

Software Compensated Robotics

CROSS-REFERENCE TO RELATED APPLICATIONS

[001] This application claims priority to U.S. provisional patent application Ser. No. 62/854,071 filed May 29, 2019 and is a Continuation-in-part of U.S. non-provisional patent application Ser. No. 16/237,721 filed January 1, 2019, the disclosures of which are hereby incorporated herein by reference.

BACKGROUND

[002] *Field of the invention*

[003] The invention is in the field of robotics, and in some embodiments the field of vision-controlled robotics.

[004] *Related art*

[005] Control of a robot typically involves sending an electronic signal and activating an actuator based on the electronic signal. The actuator can include a DC motor, hydraulic device, synthetic muscle, pneumatic device, piezoelectric, a linear or rotational actuator, or other movement generation device. The generated movement may be scaled up or down using a gear box or lever, and then used to move a part of the robot. The amount of movement is optionally detected using an encoder. The encoder and other components are optionally embodied in a servo motor or other actuator. A robot having multiple degrees of freedom, e.g., 6-degrees, typically require at least one movement generation device for each degree of freedom.

[006] Reaching a desired “pose” for a robot requires specification of both a location (x, y, z) and a set of angular values (α , β , γ). Reaching a desired pose depends on knowing an existing pose of the robot and applying motion to six movement generation devices to move from the current pose to a desired pose. Such movement is typically achieved by using a target pose and a model of the robot to calculate a movement needed in each degree of freedom. The precision and accuracy of reaching the desired pose is dependent on inverse kinematics, which requires knowledge of the initial pose

and accuracy and precision of the movement. Achieving high precision and accuracy can require expensive components, particularly when heavy loads are involved. Requirements for precision and accuracy also preclude, in many applications, use of some types of movement generation devices which may change over time, such as tendon mechanisms. Finally, in many applications, use of some types of materials are precluded for use in robotics for similar reasons.

SUMMARY

[007] Vision based robot control includes a real-time feedback loop which compensates for variations in actuator response and/or models of the robot using data collected from cameras and other input devices. Images of actual robot movement in response to control signals are used to determine future control signals need to achieve desired robot movements. A computer vision software pipeline, which may be implemented as a multi-stage neural network, is configured to process received images and to generate control signals for reaching a desired movement goal of the robot. When implemented using a neural network, such a network may include at least one neural network block having a stored state that allows for dynamic temporal behavior. Specifically, such a neural network is configured such that images are the primary input used to control movement of the robot toward a specified goal, though other inputs, such as from servo encoders, potentiometers, contact sensors, and/or force sensors may also be included. Together, these inputs are used to detect responses of the robot to a prior set of control signals. The stored state of the neural network enables the incorporation of past responses in the prediction of future responses.

[008] Various embodiments of the invention include a robotic system comprising: a movement generation device; a transmission coupled to the movement generation device and to a robotic manipulator, the transmission being configured to move the robotic manipulator in response to the movement generation device; an end effector attached to the robotic manipulator, a pose of the end effector being dependent on movement of the robotic manipulator; a camera configured to generate an image of the end effector; a multi-stage neural network including: a perception block configured to receive the image and generate an image processing output representative of a state

of an object within the image, a policy block configured to generate command signals for movement of the end effector, the generated command signals being based on at least i) a goal for the end effector, ii) the image processing output and optionally iii) a time dependent internal state of the policy block, and a compensation block configured to provide an output for control of the movement generation device based both the command signals and the image processing output; and control logic configured to provide the goal for the end effector to the policy block, or to select the policy block based on the goal for the end effector.

[009] Various embodiments of the invention include a method of controlling a robot, the method comprising: capturing an image using a camera, the image optionally including an end effector connected to a robotic manipulator; processing the captured image to produce a representation of objects within the image, as well as a state of the robot itself; applying a policy to the representation of objects to produce command signals, the production of command signals being based on at least a goal and the representation of objects; compensating for a change in response of the robotic manipulator to command signals, to produce compensated control signals, the compensation being based on prior command signals and the representation of objects; and activating the robot using the compensated control signals.

[0010] Various embodiments of the invention include a method of calibrating a robot, the method comprising: generating control signals; providing the control signals to a robot, the control signals optionally being configured to generate an expected movement of an end effector attached to the robot; capturing an image showing a response of the robot to the control signals; generating second control signals; changing a state of the neural network responsive to the image and the expected movement; and generating second control signals; compensating the second control signals to produce compensated control signals using the neural network, the compensation being responsive to the changed state of the neural network, the compensation being configured to reduce a difference between the expected movement and a movement of the end effector indicated by the image.

[0011] Various embodiments of the invention include a robotic system comprising: an end effector comprising: a digit having at least three segments separated by at least first and second joints, the three segments including a proximal segment, a medial segment and a distal segment, the proximal segment being attached to a robotic manipulator via a third joint, a first transmission configured to flex the third joint, a second transmission configured to flex both the first and second joints, wherein the relative angles of the first and second joints are dependent on contact between an object and the medial segment or between the object and the distal segment, and a first elastic element configured to extend the first joint; one or more movement generation devices configured to move the first and second transmission independently; and a camera configured to generate an image of the end effector; a neural network configured to provide movement command signals to the movement generation device, the movement command signals being compensated for variations in relative movements of the first and second joints, the compensation being based on the image.

[0012] Various embodiments of the invention include a method of controlling a multi-joint robotic end effector, the method comprising: moving, e.g., pulling, a first transmission to flex a first joint; capturing an image of a digit of the end effector, the first joint separating the digit of the end effector from a robotic manipulator, the digit including at least two or three segments separated by at least second and third joints, the three segments including a proximal segment, a medial segment and a distal segment, the proximal segment being attached to a robotic manipulator by the first joint; generating command signals configured to move the distal segment to a desired location; compensating the generated command signals for variation in movement of the distal segment in response to moving of the first transmission, the compensation being based on processing of the image using a neural network; and moving a second transmission to flex both the second and the third joints, flexed angles of the second and third joints being dependent on contact between an object and the medial segment.

[0013] Various embodiments of the invention include a robotic system comprising: an end effector; a robotic manipulator configured to support the end effector; one or more movement generation

devices configured to move the end effector in response to movement command signals; a camera configured to generate an image of the end effector; a memory storage configured to store long-term memory data; and a neural network configured to provide the movement command signals to the movement generation device, the movement command signals being compensated for non-deterministic variations in movements of the end effector, the compensation being based on the image, wherein generation of the command signals is based on the long-term memory data and compensation of the generated command signals is based on a short-term memory in the neural network.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 illustrates a robotic system, according to various embodiments of the invention.

[0015] FIG. 2 illustrates a robot, according to various embodiments of the invention.

[0016] FIG. 3 illustrates a neural network at different times, according to various embodiments of the invention.

[0017] FIG. 4 illustrates a neural network including one or more multiplex layer, according to various embodiments of the invention.

[0018] FIG. 5 illustrates methods of controlling a robot, according to various embodiments of the invention.

[0019] FIG. 6 illustrates an end effector, according to various embodiments of the invention.

[0020] FIG. 7 illustrates method of controlling a robotic joint, according to various embodiments of the invention.

DETAILED DESCRIPTION

[0021] As used herein, a “movement generation device” is a device that causes movement or force.

For example, a movement generation device can include a DC motor, an AC motor, a pneumatic device, a piezoelectric, an electro-magnetic driver, a stepper motor, a servo, and/or the like.

[0022] As used herein, an “actuator” includes a movement generation device, circuitry configured to control the movement generation device and an optional encoder configured to measure movement and/or force of the movement generation device.

[0023] As used herein, an “end effector” is a device configured to interact with or operate on an object. Examples of end effectors include, a cutting tool, a gripping tool, a suction tool, a pushing tool, a pulling tool, a lifting tool, a welding tool, a gripping tool, an attachment tool, a heating tool, a soldering tool, a pressing tool, and/or the like. Tools need not make direct contact with an object. For example, a camera, laser, paint gun or a heat lamp may be used as an end effector. In some embodiments, an end effector includes a robotic hand, which has two or more fingers configured to manipulate objects, such as other tools and/or work pieces.

[0024] As used herein, “logic” is used to refer to hardware, firmware, and/or software stored on a computer non-transitory readable medium. Logic includes computing instructions and electronic circuits configured to execute these instructions.

[0025] FIG. 1 illustrates a Robotic System 100, according to various embodiments of the invention. Robotic System 100 can include a wide variety of alternative devices. For example, Robotic System 100 can include manipulators configured to move large objects or extremely small devices configured to perform delicate operations such as vascular surgery. Robotic System 100 can include self-guided vehicles such as drones. Robotic System 100 may include a human exoskeleton, or a prosthesis.

[0026] Robotic System 100 includes at least one Movement Generation Device 110 optionally configured to generate movement of at least one Transmission 120. Movement Generation Device 110 can include any of the movement generation devices discussed herein. Movement Generation Device 110 is optionally coupled with a control circuit and/or encoder configured to control or measure movement respectively. Movement Generation Device 110 is optionally coupled with a device configured to measure appearance, temperature, pressure, strain, current, or some other indicator of a state of Movement Generation Device 110. Movement Generation Device 110

optionally includes a “feedback mechanism” configured to measure a movement, torque, force and/or other aspect of the generated movement – and generate a corresponding signal. The feedback mechanism can include an encoder (either internal or external to a motor) or a resolver. Movement Generation Device 110 can include more than one feedback mechanism.

[0027] The term “transmission,” e.g., Transmission 120, is used to refer to a mechanical power transmission device configured to convey movement, power or energy between two points. Optionally Transmission 120 is a movable linkage such as a hydraulic coupling, a pneumatic coupling, a rope, a lever, a cable, a chain, a gear, a cam, a driveshaft, a rope, a screw drive, a belt, a pulley, and/or the like. Transmission 120 can include natural or synthetic fibers. Transmission 120 is coupled to Movement Generation Device 110 and at least one robotic Manipulator 130. Each Transmission 120 is configured to convey movement from an instance of Movement Generation Device 110 to one or more respective robotic Manipulators 130. For example, movement generated by an electric motor may be conveyed to a robotic manipulator via a pulley and cable. In various embodiments, Transmission 120 may experience changes in length due to load, temperature, age, and/or other factors. Various embodiments include Transmissions 120 configured in opposition. For example, a first Transmission 120 may be configured to rotate a joint in a first direction while a second Transmission 120 may be configured to rotate a joint in a second direction. Transmissions 120 optionally comprise one or more polymer fibers such as Nylon® and/or Spectra line®. For example, some embodiments of Transmissions 120 include multiple Nylon fibers woven into a rope or cord. Transmissions 120 including metal or polymer fibers may be referred to herein as “tendons.” Transmissions 120 can include metal, polymer, nano-structures, and/or the like.

[0028] In some embodiments, an instance of Transmission 120 includes a device having multiple linkages. For example, a single Transmission 120 can include multiple connections between anchor points. In another example, a single Transmission 130 can include a set of linkages or gears. These multiple cables may even take different routes. As used herein, a “single transmission” is characterized by a set of points at which Transmission 120 applies forces. For example, an instance

of Transmission 120 can include two cables (each of which may include multiple fibers) that take different paths between one or two or more co-located Movement Generation Devices 110 and location(s) at which both the cables apply force. This may be considered a “single transmission” because of the forces generated by the cables are applied at the same locations and, thus, the motions caused by the two cables are the same.

[0029] Manipulator 130 is typically a load bearing element, such as a wheel, robotic arm or truss. At least one of the one or more Manipulators 130 is configured to be attached to an End Effector 140. A pose of End Effector 140 is dependent on movement of Manipulator 130. Transmission 120 is configured to move Manipulator 130 or End Effector 140 in response to Movement Generation Device 110. Manipulator 130 is optionally a member of a plurality of robotic manipulators configured to manipulate End Effector 140 in the six-dimensional space of pose. Minimally this implies six degrees of freedom, however a robotic system may have more degrees of freedom than this minimal number.

[0030] Robotic System 100 optionally further includes one or more Camera 150 configured to generate an image of End Effector 140, Manipulator 130, and/or other objects within a three-dimensional environment. A pose of Camera 150 is optionally dependent on movement of an instance of Manipulator 130. As such, Camera 150 may be positioned in a way similar to other examples of End Effector 140. Some embodiments of Robotic System 100 include a first Manipulator 130 configured to move Camera 150 and a second Manipulator 130 configured to move End Effector 140 within a field of view of Camera 150. Camera 150 is optionally a stereoscopic camera or a color camera. In some embodiments, Camera 150 is replaced or augmented by an alternative detector such as a laser range finder, time-of-flight sensor, radar, sonic device, or other sensor capable of measuring depth either separately or in addition to color or grayscale imagery. Output from any such sensors or detectors may be processed along with or as an alternative to images generated by Camera 150.

[0031] Robotic System 100 further includes a Neural Network 160. Neural Network 160 is a multi-stage neural network including at least a perception block, a policy block and a compensation block, (see FIG. 3). These various blocks are optionally combined in one or more neural network stages configured to perform the functionality of the discrete blocks discussed herein for clarity. For example, all three blocks can be combined in a single neural network stage, or any two of these blocks can be combined in a particular stage. Particular neural network nodes may provide functionality of more than one of the blocks. As such, the boundaries between blocks may not be distinct and one or more neural network system including the functionality described as being provided by each block may be considered to include each of the perception, policy and compensation blocks. As used herein, neural network “blocks” may or may not be distinct from each other.

[0032] The perception block is configured to receive an image generated by Camera 150, and optionally other inputs, and to generate an image processing output representative of a state of an object with the image. For example, the perception block may be configured to receive signals from one or more of Camera 150 and also to receive signals from a feedback mechanism included in Movement Generation Devices 110. The signals received from the feedback mechanism may or may not be used to estimate a pose prior to being received by the perception block. If a pose is estimated prior to reception by the perception block, then this estimate may be provided to the perception block in addition to or instead of the direct signals received from the feedback mechanism. In various embodiments, the contributions to the image processing output can vary considerably. For example, in one embodiment the output is dependent purely on the image from Camera 150. In other embodiments, the output is dependent almost entirely (>90%) on signals received from the feedback mechanism(s), mostly dependent (>50%) on signals received from the feedback mechanism(s) or substantially dependent (>75%) on signals received from the feedback mechanism(s). In various embodiments, the contributions of images from Camera 150 to the image processing output can be at least 1, 3, 5, 10 or 15%, or any range there between. The perception

block is optionally configured to vary this dependence based on prior states of Neural Network 160 and/or perceived accuracy of the signals received from the feedback mechanism.

[0033] The policy block is configured to generate command signals for movement of End Effector 140 (or Camera 150). The generated command signals are based on i) a goal for End Effector 140, e.g., a desired pose or movement, ii) the image processing output, iii) optionally signals received from the feedback mechanisms of Movement Mechanisms 110, and optionally iv) time-dependent internal states of the policy block and/or compensation block. Signals received from feedback mechanisms of Movement Mechanisms 110 are optionally received directly by the policy block. For example, the policy block may receive encoder data indicative of position or current data indicative of applied torque. The compensation block is configured to provide an output for control of one or more of Movement Generation Device 110 based both the command signals and the image processing output. This output is typically an adapted version of the command signals generated by the policy block. Any of the perception block, policy block and compensation block can include recurrent neural network nodes and/or have other mechanisms for storing a “memory” state.

[0034] Robotic System 100 further includes Control Logic 170. Control Logic 170 is configured to provide a goal for movement of End Effector 140, to the policy block. Alternatively, Control Logic 170 may be configured to select a particular policy block configured to execute a specific goal. In specific examples, Control Logic 170 is configured to receive a set of instructions to move an object from a first location to a second location. This task is optionally divided into multiple steps each represented by a goal. The specific goals may be to 1) move a gripping tool adjacent to the object, 2) grasp the object using the gripping tool, 3) lift the object using the gripping tool to a first intermediate position, 4) move the object to a second intermediate position, and 5) place the object on a designated surface. Control Logic 170 is optionally configured to divide a task in to specific goals. Each of these goals is optionally performed by a different policy block.

[0035] In some embodiments, a particular policy block is configured to perform multiple goals and/or specific classes of goals. In these embodiments, a specific goal is provided to the policy block

at execution time. Control Logic 170 is optionally configured to select a policy block based on a specific goal class. For example, a specific policy block may be configured to execute “linear movement goals.” This policy block may receive a destination and a velocity; or a vector, velocity and distance, and use this information to perform a specific movement goal. Other specific policy blocks may be configured to execute “gripping goals,” “attachment goals,” “rotation goals,” “position relative to goals,” “insert goals,” “cut goals,” and/or the like.

[0036] In some embodiments, Control Logic 170 is configured to include default goals, such as avoiding a collision between Manipulator 130 and a person nearby, or avoiding contact between two different instances of Manipulator 130. Control Logic 170 may further be configured to select between different available end effectors for a task, for example between a gripping tool and a cutting tool. These different end effectors may be attached to different instances of Manipulator 130 or be alternatively attached to the same instance of Manipulator 130. Control Logic 170 may be configured to provide goals related to movement of Camera 150 and/or goals related to identifying a particular object. For example, Control Logic 170 may provide goals to identify male and female parts of a connector and positioning Camera 150 such that insertion of the male part into the female part can best be observed. Other goals provided by Control Logic 170 can include object recognition goals, movement goals, gripping goals, cutting goals, attachment goals, insertion goals, heating goals, positioning goals, activation goals (e.g., press the ON button), rotation goals, lifting goals, releasing goals, placement goals, and/or goals relating to any other interactions between End Effector 140 and on object.

[0037] Goals generated by Control Logic 170, and thereby selection of policy blocks, optionally depend on outputs of a perception block. For example, the outputs of a perception block may be used to identify a location, orientation and/or identity of an object. In one example, an orientation of an object may result in a goal of rotating the object to a different orientation. In another example, identification of a human hand by a perception block may result in a goal to avoid the hand

or interact with the hand. In a specific example, the goal may be to avoid contact between a cutting tool and a moving hand or to accept an object from the hand.

[0038] In some embodiments, a goal generated by Control Logic 170 is configured for calibration of the compensating block. For example, Control Logic 170 may generate a series of movement goals for the purpose of observing a resulting movement of End Effector 140. In this case, Camera 150 and the perception block are used to determine actual movements in response to control signals generated by the compensating block. Such movements and measured results cause a change in state of the compensating block and/or policy block, making the compensating block and/or policy block better able to generate command signals that will result in a desired movement.

[0039] In some embodiments, Control Logic 170 is configured to divide a task into goals of different magnitude. For example, a task of moving a gripping tool in position to grip an object may include a goal of moving a first distance, a goal of moving a second distance and a goal of moving a third distance. The first distance being larger than the second distance and the second distance being larger than the third distance. The goal of moving the second distance may be generated before or after execution of the goal of moving the first distance. More specifically, a task of moving approximately 11 cm may be divided into a goal of making a 10 cm movement, a goal of making a 1 cm movement and one or more goals of making sub-1 mm movement. A result of executing 1st goal is considered in defining the requirements of the 2nd goal and a result of executing the 2nd goal is considered in the number and requirements of subsequent goals. Such a task may be used, for example, to precisely place a pin in a hole.

[0040] A task performed using Control Logic 170 can include operation or activation of a machine. For example, a task may include electropolishing a part. Control Logic 170 can divide this task into goals such as picking up the part, attaching an electrode to the part, closing a protective cover, placing the part in an electropolishing bath, activating (turning on) an electropolishing circuit, opening the cover, removing the part from the bath, disconnecting the electrode, and/or placing the part on a transport device to be taken to a location of the next task to be performed on that part.

Activating the electropolishing circuit can include pressing a button using an instance of End Effector 140, activating a circuit using a command issued by Policy Block 320, and/or the like. Machine activation as part of a task performed using Control Logic 170 can include activating a washing device, a heating device, a cutting device, a spraying device, drilling device, a mixing device, a pressing device, a deposition device, a programming device, and/or any other device used in logical, mechanical or chemical processing of an object.

[0041] In some embodiments the start or completion of a goal are determined by visual input from Camera 150. For example, one or more images from Camera 150 may indicate that a gripping tool is in position to grip a target object, and subsequently that the gripping tool is in contact with the object. These images may be used to represent the completion of a positioning goal, the start of a gripping goal and the completion of a gripping goal. The one or more images are used to determine relative relationships between the objects, not necessarily absolute positions of the objects. This allows goals to be defined in terms of relative relationships between objects. For example, a goal may include moving a gripping tool to a pose (+/- some margin of distance error) relative to a target object. This goal can then be achieved even if the location and/or orientation of the target object changes as the goal is being executed.

[0042] Robotic System 100 optionally further includes a Memory Storage 180 configured to store long-term memory data. Long-term memory data is data that may be used to alter the state, e.g., operation of Neural Network 160 in response to a specific goal, sensor data, task, and/or image content. For example, long-term memory data may be used to change the generation of command signals in response to the identification of a particularly heavy or delicate object within an image obtained using Camera 150. The memory data is "long-term" in that one or more hours, days, weeks or any arbitrary time period may pass between its use. Long-term memory data may be used to abort a task or goal based on identification of a person within an image, or otherwise detected near Tool 140. In some embodiments, long-term memory data is used to override planned movements or change priorities. For example, long-term memory may be used to alter the state of

Neural Network 160 by being provided as input to nodes of Neural Network 160. These nodes can be in any of the blocks discussed elsewhere herein. Typically, long-term memory data is not dependent on immediately prior command signals and immediately prior captured images. For example, the content of long-term memory data is usually not dependent on movements or other results of the most recent sets of adapted command signals generated using Neural Network 160.

[0043] Long term memory data may be provided to Neural Network 160 using a variety of approaches. For example, data may be retrieved from random access memory, non-volatile memory, and or the like. The long term memory data can be provided as operand data inputs to nodes of Neural Network 160 (at any of the blocks discussed herein). Alternatively, the long term memory data may be used to change weighting or operation of specific nodes. In some embodiments, long term memory data is stored in a neural memory accessed dynamically by Neural Network 160 during data processing. These embodiments may result in a neural Turing machine or differentiable neural computer

[0044] Long-term memory data can be contrasted with short-term “memory” (adjustments to neural network state due to recent events) that results from nodes of Neural Network 160 configured to receive a past state or “memory,” e.g., recurrent nodes. The short-term memory is a result of immediately preceding states of the nodes. Various embodiments of Neural Network 160 may, thus, have both short-term memory based on the most recent results of command signals, and also long-term memory that can be used applied at times that are minutes, hours or days, etc. apart. These memories of different time scales are optionally used to compensate for different types of errors that can occur in Robotic System. For example, short-term memory can be used to compensate for “play” or “lag” in the movement of Transmission 120 in response to signals received by Movement Generation Device 110. While long-term memory can be used to compensate for appearance of an unexpected object within an image or stretching of Transmission 120 resulting from manipulating a particularly heavy object. Long-term memory is configured to operate at a longer time period relative to short-term memory. Other uses of long-term memory include

controlling a force used to manipulate and/or modify a particular material, adjusting a time for an operation, restoring defaults for a new Transmission 120.

[0045] In various embodiments, Control Logic 170 is configured to add long-term memory data to Memory Storage 180 in response to particular events. For example, if Neural Network 160 fails to properly generate command signals to perform a task, if the adaption of command signals includes a change beyond a predetermined threshold, or if a discrete event (such as replacement of Transmission 120), then Control Logic 170 may be configured to store memory data in Memory Storage 180 and associated this data with a corresponding event. Using long-term memory data stored in Memory Storage 180.

[0046] FIG. 2 illustrates a Robot 200, according to various embodiments of the invention. The Robot 200 is meant as an illustrative example. Various embodiments of the invention include a wide variety of robotic architectures, designs and structures in addition to or instead of those illustrated in FIG. 2, which is for illustrative purposes. Robot 200 can include a system of arbitrary complexity and may include multiple End Effectors 140 of any type known in the field of robotics. Robot 200 can include both robot arms, e.g., one or more Manipulators 130 and robot hands, e.g. End Effectors 140, having one, two or more “fingers.” By using image input, the systems and methods described herein can be used to control both the robot arms and robot hands. Generated images detect the result of movement of both the “arms” (Manipulators 130) and “hands” (End Effectors 140) of Robot 200. As a result, a neural network trained using such images inherently provides an optimal balance between control of the movement of the arms and hands. For example, the generated movement of the arms and hands can have an optimal relative magnitude optimized to achieve a goal. In a specific case, picking up an object using a robot hand, e.g. End Effector 140, can include movement of both an arm, e.g., one or more Manipulators 130, and fingers of the hand. The neural network system described herein, trained based on images generated using Camera 150, can result in an optimal movement. The optimization being with regard to minimal error toward achieving a desired goal, minimal total movement, minimal energy usage, most probable goal achievement, minimal

adverse effects (e.g., damage to a target object or person), and or the like. Robot 200 can include large scale robots configured to manipulate heavy loads, small scale robotics configured to perform surgery or modify integrated circuits, mobile robots, and/or anything in between.

[0047] Robot 200 includes a Base 210 configured to support other elements of Robot 200. Base 210 can be fixed, movable, or mobile. For example, in some embodiments Base 210 includes propulsion, a conveyor, legs, wheels or tracks, and movement of an End Effector 140 optionally includes movement of Base 210. Alternatively, Base 210 may be configured to be bolted or otherwise fixed to a floor and to support heavy loads manipulated by one or more End Effectors 140. Alternatively, Base 210 may include a body of a walking robot in which End Effectors 140 include tracks, pads or feet. Alternatively, Base 210 may include a body of a floating or submersible embodiment of Robot 200. Base 210 may be configured to support multiple robotic arms and End Effectors 140.

[0048] Robot 200 further includes at least one Movement Generation Device 110. Movement Generation Device 110 is configured to generate movement, e.g., rotational and/or linear movement. In some embodiments, Movement Generation Device 110 is attached to a Transmission 120, Transmission 120 is attached to a Manipulator 130, and Manipulator 130 is attached to End Effector 140, such that the pose of End Effector 140 is responsive to movement generated by Movement Generation Device 110. End Effector 140, Manipulator 130 and/or Movement Generation Device 110 are optionally separated by at least one Robotic Joint 225. In various embodiments, an instance of Movement Generation Device 110 is connected to a particular End Effector 140 by a Transmission 120 that traverses one, two, three or more Robotic Joints 225.

[0049] Robotic Joint 225 can include, for example, linear joints, orthogonal joints, rotational joints, twisting joints, or revolving joints. Instances of Robotic Joint 225 can be configured to couple Base 210, Manipulators 130, and/or End Effectors 140. In various embodiments, End Effector 140 and/or Manipulator(s) 130 are separate by one or more Robotic Joints 225. Transmission(s) 120 are optionally configured to traverse these Robotic Joints 225. For example, as illustrated in FIG. 2,

Transmission 120 can extend from Movement Generation Device 110, past one or more Robotic Joints 225, past one or more Manipulators 130, to terminate in a connection to one of Manipulators 120 or End Effector 140.

[0050] FIG. 3 illustrates instances of Neural Network 160 at different times, according to various embodiments of the invention. Neural Network 160 includes at least a Perception Block 310, a Policy Block 320 and a Compensation Block 330. Neural Network 160 is configured to receive images, and based on those images generate command signals configured to control Movement Generation Device 110. The command signals are generated to complete a goal, such as movement or operation of End Effector 140.

[0051] Perception Block 310 includes a neural network configured to receive an image, and/or series of images, and generate an image processing output representative of the state of an object within the image. The image processing output can include object features, e.g., corners, edges, etc., identified within an image; and/or relationships there between. In a specific example, the image processing output can include joint angles and positional coordinates of the fingers of a robotic hand, and distances between these fingers and an object. The image processing output can include classifications and/or identifications of objects within an image. The image processing output can include data characterizing differences between two images, for example, a number of pixels an object has moved between images, or numbers of pixels particular object features have moved between images. In various embodiments Perception Block 310 is configured to generate an image processing output based on a stereoscopic image, a light field image, and/or a multiscopic image set generated by two or more Camera 150 with overlapping fields of view. In various embodiments Perception Block 310 is configured to determine spatial relationships between objects. For example, Perception Block 310 may be configured to generate an image processing output representative of a distance between a target object and End Effector 140. The image processing output optionally includes a representation of a pose of an object within the image and/or a pose of End Effector 140.

[0052] Perception Block 310 optionally includes a recurrent neural network in which the processing of an image results in a change in state in of the neural network, and/or an alternative method of storing and using past states of the neural networks. The change in state is typically represented by a change in operation of specific nodes within the neural network. This change in operation is, optionally, a result of a previous (e.g., recurrent or memory) output of that specific node or other nodes within the network. Specifically, a previous output may be included as a current input to the operation of the node. Specific nodes, sets of nodes, levels of nodes, and/or entire blocks of nodes may be responsive to any previous output, and thus their operational state may change over time. Optionally, a recurrent instance of Perception Block 310 may be used to detect changes between images. For example, movement of objects as seen in different images or a change in viewpoint from which the image is obtained.

[0053] In some embodiments, Neural Network 160 includes a plurality of Perception Blocks 310. Each of these Perception Blocks 310 are optionally associated with a different camera, the different cameras having overlapping fields of view such that they can be used to view an object from different viewpoints. Alternatively, a particular Perception Block 310 may be configured to receive images from two or more cameras. As discussed elsewhere herein, e.g., with reference to FIG. 4, a multiplex layer is optionally used to selectively communicate image processing outputs from each of the Perception Blocks 310 to one or more Policy Block 320. The different Perception Blocks 310 are optionally configured to process images in different ways. For example, one Perception Block 310 may be configured to read barcodes, another Perception Block 310 may be configured to recognize particular objects, e.g., faces or end effectors, and/or another perception block may be configured to measure distances based on a stereo image pair. One of Perception Block 310 may be configured to detect geometric objects such as a bolt or an integrated circuit while another Perception Block 310 is configured to identify people, e.g., a hand in a work area. Perception Blocks 310 may process images in parallel or serially. For example, in parallel processing, a first Perception Block 310 may

process an image at the same time that a second Perception Block 310 is processing the same image or a different image.

[0054] In various embodiments, image processing outputs of Perception Block 310 include a representation of a distance between End Effector 140 and an object as seen within a processed image, and/or a distance between two objects with the image. The outputs can include a representation of an object within a three-dimensional environment. In various embodiments, image processing outputs include a representation of a change in state of an object within a processed image, as compared to a prior image. For example, the outputs can include information regarding translation or rotation of an object, a change in color of an object, filling of a seam, hole, or gap (as in a welding operation), addition of a material (as in a soldering operation), alignment of objects or surfaces (as in positioning of an object at a desired place or a screw over an opening), insertion of one object into another, and/or the like.

[0055] In some embodiments, image processing outputs of Perception Block 310 include estimates of positions of objects that are occluded by other objects within an image. For example, if a first object is moved in front of a second object, a position of the second object may be estimated from data received in prior images. The “memory” of the position of the second object can be retained in a state of the Perception Block 310, where Perception Block 310 includes one or more recurrent or other types of “memory” layers. Such memory may be otherwise stored in an external memory that is accessed by the neural network, such as with a Differentiable Neurocomputer.

[0056] Policy Block 320 is configured to generate command signals for movement of End Effector 140. The generated command signals are based on at least: 1) a goal for movement of End Effector 140, 2) the image processing output received from Perception Block(s) 310, optionally 3) a time dependent internal state of Policy Block 320, and optionally 4) feedback received from Compensation Block 330. Neural Network 160 optionally includes multiple Policy Block 320. Optionally, different instances of Policy Block 320 are configured to perform different tasks and/or goals. For example, one instance may be configured for accomplishing a welding goal while other

instances are configured for accomplishing moving or gripping goals. An instance of Policy Block 320 may be configured to accomplish any one or more of the goals discussed herein. Selection of a particular instance of Policy Block 320 for processing a particular image is optionally responsive to a type of goal for movement of End Effector 140. For example, an instance of Policy Block 320 configured to accomplish a gripping goal may be configured to generate commands that result in applying a particular force using an instance of End Effector 140 configured for gripping. Instance of Policy Block 320 can, thus, be configured to generate command signals for a wide variety of different specific actions.

[0057] Policy Blocks 320 may be configured to generate command signals for a specific task, for classes of tasks, or in some embodiments an instance of Policy Block 320 is configured to generate command signals for general tasks. For example, one instance of Policy Block 320 can be configured to generate command signals for a movement task while another instance of Policy Block 320 is configured to generate command signals for driving a screw. A Policy Block 320 for a specific task or class of tasks may be selected using Control Logic 170, from a plurality of alternative Policy Blocks 320, for processing of an image processing output received from Perception Block 310. The selected policy being based on a current goal and/or task. A policy selection may occur external to the system illustrated in FIG. 3. For example, a policy may be selected using an embodiment of Control Logic 170 including a finite state machine. Alternatively, selection of a policy may be performed by a separate neural network or portion of the policy network configured so as to respond to different visual or other cues to determine the relevant policy or policy phase to execute at some particular moment.

[0058] Policy Block 320 optionally includes recurrent layers or other memory dependent mechanism in which a state of Policy Block 320 is changed through processing of image processing output. These changes in state impact how the next image processing output is processed using the same Policy Block 320. The processing of image processing output can, thus, be dependent on prior states of Policy Block 320. In some embodiments, Policy Block 320 is configured to receive outputs

(image processing outputs) from multiple Perception Blocks 310. These outputs can be received in parallel or serially. For example, in some embodiments, Policy Block 320 is configured to receive outputs from a first Perception Block 310 configured to determine distances between objects, a second Perception Block 310 configured to detect orientation of objects and a third Perception Block 310 configured to detect presence of a person. These outputs may be received at essentially the same time or one of the outputs at a time. Command signals generated by Policy Block 320 can be configured to move End Effector 140, to move an object, to grasp an object, to apply a pressure, to rotate an object, to align objects, and/or any other action disclosed herein.

[0059] In some embodiments, multiple Policy Blocks 320 are configured to process image processing outputs in a serial manner. For example, a first Policy Block 320 may receive the image processing output from Perception Block 310 and determine if a goal has been achieved. If the goal has not been received the image processing output is provided to a second Policy Block 320 configured to generate control signals for moving End Effector 140 to a new pose – based on the goal. These control signals, and optionally the image processing output are then received by a third Policy Block 320 configured to adjust these control signals, if necessary, such that the movement of End Effector 140 does not result in a collision with a person or other object.

[0060] One or more of Policy Block 320 is optionally configured to receive image processing output based on images received from multiple Cameras 150, and to generate the command signals based on the multiple images. The images may be received and/or processed in serial or parallel. For example, Cameras 150 may be disposed to view an environment and/or object from several different vantage points and Policy Block 320 may use images generated by these Cameras 150, in combination, to generate control signals. Cameras 150 are optionally disposed in stereoscopic or multiscope configurations. For example, clusters of Cameras 150 may be disposed in different locations, each cluster including Cameras 150 configured such that their images can be combined to achieve three dimensional information, expanded field of view, multi-viewpoints, variations in resolution, and/or the like.

[0061] In a specific example, command signals generated by Policy Block 320 may be configured to move End Effector 140 10 mm. However, image processing output indicates that the previous command signals underachieved a desired movement, e.g., command signals which were intended to move End Effector 140 by 15 mm resulted in a movement of only 13.5 mm. In response to this information about prior results of command signals, Compensation Block 330 is configured to adjust the current command signals received from Policy Block 320 such that the resulting movement of End Effector 140 is closer to the desired 10 mm (relative to the movement that would result from uncompensated command signals). Compensation Block 330 uses differences between expected movement (or other operation) of End Effector 140 and actual detected movement to adjust future command signals such that the adjusted command signals better result in the currently desired movement. Adjustment for operations other than movement are compensated for in a similar fashion.

[0062] The command signals generated by Policy Block 320 are typically sent to Compensation Block 330 for adjustment. Compensation Block 330 is configured to adjust the command signals based on at least the image processing output (generated by Perception Block(s) 310), and to produce a resulting output for control of Movement Generation Device 110. Compensation Block 330 is responsive to both the image processing output, as generated by Perception Block(s) 310, and command signals generated by Policy Block(s) 320. Optionally, Compensation Block 330 is configured to receive a copy of the image processing output that has not been processed by Policy Block 320.

[0063] A purpose of the dependence on the image processing output is so that Compensation Block 330 can adjust the command signals responsive to changes in the environment which occurred as a result of recent, e.g., the last, actions by Movement Generation Device 110. Specifically, Compensation Block 330 is configured to use the image processing output to modify control signals sent to Movement Generation Device 110, where the modification is responsive to how Movement Generation Device 110 responded to recent control signals as indicated by the image processing

output. Compensation Block 330 is optionally configured to determine a result of a prior set of control signals provided to Movement Generation Device 110 based on the image processing output, and to adapt subsequent control signals responsive to this result.

[0064] Compensation Block 330 is, thus, able to adjust command signals over time to compensate for inaccuracies in the expected physical dimensions and other properties of Robot 200, physical changes in parts of Robot 225, changes that occur over time, changes in environment in which Robot 225 is operated, and/or the like. These changes can include changes in length of Transmissions 120 or Robotic Manipulator 130, wear in gears and/or backlash resulting from wear, Robotic Joints 225 or actuators, temperature changes, changes in spring strength, changes in hydraulic or pneumatic system response, loads on Movement Generation Device 110, weights and balance of objects being manipulated, changes in motor power, and/or the like. For example, Compensation Block 330 is optionally configured to compensate for weight of an object lifted by the end effector, by adapting the output for control of Movement Generation Device 110. This adaptation may occur in real-time based on the identity of an object or failure to move the object as expected using a prior command. Such an adaptation can include, for example a change in a selected voltage, current or digital input provided to Movement Generation Device 110.

[0065] Any of Perception Block 310, Policy Block 320, and/or Compensation Block 330 may be configured to receive memory data from Memory Storage 180 and, thus, change state.

[0066] The system may possess memory, either explicitly, or implicitly through the configuration of recurrence or other properties of a neural network, which has the function of associating changes in control policy with different objects the robot is intended to manipulate. These changes may be effective when an object is seen, when it is grasped, when it is lifted, or at any other subset of the overall task of manipulating the object. These changes may affect how actuators are used, what limitations are placed on the actuator motion or energy, force applied, or any other thing material to the strategy for manipulation, including such actions as might be used to pre-tension elastic elements of the system or changes in the grasping or lifting strategy (e.g. grasp around a light object

and lift transverse to the grasp forces (i.e. relying on friction) vs. grasping beneath a heavy object and lifting in the direction of the grasp forces (i.e. presuming friction to be unreliable).

[0067] Optionally some Policy Blocks 320 are configured for calibration of Compensation Block 330.

These Policy Blocks 320 generate command signals specifically selected to clearly detect resulting actions (e.g., movements) of End Effector 140 and, as such, alter the state of Compensation Block 330 to improve adjustments of command signals made by Compensation Block 330. The state of Compensation Block 330 is, thus, optionally representative of a prior response of the Robotic Manipulator to (adapted) command signals sent to Movement Generation Device 110.

[0068] Any combination of Perception Block 310, Policy Block(s) 320, and/or Compensation Block 330 may be trained together or separately. For example, in some embodiments, Perception Block 310 is first trained to generate processed image outputs. These outputs are then used to train Policy Block 320. Finally, all three blocks may be further trained as a unit.

[0069] FIG. 3 illustrates Neural Network 160 at a “Time A” and a “Time B.” Perception Block 310, Policy Block 320, and/or Compensation Block 330 may have different states at these different times, the states being indicated by “310A” and “310B” etc. A first image is processed at Time A and a next or subsequent image is processed at Time B. Because of the recurrent or memory dependent layers in at least the Compensation Block 330, the processing of the first image effects how the second image is processed. Specifically, the change in state may be reflected by changes in operation of nodes in the neural network and these changes impact the processing of later images. The system is adapted (learns) in real time using received images.

[0070] Arrows within FIG. 3 represent examples of movement of image processing output (340), movement of command signals (350) and possible movement of state information (360).

[0071] FIG. 4 illustrates embodiments of Neural Network 160 including one or more multiplex layer. In these embodiments a multiplex block (Mux 340) is configured to receive image processing outputs from several Perception Blocks 310 (indicated 310A, 310B and 310C, etc.) and to communicate these image processing outputs to one or more Policy Block 320. The image processing outputs are

optionally generated based on images and/or other sensor data, received from different cameras and/or other sensors. Mux 410 may be configured to provide these outputs in parallel or serially. In some embodiments Mux 410 is configured to generate a three-dimensional representation of an environment in which Robotic System 100 operates based on the received image processing outputs, and then provided that representation to Policy Block 320.

[0072] In some embodiments, Perception Blocks are used selectively. For example, for achieving a particular goal, the output of Perception Block 310B may not be relevant. In this case Perception Block 310B may not be used to process an image. In a more specific example, if Perception Block 310B is configured to receive an image from a camera that does not have a useful view of End Effector 140 and an object, then an image from that camera may not be computed and/or results of any processing of that image may not be passed to any of Policy Blocks 320.

[0073] Alternative embodiments, not shown, Mux 410 may be configured to receive command signals from multiple Policy Blocks 210 and provide these command signals to Compensation Block 330. These embodiments of Mux 410 are optionally configured to process or combine the received command signals. For example, if a first Policy Block 210 is configured to generate an output to move End Effector 140 and a second Policy Block 210 is configured to avoid having End Effector 140 hitting a nearby person, then Mux 410 may be configured to assure that command signals are communicated such that the goal of not hitting a person takes priority. In another example, Mux 410 is optionally configured to combine command signals received from two different instances of Policy Block 210, where a first instance of Policy Block 210 is configured to generate control signals to a first instance of Movement Generation Device 110 (optionally coupled to a first Robotic Manipulator 130) and a second instance of Policy Block 210 is configured to generate control signals to a second instance of Movement Generation Device 110 (optionally coupled to a second Robotic Manipulator 130). The first and second Robotic Manipulator 130 may be part of a same robotic arm, optionally separated by an instance of Robotic Joint 225, configured to move a single End Effector

140 as illustrated in FIG. 2, or may be attached to separate End Effectors 140 configured to work together on an object.

[0074] In some embodiments, a single Policy Block 210 is configured to control multiple Robotic Manipulators 130. These Robotic Manipulators 130 may be part of a single robotic arm or part of separate robotic arms. For example, a single Policy Block 210 may be configured to control a Robotic Manipulator 130 used to position a screw and also to control a Robotic Manipulator 130 used to rotate a screwdriver. By using a single Policy Block 210 the two (or more) robotic arms may be operated in a coordinated fashion. Likewise, if a single Policy Block 210 is used to control Robotic Manipulators 130 which are part of the same robotic arm, their movement can be coordinated to achieve a goal.

[0075] FIG. 5 illustrates methods of controlling a robot, according to various embodiments of the invention. The methods illustrated in FIG. 5 are optionally performed using Robotic System 100 and/or Neural Network 160. The methods include using images, and optionally other sensor data, as the primary input to control positioning and/or use of an end effector, such as End Effector 140. Recurrent or memory dependent layers within Neural Network 160 are configured such that control signals can be adapted based on the results of prior control signals as indicated in the images

[0076] In a Receive Task Step 505, a task for the operation of Robotic System 100 is received. As described elsewhere herein, this task can be, for example, to place an object in a particular position, to pick up an object, to connect two objects, to apply heat or other processing to an object, and/or the like. In an illustrative example, tasks may include 1) placing an adhesive on a first object (with a certain application profile), 2) placing a second object against the first object, and 3) removing excess adhesive. The goals are optionally received by Control Logic 170 from a source external to Robotic System 100. For example, a human user may enter a series of tasks via a user interface displayed on a client device.

[0077] In an optional Divide Task Step 510, the one or more tasks received in Receive Task Step 505 are divided into specific goals. As noted elsewhere herein, goals are specific steps that may be

performed to complete a task. For example, the above task of placing an adhesive on a first object may be divided into goals of: a) positioning the object, b) picking up a glue dispenser, c) positioning the glue dispenser relative to the object, d) compressing the glue dispenser to cause glue to be released onto the object, and e) moving the glue dispenser (or object) as the glue is released. Each of these steps can be performed using camera-based monitoring and real-time feedback via Neural Network 160. For example, compressing the glue dispenser may be monitored using an instance of Policy Block 320 specifically configured to receive and use criteria for how a desired bead of glue should appear on the object. Divide Task Step 510 is optionally performed using Control Logic 170.

[0078] In some cases, the goals themselves may be understood by the policy block using input from the Perception Block 310. For example, the robot may be presented with an image or video that describes, demonstrates, or shows, the correct behavior, or some aspect of it. As a concrete example, the robot may be presented with the image of a correctly assembled part and given the components. Based on the image, the robot's task (and goals) may be implicitly defined, i.e. to put the pieces together to form the assembled part shown in the image. Policy Block 320 is optionally trained using inverse reinforcement learning.

[0079] In a Capture Image Step 515, one or more images are captured using Camera 150. The one or more images typically include End Effector 140 and/or an object to be manipulated by End Effector 140. As noted elsewhere herein, in various embodiments, Camera 150 may be supplemented by other sensing devices such as a laser range finder, radar, an acoustic range finder, a pressure sensor, an electrical contact sensor, a current or voltage sensor, a magnetic sensor, an encoder, any other sensor or detector discussed herein, and/or the like. In these embodiments, respective sensor data may also be received from one or more of these devices in Capture Image Step 515.

[0080] In a Process Image Step 520, the image(s) and optionally other sensor/detector data received in Capture Image Step 515 are processed using Neural Network 160. This processing results in at least one "image processing output" as discussed elsewhere herein. The image processing

output can include information derived from the one or more images and/or from any of the other sensor data from any of the other sensors discussed herein. In various embodiments, the image processing output includes features of a processed image, and/or differences between different images. In some embodiments, image processing outputs include a representation of objects within a three-dimensional environment. For example, the image processing output can indicate object orientation and/or spatial relationships between objects in three dimensions. Process Image Step 520 is optionally performed using Perception Block 310 and can result in any of the image processing outputs taught herein to be generated by Perception Block 310.

[0081] In an Apply Policy Step 530, the one or more image processing outputs generated in Process Image Step 520 are used to generate control commands configured for the control of a robotic device, such as Robot 200. The control signals are generated in response to a goal, as well as the image processing output. The goal can be a subset of a task, as described elsewhere herein. The control commands are optionally configured to cause operation of Movement Generation Device 110.

[0082] Apply Policy Step 530 optionally includes selection of one or more Policy Block 320 from a plurality of Policy Blocks 320, for the processing of the image processing outputs. Specific instances of Perception Block 310 may be associated with specific instances of Policy Block 320. For example, different associated pairs (or sets) of Perception Blocks 310 and Policy Blocks 320 may be configured to perform alternative tasks of “picking up an object,” “placing an object” or “pushing a button.” In another example, an instance of Perception Block 310 configured to detect presence of a person in an image may be associated with an instance of Policy Block 320 configured to assure that End Effector 140 does not come in contact with a person. This specific pair of Perception Block 310 and Policy Block 320 may have priority over other pairs of blocks operating in parallel, rather than being alternative combinations. In another example, a Perception Block 310 configured to analyze an image of a glue bead or a metal weld may be associated with a Policy Block 320 configured to generate command signals to deposit the glue bead or metal weld, respectively. Outputs of this

Perception Block 310 are sent to, at least, those Policy Blocks 320 with which they are associated, optionally among other Policy Blocks 320.

[0083] In Apply Policy Step 530, Policy Blocks 320 are optionally selected for further processing of specific image processing outputs based on the contents of these image processing outputs. For example, if a specific object is identified as being within an image, then a Policy Block 320 configured (e.g., trained) to generate command signals to manipulate the identified object may be selected. For example, an image including a flask of a liquid may result in an image processing output identifying the flask and liquid carrying capacity, and this output may be assigned to an instance of Policy Blocks 320 specifically configured to move flasks of liquids.

[0084] Selection of one or more specific Policy Block(s) 320 is optionally included in Process Image Step 520. A particular image or other sensor data may be processed using multiple Policy Blocks 320, each of the Policy Blocks 320 being trained for a different purpose. For example, an image may be processed by a first Policy Block 320 configured to monitor an amount of glue applied to an object and also processed by another Policy Block 320 configured to monitor movement of a glue dispenser relative to the object.

[0085] Command signals generated by Policy Block(s) 320 may indicate that a goal has been achieved. For example, if completion of a goal requires no additional action by Robot 200 and/or End Effector 140, then a goal may be considered complete. In some embodiments, the completion of a goal is indicated by a particular sensor state. For example, a sensor configured to detect external temperature may indicate that a desired temperature of a workpiece has been reached or a current sensor may indicate that two conductors are in contact with each other. This sensor state may be recognized by Perception Block 310 as indicating that the goal has been completed. Policy Block 320 may be trained with the objective of reaching this sensor state. Specifically, in some embodiments, a particular sensory state, recognized by the Perception Block 310, is sufficient to distinguish a completed goal from an incomplete goal. In such a case, the Policy Block is trained to recognize this state and terminate the policy.

[0086] In an optional Goal Achieved? Step 540, a determination is made as to whether the current goal has been achieved. Achievement may be indicated by location, orientation, and/or other characteristic of an object; connections between objects; and/or completed modification of one or more objects. The determination of whether a goal has been achieved is typically based at least in part on the representation of object embodied in the image processing output. In some embodiments, a goal may be explicitly aborted instead of being completed.

[0087] Goal Achieved? Step 540 is optionally included in an early part of Apply Policy Step 530. As such the determination of whether a goal has been achieved can be made prior to further processing of an image processing output by Policy Block 320.

[0088] In an optional Request Goal Step 545, a new goal is requested. The new goal can be part of an existing task or a new task. The new goal is typically provided by Control Logic 170. New goals may be requested when a prior goal is completed or aborted.

[0089] In a Compensate Step 550, the command signals provide by one or more of Policy Blocks 320 are adjusted to produce compensated control signals. In various embodiments, the compensation is based on any combination of: the received command signals, past command signals, image processing output, goals, safety requirements, a current state of Compensation Block 340, one or more prior states of Compensation Block 340, and/or the like. Compensation Step 550 is optionally performed using Compensation Block 340. The compensation can include, for example, an adjustment in a current, voltage, distance, pressure, digital command, time period, and/or any other aspect of control signals. In some embodiments, the compensation is for a change in response of Robotic Manipulator 130 and/or End Effector 140 to prior (optionally compensated) control signals. This change can occur over time or can be in response to a load on End Effector 140, e.g., lifting of a heavy object or cutting a tough object.

[0090] In some embodiments, Compensate Step 550 uses one or more recurrent or memory dependent layers within Compensation Block 340 in order to make the compensation dependent on past commands and observed responses to these commands by Robotic System 100. The layers are

configured, e.g., via training, such that differences, between expected responses and observed responses (as observed in the images processed by Perception Block(s) 310) of Robotic System 100 to received command signals, result in changes to the state of Compensation Block 340. These changes are configured such that the response of Robotic System 100 to future compensated command signals is closer to a desired and/or expected response.

[0091] For example, if a goal is to move End Effector 140 a distance of 20 cm in a direction X over a period of 3 seconds and control signals to perform this movement result in movement of only 18 cm in a direction X+10 degrees over 5 seconds, as observed by Camera 150, then the state of Compensation Block 340 is changes such that the next compensated command signals generated for this same goal result in a movement closer to 20 cm (relative to 18 cm), closer to direction X (relative to direction X+10 degrees), and/or a time of movement closer to 3 seconds (relative to 5 seconds). Direction X may be defined in a two or three-dimensional coordinate system. In some embodiments, adjustments to command signals made by Compensation Block 330 are optionally provided as feedback to one or more of Policy Blocks 320 in order to change a state of Policy Blocks 320. This change in state is also typically configured to adjust future command signals to be more likely to produce desired responses in Robotic System 100. In these embodiments, any of the various features described herein as being included in Compensation Block 340 are optionally included in Policy Block 330.

[0092] Compensation Block 330 is optionally configured to compensate for a change in the length of one or more Transmissions 120. These Transmissions 120 may be configured to move Robotic Manipulators 130 and/or End Effectors 140. The Robotic Manipulators 130 may be part of the same or different robotic arms. In a specific example, Compensation Block 330 is configured to generate compensated command signals to coordinate movement of two or more End Effector 140 to desired relative poses. Compensation Block 330 is optionally configured to compensate for variations in the length of Transmission 120 of at least 0.25%, 0.5%, 1%, 2%, 3%, 10%, or any range there between. Compensation Block 330 is optionally configured to compensate for play in positioning of Robotic

Manipulators 130 and/or End Effectors 140 that results from changes in lengths or effective lengths of Transmissions 120 configured in opposition, and/or for play or hysteresis in other movement coupling devices. For example, in various embodiments, Transmissions 120 that consist of a set of gears and cams have play and hysteresis that can be compensated for by Compensation Block 330. In another example, Policy Block 320 may be configured to detect play or hysteresis in the positioning of End Effector 140 and Compensation Block 330 may be configured to adjust for this hysteresis by adapting control signals such that the hysteresis is compensated for or eliminated. Compensation Block 330 may, thus, auto-tension Transmissions 120 in real-time. Hysteresis can be dependent on immediately preceding motions, command signals or events, and thus have a temporal and/or state dependence. In some embodiments, recurrent or memory dependent layers of Compensation Block 330 are used to account for and compensate for this temporal and/or state dependence.

[0093] In an Activate Step 560, one or more of Robotic Manipulators 130 are activated using the compensated command signals generated by Compensation Block 340. This activation can include sending the compensated command signals to Robot 200 from Neural Network 160 via a communication network. Following Activate Step 560, the methods illustrated in FIG. 5 optionally return to Capture Image Step 515 and the process repeated.

[0094] FIG. 6 illustrates an End Effector 140, according to various embodiments of the invention. In these embodiments, End Effector 140 includes one, two or more Digits 610. For example, multiple Digits 610 may be connected to a Palm 620 and be configured to mimic functionality of a human hand. One or more of the Digits 610 may be arranged in opposition each other. Each of Digits 610 includes two, three or more Segments 630 including a Proximal Segment 630A, optional Medial Segment(s) 630B, and a Distal Segment 630C. The designation of “Proximal” and “Distal” are relative to Palm 620. Palm 620 may be an example of a robotic Manipulator 130. While the End Effector 140 is illustrated as having three segments, the systems and methods discussed here can be applied to

systems having 4 or more digits. Further, the position of more than two joints may be controlled by a single Transmission 120.

[0095] Segments 630 are separate by Joints 640 at which Segments 630 may rotate and/or translate relative to each other. Joints 640 typically comprise some sort of transmission (not shown), such as a hinge joint, pivot joint, ball-and-socket, slip joint, saddle joint, and/or the like. At least one of Joints 640 is configured to attach Proximal Segment 630A to Palm 620. Like Segments 630, Joints 640 may be designated as a Proximal Joint 640A, optional Medial Joint(s) 640B, and a Distal Joint 630C. The relationships between these Joints 640 and Segments 630 are illustrated in FIG. 6. A position of End Effector 140, as illustrated in FIG. 6, is characterized in part by Angles 650A-650C representative of spatial relationships between Segments 630 and Palm 620. These Angles 650 are optionally measured relative to axes or features of Segments 630 other than those shown.

[0096] The illustrated embodiments of End Effector 140 include a first Transmission 120A configured to control at least two of Angles 650 between Segments 630. Optionally, the at least two Angles 650 include an Angle 650C between Distal Segment 630C and an adjacent Segment 630. In a two-segment Digit 610 the adjacent Segment 630 is Proximal Segment 630A. In a three (or more) segment Digit 610 the adjacent Segment 630 is the one of Medial Segment(s) 630B closest to Distal Segment 630C. The at least two Angles 650 also include the next closest Angle 650C to Distal Segment 630C. Specifically, in the three-segment Digit 610 illustrated in FIG. 6, Transmission 120A is configured to control both Angle 650B and Angle 650C. In some embodiments single Transmission 120A is the only Transmission 120 configured to control these multiple angles. The positions of Transmissions 120 within End Effector 140 are shown for illustrative purposes only, in practice Transmissions 120 may be disposed in a wide variety of locations within End Effector 140. Likewise, the length, shapes and sizes of Segments 630 may vary widely.

[0097] Transmissions 120 can be configured to flex (curl tighter) and/or extend Digit 610. In some embodiments, Transmission 120A includes two transmissions configured to apply opposing forces to both flex and extend one of Digits 610. In some embodiments, End Effector 140 includes one or

more elastic element (not shown) configured to apply a force opposing pulling on Transmissions 120 using Movement Generation Device 110. The elastic element can include a spring, a coil, a rubber band, an elastic membrane, a pneumatic, an electrical coil configured to generate an electro-magnetic force, a magnet, a device configured to change shape in response to a voltage or current, a variable stiffness actuator, and/or the like. The elastic elements may be located at joints and/or elsewhere along the path of Transmissions 120. For example, each joint within Digit 610 can include an elastic element configured to extend Digit 610. Alternatively, an elastic element may be located at an alternative location and connected to a joint via an additional Transmission 120 or other linkage.

[0098] In embodiments, a single Transmission 120 is configured to control more than one Angle 650 between Joints 640. For example, in FIG. 6, Transmission 120A can be configured to flex both Joint 640B and Joint 640C. In this case, the Angles 650B and 650C are not deterministic functions of the state of Transmission 120A. The ratio between Angle 650B and Angle 650C can dependent on forces experienced by Distal Segment 630C and Medial Segment 630B. For example, if Medial Segment 630B comes in contact with an Object 660, then further movement of Transmission 120A will result in more flexing at Joint 640C relative to Joint 640B. The forces on Segments 630 can occur from a wide variety of sources, including the weight of a held Object 660, touching Object 660, softness or malleability of Object 660, and/or the like.

[0099] Digits 610 optionally include one or more Sensors 670. Sensors 670 may be configured to detect relationships between Digits 610 and other object within the environment of Robotic System 100. Sensors 670 can include photo sensors, Camera 150, force/pressure sensors, chemical sensors, electrical sensors, temperature sensors, and/or any other known type of sensor. Sensors 670 can be included on any part of Digits 610. Outputs of Sensors 670 are optionally provided to Neural Network 160 for use in generating movement command signals.

[00100]

[00101] FIG. 7 illustrates method of controlling a robotic joint, according to various embodiments of the invention. In these methods, the position of Digit 610 is not necessarily determinable from the state of one or more Transmissions 120 configured to flex Digit 610. This non-deterministic position can result when a single Transmission 120 is used to flex more than one Joint 640. Non-deterministic positions of Digit 610 can also result from changes in the length of Transmissions 120, from lag or play in the movement of Transmissions 120, from changes in elasticity of Transmissions 120, from weight of object manipulated by Digits 610, from outside forces, and/or the like. The illustrated methods include the use of images, and optionally other sensor data, to compensate command signals for an actual position of End Effector 140. These methods are optionally performed using Robotic System 100, and the steps may be performed in alternative orders.

[00102] In a Flex Step 710, Digit 610 is flexed by using Movement Generation Device 110 to move Transmission 120A or Transmission 120B. This flexion results in movement of at least two of Segments 630, e.g., Segment 630B and Segment 630C. These movements may further result in changes in at least one or at least two of Angles 640, e.g., Angle 640A; Angle 640A and 640C; Angle 640B and 640C; or Angle 640A and Angle 640B.

[00103] For the various reasons discussed elsewhere herein, the relative changes in the at least two of Angles 640 may not be determinable based on a magnitude of the movement of Transmission 120A, movement of Transmission 120B, and/or action of Movement Generation Device 110. For example, in Flex Step 710, one or more of Segments 630 may come in contact with Object 660 or experience some other external force. This can result in variations in the relative magnitude by which each of Angles 640 changes, variations in the relative movements of Joints 640 and, thus, variations in the relative movements of the Segments 630 within Digit 610.

[00104] In a Capture Image Step 720, one or more Camera 150 is used to capture an image of End Effector 140. Capture Image Step 720 optionally includes receiving signals from any of the other types of sensors discussed herein.

[00105] In an optional Select Step 730, long-term memory data is selected from Memory Storage 180. The selection can include an initial analysis of the image captured in Capture Image Step 720 using Neural Network 160. For example, initial analysis to determine characteristics of an object being manipulated. The long-term memory data may be selected based on a goal or task, and/or on characteristics (type, weight, position, size, shape, etc.) of the identified object. Neural Network 160 is optionally configured to make this selection based on the image. In one example, a particular heavy object is identified and long-term memory data configured for better manipulation of a heavy object is selected in response. In another example, a particular heavy object is identified and a neural network, neural network parameters (e.g., weightings), and/or a neural network configuration is selected in response. In one example, a particularly delicate object is identified and long-term memory data configured for better manipulation of a delicate object is selected in response. In one example, a task for the movement of End Effector 140 with unusually high precision is received and long-term memory data configured to achieve this high precision is selected in response. Select Step 730, and/or other steps illustrated in FIG. 7, are optionally included in the methods illustrated in FIG. 5.

[00106] In a Generate Step 740, command signals configured to move one or more of Segments 630 to desired location(s) are generated. These command signals are typically generated by using Neural Network 160 to process the image generated in Capture Image Step 720. The command signals may also be generated based on specific goals, tasks, and/or other sensor data as described herein. Further, Generate Step 740 optionally includes the use of long-term memory data, selected in Select Step 730, to alter the state and operation of Neural Network 160. For example, this data may be provided as input to specific nodes of Neural Network 160 in order to affect the generation of command signals at these nodes and other downstream nodes dependent on the input. The generated command signals may be configured to move any of Segments 630 within one or more Digits 610 of End Effector 140.

[00107] In a Compensate Step 750, the command signals generated within Neural Network 160 are compensated based on responses to prior command signals as indicated in the captured image and/or other sensor data. For example, the compensation may be based on the movement resulting from moving Transmission 120A and/or Transmission 120B in Flex Step 710. Compensate Step 750 is optionally performed using Compensation Block 330, as described elsewhere herein. Compensate Step 750 is optionally responsive to both long-term memory data retrieved from Memory Storage 180 and a short-term memory resulting from recurrent or memory enabled nodes of Neural Network 160. Dependence on long-term memory allows the compensation to be responsive to images captured at different times, e.g., minutes, hours, days or longer periods apart.

[00108] In various embodiments, Generate Step 740 and Compensate Step 750 are performed as a single step, with no discrete demarcation between generation and compensation of the command signals. Discrete uncompensated command signals need not be generated as an output of Neural Network 160. . These un-compensated command signals may then be compensated and/or further compensated by later nodes of Neural Network 160. The un-compensated signals may or may not be directly useable to control Movement Generating Devices 110.

[00109] In a Move 1st Transmission Step 760, one of Transmissions 120 are moved using an instance of Movement Generation Device 110, in response to the compensated command signals. In an optional Move 2nd Transmission Step 770, another of Transmissions 120 is moved using an instance of Movement Generation Device 110, in response to the compensated command signals. Steps 760 and 770 may be performed in parallel or in series. Each step results in movement of Digits 610 and can include moving different combinations of Segments 630. In some embodiments, Move 1st Transmission Step 760 and Move 2nd Transmission Step 770 can include pulling on a tendon embodiment of Transmissions 120. The amount of movement of a particular Segment 630 may be dependent on contact between Digits 610 and one or more Object 660.

[00110] In an optional Extend Step 780, one or more of Digits 610 is extended. The extension is optionally performed using an opposing transmission (e.g., tendon) and/or an elastic element as

discussed elsewhere herein. Extend Step 780 may include release of one or more Transmission 120 using Movement Generation Device 110.

[00111] Several embodiments are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations are covered by the above teachings and within the scope of the appended claims without departing from the spirit and intended scope thereof. For example, the systems and methods discussed herein can be applied to an exoskeleton, a prosthetic device, a vehicle, and/or a system configured to interact with a human. For example, Robotic System 100 could be configured to hand an object to a person, or to control a prosthetic limb. While the examples provided herein are focused on “images” collected by a camera, these described systems may be configured to operate using any type of sensor data, e.g., data generated by a strain gauge, a pressure gauge, a medical sensor, a chemical sensor, radar, ultrasound, and/or any other sensor type discussed herein. The Transmissions 120 discussed herein may be substituted for or include other movement coupling components such as tendons, cables, fibers, encoders, gears, cams, shafts, levers, belts, pulleys, chains, and/or the like.

[00112] The embodiments discussed herein are illustrative of the present invention. As these embodiments of the present invention are described with reference to illustrations, various modifications or adaptations of the methods and or specific structures described may become apparent to those skilled in the art. All such modifications, adaptations, or variations that rely upon the teachings of the present invention, and through which these teachings have advanced the art, are considered to be within the spirit and scope of the present invention. Hence, these descriptions and drawings should not be considered in a limiting sense, as it is understood that the present invention is in no way limited to only the embodiments illustrated.

[00113] Computing systems referred to herein can comprise an integrated circuit, a microprocessor, a personal computer, a server, a distributed computing system, a communication device, a network device, or the like, and various combinations of the same. A computing system may also comprise volatile and/or non-volatile memory such as random access memory (RAM), dynamic random access

memory (DRAM), static random access memory (SRAM), magnetic media, optical media, nano-media, a hard drive, a compact disk, a digital versatile disc (DVD), and/or other devices configured for storing analog or digital information, such as in a database. The various examples of logic noted above can comprise hardware, firmware, or software stored on a computer-readable medium, or combinations thereof. A computer-readable medium, as used herein, expressly excludes paper. Computer-implemented steps of the methods noted herein can comprise a set of instructions stored on a computer-readable medium that when executed cause the computing system to perform the steps. A computing system programmed to perform particular functions pursuant to instructions from program software is a special purpose computing system for performing those particular functions. Data that is manipulated by a special purpose computing system while performing those particular functions is at least electronically saved in buffers of the computing system, physically changing the special purpose computing system from one state to the next with each change to the stored data.