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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 September 2015 (24.09.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/143273 A2

(51) International Patent Classification:
G06F 19/00 (2011.01)

(21) International Application Number:
PCT/US2015/021672

(22) International Filing Date:
20 March 2015 (20.03.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/968,826 21 March 2014 (21.03.2014) US

(71) Applicant: **RENSSELAER POLYTECHNIC INSTITUTE** [US/US]; 110 8th Street, Troy, NY 12180 (US).

(72) Inventors: **WEN, John, T.**; 168 Meadows Drive, Melrose, NY 12121 (US). **BRAASCH, Jonas**; 6 Colonie Avenue, Latham, NY 12110 (US). **SINHA, Utkarsh**; 2727 Baton Rouge Drive, San Jose, California 95133 (US). **CUNNINGHAM, Andrew**; 6 Ponderosa Drive, Voorheesville, NY 12186 (US). **KEDDY-HECTOR, William, Henry**; 67 North Street, Saratoga Springs, New York 12866 (US). **KRUSE, Daniel, Charles**; 167 4th Street, Apt. 3, Troy, NY 12180 (US). **WHALEN, David**; 4 Compton Place, Scotia, NY 12302 (US). **LU, Lu**; 2439 22nd Street, Apt. 6, Troy, NY 12180 (US).

(74) Agent: **HOFFMAN, Michael, F.**; Hoffman Warnick LLC, 540 Broadway, Albany, NY 12207 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: MOBILE HUMAN-FRIENDLY ASSISTIVE ROBOT

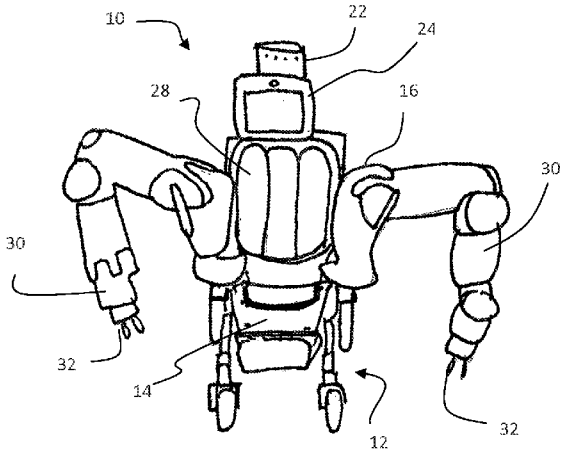

FIG. 1

(57) Abstract: A robotic assistant, associated software and methodology for operating the same. The described robotic assistant includes: a motorized base having at least two motor driven wheels controlled by a first control platform; a dual arm robot mounted on the motorized base, the dual arm robot having a first arm and a second arm controlled by a second control platform; a remote sip and puff mouth controller having three degrees of operational freedom; and a computer system that receives command signals from the remote sip and puff mouth controller, and includes an algorithm that translates the command signals into a first type of control signal for directing the motorized base to move, and a second type of control signal for directing the dual arm robot to move.

WO 2015/143273 A2

MOBILE HUMAN-FRIENDLY ASSISTIVE ROBOT

TECHNICAL FIELD

[0001] The subject matter of this invention relates to mobile assistive robots, and more particularly to dual arm mobile assistive robot devices integrated into a motorized wheelchair for assisting mobility impaired individuals.

BACKGROUND

[0002] Mobility disabled individuals usually require full-time caretakers to be present with them to help with daily activities, which can be very expensive in the long run. One approach in addressing this need is to develop a robotic assistant that could help the user perform daily chores that they are unable to perform themselves. These robotic assistants can also enable remote caretaking through the use of telepresence and remote teleoperation, and also provide means to watch over a person and alert relevant personnel in case of emergencies.

[0003] Despite the proliferation of industrial robots in fields such as automotive and semiconductor manufacturing, food processing, and material handling, existing robots largely operate in a teach-and-repeat mode, with limited human interaction. For less structured environments, such as in assistive living, human guidance and intervention is essential.

[0004] Current robotic solutions for mobility impaired individuals mostly consist of a single robotic arm mounted on a wheelchair, which limits its usage to simple tasks like 'pick and place'. However, these arms are usually controlled via a joystick and as a result, are not appropriate for individuals who are paralyzed from the neck or chest down (approximately 5000 per year in US), or have limited use of their hands.

Furthermore, adapting existing robots that are controllable via human guidance for mobility impaired individuals are faced with numerous economic and technological hurdles that have yet to be fully addressed by the prior art.

[0005] Research work is also being done to develop fully automated robots that can navigate and pick/place objects by themselves. However, these systems only exist in academia and are tailored for specific scenarios or workspace. Furthermore, they do not handle interaction with the environment well, such as opening doors or drawers, or performing dual-arm tasks such as holding large objects such as a serving tray or manipulating flexible objects such as a tablecloth. As such, solutions that could potentially be useful for mobility impaired individuals are not available. Accordingly, a need exists for a robotic assistant that can help severely handicapped individuals.

SUMMARY

[0006] The present disclosure describes a robotic assistant that is more versatile, cheaper, and which can be remotely controllable by anyone whose mobility is impaired. The disclosed robotic assistant generally comprises a motorized base and dual arm robot mounted thereon. The robotic assistant is designed to be utilized by mobility impaired individuals through a sip-and-blow interface, tongue command device, eye/blink tracker, a joystick, etc., to, for example, command the motion of the assistant to pick up objects from floor or shelf, to put food or drink into microwave oven, to perform light cleaning tasks, to inspect suspicious noise, etc.

[0007] The assistive robot can also serve as a surrogate for medical staff/caregivers at a remote location, e.g., via the Internet, to inspect and examine the individual through vision and touch (with possible haptic feedback) and deployment of different biometric sensors such as a thermometer, stethoscope, oximetry, ophthalmoscope, etc.

[0008] Sensors (e.g., vision, proximity, touch) on the robotic assistant may be used to provide feedback to the user to facilitate task execution (e.g., warning signal or automatic avoidance for impending collision, etc.). The multi-arm feature of the platform may also be used in a social companion/assistant context, to encourage exercise, play games, etc.

[0009] The platform may be connected to the Internet to allow remote operation by medical staff/caregivers to inspect/examine/interact with a patient from remote surroundings. Through the distributed software architecture, the platform may be integrated into sensors/actuators in the user environment (e.g., a smart home) as a mobile sensor for home lighting, HVAC, and security systems.

[0010] The described embodiments have advantages over existing solutions, including: (1) The multi-arm capability allows a wider range of tasks to be performed such as handling large and bulky objects, manipulating flexible objects such as clothing, cables and sheets, and inherent two arm tasks such as opening jars, preparing food, etc.

(2) The interface is adapted to handle a suite of input devices, such as mouth controllers, joysticks, etc., accessible to anybody who is mobility impaired.

(3) The robotic assistant can serve as a telepresence platform. Current telepresence systems do not have manipulation capabilities (let alone dual-arm capabilities). This platform can provide a much wider range of diagnostic and examination options, including touch and feel (through haptic feedback), and deployment of diagnostic devices.

(4) With the arms, this platform is anthropomorphic, allowing it to provide a social companion function. Because the system is teleoperated, the solution is an order of magnitude cheaper compared to systems that utilize fully automated robots.

(5) The automatic regulation of the force of interaction from the arms interacting with the environment through contacts, or multiple arms holding a jointly-held object, allows the user to safely and efficiently execute tasks such as door opening, cleaning, transporting large trays, etc.

(6) The user can map the user input to any specified degrees of freedom, such as the position or orientation of each arm, the position and orientation of the mobile base, or the position or orientation of the midpoint between the arms for two-arm manipulation, and the remaining degrees of freedom of the system will be automatically determined to best match with the user command while satisfied the collision avoidance and other constraints.

[0011] A first aspect provides a robotic assistant, comprising: a motorized base having at least two motor driven wheels controlled by a first control platform; a dual arm robot mounted on the motorized base, the dual arm robot having a first arm and a second arm controlled by a second control platform; a remote sip and puff mouth controller having at least three degrees of operational input; and a computer system that receives command signals from the remote sip and puff mouth controller, and includes a control system provides for coordinating movement of the motorized base and the robot arms by translating the command signals into a first type of control signal for directing the motorized base to move, and into a second type of control signal for directing the dual arm robot to move.

[0012] A second aspect provides a control platform for controlling the operation of a robotic assistant, comprising: a control system that controls movement of a motorized based and a dual arm robot mounted on the motorized base, wherein the control system implements a first type of control signal to control the motorized base, and a second type of control signal to control the dual arm robot; an input for receiving

command signals from a remotely located controller, wherein the command signals include a proposed movement of a selected frame in a three dimensional space; and a software algorithm that coordinates the motion of the both the motorized base and dual arm robot based on the proposed movement of the selected frame.

[0013] A third aspect provides a method for controlling arms of a robotic assistant having a motorized based and a dual arm robot mounted on the motorized base, comprising: receiving user commands from a remote input device to cause a gripper of a first arm to grasp an object; receiving user commands from the remote input device to cause a gripper of a second arm to also grasp the object; selecting a frame relative to the first arm and second arm; selecting between a position mode and an orientation mode; in response to selecting position mode, maintaining an orientation of the frame while allowing a user to move the point via the remote input device to effectuate movement of the object; and in response to selecting orientation mode, maintaining a position of the frame while allowing the user to reorient the frame via the remote input device to effectuate movement of the object.

[0014] A forth aspect provides a method for controlling a robotic assistant having a motorized based and a dual arm robot mounted on the motorized base, comprising: receiving user commands from a remote input device to move an arm of the dual arm robot; causing a contact that results in a resistance being detected by a sensor on the arm; establishing a force set point by causing a desired amount of force to be exacted in a determined direction; and receiving user commands to effectuate a movement of the arm while maintaining the force set point.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] These and other features of this invention will be more readily understood from the following detailed description of the various aspects of the invention taken in conjunction with the accompanying drawings in which:

[0016] Figure 1 shows a front view of a robotic assistant according to embodiments.

[0017] Figure 2 shows a rear isometric view of the holder according to embodiments.

[0018] Figure 3 shows an illustrative mouth input device according to embodiments.

[0019] Figure 4 shows the kinematic model of a robotic assistant according to embodiments.

[0020] Figure 5 shows an illustrative operational architecture according to embodiments.

[0021] Figure 6 shows an illustrative graphical representation of a robotic assistant according to embodiments.

[0022] Figure 7 depicts a computing infrastructure of the robotic assistant according to embodiments.

[0023] Figure 8 depicts a graph showing inequality constraints according to embodiments.

[0024] Figure 9 shows a block diagram of a hybrid control structure according to embodiments.

[0025] Figure 10 shows a flow diagram showing a process for operating the robotic assistant according to embodiments.

[0026] Figure 11 shows a flow diagram showing a process for operating both arms of the robotic assistant according to embodiments.

[0027] Figure 12 shows a flow diagram showing a process for employing force as a constraint to operate the robotic assistant according to embodiments.

[0028] The drawings are not necessarily to scale. The drawings are merely schematic representations, not intended to portray specific parameters of the invention. The drawings are intended to depict only typical embodiments of the invention, and therefore should not be considered as limiting the scope of the invention. In the drawings, like numbering represents like elements.

DETAILED DESCRIPTION

[0029] Referring now to the drawings, Figure 1 depicts a front view and Figure 2 depicts rear isometric view of a mobile assistive robot (hereinafter "robot" or "robotic assistant") 10 that can be remotely controlled, e.g., by mobility disabled individuals to improve their quality of life. Recent research efforts in the field of manipulation-based assistive robotics are centered on creating autonomous robotic assistants. In contrast to this common paradigm, the present approach focuses on providing a platform for allowing a handicapped individual to manually control the robotic assistant 10. Namely, a platform is provided so that the mobility disabled person is able to control the robotic assistant 10 using an interface in order to increase the individual's independence and to maximize the number of tasks the robotic assistant 10 can complete.

[0030] Robotic assistant 10 generally comprises a motorized base 12 with at least two independently driven wheels and at least two additional small caster wheels for balance, an on-board manipulator that generally comprises a dual arm robot 16 sensors, a distributed control platform and/or other input/output devices. Robotic assistant 10 is controlled by a remote a human input device such as mouth controller, joystick, or other device.

[0031] In one embodiment, the robotic assistant 10 is an integration of a commercially available retrofitted power wheelchair, an off the shelf dual arm robotic system such as the dual arm BAXTER™ robot by Rethink Robotics™, and a JAMBOXX™ sip and puff mouth controller. The control platform is generally managed by a computing system, such as a general purpose computer (e.g., a laptop 26) or special purpose computing device. Although various embodiments are described in terms of commercially available systems, such as a retrofitted wheelchair, Baxter robot and Jamboxx device, it is understood that other commercially available systems could likewise be utilized, as well as custom built systems. For the purposes of this disclosure, however, the trade names "Baxter" and "Jamboxx" are intended to generally describe a dual arm robot 16 and mouth controller 40, respectively, and may therefore be used as such without limiting the scope of the invention to a particular implementation.

[0032] While dual arm robots are regularly used for various industrial applications, their use is primarily limited to preprogrammed repetitive tasks. The present invention draws on tools from both industrial robotics for object manipulation and handling, and social robotics, where robots provide human communication and assistance by performing more than simple repetitive tasks. To achieve this, a human-scale lightweight dual arm robot system 16, such as the Baxter robot, is utilized. Such a robotic system can safely operate in the presence of humans and utilizes a 120V power supply, which can easily integrate with the battery power supply on the mobile base through an inverter 20.

[0033] As shown in Figures 1 and 2, robotic assistant 10 generally includes a retrofitted power wheel chair base (i.e., motorized base) 12 that includes a motorized drive and wheels, a battery pack compartment 18, a mounting adaptor plate 14 for

mounting the dual arm robot 16, and a back plate 34 for mounting equipment and accessories. The use of existing power wheelchair technology as the motorized base 12 for the robotic assistant 10 greatly reduces the cost, while providing a proven solution. The motorized base 12 may for example be retrofitted from a QUICKIE™ s646 or other types of wheelchair powered by two lead-acid batteries (and/or other power sources such as fuel cell stacks) which can easily be modified for computer interface and control.

[0034] Battery pack compartment 18 may utilize any type of system (e.g., lead or lithium batteries, fuel cells, etc.) for providing power to both the wheel chair base 12 and dual arm robot 16. Because the Baxter robot operates on 120V AC, an onboard inverter 20 is utilized to convert the power for the dual arm robot 16. The dual arm robot 16, such as the Baxter robot, generally includes a torso 28, left and right arms 30, grippers 32, a display screen 24, plurality of cameras (e.g., proximate the grippers 32 and screen display 24), and head sonar distance sensors 22. An onboard laptop computer 26 is provided along with additional sensors, such as infrared or the like. In addition, a positional data collector may be implemented within the motorized base 12 and laptop 26 to track the position of the robotic assistant in a space over time.

[0035] The dual armed robot 16 as shown includes a total of fourteen degrees of freedom with seven degrees of freedom in each arm 30 and is designed to be safe to operate near humans. The robotic assistant 10 as a whole is controlled by receiving remote commands from the operator to the laptop 26 mounted on the robot 10. Once a command is received, it is interpreted by the laptop 26 and sent to either the dual armed robot, if it is a manipulation command, or the wheelchair circuitry within the motorized base 12, if it is a motion command.

[0036] As noted, the Baxter robot consists of two 7-degree-of-freedom arms fixed to a rigid base, along with a suite of sensors found along the torso and on each arm. Each arm's kinematics consists of a collocated wrist, a near-collocated shoulder with a small offset between the first and second joints, and an elbow. The kinematics for the assistant are shown in Figure 4 and are described in further detail elsewhere herein. Each joint servo is connected to an encoder and torque sensor for closed loop control and to detect servo loading from contact. Additionally, at the "end-effector" of each arm is a parallel jaw gripper 32, an infrared ranger, and a camera. On the 'head' of the robot is a ring of sonar sensors 22, a display screen 24 and rotatable camera.

[0037] In the described embodiment, the mobile base with batteries weighs approximately 118.8 kg while the Baxter robot weighs an additional 74.8 kg – similar to a normal adult and well within the payload range of a power wheelchair. One of the chief concerns is to ensure the safety of those around the system during use. For the mechanical design, this manifested itself into making sure that the combined system would not be in danger of tipping during normal operation. Of the possible causes for instability, the two most likely scenarios are tipping from rapid acceleration of the motorized base 12, and tipping due to center of gravity displacement. The first of these is resolved by establishing speed limits on the motorized base 12. A 'Safe' arm configuration is also chosen for rapid motion. In order to ascertain how the system's stability would be affected by the Baxter's arm movements, the center of gravity for the entire system was calculated in the most extreme possible arm configurations. It is important for the system to remain stable in all arm configurations without limiting the work envelope or payload capacity. The most effective mounting location was then chosen based on the results from these

calculations. The adapter plate 14 was designed in order to locate the combined center of gravity as close to the center of the wheelbase as possible while still allowing full arm movement with the 2.3 kg max payload for which Baxter is rated.

[0038] With the chosen mounting location, the arms from Baxter have a working envelope that extends from the ground directly in front of the wheelchair to a maximum working height of 1.63 m off the ground. Each arm is also able to extend 1.04 m out to either side of the robot or 0.68 m in front of the motorized base 12. The combined footprint of an illustrative system is only 0.914 m in length and 0.673 m in width making it only 2 cm wider than the power wheelchair itself and equally as long. This is an important feature due to door and hallway width restrictions that are likely to exist in many households. In addition, a rear mounting surface 34 was designed in order to accommodate the equipment needed to mobilize the system. This provides a mounting location for the inverter 20, control laptop 26, as well as any future hardware. By locating these objects on the opposite side of the center of mass from the robotic end effecters, the stability margin was further improved particularly for the scenario when heavy objects need to be retrieved from the ground in front of the system.

[0039] There are several devices on the robot that need to be powered including the Baxter 16, laptop 26 and motorized base 12. Both the Baxter 16 and the laptop 26 are generally designed to be powered through a typical wall outlet while the motorized base 12 requires a 24V DC line. The power architecture is structured around the power source used in the robot, e.g., two 12V/69Ah gel-cell batteries. The batteries are connected in series to power the mobile base 12 and in parallel to inverter 20. The inverter 20 converts the DC power into AC power.

[0040] The overall software architecture is shown in Figure 5. The internal control system for the dual arm robot 16 may be implemented with a Robot Operating System (ROS) 33 on the robot computer 25. The robot control platform 31 subscribes to the joint positions and publishes joint velocities at a rate of 100 Hz. There are additional services 35 running onboard the robot computer 25 to prevent self-collision, publish gravity compensation torques to the servos, and, most importantly, maintain safety by halting the servos during an unexpected impact. A custom script using, e.g., the Rethink Robotics Python SDK, may for example be utilized to make the Baxter robot subscriptions and publications, while wrapped in a Robot Raconteur 37 service to host the relevant joint information on a local network. This allows for rapid development in high-level languages, distribute computational effort among a network of computers, and incorporate high-performance visualizations into the control architecture.

[0041] Controlling the robotic assistant as a whole requires simultaneous and coordinated control over both the motorized base 12 and the dual arm robot 16. To accomplish this, a laptop 26 is mounted on the rear mounting surface 34 to send the appropriate control signals to a corresponding device. When the laptop 26 receives a message from a user via a remote computer 39, it sends a command to either the Baxter 16 or the motorized base 12. The laptop 26 is connected to the Baxter robot 16 through Ethernet and has the capability to send commands directly to the computer 25 inside Baxter 16. The laptop 26 receives task space commands (left, right, up, down, etc.) for moving Baxter's arms wirelessly from the remote computer 39 via an input device 41. It then uses a Jacobian based damped least-squares controller to calculate the joint velocities required to move the arms in the desired direction. These joint velocities are then transmitted via a custom Python script to read Robot Raconteur

messages and publish ROS messages to the corresponding rostopics on Baxter's computer 25. This script acts as a Robot Raconteur ROS bridge 43.

[0042] In contrast, the mobile base 12 is controlled through intermediate circuitry. The laptop 26 is connected to an Arduino 45 (via USB) which is then connected to the controller 47 for the motorized base 12. When the laptop 26 receives a command to move the mobile base 12, it converts the command into a serial byte to send to the Arduino 45. Once the Arduino 45 receives this byte, it sends a control signal to a digital potentiometer to mimic a physical joystick. This signal is passed to the mobile base's control system and functions as if it were receiving commands from a physical joystick.

[0043] Referring now to Figure 7, an illustrative control platform is shown in which laptop 26 is provided with a robot manager 58 that includes: (1) a set of communication interfaces 60 for communication with various systems, and (2) a control system 62 for interpreting commands and controlling the robotic assistant 10. In this example, robot manager 58 is adapted to receive remote interface commands 64 (e.g., from a Jamboxx) or Internet based commands 66 (e.g., from a health care provider). Whenever a command is received, it is interpreted by the control system 62 to move or control aspects of the robotic assistant 10. As shown, the robot manager 58 is capable of sending and receiving command signals to the motorized base 12, dual arm robot 16, sensors 68 or other devices 70. As noted, the motorized base 12 includes its own control platform having software for causing the motorized base 12 to move/respond according to received commands. The dual arm robot 16 likewise has its own unique control platform and software for causing its arms to move/respond. Because the control platforms for the two may be distinct, each is controlled with unique input commands. In addition, sensors 68 and other devices 70

may also be controlled by the control system 62. The control system 62 includes a control algorithm 63, described elsewhere herein, for coordinating movements of the motorized base 12 and dual arm robot 16, based on inputs, constraints, etc.

[0044] In one illustrative embodiment, the user is able to select a "frame" which may comprise any point on an arm of the Baxter 16, or any point relative to the two arms. In one typical case, the frame may comprise an end-effector on an arm. Once selected, the user can manipulate the frame to effectuate a movement of the robotic assistant 10. For example, using an input device 41, the user is able to control and move a selected end-effector in a three dimensional space. The computing platform will determine the best way to control the robotic assistant's movement in response to the inputted commands, including moving the mobile base 12 when necessary.

[0045] Any type of input device 41 may be utilized to operate the robotic assistant 10, including, e.g., a Jamboxx or a joystick. Using such devices, two possible control schemes may be implemented. The first control scheme involves placing all functionality required to control the robot 10 on a single input device. This results in the ability to drive the mobile base and operate the arms at the same time with ease.

[0046] One example of a mouth controller 40 is depicted in Figure 3, which utilizes a sip and puff controller with three degrees of freedom including: (i) a slider that can be shifted left and right with the mouth, (ii) the rotation of the same slider up and down, and (iii) a pressure sensor attached to the slider used as suck/puff interface (breath controller). The design focuses on high accuracy and low movement resistance so it can be used over longer periods of time. Using this device, an impaired user can control a computer with the precision of a computer mouse.

[0047] A second control scheme involves a hybrid approach and utilizes both the Jamboxx and joystick. This scheme was tailored towards for use by quadriplegics as

they are able to proficiently use the Jamboxx and can often control a joystick, like the ones mounted on a wheelchair, as well. In one embodiment, Baxter is controlled by the Jamboxx while the motorized base 12 is controlled by a joystick.

[0048] In addition to the controller layout, feedback from the system is delivered to the user in two ways. First, the Baxter robot 16 is equipped with three onboard cameras which can stream video to the operator. There is a head camera which provides an overhead view of the motion of the system, as well as two cameras located on the end-effectors of the two arms. By using a combination of the viewpoints, the operator is able to see how the robotic assistant 10 is positioned in order to complete the desired task.

[0049] A custom MATLAB simulation package may be utilize to give the operator real-time feedback of the arm positions. This may be done by approximating the joints and linkages of the robot with a set of primitive 3D shapes. By using the known forward kinematics of the system, the current arm configuration is drawn by passing the drawing package a series of joint angles for each arm. A simulated configuration is shown in Figure 6. Having the capability to simulate the robotic system also provides the dual functionality of allowing the designers to test different control layouts, or allowing users to get comfortable operating the robotic assistant 10, without needing to activate the physical system. This will be particularly useful when individuals with different physical disabilities require custom control interfaces.

[0050] In the first control scheme noted above, both the mobile base 12 and dual arm robot 16 are controlled by a single input device, such as a Jamboxx. There are two typical operations: local tasks, which are within the arm workspace, and non-local tasks, which require base motion to position the task within the arm workspace. Arm and base can be controlled separately or together as an integrated unit. In the former

case, the mobile base moves the arm into the area of interest (i.e., workspace) and remains stationary while the arm performs the task. This requires switching between the two modes, placing added burden on the user. In the latter case, the arm and base may move together even when the base motion is unnecessary. Such unanticipated base motion compromises safety and, because of the weight of the system, reduces responsiveness and consumes more energy.

[0051] In the following embodiment, a general human-robot cooperative control approach is provided in which the system automatically determines if the mobile base needs to be moved to get an arm into a desired workspace. The arm (or arms) will then move when the motorized base 12 is within the work space. The arm(s) prefer to stay stationary while the motorized base 12 moves, but the process is automated so that the user does not need to switch between the two modes.

[0052] This is achieved by providing a highly redundant mobile robotic system with the human user providing motion command through an input device with a limited number of degrees of freedom. The control problem is posed as a constrained optimization at the kinematic level that determines the robot joint velocity to minimize the deviation from the human commanded motion input. A set of equality and inequality constraints is imposed based on joint and velocity limits, singularity avoidance, and collision prevention. In addition, an equality constraint is included to ensure that the base motion follows the commanded end-effector motion in a natural way by avoiding unanticipated combined arm/base motion. Since the problem is kinematic in nature, the resulting minimization becomes a convex quadratic program which guarantees a unique, and efficiently solvable, solution. This type of kinematic based algorithm is local in nature, meaning that even a globally feasible motion plan exists for the overall task, the motion may get stuck locally.

[0053] One goal is to locally follow the human command as closely as possible without violating the constraints, and rely on the human user to steer the system for the global task. A key motivation for this work is to provide assistance to severely mobility impaired individuals.

[0054] A Jamboxx 40 may be used as the input device to control both arms of Baxter 16 in a task space. Typically, task space controllers use Jacobian-based methods to map the end-effector velocities to joint velocities. The general form of the Jacobian matrix J_T is

$$J_T = \begin{bmatrix} h_1 & h_2 & \cdots & h_7 \\ h_1^{\wedge} p_{1T} & h_2^{\wedge} p_{2T} & \cdots & h_7^{\wedge} p_{7T} \end{bmatrix}, \qquad (1)$$

where each h_i is the axis of revolution for joint i, and p_{iT} is the vector from joint i to the end-effector. All vectors are represented in the Baxter base frame. Each arm on Baxter has seven joints, so the Jacobian is a 6x7 matrix. The Jacobian maps between joint velocities and task spatial velocity:

$$\begin{bmatrix} \omega_T \\ v_T \end{bmatrix} = J_T \dot{q}. \qquad (2)$$

Given a set of desired linear and angular velocity for the end-effector, it is possible to use the damped-least squares or other similar algorithms to command the joint motion:

$$\dot{q} = \underbrace{J_T^T (J_T J_T^T + \beta I)^{-1}}_{J_T^{\dagger}} \begin{bmatrix} \omega_T \\ v_T \end{bmatrix}. \qquad (3)$$

[0055] The coupling between position and orientation of the robot end-effector may be non-intuitive for user to command. Instead, we allow the user to command

position and orientation separately. The overall control consists of the following steps:

1) The user uses Jamboxx to select from a variety of options (see Figure 4):

– Desired direction of movement for end-effector (forward, backward, left, right, up, down)

– Desired wrist joint movement in the positive or negative direction

– Opening or closing of the gripper

2) The selected option is then translated into a fixed linear end-effector velocity (v_T) or a fixed wrist joint velocity (q_i, for i = 5, 6, or 7).

– For linear motion, the angular velocity (wT) is set at zero and the damped-least squares controller is used to generate the desired joint velocity vector, which is then published to Baxter.

– For angular motion, the wrist joint velocities are directly published to Baxter.

[0056]

[0057] Figure 4 shows a schematics of the system at the zero configuration, where all the joint angles are at zero, and the motorized base 12 is centered at the origin and facing the positive x-direction of the inertia frame. The numerical values of the robot parameters are:

a = 0.069m; b = 0.364m; c = 0.069m; d = 0.374m e = 0.01m;

$pBO_L = [0{:}004, 0{:}259, 0{:}390]^T m$.

[0058] Only the left arm is labeled; the right arm is identical to the left, symmetrical with respect to the x-z plane of the base frame. The link parameters of the arms are a, b, c, d as shown in the diagram. The displacement of the base of the arms from the motorized base 12 coordinate center is given by the vector pBO_L and pBO_R. The joint

angles of the left and right arms are denoted by the vectors q_L and q_R, respectively, where q_L consists of $\{q_{L1}, \ldots q_{L7}\}$ and q_R consists of $\{q_{R1}, \ldots q_{R7}\}$. Assume no wheel slippage, the kinematics of the motorized base 12 is given by the standard unicycle model:

$$\underbrace{\begin{bmatrix} \dot{\theta}_B \\ \dot{x}_B \\ \dot{y}_B \end{bmatrix}}_{\dot{q}_B} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & \cos\theta_B \\ 0 & \sin\theta_B \end{bmatrix}}_{J_B} \begin{bmatrix} \omega_B \\ v_B \end{bmatrix}$$

where (x_B, y_B) is the motorized base 12 center with respect to inertia frame, Θ_B is the orientation of the motorized base 12, and (w_B, v_B) are the turning and linear velocities of the motorized base 12. In this discussion, only one arm is considered at a time, e.g., the left arm. The differential kinematics of the left arm/base system in the mobile base frame is given by

$$V_T = J_{TL} q_L + \begin{bmatrix} e_z \omega_B \\ -(p_{BO_L} + p_{O_L T_L}) \times e_z \omega_B + e_x v_B \end{bmatrix}$$

where $VT = [wT, vT]^T$ is the spatial velocity (angular and linear) of the task frame at the end-effector, J_{TL} is the corresponding arm Jacobian, p_{OLTL} is the arm base to end-effector vector, $e_x = [1,0,0]^T$ and $e_z = [0,0,1]^T$ are the unit vectors in the base frame.

[0059] Combining these two equations, the overall kinematics is provided in a compact form:

$$V_T = J_T(q)u = \begin{bmatrix} J_{TL} & \begin{matrix} e_z \\ -(p_{BO_L} + p_{O_L T_L}) \times e_z \end{matrix} & \begin{matrix} 0 \\ e_x \end{matrix} \end{bmatrix} u$$

$$\dot{q} = J_q(q)u = \begin{bmatrix} I & 0 \\ 0 & J_B \end{bmatrix} u$$

where

$$q = \begin{bmatrix} q_L \\ q_B \end{bmatrix}, \; u = \begin{bmatrix} \dot{q}_L \\ \omega_B \\ v_B \end{bmatrix}.$$

[0060] Consider the scenario where the human user provides the desired motion, in terms of either position or velocity, through an input device. The goal is to determine the mobile manipulator joint motion (including both the manipulator and the mobile base) to stay close to the human command while satisfying a set of equality and inequality constraints. The user would typically use velocity command for large motion (the input is like a gas pedal) and position command for close-up, more precise motion. The constraints could be due to environmental constraints, singularity avoidance, or specified joint pose. There is also the more qualitative requirement (for now) that the robot motion should be "natural", avoiding unanticipated internal arm motion or the base motion.

[0061] To be more precise, denote the Euclidean frame (orientation and position) the human user is commanding by $\{E_A,O_A\}$. This could be the end-effector frame or a frame affixed somewhere on the robot, e.g., the arm elbow joint or the base. The differential kinematics is given by $V_A = J_A(q)u$, where V_A is the spatial velocity of frame A, J_A the Jacobian, q the robot configuration (manipulator joint displacement and the base position and orientation), and u the commanded velocity (manipulator joint velocities and mobile base heading velocity and angular velocity). The user may command just a subset of frame A, e.g., translation or rotation. For the purpose of

this discussion, only translation is considered, as it is the most common type of operation. Rotation may be useful in dual arm operation as well, but is omitted for brevity. Denote the corresponding linear velocity, in the mobile base frame, by v_H, which may be a direct velocity command from the user or the velocity control for the position command. The human-directed velocity v_H is related to the v_A through a selection matrix S:

$$v_H = Sv_A = SJ_A(q)u := J_H(q)u.$$

[0062] The user can also specify some joints or frames in the robot to be in a given configuration, e.g., elbow up or down to avoid bumping into objects, or fixed wrist orientation to hold a tray level. It is also important to observe joint limits, avoid singularities which would reduce mobility, and maintain a safe distance from obstacles. All of these are holonomic constraints on the robot configuration. To be specific, the problem that addressed is to determine u to best achieve the commanded velocity v_H while observing imposed equality and inequality constraints:

$$\min_u \|J_H u - v_H\| \text{ subject to}$$
$$h_E(q) = 0$$
$$h_I(q) \geq \eta > 0.$$
$$\dot{q} = J_q(q)u, \; q(0) = q_0.$$

[0063] As q is dynamically related to u, this is an optimal control problem over a specified time period [0,T] (with a given $\{v_H(t) : t \in [0,T]\}$), and the norm in the above formulations is a functional norm. The problem is relaxed by converting the equality constraint to a soft constraint (becoming part of the control objective) and the

inequality constraint to a barrier function. For the equality constraint, consider a Jacobian-based velocity control:

$$v_E := J_E(q)u = -k_E h_E(q),$$

where J_E is the Jacobian corresponding to $h_E(q)$ and k_E is the feedback gain. If the number of equality constraints is less than the number of joint velocity input, and the robot is not at a singular configuration with respect to J_E, then J_E is full row rank and this equation always has at least one solution.

[0064] For the inequality constraint, the input is required to steer the system towards the feasible region if it is infeasible, and remain in the feasible region if the system is already within it. This becomes a linear inequality constraint on u:

$$\frac{dh_{l_i}}{dt} = \frac{\partial h_{l_i}}{\partial q} J_q(q)u \geq \sigma_i, \; i = 1, \ldots, n_i$$

where Jq is the arm Jacobian (needed for the nonholonomic base constraint), n_i is the number of inequality constraints, and σ_i, shown in Figure 8, is given by

$$\sigma_i = \begin{cases} -\tan(c_i \pi/2), & \text{if } h_{l_i} \geq \eta_i + \varepsilon_i \\ -\tan(c_i \pi(h_{l_i} - \eta_i)/2\varepsilon_i) & \text{if } \eta_i < h_{l_i} < \eta_i + \varepsilon_i \\ e_i(\eta_i - h_{l_i})/\eta_i & \text{if } 0 \leq h_{l_i} \leq \eta_i \\ e_i & \text{else} \end{cases}$$

with n_i specifying the region in which the velocity constraint becomes active, e_i ensures that an infeasible inequality constraint is moved towards the feasible region, and $c_i < 1$ but close to 1 to avoid possible large amplitudes of σ_i. The dynamic

optimal control problem now becomes a static optimization problem to determine the input u at each time instant:

$$\min_{u} \|J_H u - v_H\|$$

$$J_E(q)u = -K_e h_E(q)$$

$$\frac{\partial h_L}{\partial q} J_q(q)u \geq \sigma_i,\ i = 1, \ldots, n_i.$$

[0065] A regularization term $\|u\|$ may need to be added to the cost function to ensure that the solution is unique. This problem is a convex quadratic program which may be efficiently solved using commercially available tools. If a solution exists, then the robot would remain within the feasible region while staying close to the human commanded motion. However, there could be the local minimum issue when the algorithm is "stuck" (u = 0 before the goal is reached). For the purposes of brevity, only a local solution of u is considered and reliance is on the human user to steer out of local minima, if encountered.

[0066] The design of the inequality and equality constraints is critical to well coordinated motion under human command. A set of design requirements are formulated below to guide the choice of these constraints.

1) When the user does not give commands (v_H is zero), both the arm and the motorized base 12 do not move. This requirement is obviously necessary to avoid any unintended surprise motion.

2) When the arm is inside the interior of the workspace, only the arm moves but not the motorized base 12. This requirement is to avoid excessive base motion when arm alone could follow the user command.

3) When the arm is at the workspace boundary, and the user commanded velocity is the direction that would move beyond the arm workspace, then the motorized base 12 should move to achieve the desired user motion commands with the end-effector orientation locked. This requirement allows base motion only when it is necessary to follow the user command.

4) When the arm is at the workspace boundary, the base motion should result in the end-effector motion in the same direction as the user commanded motion. This requirement will avoid the unnatural motion where the base seems to not follow the user command (with the arm also moving to compensate for the undesired base motion). Note that during the transient (as in t = 2.5 sec and t = 5 sec snapshots), the base rotates in the opposite direction as the user commanded direction. Eventually the base rotates in the right direction, but such unnatural transient motion should be avoided.

5) When the obstacle is close to the mobile base, the mobile base speed is reduced, and eventually becomes zero as the minimum distance to the obstacle decreases. This requirement is necessary for safety.

The constrained optimization problem is refined step-by-step to satisfy the above requirements. The first step is to remove the equality constraint through a reparameterization of u:

$$u = J_E^{\perp}\xi + J_E^{\dagger}v_E$$

[0067] Where the columns of $J_E^{\perp}$ is a basis of the null space of J_E and $J_E^{\dagger}$ is the Moore-Penrose pseudo-inverse of J_E. The previous becomes

$$\min_{\xi} \left\| J_H (J_E^{\perp} \xi + J_E^{\dagger} v_E) - v_H \right\|$$

subject to

$$\frac{\partial h_i}{\partial q} J_q(q)(J_E^{\perp} \xi + J_E^{\dagger} v_E) \geq \sigma_i, \; i = 1, \ldots, m_i.$$

If the matrix $J_H J_E^{\perp}$ is not of full column rank or is ill conditioned, there could be large changes in u in each iteration. One solution is to add a penalty term to prevent u to deviated too far from its previous value:

$$\min_{\xi} \left\| J_H (J_E^{\perp} \xi + J_E^{\dagger} v_E) - v_H \right\| + \lambda \left\| (J_E^{\perp} \xi + J_E^{\dagger} v_E) - u_{prev} \right\|$$

subject to the above, where u_{prev} is the input in the previous iteration, and λ is a weighting factor for the relative emphasis between changes in u versus following the user command. The problem with this approach is that even when the user is not moving, i.e., $v_H = 0$, the robot could continue to move (violating Requirement 1). An alternate approach is therefore adopted by adding a regularization term to the cost function. Let

$$J_{HE} = \begin{bmatrix} J_H \\ J_E \end{bmatrix}^{\perp}$$

Replacing the optimization problem above by

$$\min_{\xi} \left\| J_H (J_E^{\perp} \xi + J_E^{\dagger} v_E) - v_H \right\| + \lambda \left\| (J_{HE}^{T} (J_E^{\perp} \xi + J_E^{\dagger} v_E) \right\|$$

subject to the same constraint. Since

$$\begin{bmatrix} \begin{bmatrix} J_E \\ J_H \end{bmatrix} \\ J_{HE}^T \end{bmatrix}$$

is square invertible,

$$\begin{bmatrix} \begin{bmatrix} J_E \\ J_H \end{bmatrix} \\ J_{HE}^T \end{bmatrix} J_E^{\#} = \begin{bmatrix} \begin{bmatrix} I & 0 \\ J_H J_E^{\#} \end{bmatrix} \\ J_{HE}^T J_E^{\#} \end{bmatrix}$$

is of full column rank. This implies

$$\bar{\mathcal{N}}((J_H + \bar{\lambda} J_{HE}^T) J_E^{\#})^{-} = \{0\}$$

where *N* denotes the null space. Hence the minimization problem is strictly convex, meaning that there will be no unwanted motion, and at the same time, the user command tracking or motion imposed by the equality constraints are not be unduly penalized. The constant λ is again a weight constant between motion in the null space, and user command following. It is preferable to move the arm, instead of the base, whenever possible, as the arm is lighter weight and flexible, and therefore more responsive, consumes less energy, and less threatening to the user. To emphasize arm motion over base motion, a penalty term is added to the base motion. Let C be a matrix that select the base velocity command from u:

$$\begin{bmatrix} v_B \\ \omega_B \end{bmatrix} = Cu.$$

Define

$$J_{HEC} = \begin{bmatrix} J_H \\ J_E \\ C \end{bmatrix}^{\perp}$$

The above is replaced by

$$\min_{\xi} \left\| J_H(J_E^{\perp}\xi + J_E^{+}v_E) - v_H \right\| + \lambda_1 \left\| (J_{HEC}^{T}(J_E^{\perp}\xi + J_E^{+}v_E) \right\| + \lambda_2 \left\| C(J_E^{\perp}\xi + J_E^{+}v_E) \right\|$$

where λ1 is the weighting for the null space motion and λ2 is the weighting for the base motion. If the user command and equality constraints may be satisfied with arm motion alone without violating the inequality constraints, this method ensures that there would be no base motion, satisfying Requirements 2 and 3. In the specific case when the user commands a fix point on the arm to translate in certain direction and the arm reaches its boundary, it is more natural to have the base follow the desired direction of motion (projected onto the x-y plane), rather than having other combined base/arm motion. In this case, a single equality constraint is added

$$e_z^T(v_H \times (\omega_B \times p_{0T} + v_B)) = J_{EB}u = 0$$

where p_{0T} is the vector from center of the wheel axle to 0T, and J_{EB} is the matrix that captures the constraint. The equality constraint is only enforced in the ez direction since one is only interested in the x-y components of v_H. This additional constraint is simply appended to the existing equality constraint

$$\underbrace{\begin{bmatrix} J_E \\ J_{EB} \end{bmatrix}}_{E} u = \underbrace{\begin{bmatrix} v_E \\ 0 \end{bmatrix}}_{\beta}.$$

This ensures that we satisfy Requirement 4. Requirement 5 follows from the inequality constraint defined above. Summarizing the above, the proposed control algorithm for the kinematic control u for a given user velocity command and equality and inequality constraints is given below:

Algorithm: Consider a mobile manipulator at a configuration q, with the user specified velocity v_H, equality constraint, and inequality constraint. Choose the robot motion input u as provided above, with x solved from the following quadratic programming problem:

$$\min_{\xi} \left\| J_H(E^{\perp}\xi + E^{+}\beta) - v_H \right\| + \lambda_1 \left\| (J_D^T(E^{\perp}\xi + E^{+}\beta) \right\| + \lambda_2 \left\| C(E^{\perp}\xi + E^{+}\beta) \right\|$$

subject to

$$\frac{\partial h_i}{\partial q} J_q(q)(E^{\perp}\xi + E^{+}\beta) \geq \sigma_i, \; i = 1, \ldots, n_i$$

Where C is given by the equation above, λ_1 and λ_2 are chosen by weighting factors,

$$J_D = \begin{bmatrix} J_H \\ E \\ C \end{bmatrix}^{\perp}.$$

And σ_i is given by equation above.

[0068] Note that the minimization problem is strictly convex as the matrix

$$\begin{bmatrix} J_H E^{\perp} \\ \lambda_1 C E^{\perp} \\ \lambda_2 J_D^T E^{\perp} \end{bmatrix}$$

is of full column rank. Therefore a unique solution always exists and may be efficiently found by using the gradient descent type of methods.

[0069] The following describes the equality and inequality constraints imposed on the system. Consider the two arms of the robot separately. Without loss of generality, consider only the left arm of the system. The kinematics as provided above given with $u \in R^9$ (7 for the arm and 2 for the motorized base), and $q \in R^{10}$. Focus only on user commanding the translation of the end-effector. Hence,

$$J_H = J_T^p$$

the translational portion of J_T. The choice of the inequality and equality constraints is next described below.

1) Arm Workspace: There are two types of workspace constraints: joint limits and end-effector boundary constraint. The ranges of the joint angles are specified as follows (inequalities are interpreted component-wise):

$$\begin{bmatrix} -\pi/4 \\ -\pi/3 \\ -2\pi/3 \\ q_{4min} \\ -\pi/2 \\ -\pi/2 \\ -\pi/2 \end{bmatrix} \leq q_L \leq \begin{bmatrix} \pi/2 \\ \pi/3 \\ -\pi/4 \\ 2\pi/3 \\ \pi/2 \\ \pi/2 \\ \pi/2 \end{bmatrix},$$

[0070] The values are provided by the manufacturer, except for q_{4min} which is to be specified based on the distance between the current position of the motorized base to the target. When the motorized base is far from the target, the arm should bend close to the body while the motorized base moves forward (to avoid bumping into nearby objects). When the motorized base is close to the target, the arm can stretch to reach to forward. Thus, let $q_{4min} = \pi/2$ (arm bent upward) when the distance between the

motorized base 12 and target is larger than a threshold value (e.g., 1 m), and let q_{4min} = 0 (arm outstretched) when the distance is smaller than that threshold. The target location may either be specified by the user as the robot gets close, or detected automatically by the onboard cameras. To avoid self collision, the coordinate of the end-effector in x-direction is specified,

$$e_x^T p_{OT}$$

to be larger than 0.3 m. This is particularly important when the motorized base is backing up. Combining the above, we have 15 inequality constraints:

$$h_1 = \begin{bmatrix} q_L - q_{L_{min}} \\ q_{L_{max}} - q_L \\ e_x^T p_{OT} - 0.3 \end{bmatrix}.$$

The corresponding Jacobian is

$$J_1 = \begin{bmatrix} I_7 & 0_{7\times 2} \\ -I_7 & 0_{7\times 2} \\ e_x^T J_{T_L}^p & 0_{7\times 2} \end{bmatrix}$$

where

$$J_{T_L}^p$$

is the translational portion of the left arm end-effector Jacobian, J_{TL}, I_7 is the 7 x 7 identity matrix. The parameters in the σ function are hand-tuned to be η = ε = 0.15 rad for the joint limit constraints, η = 0.1 m, ε =0.15 m for the self collision constraint, c is set to 0.45 in all cases. In all cases except for q_4, the constraints are always satisfied, so we set e_i to a small number 0.01. For q_4, the constraint region switches (due to q_{4min}), so the corresponding e_i is chosen larger to be 0.1.

2) End-effector Orientation Control: To maintain the end-effector orientation at a specified configuration, we use a proportional feedback control kinematic control law:

$$\omega_T = -K_T q_v(R_{OT}^{des^T} R_{OT}),$$

where

$$R_{OT}^{des} \text{ and } R_{OT}$$

are the desired and actual end-effector orientation matrix, K_T is the feedback gain and q_v is the vector quaternion. Thus, the first set of equality constraint is given with the corresponding Jacobian

$$J_E = \begin{bmatrix} J_{T_L}^R & 0_{3\times 2} \end{bmatrix}$$

where

$$J_{T_L}^R$$

is the rotational portion of the end-effector Jacobian J_{TL} .

3) Motorized base 12 Movement Constraint: As described, an additional equality constraint is imposed to ensure the base follows the user commanded motion when the arm reaches the workspace boundary. In the component form,

$$(-v_H^{(y)} p_{OT}^{(y)} - v_H^{(x)} p_{OT}^{(x)})\omega_B + v_H^{(y)} v_B = 0$$

with the corresponding constraint Jacobian given by:

$$J_{EB} = \begin{bmatrix} 0_{1\times 7} & -v_H^{(y)} p_{OT}^{(y)} - v_H^{(x)} p_{OT}^{(x)} & v_H^{(y)} \end{bmatrix}.$$

4) Obstacle Avoidance: Obstacles in the workspace may be detected by appropriate sensors, e.g., ultrasound range sensor, time-of-flight sensor, cameras, etc. In our current implementation, we use four IR sensors placed at the front side of the motorized base 12 to measure the distance of the obstacle ahead, d_i, i = 1,..., 4. Suppose the minimum distance threshold for the sensors are d_{imin} , i = 1,…, 4. Then the inequality constraints are

$$h_I = \begin{bmatrix} d_1 - d_{1\min} & d_2 - d_{2\min} & d_3 - d_{3\min} & d_4 - d_{4\min} \end{bmatrix}^T.$$

The cooresonding Jacobian is

$$J_I = \begin{bmatrix} 0_{4\times 8} & [-1,-1,-1,-1]^T \end{bmatrix}$$

since

$$\dot{d}_i = -v_B$$

(forward velocity decreases the distance to obstacle). In order the forward moving velocity to decrease when the motorized base 12 is close to the obstacle, a relatively small value of c and large value of ε is set in σ, e.g., c = 0.5, ε = 1.2 m, so that the speed reduction starts at 1.2 m away from the obstacle. The size of the buffer zone is set to η= 0.1 m, so the motorized base 12 starts to decelerate as far as 1.3 m from obstacle. Since one expects the inequality constraints always to be satisfied, e_i is chosen as a small number, 0.01.

FORCE

[0071] With the increasing availability of affordable and powerful sensors, actuators, processors, and communication devices, many robots are now capable to

operate in human-centric environments such as hospitals, offices and homes. These environments are typically "unstructured", where the robot does not have the a priori knowledge of the surrounding in advance, and the objects and obstacles in the environment may change. There are also many tasks involving physical contact between the robot end-effector and the environment, e.g., cleaning and polishing, door and drawer opening, etc. For such tasks, force control is required in the constrained direction, and motion control in the complementary direction. Hybrid force/position control to date mostly involves autonomous robot control with known environment geometry. When the environmental geometry is unknown, the constraint surface may be estimated from measurements, e.g., a combination of vision/force sensors, force/velocity sensors, etc. Numerous control algorithms have been proposed, including adaptive control, iterative learning control and neural network control. All these approaches aim to maintain a desired contact force along the (stiff) constrained direction and control the position/velocity along the complementary unconstrained direction.

The work mentioned above addresses autonomous robot force control. In the assistive robotics context, as in the present case, the human user provides certain level of input typically using an input device. As described, an optimization based redundancy resolution may be implemented to complement the user command for a mobile manipulator. This allows a disabled user to accomplish certain daily tasks, e.g., fetch and retrieve of items from different parts of the living space. The user commands the end-effector linear velocity using a sip-puff device, while the control algorithm determines the overall motion of the highly redundant system to achieve obstacle circumvention, and arm singularity and joint limit avoidance, with intuitive base motion.

The approach to contact tasks is extended to cases where the arm end-effector interacts with an environment. While the contact force is measurable, the exact geometry and stiffness of the environment constraint is generally not known a priori. As with the above discussions, the human user provides three degrees of freedom commands to the robot end-effector. In free space (non-contact) motion, the input is interpreted as a linear velocity command as before. When the contact occurs, generalized damper type of impedance control is used for the force control in the constrained direction (determined based on the measured motion). The user input in the constrained direction, e.g., pressing further into and backing away from the constraint, adjusts the contact force setpoint. The user input in the complementary unconstrained direction is used as a velocity command as in free space motion. There are two key features of the present approach that enhances the usability of such assistive robotic systems. First, the user can seamlessly switch between the free space mode and contact mode aided by a hysteresis function with the measured contact force magnitude as the input. Second, the user can either move the end-effector in the unconstrained direction while keeping the contact force constant, or change the desired contact force.

[0072] This is achieved through a "user intention function" that maps the three degrees of freedom input command to the two different cases. For example, an input device may be used to command the robot holding an eraser to approach a whiteboard, press the eraser into the whiteboard to ramp up the contact force, and then move the eraser to clean the board while maintaining the contact force.

[0073] Baxter is for example equipped with three cameras, two on each arm and one on the head which can be used to detect targets and obstacles. It also has joint

torque sensors based on which the end-effector force may be estimated, though, due to inexact joint gravity and spring load, the estimate is noisy and not very accurate.

[0074] In one illustrative embodiment, a 3-dof translational movement of the end-effector may be implemented. The end-effector orientation is fixed with respect to the motorized base. The user provides 3-dof input command v_H to the system, e.g., from a joystick or Jamboxx. The actual control input u is:

$$[\dot{q}_1 \cdots, \dot{q}_7]^T$$

for single-arm (left-arm) manipulation or

$$[\dot{q}_1 \cdots, \dot{q}_7, \dot{q}_1 \cdots, \dot{q}_7]^T$$

for dual-arm manipulator. To allow the user freely and comfortably manipulate the end-effector in free space or in contact with the environment, control input u is designed such that:

1. The user should be able to easily control the end-effector both in free space movement and in contact with environment constraint

2. When the end-effector is in contact with environment, the user should be able to either move the end-effector in unconstrained direction while keeping the contact force constant, or change the desired contact force.

[0075] When the human user operates the robot end-effector, it may either move in free space or in contact with environment. The goals for free space control and contact control are different: the former is to regulate the 3-dof end-effector motion only, and the latter is to regulate both the contact force in the constrained dimension and the motion in the complementary unconstrained directions.

[0076] One approach is illustrated in Figure 9. At a given time instant, if the system is in the free space mode, the corresponding free space motion controller will be

implemented, and the switching condition 1 is tested to determine if a transition to the contact mode is necessary. If the answer is yes, the desired contact force is reset to the initial value and the system goes to the contact mode. If the answer is no, then the system will remain in the free space mode. Similarly, if the system is in the contact mode, the corresponding contact controller will be implemented, and the switching condition 2 is tested to determine if the transition to free space mode should take place. As shown in the block diagram of Figure 9, the control system consists of five modules:

1. Free space motion controller
2. Switching condition 1
3. Desired contact force reset
4. Contact controller
5. Switching condition 2

A. Free Space Motion Controller

[0077] As described herein, a human-directed control law is proposed for the free space motion of the end-effector of a mobile manipulator. The user provides the desired end-effector velocity command $v_H \in R^3$. The kinematic control input u is generated by solving a constrained optimization algorithm in each control instant that minimizes the difference between v_H and the actual end-effector velocity v_T while satisfying certain equality and inequality constraints for the purpose of obstacle and singularity avoidance as well as minimizing the base motion. The optimization algorithm is formulated as follows:

$$\min_{u} \{ \| v_H - J_H u \| + \lambda \| v_S - J_S u \| \}$$
$$\text{subject to}$$
$$J_E u = v_E$$
$$J_I u \geq \sigma,$$

where

$$\| v_H - J_H u \|$$

is the difference between the user commanded velocity and actual end-effector velocity, with J_H being the corresponding Jacobian matrix. The second term in the optimization,

$$\lambda \| v_S - J_S u \|$$

is the additional soft constraint term to ensure strictly convexity and to minimize the base motion when it is not necessary. The inequality constraints are given by $J_E u = v_E$, and the inequality constraint is captured by the componentwise inequality $J_I u \geq \sigma$. The same constrained optimization problem formulation as noted above is used. However, since base motion is not considered, the soft constraint matrix J_S only contains the null space vectors of

$$\begin{bmatrix} J_H \\ J_E \end{bmatrix}$$

[0078] The constraints related to the base motion are no longer needed. This problem is a simple quadratic programming problem and may be easily solved by using commercial software packages such as MATLAB to generate the kinematic control law u.

B. Switching condition 1

[0079] To determine the transition to the contact mode, the magnitude of the measured end-effector force is used. Due to the acceleration and deceleration of the system as well as noises and bias of the signal, the measured end-effector force may be very different from the actual value. Indeed, even when the end-effector is not in contact with any environment, the measured end-effector external force may still have non-negligible magnitude. To address this issue, a deadzone $F_h > 0$ is added for the measured end-effector force, i.e., the switching condition is given by

$$\|F_T\| > F_h.$$

[0080] A large deadzone value means greater robustness, i.e., the transition is not likely to be affected by inaccuracies in the end-effector force measurement. However, it also reduces the sensitivity of contact, which means that the contact force needs to be high enough for the system to be treated as in the contact mode. In practice, the value of F_h is set based on the inaccuracy level of the force measurement and the required sensitivity level of the contact.

C. Desired Contact Force Resetting

[0081] In the contact mode, the end-effector contact force is controlled to track its desired value. The rate of change of the desired contact force may be controlled by the user. When the system enters into the contact mode, the initial desired contact force is reset to $F_{Td} = F_h$, so that it matches with the value of measured F_T the moment the system enters into the contact mode. Note that the desired contact force is reset every time the system enters into the contact mode. After that, the changing rate of the desired contact force will be regulated by the user input command as shown in the next subsection.

D. Contact Controller

[0082] The main objective of contact controller design is twofold:

1. when the user decides to move the end-effector in unconstrained direction, the contact force should be kept constant around its desired value;

2. when the user decides to change the desired contact force instead of moving the end-effector in unconstrained direction, the rate of change of the desired force can be directly regulated by the user.

[0083] The key issue is to determine the user's intention between the two cases based on the 3-dof input from the user. Define the two cases of user intention as follows:

Case 1: Move the end-effector along unconstrained direction while keeping the contact force around its desired value.

Case 2: Changing the desired contact force value.

A mapping function f is defined from human input vH to the user intention:

$$f(v_H) = \begin{cases} Case\ 1, & \text{if } v_H \in \Omega_1 \\ Case\ 2, & \text{if } v_H \in \Omega_2, \end{cases}$$

where Ω_1 and Ω_2 are two non-intersecting complementary sets in the 3D Euclidean space, with Ω_2 defined as follows:

$$\Omega_2 \triangleq \left\{ v_H \middle| \frac{F_T' v_H}{\|F_T\|} > \varepsilon_1 v_{H max}, \frac{\|F_T' v_H\|}{\|F_T\| \|v_H\|} > \varepsilon_2 \right\} \cup \left\{ v_H \middle| \frac{F_T' v_H}{\|F_T\|} < -\varepsilon_1 v_{H max}, \frac{\|F_T' v_H\|}{\|F_T\| \|v_H\|} > \varepsilon_2 \right\},$$

where ε_1 and ε_2 are two positive constants between 0 and 1.

The set Ω_1 is the complement of Ω_2.

[0084] The region Ω_2 is defined within a small cone of the direction of F_T with big enough magnitude. Such specification is intuitive to the user since when the user pushes the end-effector along the direction of contact force, the intention is likely to change the desired contact force. If the command is to push into the constrained direction, the intent is interpreted as to increase the desired contact force. If the command is to pull away, then the intent is interpreted as to decrease the contact force, and eventually break off the contact. If the user pushes the end-effector in the complementary direction of the constrained direction, the desired force will be kept constant and only the unconstrained direction of the human input command takes effect.

[0085] This user intention mapping function allows the user to make different decisions for contact control by only providing 3-dof input command v_H without the need of using additional inputs.

[0086] The controller in the two modes of operation is given by:

Case 1 (Motion/Force Control): The joint velocity input u is given by the solution of the following constrained minimization problem:

$$\min_u \{||v_H - J_H u|| + \lambda ||v_S - J_S u||\}$$

subject to

$$J_E u = v_E, \quad J_I u \geq \sigma,$$

$$F_T^T J_H u / ||F_T|| = k(||F_T|| - F_{Td}).$$

The force setpoint remains unchanged:

$$\dot{F}_{Td} = 0.$$

Case 2 (Force Setpoint Adjustment): The joint velocity input u only focuses on satisfying the constraints by solving a modified constrained minimization problem:

$$\min_u \{||v_S - J_S u||\}$$
$$\text{subject to}$$
$$J_E u = v_E, \quad J_I u \geq \sigma,$$
$$F_T^T J_H u / ||F_T|| = k(||F_T|| - F_{Td}), \quad F_T \times J_H u = 0.$$

The force setpoint adjustment is given by:

$$\dot{F}_{Td} = \begin{cases} v_{FTd} \text{sgn}(-F_T^T v_H) & \text{if } F_{Td} < F_{Tdmax} \text{ or } F_T^T v_H \leq 0 \\ 0 & \text{otherwise.} \end{cases}$$

In Case 2, v_{FTd} is the rate of change of the desired contact force, F_{Tdmax} is the upper limit of desired contact force to prevent excessive contact force. The normal vector is estimated into the surface from the contact force as:

$$n = -F_T / ||F_T||.$$

Note that F_T is the force applied to the end-effector, so $-F_T$ is the force applied to the environment (hence the sign change in n). The force control law is based on generalized damper along the surface normal:

$$n^T J_H u = -k(-n^T F_T - F_{Td})$$

where k is the reciprocal of the specified damping and F_{Td} is the desired force setpoint. This equation then reduces to the specified force control law above. Note that to enhance robustness, F_{Td} may be scheduled based on F_T.

E. Switching Condition 2

[0087] To determine if the controller should transition to the free space mode, the contact force condition is again used:

$$\|F_T\| < F_l.$$

Note that F_l needs to be smaller than F_h to avoid chattering between the two modes.

ILLUSTRATIVE IMPLEMENATIONS

[0088] Figure 10 depicts a generalized flow diagram showing a process for controlling the coordinated movement of the motorized base 12 and dual arm robot 16 of the robotic assistant 10 using a single input device. At S1, the operator uses the input controller (e.g., Jamboxx) to select a frame of the robotic assistant to control. In one typical example, the operator would select an end-effector, which is located proximate the grippers at the end of an arm. However, the operator may select other points, e.g., a wrist, shoulder, etc. Next at S2, the operator uses the input controller to command the selected frame to move within a three dimensional space. For example, the operator can use the Jamboxx to command the end-effector to move towards an object on a table.

[0089] The process next automatically determines at S3 if the selected arm is within the workspace, i.e., can the end-effector reach a desired point without the need to move the motorized base. If no, the motorized base is moved within the workspace at S3, without preferably moving the arm. Once the arm is in the workspace, a determination is made at S5 whether there are any constraints. For example, is there an object blocking the path, etc. If no, then the arm is independently moved as commanded by operator at S6. If yes, then the arm movement is limited or altered based on the constraint.

[0090] Figure 11 depicts an illustrative flow diagram showing a process for manipulating an object, e.g., a tray, a broom handle, etc., using both arms of the dual arm robot. At S11, the user commands a first robot arm to grasp an object, by for example manipulating the associated end-effector. At S12, the user switches control to the end-effector of the second arm and commands the second robot arm to also grasp the object. At S13, the user switches control, e.g., from the second end-effector to a frame (e.g., midpoint between the two end-effectors). Next, at S14 the user selects either position mode S15 or orientation mode S18.

[0091] In position mode S15, the system automatically locks or fixes the orientation of the frame (e.g., midpoint) at S16. The user can then use the input device to "drag" the selected frame (midpoint) in 3D space to move the object while maintaining the orientation of the object. Using this approach, the user could for example command the robotic assistant to pick up a tray with both arms, and move the tray to another location while maintaining the orientation of the tray.

[0092] In orientation mode S18, the system automatically locks or fixes the position of the frame (e.g., midpoint) at S19. The user can then use the input device to "re-orient" the selected frame (midpoint) in 3D space to move (e.g., rotate) the object while maintaining the relative position of the object. Using this approach, the user could for example command the robot to pick up a broom handle with both arms and rotate the broom handle while maintaining a relative position of the broom handle.

[0093] In another embodiment, a mixed mode of operation may be implemented, where for example an object (e.g., a bottle) is detected within the frame of a camera. The user can command the robot to move toward the object. In this case, the object image remains in the center of the camera frame, e.g., by fixing a vector between the camera and object, and allowing the user to select an approach velocity.

[0094] Figure 12 depicts an illustrative flow diagram showing a process for controlling a force to perform some action, e.g., erase a white board, open a door, etc. At S21, the user selects a frame (e.g., end-effector) and controls/moves the frame in free space. Next, at S22, the user causes the selected frame to initiate contact with a foreign object and sense a resistance or force. For example, a gripper may be holding a whiteboard eraser, which is place on a whiteboard with a certain amount of force. Alternatively, the gripper may grasp a door handle and exert a certain amount of force directed at the hinge. At S23 the desired amount of force is maintained (i.e., fixed) in the desired direction, referred to as a "force set point." At S24, the user moves the selected frame while maintaining the force set point. Thus, in the above example, the user can maintain the whiteboard eraser with a certain amount of force against the whiteboard and move it up/down and left/right. Similarly, the user can maintain a certain amount of radial force between the door handle and hinge and for example pull backward to open the door.

[0095] As will be appreciated, various aspects of the invention may be implemented as a computer program product stored on one or more computer readable storage mediums. The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static

random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals *per se*, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.

[0096] Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[0097] Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object

oriented programming language such as Java, Python, Smalltalk, C++ or the like, and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

[0098] Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[0099] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable

data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[00100] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

[00101] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that

each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[00102] An illustrative computing system may comprise any type of computing device and, and for example includes at least one processor (e.g., CPU 54 of Figure 7), memory 52, an input/output (I/O) 56 (e.g., one or more I/O interfaces and/or devices), and a communications pathway. In general, processor(s) execute program code which is at least partially fixed in memory 52. While executing program code, processor(s) can process data, which can result in reading and/or writing transformed data from/to memory and/or I/O for further processing. The pathway provides a communications link between each of the components in computing system. I/O 56 can comprise one or more human I/O devices, which enable a user to interact with the computing system.

[00103] Furthermore, it is understood that the described algorithms or relevant components thereof (such as an API component) may also be automatically or semi-automatically deployed into a computer system by sending the components to a central server or a group of central servers. The components are then downloaded into a target computer that will execute the components. The components are then either detached to a directory or loaded into a directory that executes a program that detaches the components into a directory. Another alternative is to send the components directly to a directory on a client computer hard drive. When there are proxy servers, the process will, select the proxy server code, determine on which computers to place the proxy servers' code, transmit the proxy server code, then install

the proxy server code on the proxy computer. The components will be transmitted to the proxy server and then it will be stored on the proxy server.

[00104] The foregoing description of various aspects of the invention has been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed, and obviously, many modifications and variations are possible. Such modifications and variations that may be apparent to an individual in the art are included within the scope of the invention as defined by the accompanying claims.

CLAIMS

What is claimed is:

1. A robotic assistant, comprising:

a motorized base having at least two motor driven wheels controlled by a first control platform;

a dual arm robot mounted on the motorized base, the dual arm robot having a first arm and a second arm controlled by a second control platform;

a remote sip and puff mouth controller having at least three degrees of operational input; and

a computer system that receives command signals from the remote sip and puff mouth controller, and includes a control system that coordinates movement of the motorized base and the robot arms by translating the command signals into a first type of control signal for directing the motorized base to move, and into a second type of control signal for directing the dual arm robot to move.

2. The robotic assistant of claim 1, wherein the coordinated movement is determined based on user manipulation of a selected frame.

3. The robotic assistant of claim 1, wherein the motorized base comprises a retrofitted power wheelchair that includes a horizontally oriented adapter plate mounted thereon that is adapted to secure the dual arm robot.

4. The robotic assistant of claim 1, wherein the dual arm robot includes at least 6 degrees of operational freedom for each arm.

5. The robotic assistant of claim 1, wherein the command signals from the remote sip and puff mouth controller are communicated wirelessly.

6. The robotic assistant of claim 1, wherein the computer system further includes an input system for receiving command signals from a second source via the Internet.

7. The robotic assistant of claim 1, wherein control system includes an algorithm that instructs the motorized base to move only when both the first arm and the second arm are outside a defined workspace or when the user directly commands the motorized base to move.

8. A control platform for controlling the operation of a robotic assistant, comprising:

a control system that controls movement of a motorized based and a dual arm robot mounted on the motorized base, wherein the control system implements a first type of control signal to control the motorized base, and a second type of control signal to control the dual arm robot;

an input for receiving command signals from a remotely located controller, wherein the command signals include a proposed movement of a selected frame in a three dimensional space; and

a software algorithm that coordinates the motion of the both the motorized base and dual arm robot based on the proposed movement of the selected frame.

9. The control platform of claim 8, wherein the selected frame includes one of an end-effector associated with a selected arm of the dual arm robot, a specified location on a selected arm and a point fixed with respect to both arms.

10. The control platform of claim 8, wherein the software algorithm limits movement of the dual arm robot based on existing constraints.

11. The control platform of claim 8, wherein the remotely located controller includes a device selected from a group consisting of: a sip and puff device, a tongue command device, an eye/blink tracker, and a joystick.

12. The control platform of claim 8, further comprising a second input for receiving remote commands via the Internet.

13. The control platform of claim 8, wherein the control system further separately communicates with at least one sensor.

14. The control platform of claim 8, wherein each arm of the dual arm robot includes at least six degrees of robot arm movement, and wherein the software algorithm maps three degrees of operational input from the remotely located controller to the robot arm movement.

15. The control platform of claim 8, wherein the proposed movement includes at least one of a velocity or a force at the select frame.

16. A method for controlling arms of a robotic assistant having a motorized based and a dual arm robot mounted on the motorized base, comprising:

receiving user commands from a remote input device to cause a gripper of a first arm to grasp an object;

receiving user commands from the remote input device to cause a gripper of a second arm to also grasp the object;

selecting a frame relative to the first arm and second arm;

selecting between a position mode and an orientation mode;

in response to selecting position mode, maintaining an orientation of the frame while allowing a user to move the point via the remote input device to effectuate movement of the object; and

in response to selecting orientation mode, maintaining a position of the frame while allowing the user to reorient the frame via the remote input device to effectuate movement of the object.

17. The method of claim 16, wherein the remote input device provides three degrees of operational freedom.

18. The method of claim 16, wherein the frame relative to the first arm and second arm is located at a midpoint between end-effectors of the two arms.

19. The method of claim 16, wherein reorienting the frame comprises rotating the frame to effectuate a rotation of the object about a point.

20. A method for controlling a robotic assistant having a motorized based and a dual arm robot mounted on the motorized base, comprising:

receiving user commands from a remote input device to move an arm of the dual arm robot;

causing a contact that results in a resistance being detected by a sensor on the arm;

establishing a force set point by causing a desired amount of force to be exacted in a determined direction; and

receiving user commands to effectuate a movement of the arm while maintaining the force set point.

21. The method of claim 20, wherein the remote input device provides three degrees of operational freedom.

22. The method of claim 20, wherein the force set point causes an object being held by a gripper associated with the arm to be pressed against another object.

23. The method of claim 20, wherein the force set point causes a gripper door handle holding a door handle to maintain a constant rotational force with respect to a hinge.

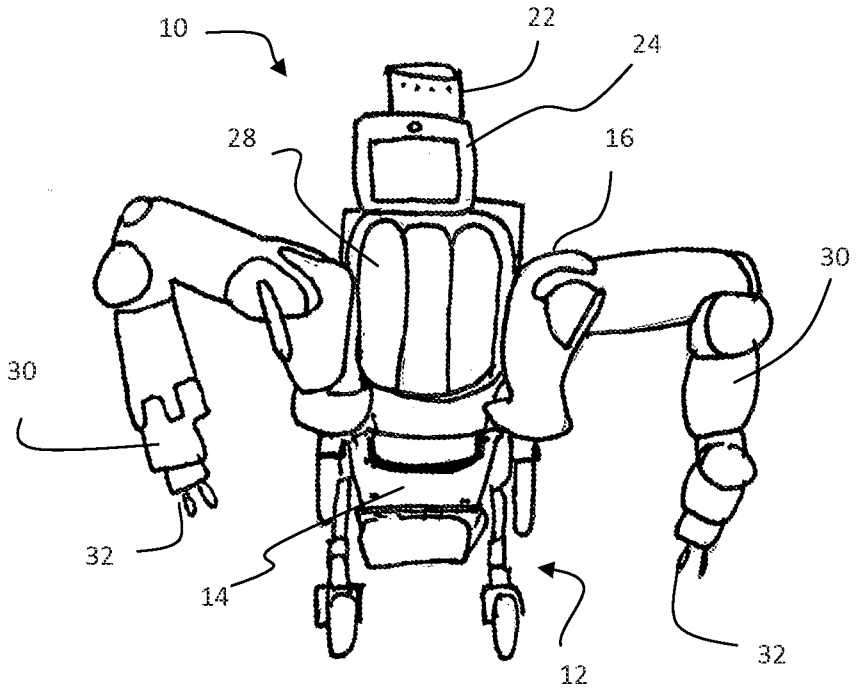

FIG. 1

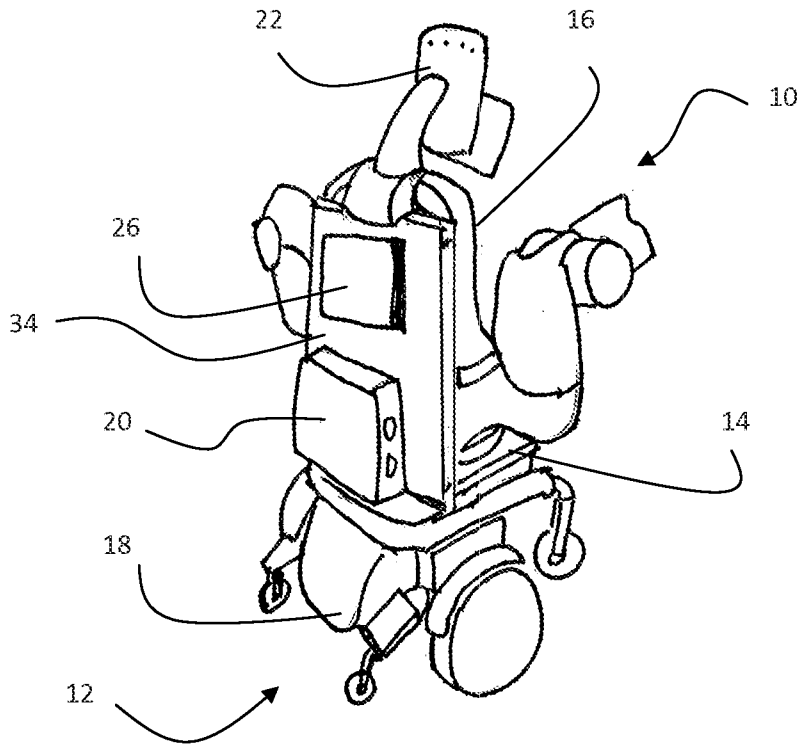

FIG. 2

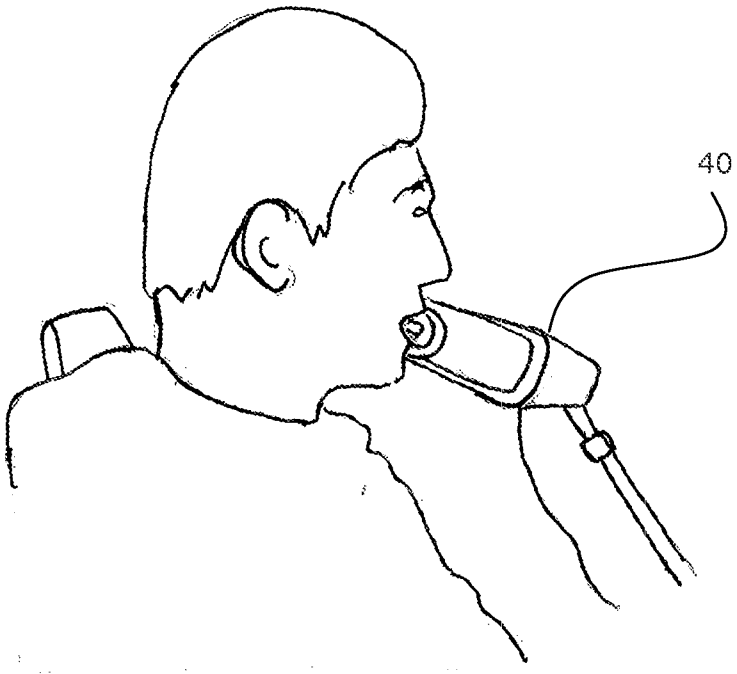

FIG. 3

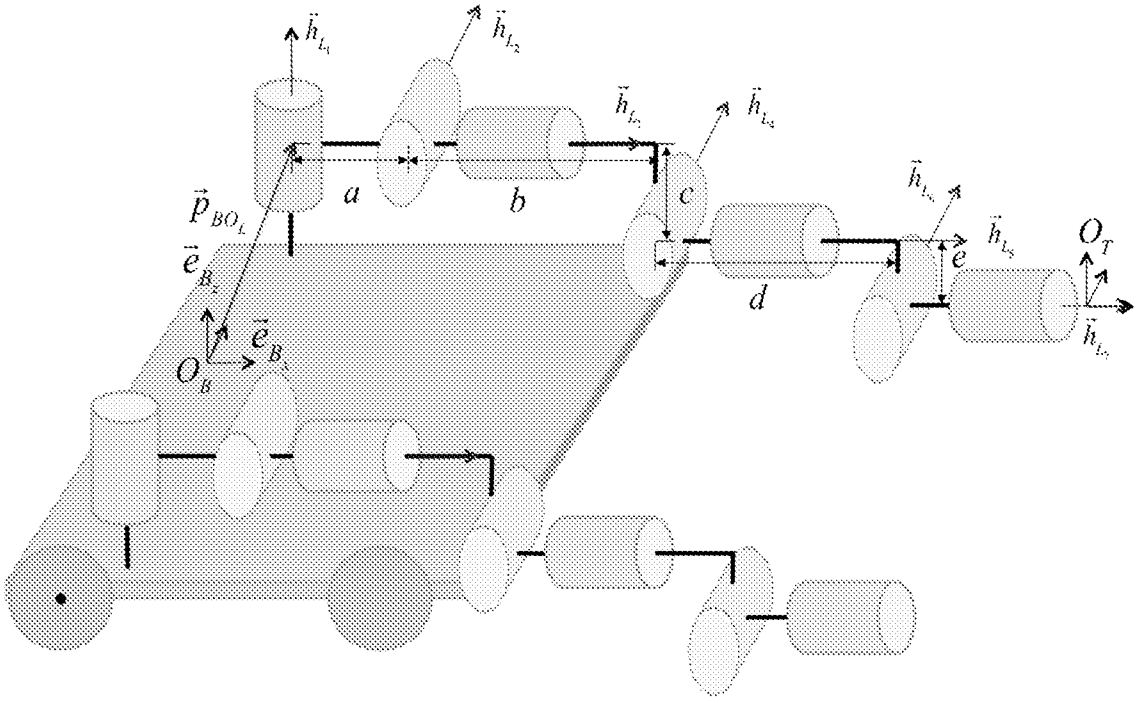

FIG. 4

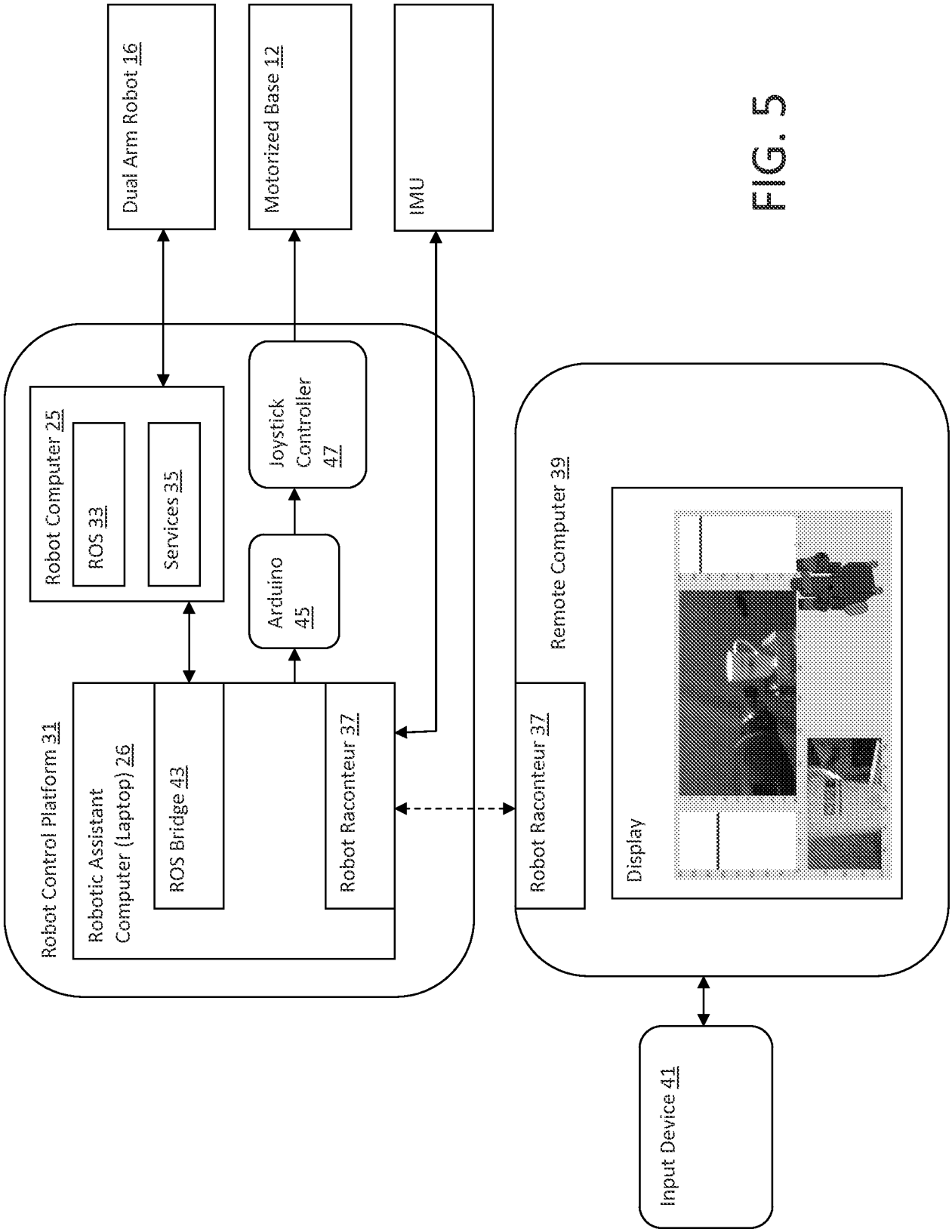
Dual Arm Robot 16
Motorized Base 12
IMU
Robot Computer 25
ROS 33
Services 35
Joystick Controller 47
Arduino 45
Robot Control Platform 31
Robotic Assistant Computer (Laptop) 26
ROS Bridge 43
Robot Raconteur 37
Robot Raconteur 37
Remote Computer 39
Display
Input Device 41

FIG. 5

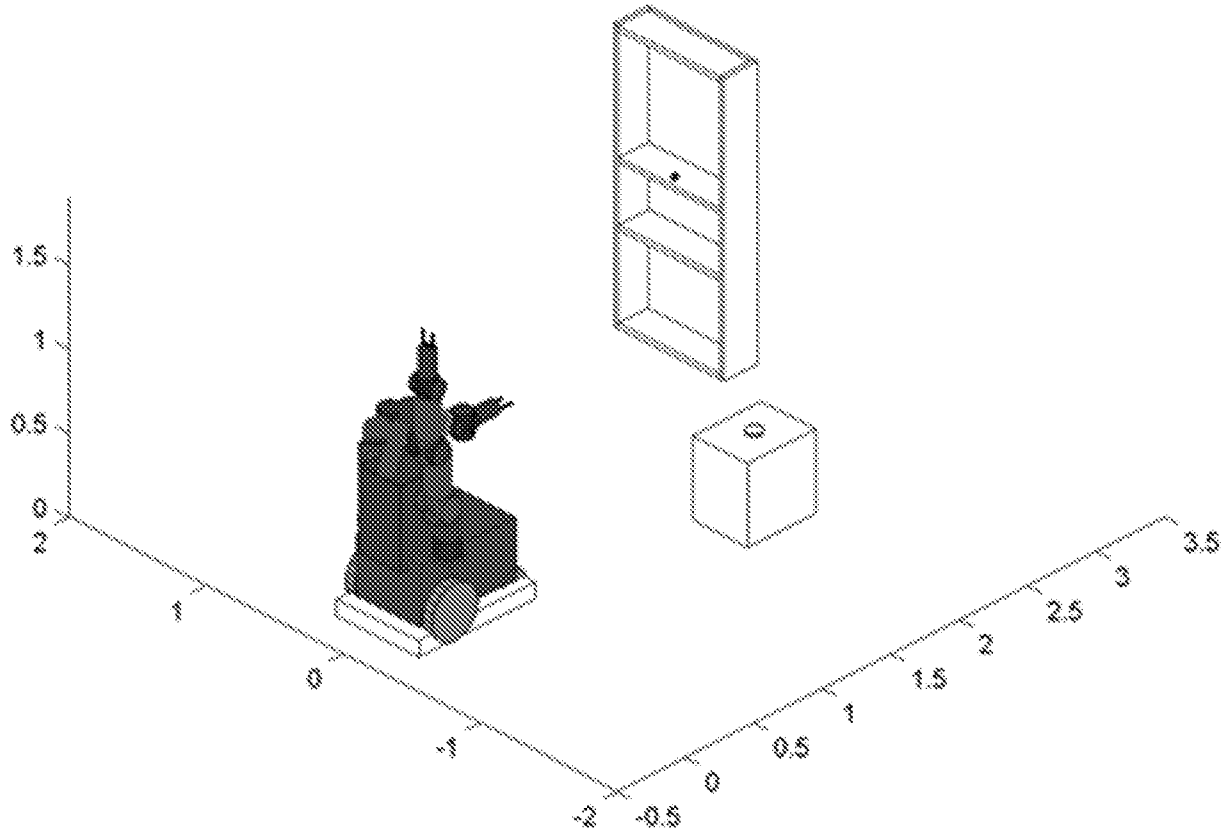

FIG. 6

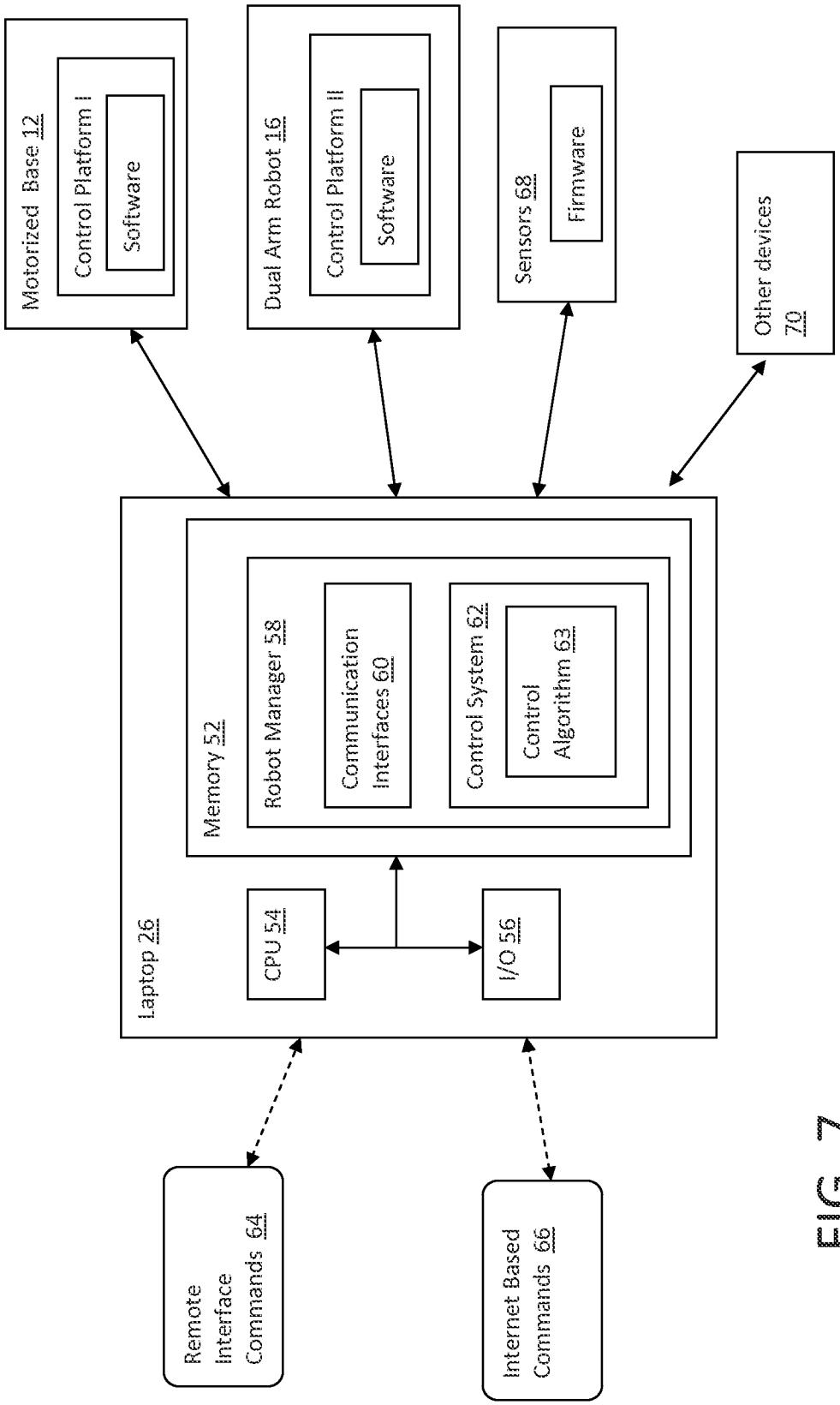
Motorized Base 12
Control Platform I
Software
Dual Arm Robot 16
Control Platform II
Software
Sensors 68
Firmware
Other devices 70
Laptop 26
Memory 52
Robot Manager 58
Communication Interfaces 60
Control System 62
Control Algorithm 63
CPU 54
I/O 56
Remote Interface Commands 64
Internet Based Commands 66

FIG. 7

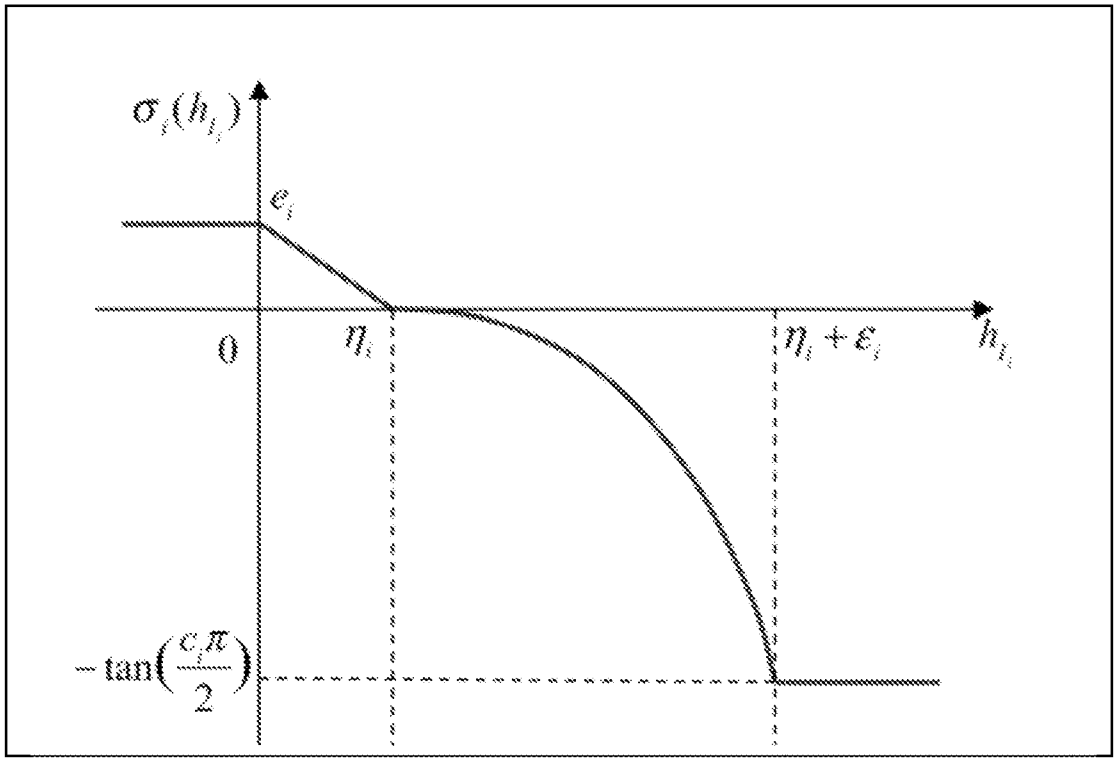
$\sigma_i(h_i)$
ε_i
0
η_i
$\eta_i + \varepsilon_i$
h_i
$-\tan\left(\frac{c_i\pi}{2}\right)$

FIG. 8

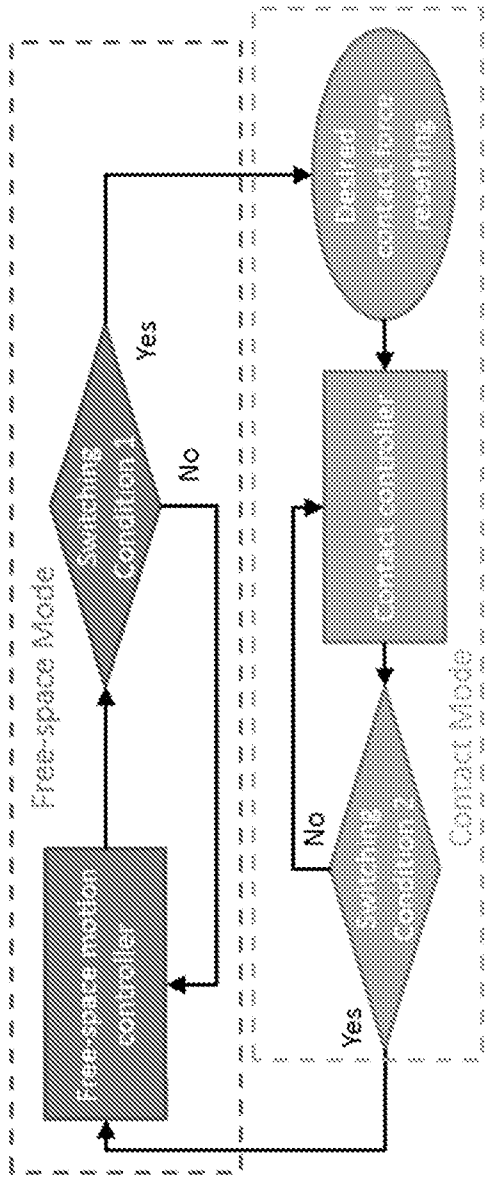
Free-space Mode
Free-space motion controller
Switching condition 1
Yes
No
Desired contact force tracking
Contact controller
Contact switching condition 1
No
Yes
Contact Mode

FIG. 9

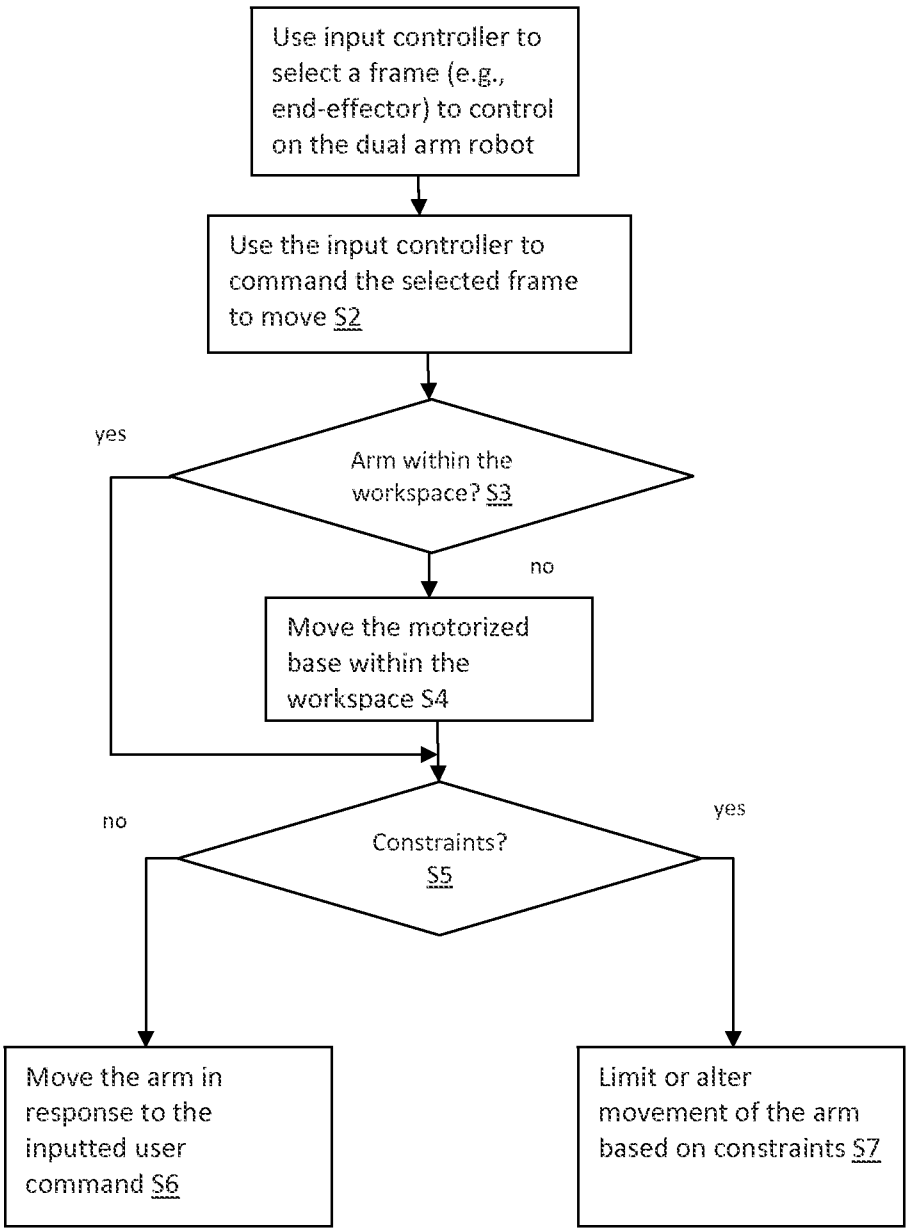
Use input controller to select a frame (e.g., end-effector) to control on the dual arm robot
Use the input controller to command the selected frame to move S2
yes
Arm within the workspace? S3
no
Move the motorized base within the workspace S4
no
Constraints? S5
yes
Move the arm in response to the inputted user command S6
Limit or alter movement of the arm based on constraints S7

FIG. 10

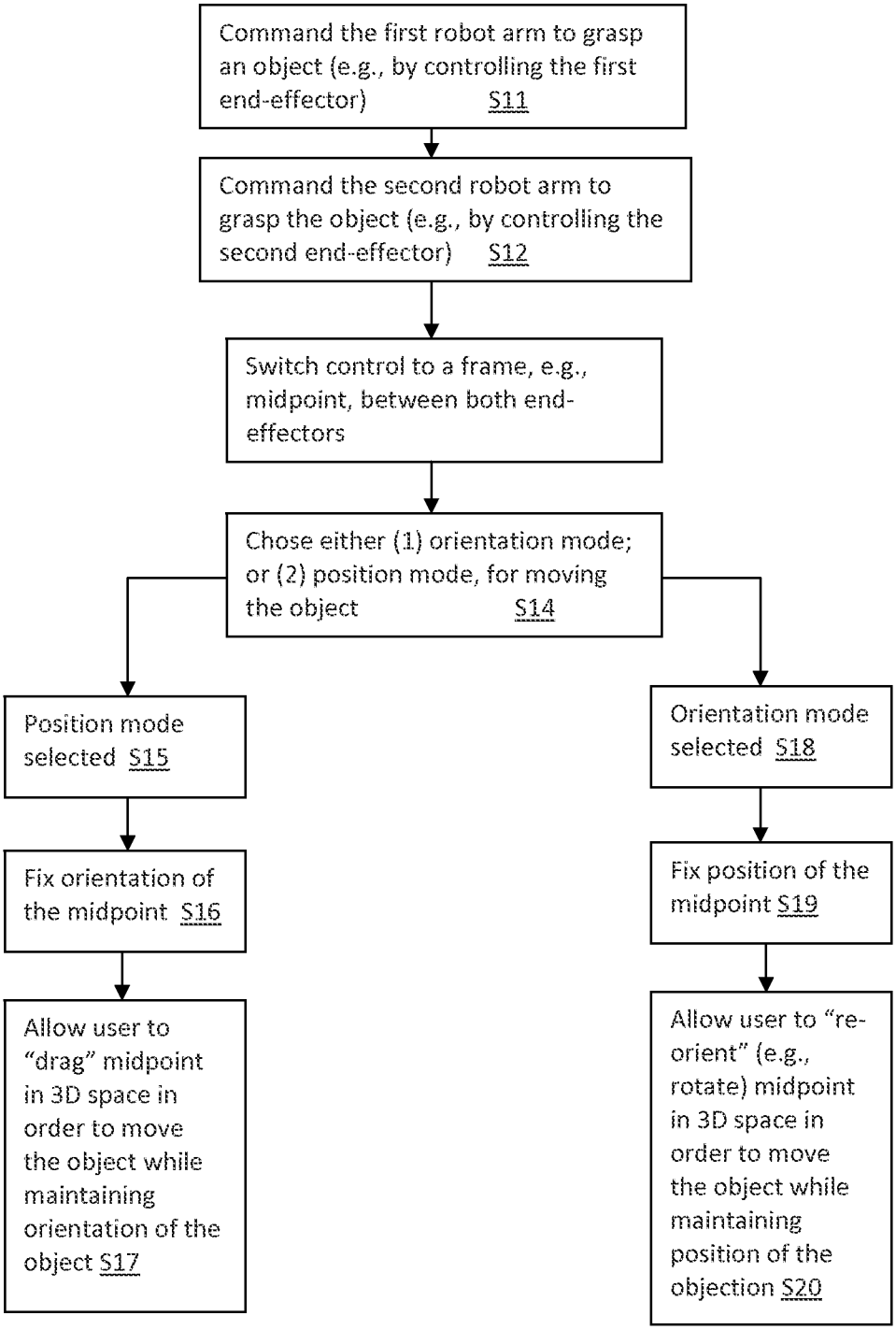
Command the first robot arm to grasp an object (e.g., by controlling the first end-effector) S11
Command the second robot arm to grasp the object (e.g., by controlling the second end-effector) S12
Switch control to a frame, e.g., midpoint, between both end-effectors
Chose either (1) orientation mode; or (2) position mode, for moving the object S14
Position mode selected S15
Fix orientation of the midpoint S16
Allow user to "drag" midpoint in 3D space in order to move the object while maintaining orientation of the object S17
Orientation mode selected S18
Fix position of the midpoint S19
Allow user to "re-orient" (e.g., rotate) midpoint in 3D space in order to move the object while maintaining position of the objection S20

FIG. 11

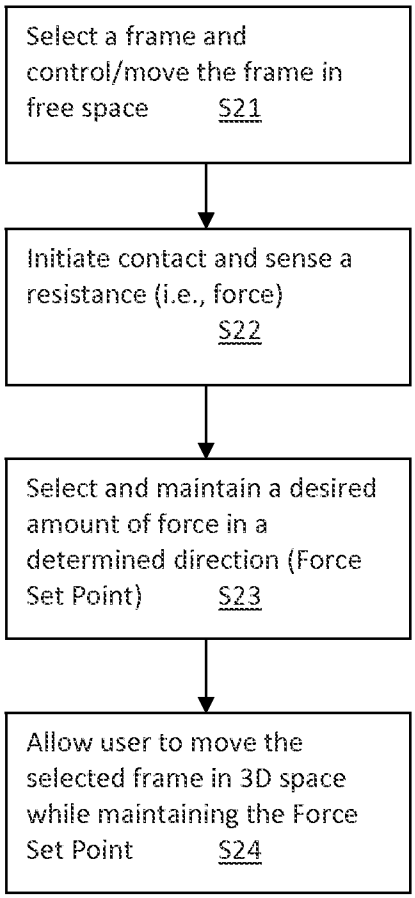
Select a frame and control/move the frame in free space S21
Initiate contact and sense a resistance (i.e., force) S22
Select and maintain a desired amount of force in a determined direction (Force Set Point) S23
Allow user to move the selected frame in 3D space while maintaining the Force Set Point S24

FIG. 12