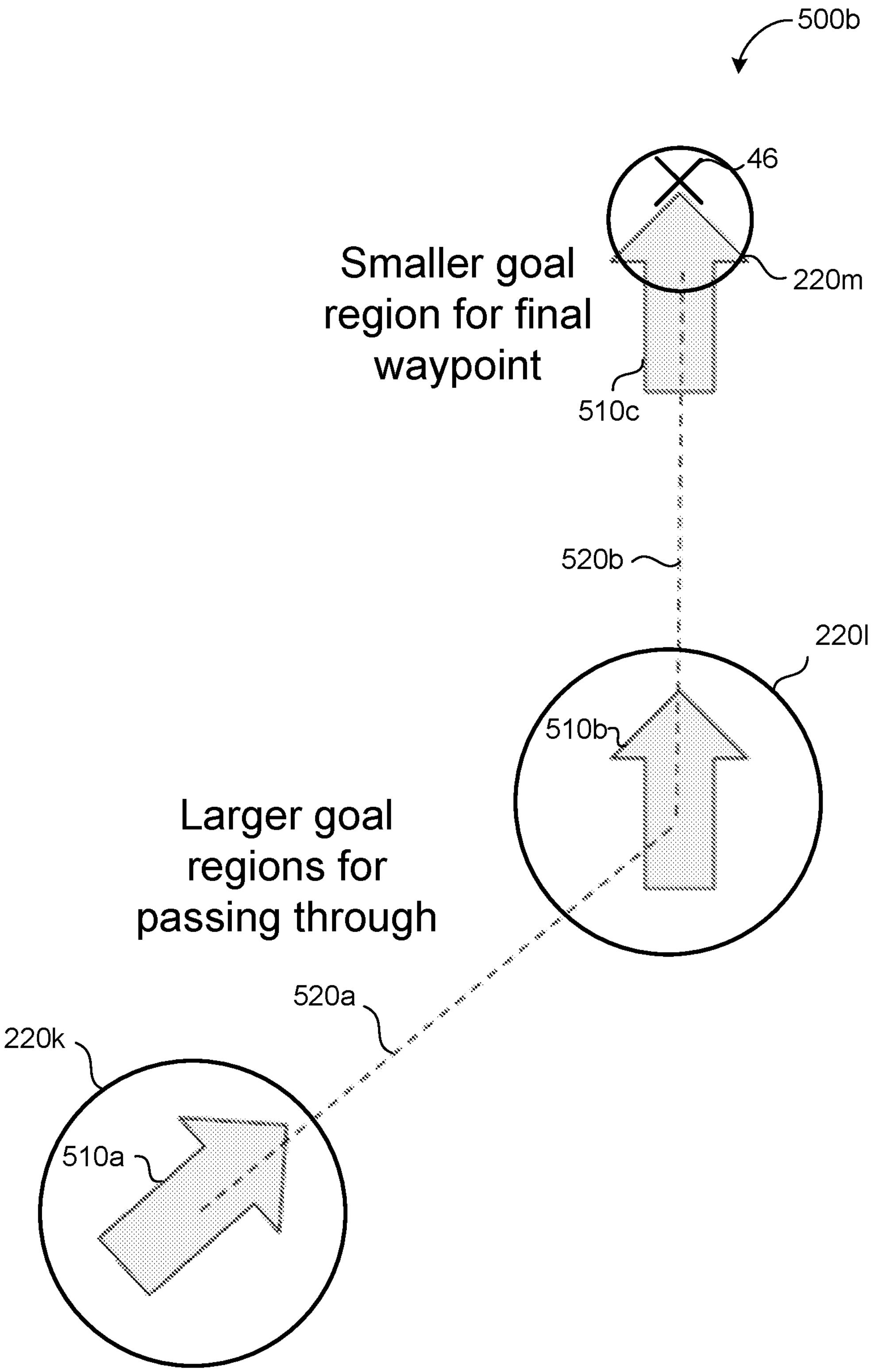

500b
46
Smaller goal region for final waypoint
220m
510c
520b
220l
510b
Larger goal regions for passing through
520a
220k
510a

FIG. 5B

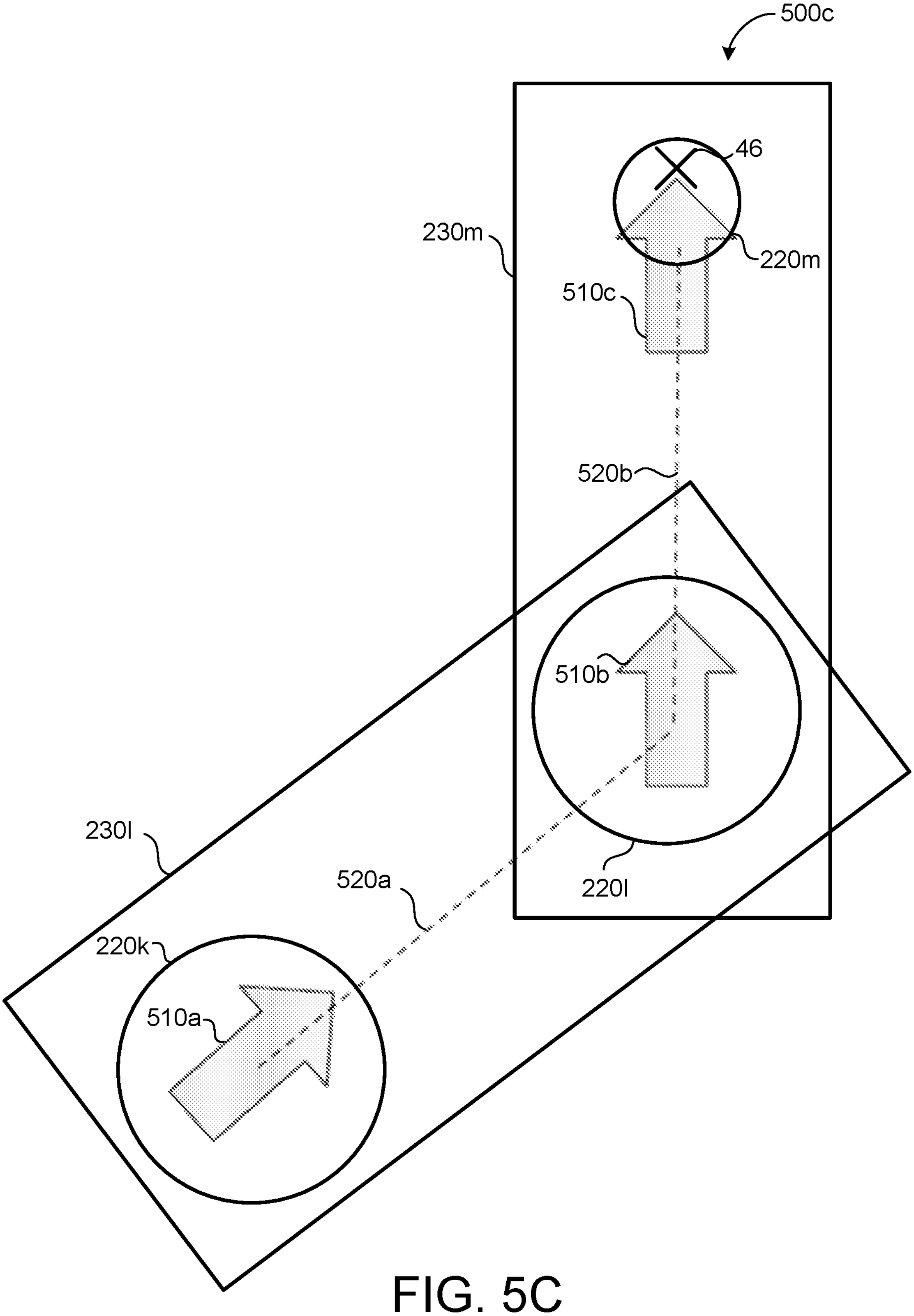


FIG. 5C

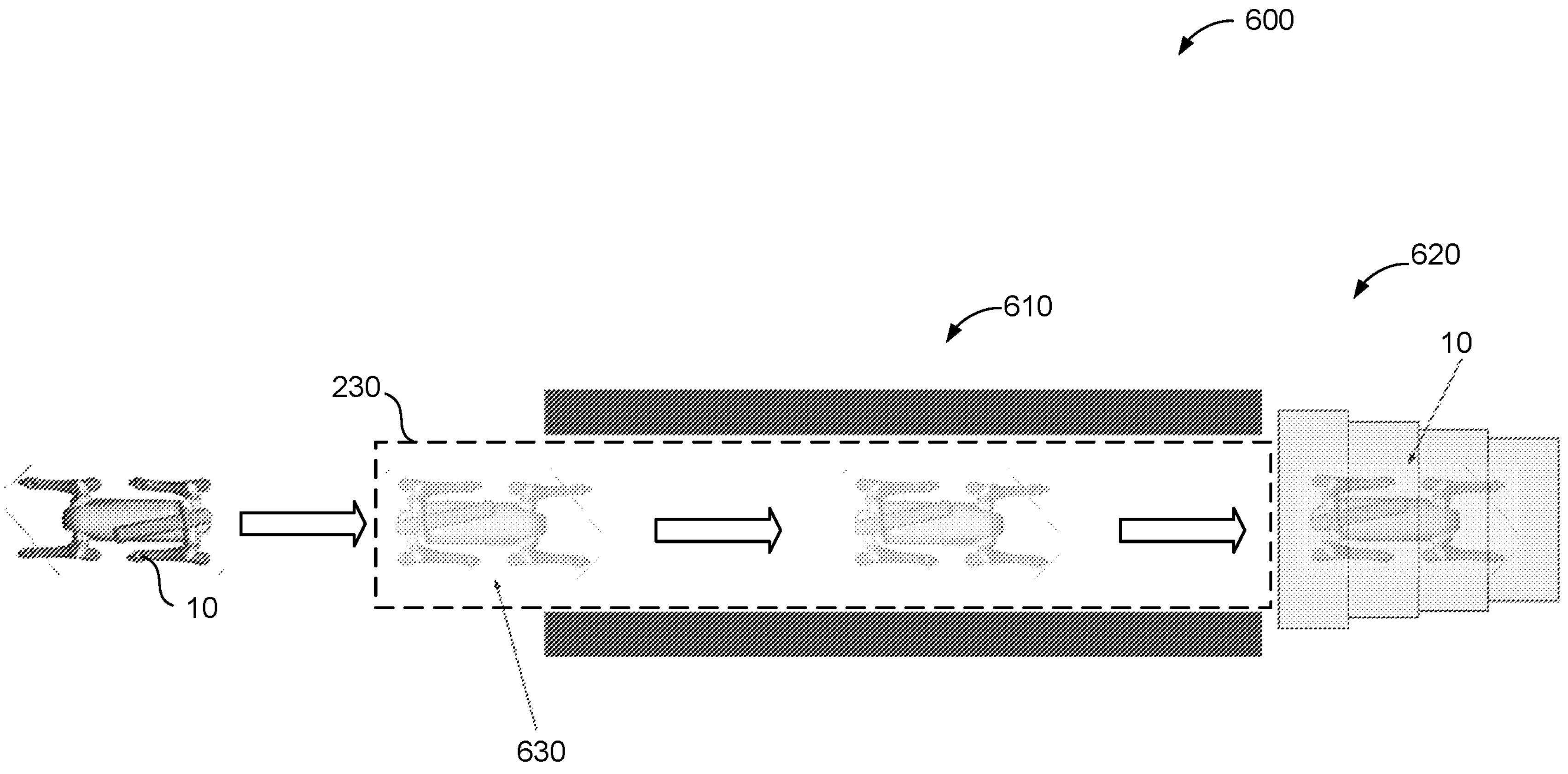


FIG. 6

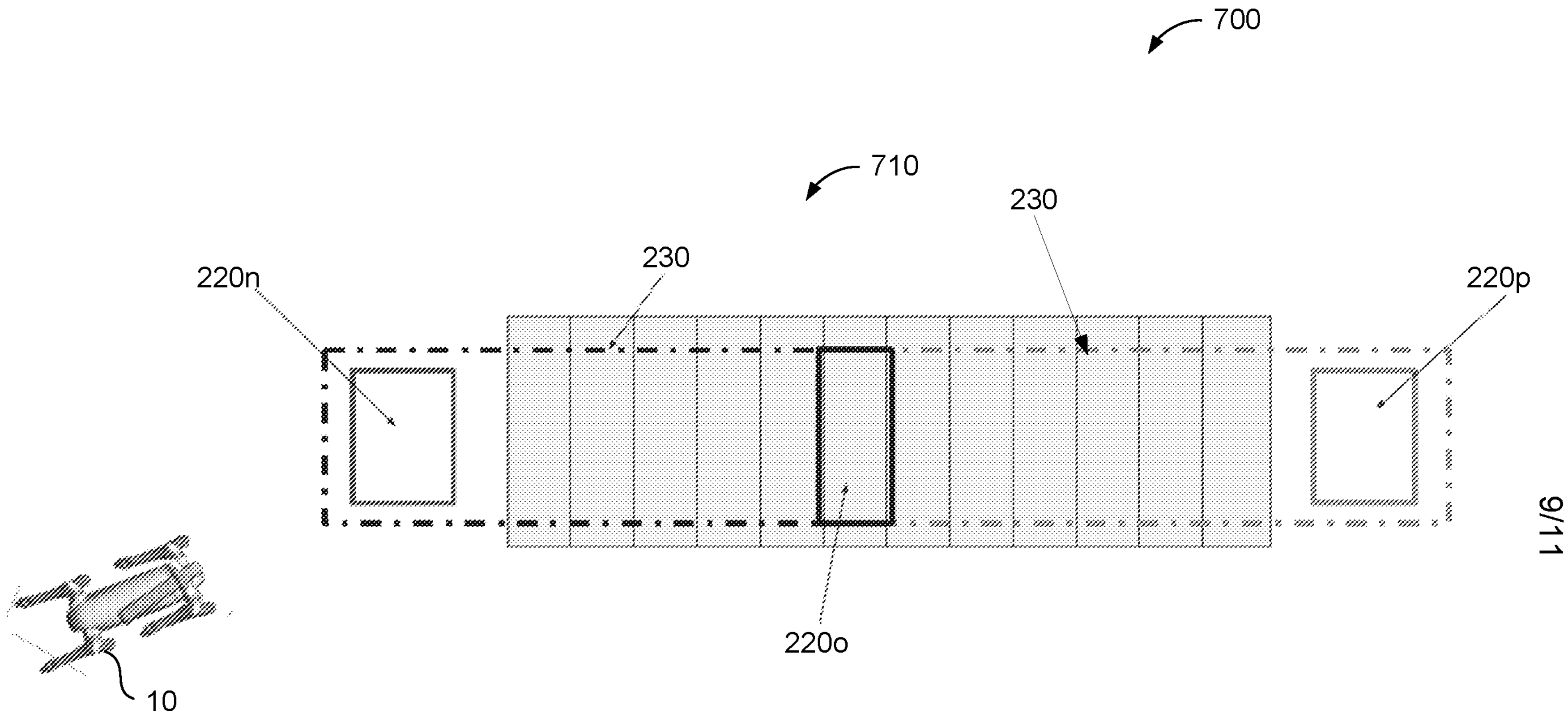


FIG. 7

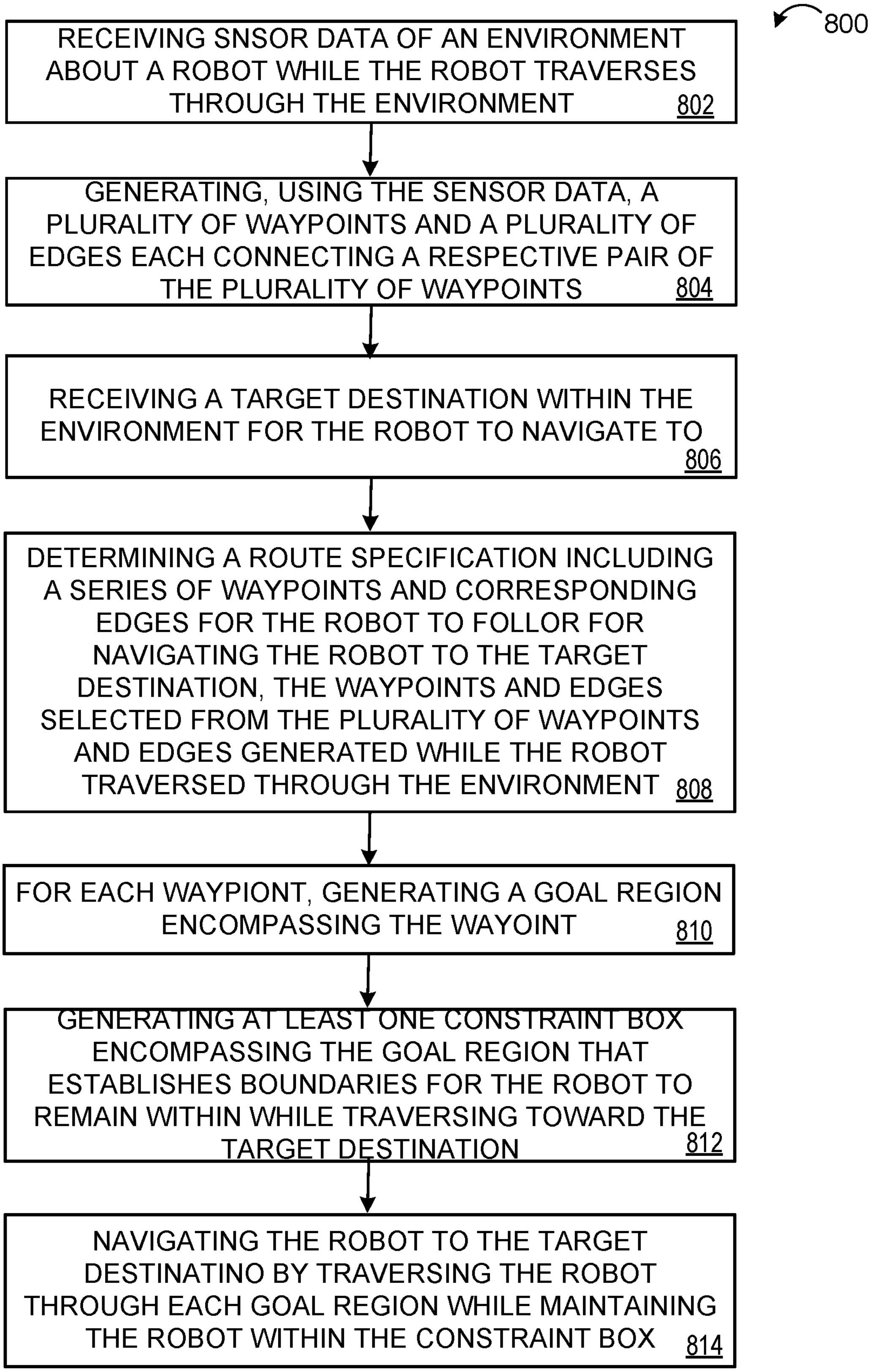

800
RECEIVING SNSOR DATA OF AN ENVIRONMENT ABOUT A ROBOT WHILE THE ROBOT TRAVERSES THROUGH THE ENVIRONMENT 802
GENERATING, USING THE SENSOR DATA, A PLURALITY OF WAYPOINTS AND A PLURALITY OF EDGES EACH CONNECTING A RESPECTIVE PAIR OF THE PLURALITY OF WAYPOINTS 804
RECEIVING A TARGET DESTINATION WITHIN THE ENVIRONMENT FOR THE ROBOT TO NAVIGATE TO 806
DETERMINING A ROUTE SPECIFICATION INCLUDING A SERIES OF WAYPOINTS AND CORRESPONDING EDGES FOR THE ROBOT TO FOLLOR FOR NAVIGATING THE ROBOT TO THE TARGET DESTINATION, THE WAYPOINTS AND EDGES SELECTED FROM THE PLURALITY OF WAYPOINTS AND EDGES GENERATED WHILE THE ROBOT TRAVERSED THROUGH THE ENVIRONMENT 808
FOR EACH WAYPIONT, GENERATING A GOAL REGION ENCOMPASSING THE WAYOINT 810
GENERATING AT LEAST ONE CONSTRAINT BOX ENCOMPASSING THE GOAL REGION THAT ESTABLISHES BOUNDARIES FOR THE ROBOT TO REMAIN WITHIN WHILE TRAVERSING TOWARD THE TARGET DESTINATION 812
NAVIGATING THE ROBOT TO THE TARGET DESTINATINO BY TRAVERSING THE ROBOT THROUGH EACH GOAL REGION WHILE MAINTAINING THE ROBOT WITHIN THE CONSTRAINT BOX 814

FIG. 8

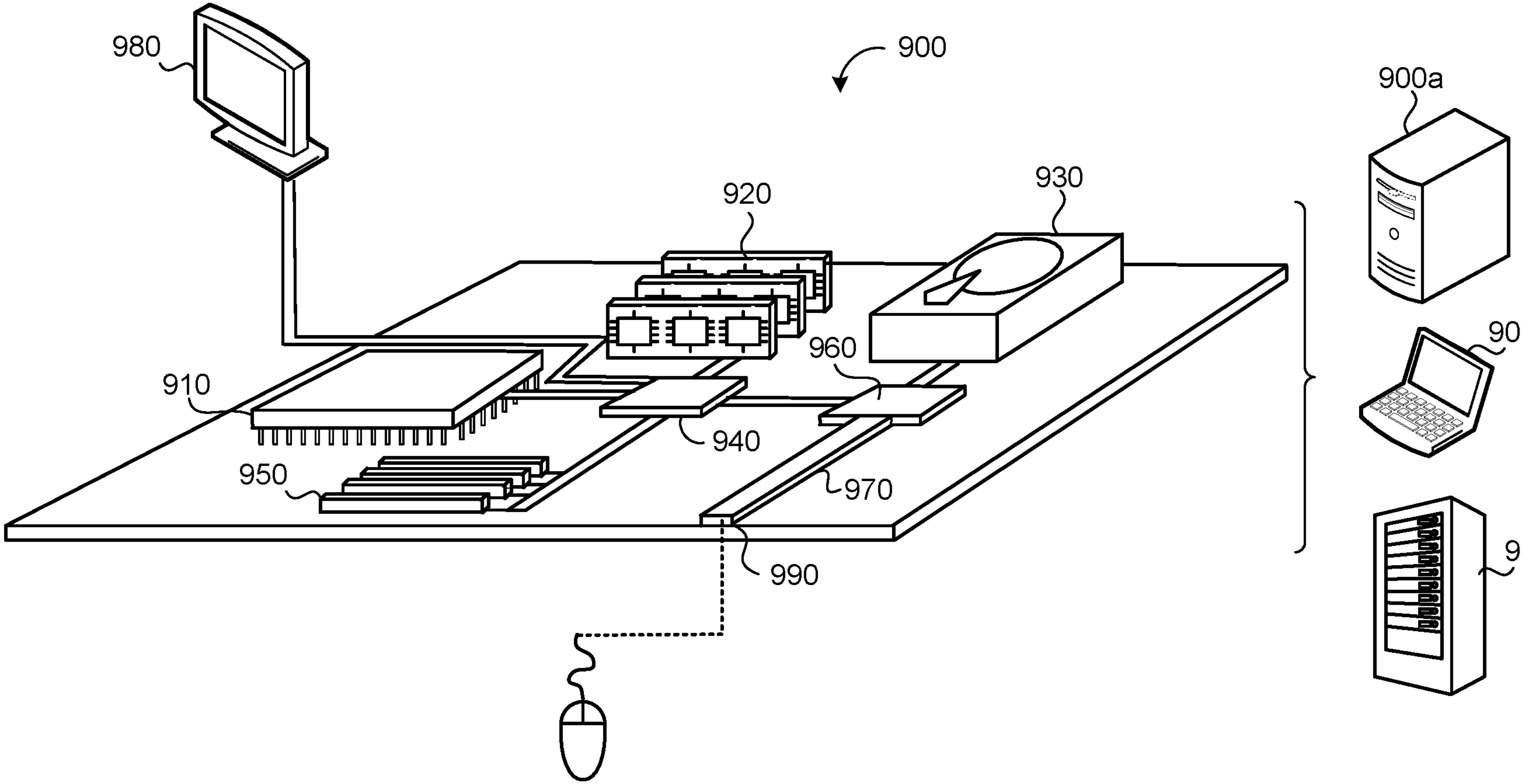


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/024785

A. CLASSIFICATION OF SUBJECT MATTER
INV. G05D1/02
ADD.

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

B. FIELDS SEARCHED

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

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2020/076422 A2 (BOSTON DYNAMICS INC [US]) 16 April 2020 (2020-04-16) paragraphs [0004] - [0018], [0038] - [0062] -----	1-28
A	US 10 471 611 B2 (IROBOT CORP [US]) 12 November 2019 (2019-11-12) the whole document ----- -/--	1-28

☒ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 14 June 2021	Date of mailing of the international search report 22/06/2021
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Etienne, Yves

2

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	NICHOLAS MARIO WARDHANA ET AL: "Enhanced Waypoint Graph for Path Planning in Virtual Worlds", CYBERWORLDS (CW), 2012 INTERNATIONAL CONFERENCE ON, IEEE, 25 September 2012 (2012-09-25), pages 69-76, XP032265621, DOI: 10.1109/CW.2012.17 ISBN: 978-1-4673-2736-7 the whole document -----	1-28
A	BRAUN T ET AL: "Topological large-scale off-road navigation and exploration RAVON at the European Land Robot Trial 2008", INTELLIGENT ROBOTS AND SYSTEMS, 2009. IROS 2009. IEEE/RSJ INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 10 October 2009 (2009-10-10), pages 4387-4392, XP031580346, ISBN: 978-1-4244-3803-7 the whole document -----	1-28
X,P	WO 2021/025707 A1 (BOSTON DYNAMICS INC [US]) 11 February 2021 (2021-02-11) paragraphs [0003] - [0017], [0071] - [0072] -----	1-28

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2021/024785

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2020076422 A2	16-04-2020	CN 112840285 A	25-05-2021
		KR 20210058998 A	24-05-2021
		US 2020117214 A1	16-04-2020
		US 2021141389 A1	13-05-2021
		WO 2020076422 A2	16-04-2020
US 10471611 B2	12-11-2019	CN 108885436 A	23-11-2018
		EP 3403146 A1	21-11-2018
		US 2017203446 A1	20-07-2017
		US 2020001475 A1	02-01-2020
		WO 2017123761 A1	20-07-2017
WO 2021025707 A1	11-02-2021	US 2021041243 A1	11-02-2021
		WO 2021025707 A1	11-02-2021