Shared-Control Teleoperation Paradigms on a Soft-Growing Robot Manipulator

Fabio Stroppa[1*], Mario Selvaggio[2], Nathaniel Agharese[3], Ming Luo[4], Laura H. Blumenschein[5], Elliot W. Hawkes[6] and Allison M. Okamura[3]

[1*]Computer Engineering Department, Kadir Has University, İstanbul, 34083, Turkey.
[2]Electrical Engineering and Information Technology Department, University of Naples Federico II, Naples, 80125, Italy.
[3]Mechanical Engineering Department, Stanford University, Stanford, 94305, CA, USA.
[4]School of Mechanical and Materials Engineering, Washington State University, Pullman, 99163, WA, USA.
[5]School of Mechanical Engineering, Purdue University, West Lafayette, 47907, IN, USA.
[6]Mechanical Engineering Department, UC Santa Barbara, Santa Barbara, 93105, CA, USA.

*Corresponding author(s). E-mail(s): fabio.stroppa@khas.edu.tr;
Contributing authors: mario.selvaggio@unina.it;
agharese@stanford.edu; ming.luo@wsu.edu;
lhblumen@purdue.edu; ewhawkes@ucsb.edu;
aokamura@stanford.edu;

Abstract

Semi-autonomous telerobotic systems allow both humans and robots to exploit their strengths while enabling personalized execution of a remote task. For soft robots with kinematic structures dissimilar to those of human operators, it is unknown how the allocation of control between the human and the robot changes the performance. This work presents a set of interaction paradigms between a human and a remote

soft-growing robot manipulator, with demonstrations in both real and simulated scenarios. The soft robot can grow and retract by eversion and inversion of its tubular body, a property we exploit in the interaction paradigms. We implemented and tested six different human-robot interaction paradigms, with full teleoperation at one extreme and gradually adding autonomy to various aspects of the task execution. All paradigms are demonstrated by two experts and two naive operators. Results show that humans and the soft robot manipulator can effectively split their control along different degrees of freedom while acting simultaneously to accomplish a task. In the simple pick-and-place task studied in this work, performance improves as the control is gradually given to the robot's autonomy, especially when the robot can correct certain human errors. However, human engagement is maximized when the control over a task is at least partially shared. Finally, when the human operator is assisted by haptic guidance, which is computed based on soft robot tip position errors, we observed that the improvement in performance is dependent on the expertise of the human operator.

1 Introduction

Robots in domestic environments, especially soft robots, have the potential to both autonomously assist humans and enable safe physical presence for remote operators [1]. The spectrum of potential operation modalities in between full autonomy and direct human operation can provide robotic systems with a rich and useful set of perception and manipulation capabilities; however, its concrete realization also creates challenges for system designs and the development of their intuitive human-in-the-loop control [2–5]. This is especially true given the challenges inherent with soft robots, where humans may have advantages in understanding and controlling the underactuated robot state. Shared control denotes operation modalities of a robotic system that serve to balance inaccuracies of both human and artificial-intelligent agents, such that they can benefit from each one's abilities [6, 7]. Employing shared control becomes essential when humans are needed to address difficult-to-automate sub-tasks, yet some autonomy is needed to improve speed and/or performance and to reduce the physical/cognitive burden on human operators. For instance, most previous studies on shared-control telerobotics rely on human intelligence to solve low-level perceptual sub-tasks that are difficult for autonomous robots,
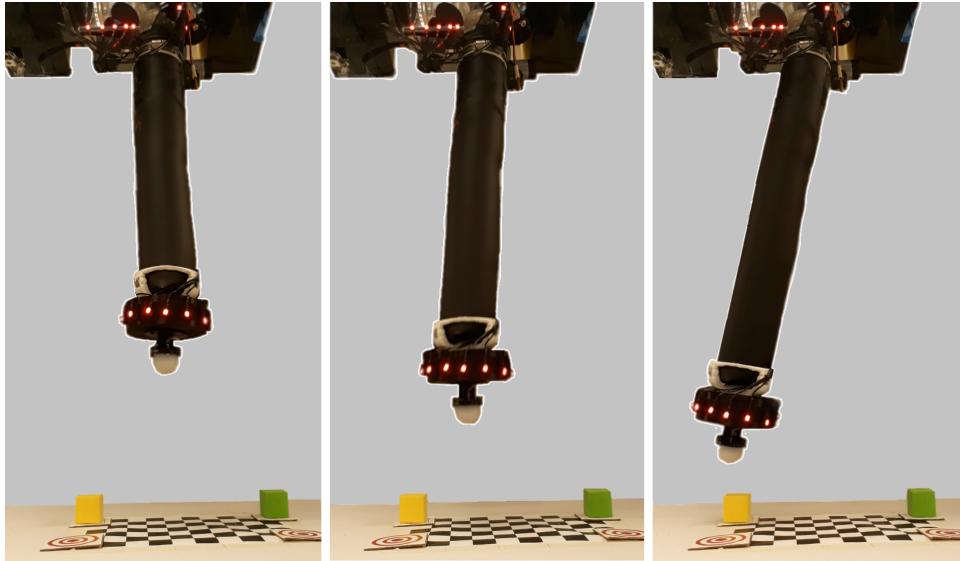
Fig. 1 The soft-growing robot manipulator can extend from a base container by growing its body and steering it towards items in the workspace to perform pick-and-place tasks by means of a soft-magnetic gripper mounted at its tip.

allocating the sub-task control either to the human or to the robot control system, or partially to both [8–11].

In this work, we present the design and the demonstration of six shared-control paradigms that can be used by human operators to remotely interact with a robotic manipulator during a pick-and-place task. As a pilot investigation of novel shared-control paradigms, we deployed them in a straightforward scenario to understand the contributions of human and artificial agents. We were interested in scenarios in which the human perspective is distorted such that the robot could provide assistance. During remote teleoperation, humans operate a device that does not align with the representation of their body, differently from a scenario with wearable exoskeletons where the human perspective is unchanged. This reveals the extent to which autonomous features change the operator's experience and impact performance. We establish how humans can effectively interact with soft-growing robotic manipulators and which interaction paradigms (and specific aspects of those paradigms) are subjectively preferred. These developments can be extended to work in more complex scenarios in which the presence of the human is a requirement and the robot can either assist or be assisted by the human (e.g., exploration of cluttered or dangerous environments in which artificial perception is limited).

Besides sub-task allocation, the proposed shared-control paradigms feature different assistance levels (in the form of autonomy and/or haptic guidance) to help the operator throughout the task execution. We consider assistance in the form of (i) guidance force rendered on a haptic device, opportunely tuned based on the abilities of the operator, and (ii) a division of the robot's degrees of freedom (DoFs) between human and artificial/autonomous control. These

forms of assistance are purposely designed to be suited when applied to a pick-and-place task and are independent of the employed robotic manipulator. We chose to demonstrate how these methods perform using a soft-growing robotic system (Fig. 1) with the aim of showing how they can be applied to this novel form robotic manipulator.

The use of shared-control teleoperation on a soft-growing robot manipulator has not been investigated yet to the best of our knowledge. Two of our proposed shared-control paradigms are novel and specifically designed for soft-growing manipulators, as they explicitly account for the shared actuation of unique DoFs. Unlike conventional rigid-body robots, the unique kinematics and compliance of *soft-growing robots* have the potential to combine computational and embodied intelligence to facilitate manipulation in complex, unknown environments; and they also offer a safer and effective solution in environments with humans [12, 13]. They imitate plant-like growth to change length and navigate environments, being able to grow and shrink through *eversion* of thin-walled material at the tip [14–17]. Their high portability makes them great candidates for applications where rigid robots cannot be employed due to the soft-growing robot properties of low inertia, low weight, and the ability to be fully stowed in small containers when retracted. They have been employed to explore an archaeological site [13], deploy reconfigurable structures [18], navigate through coral reefs [19], deliver tools [14], and burrow through granular media [20]. However, their dynamic properties and non-anthropomorphic motion capabilities demand novel remote control paradigms for such scenarios [5, 21].

2 Background on Shared Control

The term *shared control* generally refers to a scenario in which both human and robotic agents work (sometimes remotely) together to accomplish a common task. Broadly it is the spectrum of possible interactions between humans and robots, from robots having full autonomy to none at all [22]. The development of shared-control techniques must account for several aspects such as the type of interaction, the type of task being accomplished, the forms of user feedback, etc. This results in the definition of semi-autonomous control strategies with different and possibly varying levels of autonomy and user feedback [23]. Full autonomy still poses a problem for robotic systems when dealing with unknown or complex tasks in unstructured and uncertain scenarios [24]; on the other hand, shared control has been demonstrated to improve the task performance without placing the entire burden on the human operator [25]. Several works about shared control focus on the extent of human intervention in the control of artificial systems, conceding a certain amount of responsibility to each agent in the scene so as to split the control burden among all the participants [26]. The extent of human intervention, and thus robot autonomy, has been usually classified into discrete levels [9, 11, 27], with fewer studies considering a continuous domain [28, 29]. Commonly, shared-control techniques

aim to fully or partially replace a function, like identifying objects in cluttered environments [8], while others start from an autonomous robot autonomously and give control to the user only in difficult situations [9–11].

In a manipulation scenario, shared control has been used to decrease the cognitive and physical workload of the human operator, with the robot autonomously carrying out a set of secondary tasks [30]. Some works suggest that the robot can autonomously perform grasp when close to the goal [31], hold it over long time periods [32], or switch the control at some trigger [31, 33–35]. Sharing the control can also be seen as a means of collaborative interaction [3]. It can be interpreted as *assistance*, not only from the human to the robot but also vice versa. The latter is advantageous when the teleoperation is hindered by the inadequacies of the input control commands or by poor perception (e.g., remote teleoperation in which the point of view of the operator is not directly over the workspace). The robot may assist the user in accomplishing the desired task, making the teleoperation easier [36]. Therefore, the task of the robot becomes modulating the output rather than simply executing the operator's input. Some studies assist the operator by predicting their intent while selecting among different targets [36, 37], while others exploit haptic feedback/guidance techniques while moving toward a specific target [38–42]. These types of haptic assistance enforce constraints on the operator's position by applying controlled forces that are a function of the evolving behavior of the task or system [43]. Alternatively, paradigms such as *assist-as-needed* constrain the operators only when their behavior is in conflict with a known task [44–46].

3 System Description

This section describes the compliant robotic system used to study the proposed soft robot shared-control teleoperation paradigms for a pick-and-place manipulation task. The main components of the system are the soft-growing manipulator, both in real (Fig. 2(a)) and simulated scenarios (Fig. 2(b)), a kinesthetic haptic device, used to collect inputs and display haptic guidance to the operator, and two tracking systems based on motion capture and computer vision, that retrieve the robotic tip position and the items/targets of the pick-and-place task (in Cartesian pose), respectively. Figure 2(c) shows a block scheme of the system highlighting the interconnections among its modules.

The pick-and-place task consists of reaching, grasping, and placing two foam cubes in the corresponding target locations of the workspace. In the following, we will refer to the main components of the task as:

- *item*: the generic foam cube to be grasped and relocated;
- *target*: the place where the foam cube should be relocated; and
- *goal*: the current objective, which refers to a specific item when the gripper does not hold any, or to a target when the gripper is already holding an item.
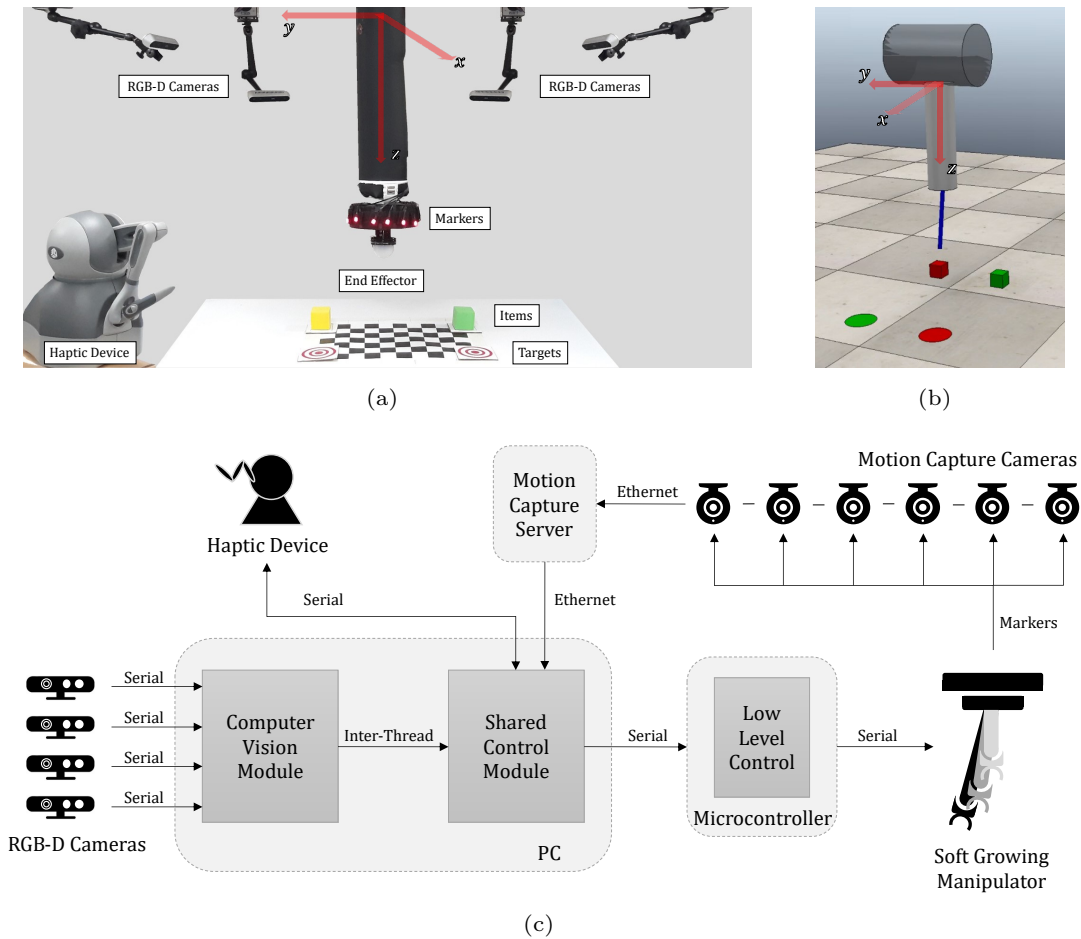
(a)   (b)

(c)

Fig. 2 (a) Picture of the real soft-growing robotic manipulation system. A coordinate system is attached to the robot base: the z axis is oriented along the gravity direction, while the x and y define the axes around which the robot steers at the base. As such, the soft-growing manipulator has 3 degrees of freedom, one associated with growth and two associated with steering. The items and targets associated with the pick-and-place task are laying on the table at the bottom. (b) Picture of the simulated soft-growing manipulator as well as items and targets. The robot and its reference system are oriented as in the real system (a). The blue vector represents the assistive force displayed to the user during assisted shared-control operations. (c) Block scheme of the overall system. The Shared Control and the Computer Vision modules run on a personal computer (PC), while Low-Level Control runs on a microcontroller connected to the manipulator. A Motion Capture system tracks the position of the manipulator while a set of RGB-D Cameras tracks the items in the environment. The human operator interfaces with the manipulator via the Haptic Device. Each connection between components indicates the type of communication (e.g., through an Ethernet cable) and whether it is one-way or two-way (single-ended or double-ended arrows, respectively).

3.1 Soft-Growing Manipulator

The robotic manipulation system consists of a soft-growing manipulator that can grow, retract, and steer in three dimensions (see Fig. 1) [13]. We used the same manipulator for our previous works on teleoperation [21]. Fig. 2(a) shows

a close-up of the manipulator alongside its reference frame: markers are placed on the end-effector to retrieve its Cartesian position.

The manipulator can carry a payload thanks to the soft-magnetic gripper attached to the end effector.The gripper can magnetically attract objects without requiring the precise positioning of the end-effector. We placed cylindrical magnets inside the gripper's silicone cover and inside the target items (with opposite polarity) so that the gripper can attract them when sufficiently close. To release items, we inflate the soft gripper such that the silicone rubber pushes the grasped item away from the magnet, increasing the distance between the two magnets until the attraction force is low enough to drop the item (Fig. 3).
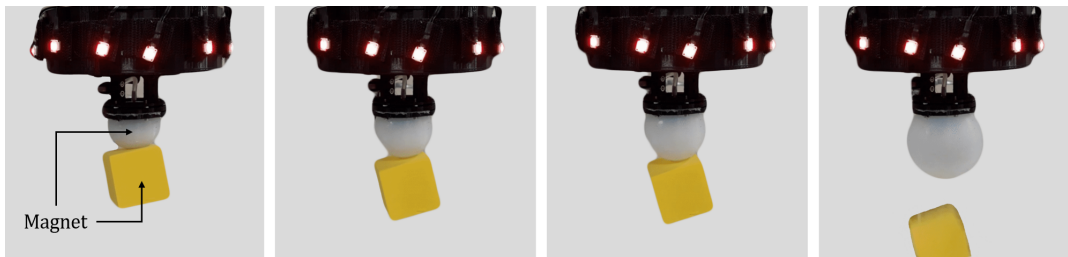
Fig. 3 Sequence of pictures showing the soft-magnetic gripper inflating and releasing an item. The gripper and the item both have a magnet inside, with opposite polarities. When the gripper is inflated, the distance between the two magnets increases until the attraction force is lower than the item's gravity and the item is dropped.

3.2 Haptic Device(s)

We used a Phantom Omni (now marketed as Geomagic Touch) to teleoperate the robot's end-effector and perform the task in the real scenario (as shown in Fig. 2(a)), and a Novint Falcon (Novint Technologies, Inc.) in the simulated one. These kinesthetic haptic devices provide force rendering up to 7 N along their linear directions. Linear displacements are used to command the manipulator's tip position in the real scenario and its tip velocity in simulation such that the haptic device's workspace is scaled to the position or velocity workspace. The haptic device is equipped with two buttons, which are used to inflate/deflate the gripper. The haptic devices are also used to render haptic forces to the user (see Sec. 4).

3.3 Tracking Modules

We used two different systems to track the manipulator's end-effector (see Sec. 3.3.1) and the items to be manipulated (see Sec. 3.3.2), based on motion capture and computer vision, respectively. Both tracking modules are transformed and expressed in the reference system depicted in Fig. 2(a) to align with the kinematics and control of the robot.

In our study, we used a controlled environment to focus on the human-machine interaction; however, the implemented motion capture system would not be available in all real-world scenarios. Methods are being developed to localize soft robots' tip position in the environment, such as using cameras placed in an eye-in-hand configuration [14, 47, 48] or shape sensors along the length [49].

3.3.1 Robot Tracking

We used the PhaseSpace Impulse X2E to track the manipulator's end-effector. This motion capture system allows real-time tracking at 270 Hz. The cameras are managed by an external server, which communicates with the main personal computer (PC) via ethernet connection (Fig. 2(c)). As shown in Fig. 2(a), eight markers are placed around the circumference of the gripper mount at the manipulator's end-effector and tracked by six cameras located on the ceiling (cameras are not shown in the picture). The position of the end-effector is given by the markers' centroid.

3.3.2 Item Tracking

Four RGB-D cameras (RealSense D415) surround the workspace and are used for item tracking. The data captured by the cameras (2D image and depth map for 3D reconstruction) is processed by a Computer Vision Module installed on the main PC (Fig. 2(c)). This module aligns the data retrieved by different cameras through automatic calibration of their extrinsic parameters (camera resectioning), such that each camera shares the same reference system. The calibration is independent of the cameras' locations, as they face the same reference system – a physical chessboard placed in the center of the workspace as shown in Fig. 2(a). Different items are identified by their color.

Although this method could potentially run in real time, we simplified the system by measuring the positions of the items only once before a trial and storing them as fixed values (when the manipulator approaches an item, it occludes the scene and the items cannot be tracked; this hinders the opportunity to identify whether the gripper is currently holding an item for full automation purposes.) Similarly, the static targets are identified only once, by positioning an item on it and storing that position as a target. Future works could expand this system to integrate state-of-the-art tracking and handle moving targets; for the purpose of our shared-control studies, we implemented the basic functionality required for evaluating the spectrum of teleoperation and quasi-autonomous operations.

3.4 Simulated Scenario

A soft-growing robot simulator was set up in CoppeliaSim [50] to develop and preliminary test our shared-control paradigms. The virtual environment is a digital twin of the real robotic system with items and targets placed in the same relative positions with respect to the manipulator (see Fig. 2(b)). The

manipulator was digitally modeled as a cylindrical body that can scale its length along its main axis to simulate eversion. When assistive paradigms are used, the haptic force rendered through the haptic devices is visually displayed to the user as a blue line segment at the robot tip as shown in Fig. 2(b).

The purpose of the simulation was to systematically tune the control parameters for the shared control paradigms featuring haptic assistance and demonstrate their performance in ideal conditions (see Sec. 5.3). Besides the unconventional kinematics of soft-growing robots, their motion fluidity is affected by several mechanical factors (actuation dynamics, friction, and external disturbances), which may have an influence on the effectiveness of shared control paradigms. Thus, we conducted a parameter-tuning simulation to establish the main effects and the interactions of the shared-control paradigm parameters in ideal simulated conditions. Then, we compared the simulated and the real system to demonstrate how these mechanical factors affect the effectiveness of our shared-control paradigms.

4 Shared-control Teleoperation Paradigms

In this section, we describe the six proposed shared-control teleoperation paradigms used to investigate the level of autonomy needed for a human to accomplish a pick-and-place task with a soft-growing robot manipulator. We included the case in which the control is not shared as a baseline for the comparison (i.e., direct teleoperation, in which the human is in full control of the system, and then built from there by gradually adding autonomy until the operator only has control over the gripper operation. The order in which the paradigms are presented follows increments in autonomy, as the control of more DoFs shifts from the human to the robot. Note that most of these paradigms are generic and can be applied to any kind of robotic manipulator that can be commanded in Cartesian space; however, two of them (see Sec. 4.4 and 4.5) are specific for soft robots in that they explicitly exploit eversion.

We will use the following terms to denote quantities measured with respect to the robot's base frame:

- $ee = \langle x, y, z \rangle \in \mathbb{R}^3$ is the current position of the manipulator's end-effector, coincident with the tip of the gripper;
- $d = \langle x, y, z \rangle \in \mathbb{R}^3$ is the desired position of the manipulator's end-effector (i.e., where the manipulator is commanded to go), coincident with the tip of the gripper;
- $c = \langle x, y, z \rangle \in \mathbb{R}^3$ is the command given by the haptic device that is to be mapped to d; and
- $g = \langle x, y, z \rangle \in \mathbb{R}^3$ is the position of the current goal.

The term d, that is the desired position of the manipulator's end-effector, is set to either c or g, or a combination of them, based on the type of shared control modality. We indicate with $m = \|g - ee\| \in \mathbb{R}_{\geq 0}$ the Euclidean distance between the goal and the end-effector at a current time and with $\dot{m} \in \mathbb{R}$ its
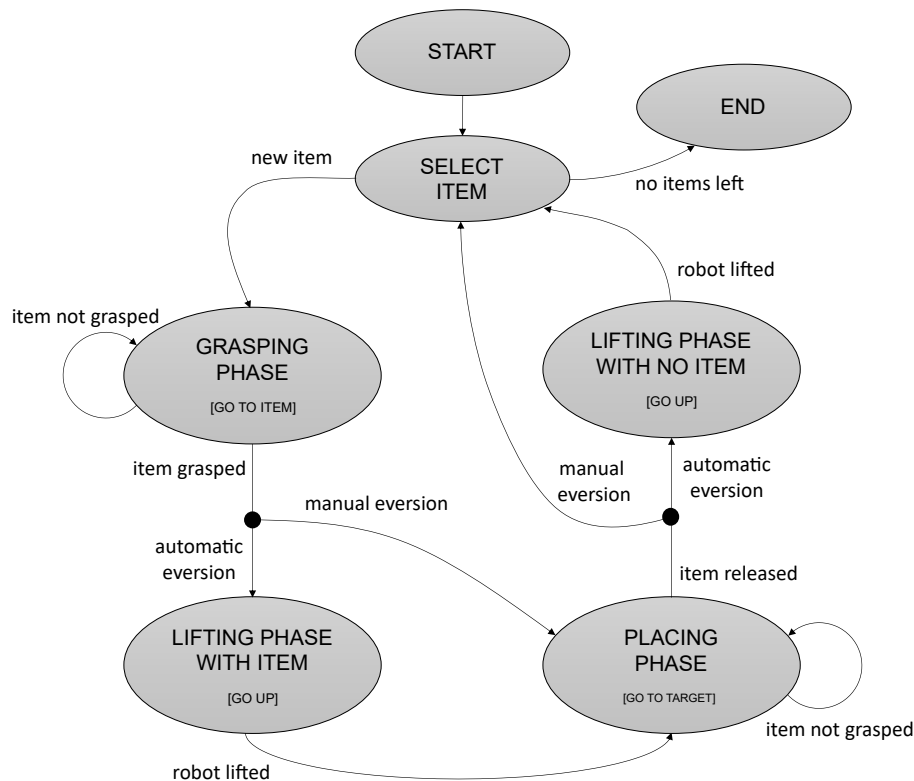
Fig. 4 State machine for the Goal Selection algorithm, used to monitor the evolution of the pick-and-place task and select the appropriate goal. The current goal can be either an object to grasp (item) or a desired position of placing (target), depending on whether the robot is currently grasping an item with the gripper. Additionally, if the eversion is automatically performed, the system will slightly lift the robot before moving on to the next goal, preventing the gripper to be dragged onto the ground.

time derivative, which is used to evaluate the motion performed by the robot. Inverse kinematics is used to compute growth and steering; whereas low-level control is used to bring *ee* towards *d*.

During the execution of a task, *g* changes based on whether the current goal is the position of an *item* or a *target*. To handle this procedure, we developed a *Goal Selection* algorithm performing the following steps: (i) select the item to be grasped (grasping phase); (ii) select the target where the current item will be placed (placing phase); and (iii) repeat until all the items are correctly placed in their respective target position (the order in which the items are selected is beyond the purpose of the algorithm.) The algorithm is implemented as a state machine, graphically represented in Fig. 4. Besides providing a set of locations that the manipulator is asked to reach, this algorithm allows the system to monitor the progress of the task. A further behavior enabled by the algorithm occurs when a goal is reached, and thus the position of the end-effector is supposedly near the ground: in such a case, it is desirable to lift the robot before moving towards the next goal, preventing problems such as dragging the tip over the ground, or hitting other items on the way. To handle this, the state machine introduces an intermediate goal: a proxy set above

the current end-effector position away from the ground, resulting in the robot being lifted before switching to the next goal (lifting phase). Note that this intermediate goal is generated only if the growth DoF is automated (as will be described in Sec. 4.5 and 4.6), because this is a location in the space that the human operator would not visualize in a real environment. The features of the six shared-control teleoperation paradigms are preliminary summarized in Table 1.

Table 1 Feature summary of the shared-control teleoperation paradigms

	Steering	Eversion	Grasping operation (Gripper open/close)	Grasping detection (Item grasped yes/no)	Haptic Guidance
Full Teleoperation	manual	manual	manual	manual	no
AAN	manual	manual	manual	manual	yes
Fixed Assistance	manual	manual	manual	manual	yes
MSAE	manual	automatic	manual	manual	no
ASME	automatic	manual	manual	manual	no
Mostly Autonomy	automatic	automatic	manual	manual	no

Due to the grasping system described in Sec. 3.1, we used the following assumption: once an item is grasped, it will only be released on a target. If the item is erroneously dropped anywhere else, the trial is discarded; although due to the nature of the soft-magnetic gripper, this never occurred during our trials. This assumption overcomes a practical limitation of our system: we have no means to track whether an item is being currently grasped. Firstly, item tracking is not performed in real time, but only at the beginning of each trial, therefore the positions of items and targets do not change over time. Secondly, since the soft-magnetic gripper may require variable time to release an item, we cannot exactly estimate when this happens without having a sensor in the gripper - which we did not include. Thanks to the aforementioned assumption, we decided that in our implementation the human is always in charge of controlling the grasping operation (whether the gripper should be 'opened' or 'closed') and of performing grasp detection (whether the gripper is currently holding an item or not). The operator can trigger the opening/closing of the gripper by pressing a button on the haptic device and indicates to the system that the item has been grasped/released by pressing a second button on the haptic device.

4.1 Full Teleoperation

In this paradigm, the human operator has full control over the manipulator, in the sense that all the Cartesian displacements of the end-effector are commanded through the haptic device commands c. Thus, this paradigm is realized by setting:

$$d = c \tag{1}$$

This paradigm is considered a baseline on which the other paradigms are built, but some could also consider it as shared control in that closed-loop control is the contribution of the robotic system.

4.2 Assist-as-Needed (AAN)

In this paradigm, the human operator still has full control over the manipulator while occasionally being assisted by a force rendered by the haptic device. The Assist-as-Needed paradigm is implemented such that the system monitors the movements of the operator, provides assistance according to this: if the operator is correctly moving toward the goal, then no assistance is needed, and any force previously generated is gradually reduced. Alternatively, if the operator is performing poorly (e.g., not moving or going in the opposite target direction), then the algorithm generates a force that increases over time while the performance is not improving. With this mechanism, sufficient force to overcome physical or perceptual impairment is provided to the operator.

For this paradigm, we define $\vec{f} \in \mathbb{R}^3$ as the haptic guidance force pushing towards the goal, expressed in N, whose value cannot exceed f_{MAX}. The direction (unit vector) of the assisting force $\hat{f} \in \mathbb{R}^3$ goes from the end-effector position to the goal. Its magnitude $|f|$ depends on k, a (variable) parameter denoting the stiffness modulating the force using Hooke's Law, whose value cannot exceed k_{MAX} (both force and stiffness are constrained). The parameter k is expressed in N/m, but it is ultimately dependent on the base unit selected for ee and g. The following equations are involved in the calculation of the haptic guidance force $\vec{f}$:

$$
\begin{aligned}
|f| &= \begin{cases} k \cdot m & \text{if} \quad < f_{MAX} \\ f_{MAX} & \text{otherwise} \end{cases} \\
\hat{f} &= \frac{g - ee}{m} \\
\vec{f} &= |f| \cdot \hat{f}
\end{aligned}
\tag{2}
$$

The procedure to modulate the stiffness parameter k according to the task performance is implemented as a state machine (see Fig. 5). Besides the starting state, there are four possible states in which the robot can be: (i) the operator has reached the goal ($m = 0$); (ii) the operator is steady ($\dot{m} = 0$); (iii) the operator is moving away from the goal ($\dot{m} > 0$); and (iv) the operator is getting closer to the goal ($\dot{m} < 0$). This strategy increases the value of k if the operator is steady or moving away from the goal, meaning that the robot should push in the direction of the goal as the operator is poorly teleoperating. On the contrary, the value of k will decrease if the operator is getting closer to the goal, as the robot is already moving in the right direction. In any case, between two consecutive iterations, k will change of at most an amount Δ, a fixed positive value expressed in N/m, which will be added or subtracted to k based on the state.
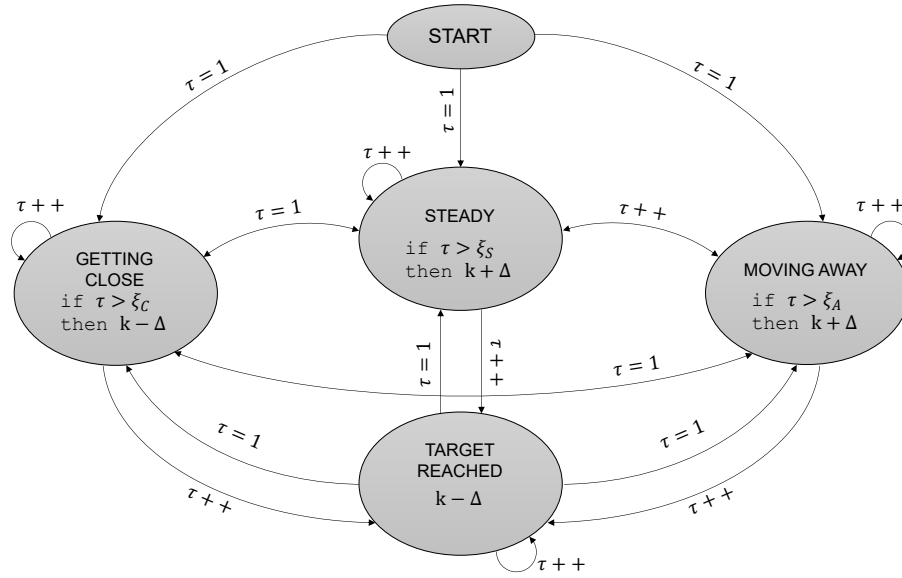
Fig. 5 State machine used to evaluate the stiffness k in the Assist-as-Needed shared-control teleoperation paradigm. The evaluation is based on the time elapsed in each state, monitored by the counter τ. Each transition between states will either increment or reset τ, and the value of k will change only when τ is greater than the limit ξ - e. g. after a certain amount of time spent in a certain state. The parameter ξ may have a single value or a different value for each state.

Since we want the operator to be in charge of the teleoperation, the logic is to wait for the operator to perform an action. This is implemented by the parameters ξ, a reaction time before the algorithm modifies the output, and τ, a counter to watch the reaction time, initialized whenever the state changes, and incremented by one at each iteration as long as the system remains the same state. Both the parameters are expressed in a number of iterations, which can be transformed to a time base unit when multiplied by the algorithm's sample rate. The reaction time could be a single parameter, or we could use a different value for each state. This is useful when we want to manage various situations differently: e.g., the algorithm will wait for a certain amount of time ξ_S before beginning to modify k when the end-effector is not moving, and a different amount of time ξ_C when the robot is moving towards the goal, etc.

The equation below shows the paradigm formulation and how the derivative of k changes based on the slope between two consecutive distances from the goal:

$$d = c$$

$$\dot{k} = \begin{cases} -\Delta & \text{if} \quad m = 0 \ \wedge \ k > 0 \\ \Delta & \text{if} \quad \dot{m} \geq 0 \ \wedge \ k < k_{MAX} \ \wedge \ \tau < \xi \\ -\Delta & \text{if} \quad \dot{m} < 0 \ \wedge \ k > 0 \ \wedge \ \tau < \xi \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

4.3 Fixed Assistance

In this paradigm, the human operator still controls the whole manipulator DoFs through the haptic device while being constantly assisted by $\vec{f} \in \mathbb{R}^3$, a force pushing towards the goal, expressed in N, whose value cannot exceed f_{MAX}. This is in contrast to the Assist-as-Needed paradigm, which only occasionally assists the human operator. The direction of the assisting force $\hat{f}$ goes from the position of the end-effector towards the goal; whereas its magnitude $|f|$ is given by Hooke's Law as shown in (4), in which: k is a fixed stiffness expressed in N/m, but it is ultimately dependent on the base unit selected for ee and g; and b is a fixed damper expressed in N·s/m, multiplied to the velocity of the end-effector $\dot{ee}$ (this velocity can refer to either the haptic device or the robot end-effector).

$$
\begin{aligned}
|f| &= \begin{cases} k \cdot m + b \cdot \dot{ee} & \text{if} \quad < f_{MAX} \\ f_{MAX} & \text{otherwise} \end{cases} \\
\hat{f} &= \frac{g - ee}{m} \\
\vec{f} &= |f| \cdot \hat{f}
\end{aligned}
\tag{4}
$$

The paradigm formulation is thus

$$
\begin{aligned}
d &= c \\
\dot{k} &= 0 \\
\dot{b} &= 0
\end{aligned}
\tag{5}
$$

To avoid sudden changes in the assisting force, a moving average filter is applied to $|f|$ over 50 samples at 100 Hz.

4.4 Manual Steering, Autonomous Eversion (MSAE)

In this paradigm, the operator only controls the steering DoFs (x and y coordinates of the reference frame), while the robot is in charge of everting towards the goal (z coordinate of the reference frame). The paradigm formulation is:

$$
d = \langle c_x, c_y, g_z \rangle
\tag{6}
$$

4.5 Autonomous Steering, Manual Eversion (ASME)

In this paradigm, the operator only controls the eversion DoF (z coordinate of the reference frame), while the robot is in charge of steering towards the goal (x and y coordinates of the reference frame). The paradigm formulation is:

$$
d = \langle g_x, g_y, c_z \rangle
\tag{7}
$$

We consider *ASME* as a paradigm that contains more autonomy than *MSAE* because eversion presents fewer DoF than steering.

4.6 Mostly Autonomous

In this paradigm, the operator has no control over the motion, and the manipulator's behavior is fully autonomous. However, as previously stated, the grasping operation and detection are still controlled by the operator. The execution of the task relies completely on the Goal Selection algorithm and the paradigm formulation is:

$$d = \langle g_x, g_y, g_z \rangle \tag{8}$$

5 Demonstrations and Results

In this section, we demonstrate the human operator teleoperation control of the soft-growing manipulation system and observe the performance and the human behavior under the different shared-control teleoperation paradigms introduced in Sec. 4, while completing a pick-and-place manipulation task, described in Sec. 5.1. We let all the paradigms be tested by an expert and by a naive user for both real and simulated scenarios, evaluated the metrics given in Sec. 5.2, and report the results in Sec. 5.5 along with their discussion. Besides being used for evaluation, the simulated scenario was also used to perform a systematic parameters tuning procedure (involving a large number of trials) for the assistive algorithms (AAN and Fixed Assistance). Results of this study are given in Sec. 5.3.

5.1 Task Description

During the execution of the task, the manipulator reaches for two items from above as shown in Fig. 2(c), and the operational workspace is a planar surface placed 700 mm below the manipulator's base. The same task is performed for each shared-control teleoperation paradigm.

We define the following phases for each item, as shown in the state machine diagram in Fig. 4:

- *grasping phase*, in which the manipulator's objective is to reach and grasp an item (item is the goal); and
- *placing phase*, in which the manipulator's objective is to reach a target and release the current item on it (target is the goal).

For the pick-and-place of two items, a trial is thus composed of two repeated grasping and placing phases, as depicted in Fig. 6. The image shows the exact progression of a trial, which will be consistent in each scenario tested during the demonstrations. This means that the order of item selection is fixed, such that both the human and the robot have the same reference for execution. We observed that the layout of the goals does not affect the performance of the robot, and we defined the layout such that the robot would have to move in a large area of the operational workspace.
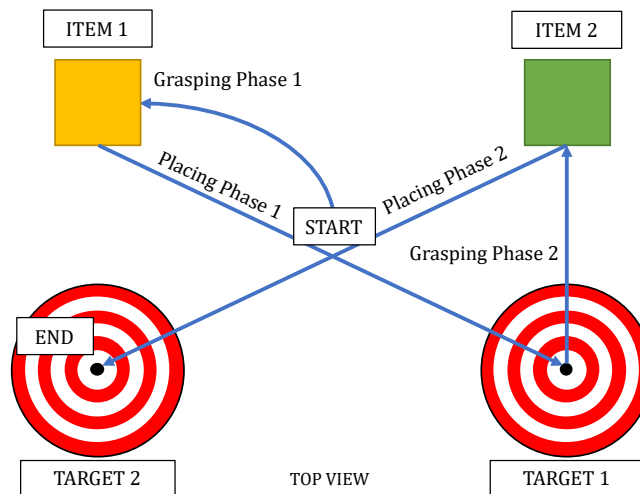
Fig. 6 Progression of the task as structured in the workspace of the robot. Each trial starts with the manipulator at the center of the operational workspace. The grasping and placing phases are executed for item 1 and then item 2. The task ends when item 2 is placed on target 2. Arrows are for illustration purposes only and do not represent real paths or trajectories. The distances between items and targets are approximately 30 cm, and are not shown to scale in this figure. Each item is a foam cube with a side length of approximately 3 cm.

5.2 Evaluation Metrics

The following metrics were used to evaluate task performance:

- *Item placement error*: the mean distance between the manipulator's end-effector and the target at the time of item release, projected on the x-y plane of the operational workspace (mm);
- *Execution time*: the time required to complete a trial (s); and
- *Average amount of assistance*: the average force rendered on the haptic device to assist the operator, only for the *Assist-as-Needed* and *Fixed Assistance* paradigms (N/iterations)

5.3 Parameter Tuning for Haptic Rendering

Assist-as-Needed and *Fixed Assistance* are the only paradigms whose parameters are directly affecting human behavior through haptic guidance. However, the assistive force rendered on the haptic device depends on several values as seen in Sec. 4.2 and 4.3, respectively. Thus, the effectiveness of these paradigms (both quantitative and qualitative) depends on the choice of parameters. For *Fixed Assistance* the choice can be straightforward, with only a few parameters to be set.

On the other hand, *Assist-as-Needed* features multiple parameters, whose influence on the task performance may not be determined a priori, requiring a systematic parameter tuning method before demonstrations. The considered parameters were the maximum amount of force rendered (f_{MAX}), the maximum amount of stiffness used to generate the force (k_{MAX}), the increment in stiffness for each loop (Δ), and the three reaction times to handle the cases of

the operator moving close to the goal (ξ_C), away from the goal (ξ_A), or neither (ξ_S). For each parameter, we defined two levels: `low` and `high`. Table 2 shows the numeric values for each parameter. We captured the aspects of the task performance with the following four performance metrics:

- $\mathcal{T}$ completion time [s]: evaluated as the time from the start to the end of the trial - coincident with the second block being placed in the corresponding target zone;
- $\mathcal{L}$ trajectory length [m]: evaluated as the integral of the robot-tip velocity norm over time as in 9;

$$\mathcal{L} = \int_0^{\mathcal{T}} \dot{e}e(t) \, \mathrm{d}t \tag{9}$$

- $\mathcal{H}$ amount of assistance [N]: evaluated as the integral of the haptic guidance force norm over time divided by the completion time as in 10; and

$$\mathcal{H} = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} \left\| \vec{f} \right\|(t) \, \mathrm{d}t \tag{10}$$

- $\mathcal{P}$ precision [m]: evaluated as the mean of the differences between the object and the target positions (projected into the horizontal plane) in the final configuration as in 11;

$$\mathcal{P} = \frac{1}{2} \sum_{i=1}^{2} \| t^{x,y} - p_i^{x,y} \| \tag{11}$$

We evaluated these metrics along a set of trials performed accounting for all the possible parameters' combinations. $2^6 = 64$ trials were needed to test all the possible combinations. Each trial was repeated 3 times for a total of 192 trials: due to this high number, we decided to perform the assessment in the simulated environment presented in Sec. 3.4.

We ran an n-way ANOVA analysis on the results to identify significant parameters ($p < 0.05$ was considered statistically significant). We report the following findings:

- setting f_{MAX} to `high` reduces the completion time of 18.184 s ($F(1, 2) = 146.078$, $p < 0.01$), reduces the trajectory length of 0.197 m ($F(1, 2) = 69.470$, $p = .014$), and increases the amount of assistance of 2.483 N ($F(1, 2) = 22.415$, $p = .042$);

Table 2 Parameters values used for the systematic parameter tuning procedure performed for the Assist-as-Needed paradigm

Parameter	f_{MAX}	k_{MAX}	Δ	ξ_S	ξ_C	ξ_A
`low`	3 N	50 N/m	2 s	1 s	1 s	1 s
`high`	7 N	100 N/m	5 s	3 s	3 s	3 s

- setting both ξ_A and ξ_S to `low` reduces the completion time of 21.58 s ($F = 66.595$, $p = 0.015$) and the trajectory length of 0.251 m ($F(1,2) = 90.909$, $p = 0.011$) and of 0.395 m ($F(1,2) = 279.946$, $p < 0.01$), respectively;
- when ξ_S is set to `high`, having k_{MAX} to `low` increases the amount of assistance provided to the operator of 9.237 N ($F = 34.317$, $p = 0.028$) - stronger and more impulsive forces will be rendered to the operator; and
- the two-way interaction between ξ_C and ξ_S ($F(1,2) = 44.046$, $p = .022$) shows a higher increase in the amount of assistance when ξ_C is set to `high` and ξ_S is `low` - with these settings, a human operator is generally haptic-guided for a longer period of time as the assistive force raises rapidly in a steady state and decreases slowly when getting close to the target; and
- the precision is not significantly affected by the choice of the factors/levels.

We used these findings to tune the parameters accounting for our design objectives: aiming to reduce both completion time and trajectory length while having a high amount of assistance. The analysis suggests to set $f_{MAX} = $ `high`, $\xi_A = $ `low`, $\xi_S = $ `low`, $k_{MAX} = $ `low`, $\xi_C = $ `high`. The parameter Δ is free to be picked. The final values we used in our setup are defined in Table 3.

5.4 Demonstrations

To compare the performance of our shared-control teleoperation paradigms we performed multiple demonstration trials. Data shown below were collected from four distinct individuals (who introduced variability in the trials) tele-operating the system. Two individuals performed the tasks in the simulated environment and the other two performed the tasks in the real environment. In each environment, one participant was a naive user who had no previous

Table 3 Final choice of the parameters for haptic rendering in assistive paradigms

Global	
refresh	100 Hz
f_{MAX}	7 N
Fixed Assistance	
k	10 N/m
b	0.1 Ns/m
f filter	50 samples
Assist-as-Needed	
Δ	20 Ns/m
th_D	30 mm
th_M	0.01 m/s
ξ_S	1 s
ξ_C	3 s
ξ_A	1 s
k_{MAX}	50 N/m

experience in remotely controlling our soft-growing manipulator, and the other participant was an expert user – one of the developers of the system. Each participant performed 5 trials of the two-item task for each of the 6 interaction paradigms, resulting in a total of 30 trials for each participant. The order of trials was randomized to prevent naive users from getting familiar with the control system as the demonstrations were carried out. We did not observe trends in performance (i.e., learning curves) during 5 trials, although in pilot testing we found that learning curves were occasionally evident for naive users after 6 trials. Because we aimed to evaluate interaction paradigms for naive users before they had substantial experience, we did not go beyond 5 trials. Due to the substantial experience of the expert users while developing and tuning the simulated and real systems, their learning curves were assumed to have plateaued before the demonstration began.

5.5 Results and Discussion

The results from the demonstration trials are presented for each metric, showing average, median, interquartile range with outliers, and max/min across all trials for each paradigm. Charts and plots are shown in Fig. 7, 8 and 9.

As in any remotely operated robotic system, the presence of time delay can affect system stability. We did not experience any instability during our experiments, but our paradigms could be combined with control strategies to address this (e.g., passivity-based control [51]).

5.5.1 Item Placement Error and Execution Time

We compared item-placement error (Fig. 7(a)) and execution time (Fig. 7(b)) against each other (Fig. 8) to emphasize performance distribution. The trend among all the paradigms is consistent: the best performances are achieved when the operator's control is limited.As expected, there is a visible difference between expert and naive users.

Full Teleoperation and *Assist-as-Needed* prioritize human control, where the operator's lack of perception is visible from the performance achieved in task execution. This is noticeable in both the simulated and real scenario: in the first, 3D environments are difficult to navigate without stereoscopy or an extensive implementation of lightning and shadows; in the second, the distance between operator and device can proportionally affect perception [52].

It is not surprising that *Mostly Autonomous* achieved the shortest and most accurate trials, as the system always knows where the goal is. This trend is visible in the other semi-autonomous paradigms to different degrees, allowing for a shorter time than in *Full Teleoperation* – thus reducing the impairment caused by limited human perception. The values of item placement error obtained in *Mostly Autonomous* are non-zero because the human operator still has full control over when and where to release items: higher values were obtained when the item was released while the manipulator was still approaching the target, both in simulated and real cases. Furthermore, the item-placing precision is
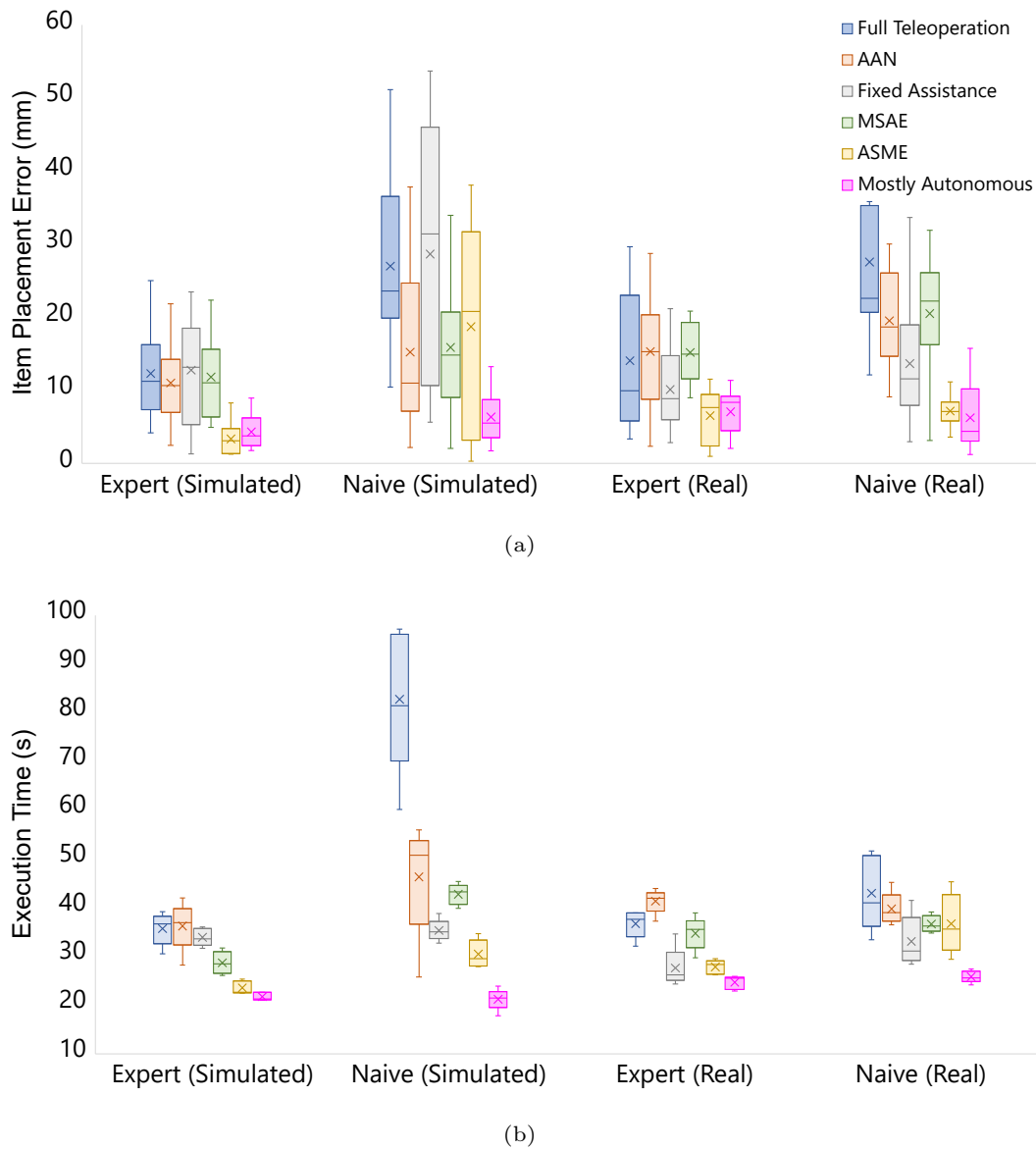
(a)

(b)

Fig. 7 Results of different metrics for each participant. (a) Error in item placement. (b) Time of execution of the task.

dependent on several factors (e.g., release time, amount of pressure used to inflate the gripper, how the item is oriented when attracted to the gripper) such that, even when the system is automated, it is not entirely repeatable. Thus, the human operator plays an important role in manually counterbalancing undesired behaviors, emphasizing the importance of having the human in the loop.

Interestingly, *MSAE* and *ASME* do not show the same performance, with the latter being better in almost all cases. This finding might suggest that controlling steering is harder than controlling eversion: steering presents more challenges in actuation and control, as it is composed of more DoFs than eversion. We hypothesize that the entity (human or robot) controlling the steering
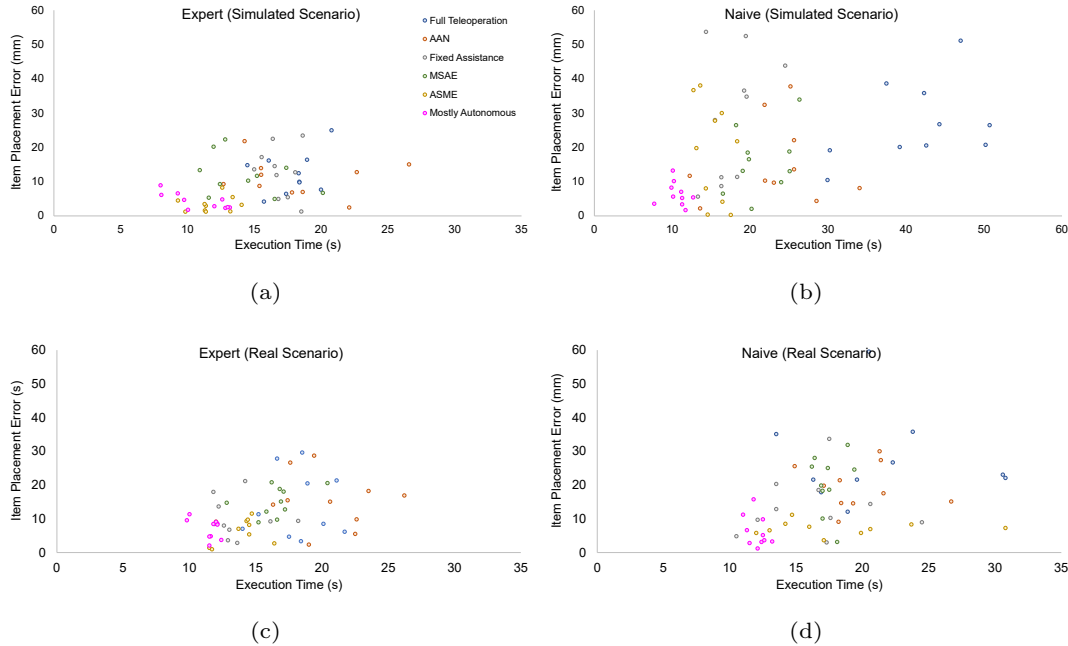
Fig. 8 Error in item placement versus execution time for each paradigm, executed by both expert and naive users for both simulated and real scenarios. Here the time refers to the duration of the execution of grasping and placing phases of a single item.

has more control over the system, and as such *ASME* is a more automated paradigm than *MSAE*.

Fixed Assistance is similar to *MSAE*, with results in the same range for expert users, and slightly worse for naive ones. This suggests that naive users might be deceived by constant haptic feedback (Fig. 9), which can be hard to accommodate when the operator is not used to it. Indeed, naive users performed slightly better in *Assist-as-Needed* than in *Fixed Assistance*, especially in the simulated scenario.
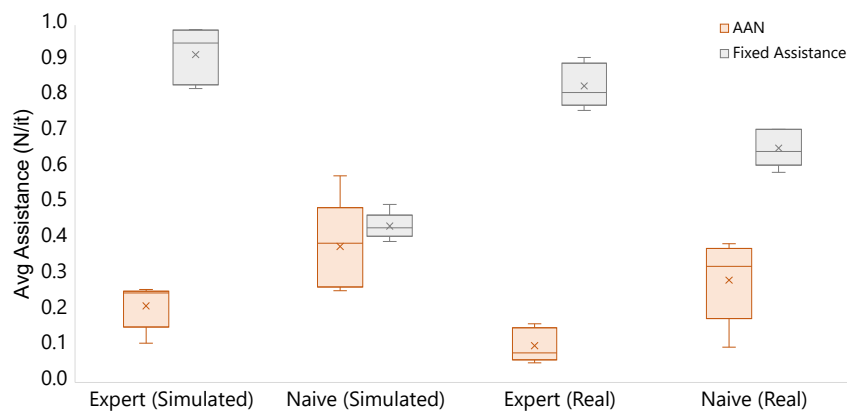

Fig. 9 Average amount of assistance rendered on the haptic interface in the assistive paradigms, executed by both expert and naive user in the simulated and the real scenario.

Finally, little difference was found between *Full Teleoperation* and *Assist-as-Needed* for experts. As shown in Fig. 9, the level of haptic assistance for *Assist-as-Needed* was barely perceivable because expert users did not need any assistance – they were highly experienced in teleoperating the soft-growing manipulator. This finding is also observed in Fig. 8: data for *Full Teleoperation* and *Assist-as-Needed* performance are similar for expert users (Fig. 8(a) and 8(c)), but different for naive users, with *Assist-as-Needed* resulting in slightly better performance (Fig. 8(b) and 8(d)). The difference between expert and naive users in the *Assist-as-Needed* paradigm shows that the method works as intended: the assistance is given and perceived only when the task was not performed correctly; expert users did not need it, and thus their results were not different from the *Full Teleoperation* case, whereas naive users occasionally required more guidance.

5.5.2 Participant's Opinions

In addition to the numerical results, we also considered how the task execution was perceived by the operators in the different interaction paradigms. Participants were asked questions to evaluate their role during the tasks. All the participants agreed that the task was difficult to accomplish under *Full Teleoperation*, especially in phases requiring accuracy such as when approaching targets. *Assist-as-Needed* was perceived differently, with naive users being correctly relieved by occasional assistance and experts barely noticing any. The opposite reaction was reported for *Fixed Assistance*, where experts found it a useful tool for guidance while naive users found it too constraining and hard to control. *MSAE* and *ASME* were both rated better than *Full Teleoperation* and described as the most collaborative paradigms in terms of interaction with the robot. *ASME* was rated easier to control but somewhat less interactive. Finally, *Mostly Autonomous* was rated as the easiest by all participants, enjoyable to observe but not very compelling in terms of interaction.

5.5.3 Paradigms for Soft-Growing Robots

The differences in performance between the *MSAE* (with the user controlling steering) and *ASME* (with the user controlling growth) paradigms suggest that the soft-growing robot's unique decoupling of DoFs can enable unique combinations of manual and autonomous control, expanding the options for shared control. In our results, *ASME* performed generally better than *MSAE*, allowing users to remain in the control loop of the robot with the possibility for increased situational awareness, while still allowing the autonomous algorithms to handle more challenging components of the task. These results suggest that sharing the control with semi-autonomous soft-growing robots increases the performance of manipulation tasks, not only in terms of the analyzed metrics but also in terms of maneuverability perceived by the operator.

6 Conclusion

In this paper, we introduced, validated, and compared the performance of six shared-control teleoperation paradigms with different autonomy and assistance levels on a soft-growing robot manipulator. The proposed paradigms were designed so as to explore a wide spectrum of potential operation modes going from full teleoperation to quasi-full autonomy, gradually reducing the role of the human operator in favor of robot autonomy. Results show that, by gradually adding autonomy to basic teleoperation, the performance improves to the detriment of human involvement. We noticed that having constant haptic guidance is a better paradigm for expert users, whereas having need-tailored guidance is preferred when users are new to the task. Having the human in control of partial DoFs might be the best solution in terms of perceived involvement and performance, whereas limiting the human control is useful when the task requires extreme precision and the system is capable of acquiring enough information from the environment to perform a correct execution. Wherever achieving this precision is not possible, the role of the human is to fine-tune position when close to targets, leaving the navigational part to the robot.

In the future, additional interaction paradigms can be developed and tested based on novel features of not only soft-growing robots but other soft robot manipulators with novel kinematics. These can be blended together to obtain a mixture of shared-control paradigms, which would lead to a continuous rather than a discrete set of control modalities that could be desirable in some applications. In this way, the user could select from a continuous range the percentage of manual versus autonomous control along each DoF of the system. For instance, considering a linear blending strategy, a real parameter can be used to weigh the control inputs coming from an autonomous controller and the human or from two different controllers along the same DoF. Blending and transitioning among control strategies could result in unexpected emergent behaviors of the system that affect usability, and should be studied carefully in future work. Additionally, we are interested in developing new paradigms exploiting the softness of our manipulator, allowing for collaborative tasks in which robots are not harmful to humans or the environment in case of undesired collisions.

Finally, due to the COVID-19 pandemic restricting human subjects' work as well as the very large number of conditions tested in the targeted user studies presented here, we had selected to demonstrate the system with only one experienced and one naive user in each scenario rather than running a complete user study. In future work, we will perform a larger user study to understand the effects of our interaction modes in scenarios that can be more complex than the one proposed in this pilot work (e.g., exploration of cluttered environments) and with further limitations (e.g., limited perception of the robot to emphasize the role of the human operator).

Supplementary information. The accompanying supplementary video shows simulated and real demonstrations carried out in this work, whereas the

accompanying supplementary document describes the details of the inverse kinematics and control of the soft-growing manipulator.

References

[1] de Graaf, M.M., Allouch, S.B., van Dijk, J.: Long-term acceptance of social robots in domestic environments: Insights from a user's perspective. In: AAAI Spring Symposia (2016)

[2] Scalise, R., Bisk, Y., Forbes, M., Yi, D., Choi, Y., Srinivasa, S.: Balancing shared autonomy with human-robot communication. arXiv preprint arXiv:1805.07719 (2018)

[3] Schilling, M., Kopp, S., Wachsmuth, S., Wrede, B., Ritter, H., Brox, T., Nebel, B., Burgard, W.: Towards a multidimensional perspective on shared autonomy. In: AAAI Fall Symposium Series, pp. 338–344 (2016)

[4] Coad, M.M., Thomasson, R.P., Blumenschein, L.H., Usevitch, N.S., Hawkes, E.W., Okamura, A.M.: Retraction of soft growing robots without buckling. IEEE Robotics & Automation Letters **5**(2), 2115–2122 (2020)

[5] El-Hussieny, H., Mehmood, U., Mehdi, Z., Jeong, S.-G., Usman, M., Hawkes, E.W., Okamura, A.M., Ryu, J.-H.: Development and evaluation of an intuitive flexible interface for teleoperating soft growing robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4995–5002 (2018)

[6] Mörtl, A., Lawitzky, M., Kucukyilmaz, A., Sezgin, M., Basdogan, C., Hirche, S.: The role of roles: Physical cooperation between humans and robots. The International Journal of Robotics Research **31**(13), 1656–1674 (2012)

[7] Corrales, J., Gomez, G.G., Torres, F., Perdereau, V.: Cooperative tasks between humans and robots in industrial environments. International Journal of Advanced Robotic Systems **9**(3), 94 (2012)

[8] Pitzer, B., Styer, M., Bersch, C., DuHadway, C., Becker, J.: Towards perceptual shared autonomy for robotic mobile manipulation. In: IEEE International Conference on Robotics and Automation, pp. 6245–6251 (2011)

[9] Kortenkamp, D., Keirn-Schreckenghost, D., Bonasso, R.P.: Adjustable control autonomy for manned space flight. In: IEEE Aerospace Conference (AeroConf), vol. 7, pp. 629–640 (2000)

[10] Sellner, B., Simmons, R., Singh, S.: User modelling for principled sliding autonomy in human-robot teams. In: Multi-Robot Systems. From Swarms

to Intelligent Automata vol. 3, pp. 197–208. Springer, ??? (2005)

[11] Dias, M.B., Kannan, B., Browning, B., Jones, E., Argall, B., Dias, M.F., Zinck, M., Veloso, M., Stentz, A.: Sliding autonomy for peer-to-peer human-robot teams. In: International Conference on Intelligent Autonomous Systems, pp. 332–341 (2008)

[12] Rieffel, J., Knox, D., Smith, S., Trimmer, B.: Growing and evolving soft robots. Artificial life **20**(1), 143–162 (2014)

[13] Coad, M.M., Blumenschein, L.H., Cutler, S., Zepeda, J.A.R., Naclerio, N.D., El-Hussieny, H., Mehmood, U., Ryu, J.-H., Hawkes, E.W., Okamura, A.M.: Vine robots: Design, teleoperation, and deployment for navigation and exploration. IEEE Robotics & Automation Magazine **27**(3), 120–132 (2020)

[14] Hawkes, E.W., Blumenschein, L.H., Greer, J.D., Okamura, A.M.: A soft robot that navigates its environment through growth. Science Robotics **2**(8), 3028 (2017)

[15] Sadeghi, A., Mondini, A., Mazzolai, B.: Toward self-growing soft robots inspired by plant roots and based on additive manufacturing technologies. Soft Robotics **4**(3), 211–223 (2017)

[16] Tsukagoshi, H., Arai, N., Kiryu, I., Kitagawa, A.: Tip growing actuator with the hose-like structure aiming for inspection on narrow terrain. International Journal of Automation Technology **5**(4), 516–522 (2011)

[17] Dehghani, H., Welch, C.R., Pourghodrat, A., Nelson, C.A., Oleynikov, D., Dasgupta, P., Terry, B.S.: Design and preliminary evaluation of a self-steering, pneumatically driven colonoscopy robot. Journal of Medical Engineering & Technology **41**(3), 223–236 (2017)

[18] Blumenschein, L.H., Gan, L.T., Fan, J.A., Okamura, A.M., Hawkes, E.W.: A tip-extending soft robot enables reconfigurable and deployable antennas. IEEE Robotics & Automation Letters **3**(2), 949–956 (2018)

[19] Luong, J., Glick, P., Ong, A., deVries, M.S., Sandin, S., Hawkes, E.W., Tolley, M.T.: Eversion and retraction of a soft robot towards the exploration of coral reefs. In: IEEE International Conference on Soft Robotics (RoboSoft), pp. 801–807 (2019)

[20] Naclerio, N.D., Hubicki, C.M., Aydin, Y.O., Goldman, D.I., Hawkes, E.W.: Soft robotic burrowing device with tip-extension and granular fluidization. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5918–5923 (2018). IEEE

[21] Stroppa, F., Luo, M., Yoshida, K., Coad, M.M., Blumenschein, L.H., Okamura, A.M.: Human interface for teleoperated object manipulation with a soft growing robot. In: IEEE International Conference on Robotics and Automation, pp. 726–732 (2020)

[22] Goodrich, M.A., Schultz, A.C.: Human-robot interaction: a survey. Foundations and Trends in Human-Computer Interaction **1**(3), 203–275 (2008)

[23] Selvaggio, M., Cognetti, M., Nikolaidis, S., Ivaldi, S., Siciliano, B.: Autonomy in physical human-robot interaction: A brief survey. IEEE Robotics and Automation Letters **6**(4), 7989–7996 (2021)

[24] Yang, G.-Z., Cambias, J., Cleary, K., Daimler, E., Drake, J., Dupont, P.E., Hata, N., Kazanzides, P., Martel, S., Patel, R.V., Santos, V.J., Taylor, R.H.: Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy. Science Robotics **2**(4), 8638 (2017)

[25] Kanda, T., Ishiguro, H.: Human-robot Interaction in Social Robotics. CRC Press, Boca Raton, Florida (2017)

[26] Schilling, M., Burgard, W., Muelling, K., Wrede, B., Ritter, H.: Shared autonomy—learning of joint action and human-robot collaboration. Frontiers in Neurorobotics **13**, 16 (2019)

[27] Bruemmer, D.J., Dudenhoeffer, D.D., Marble, J.L.: Dynamic-autonomy for urban search and rescue. In: AAAI Mobile Robot Competition, pp. 33–37 (2002)

[28] Desai, M., Yanco, H.A.: Blending human and robot inputs for sliding scale autonomy. In: IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), pp. 537–542 (2005)

[29] Anderson, S., Peters, S., Iagnemma, K., Overholt, J.: Semi-autonomous stability control and hazard avoidance for manned and unmanned ground vehicles. In: The 27th Army Science Conference, pp. 1–8 (2010)

[30] Selvaggio, M., Giordano, P.R., Ficuciello, F., Siciliano, B.: Passive task-prioritized shared-control teleoperation with haptic guidance. In: IEEE International Conference on Robotics and Automation, pp. 430–436 (2019)

[31] Kofman, J., Wu, X., Luu, T.J., Verma, S.: Teleoperation of a robot manipulator using a vision-based human-robot interface. IEEE Transactions on Industrial Electronics **52**(5), 1206–1219 (2005)

[32] Griffin, W.B., Provancher, W.R., Cutkosky, M.R.: Feedback strategies for telemanipulation with shared control of object handling forces. Presence:

Teleoperators & Virtual Environments **14**(6), 720–731 (2005)

[33] Li, M., Okamura, A.M.: Recognition of operator motions for real-time assistance using virtual fixtures. In: IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 125–131 (2003)

[34] Kragic, D., Marayong, P., Li, M., Okamura, A.M., Hager, G.D.: Human-machine collaborative systems for microsurgical applications. The International Journal of Robotics Research **24**(9), 731–741 (2005)

[35] Shen, J., Ibanez-Guzman, J., Ng, T.C., Chew, B.S.: A collaborative-shared control system with safe obstacle avoidance capability. In: IEEE Conference on Robotics, Automation and Mechatronics, vol. 1, pp. 119–123 (2004)

[36] Dragan, A.D., Srinivasa, S.S.: A policy-blending formalism for shared control. The International Journal of Robotics Research **32**(7), 790–805 (2013)

[37] Javdani, S., Srinivasa, S.S., Bagnell, J.A.: Shared autonomy via hindsight optimization. In: Robotics Science and Systems. NIH Public Access, ??? (2015)

[38] Crandall, J.W., Goodrich, M.A.: Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1290–1295 (2002)

[39] Aarno, D., Ekvall, S., Kragic, D.: Adaptive virtual fixtures for machine-assisted teleoperation tasks. In: IEEE International Conference on Robotics and Automation, pp. 1139–1144 (2005)

[40] Selvaggio, M., Notomista, G., Chen, F., Gao, B., Trapani, F., Caldwell, D.: Enhancing bilateral teleoperation using camera-based online virtual fixtures generation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1483–1488 (2016)

[41] Selvaggio, M., Abi-Farraj, F., Pacchierotti, C., Robuffo Giordano, P., Siciliano, B.: Haptic-based shared-control methods for a dual-arm system. IEEE Robotics & Automation Letters **3**(4), 4249–4256 (2018)

[42] Selvaggio, M., Ghalamzan E., A.M., Moccia, R., Ficuciello, F., Siciliano, B.: Haptic-guided shared control for needle grasping optimization in minimally invasive robotic surgery. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3617–3623 (2019)

[43] O'Malley, M.K., Gupta, A., Gen, M., Li, Y.: Shared control in haptic systems for performance enhancement and training. Journal of Dynamic Systems, Measurement, and Control **128** (2006)

[44] Squeri, V., Basteris, A., Sanguineti, V.: Adaptive regulation of assistance 'as needed'in robot-assisted motor skill learning and neuro-rehabilitation. In: IEEE International Conference on Rehabilitation Robotics, pp. 1–6 (2011)

[45] Guidali, M., Schlink, P., Duschau-Wicke, A., Riener, R.: Online learning and adaptation of patient support during adl training. In: IEEE International Conference on Rehabilitation Robotics, pp. 1–6 (2011)

[46] Stroppa, F., Marcheschi, S., Mastronicola, N., Loconsole, C., Frisoli, A.: Online adaptive assistance control in robot-based neurorehabilitation therapy. In: IEEE International Conference on Rehabilitation Robotics, pp. 628–633 (2017)

[47] Greer, J.D., Morimoto, T.K., Okamura, A.M., Hawkes, E.: A soft, steerable continuum robot that grows via tip extension. Soft Robotics **6**(1), 95–108 (2019)

[48] Coad, M.M.: Design, modeling, and control of vine robots for exploration of unknown environments. PhD thesis, Stanford University (2021)

[49] Gruebele, A.M., Zerbe, A.C., Coad, M.M., Okamura, A.M., Cutkosky, M.R.: Distributed sensor networks deployed using soft growing robots. In: IEEE International Conference on Soft Robotics (RoboSoft), p. (2020)

[50] Rohmer, E., Singh, S.P., Freese, M.: V-rep: A versatile and scalable robot simulation framework. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1321–1326 (2013)

[51] Nuño, E., Basañez, L., Ortega, R.: Passivity-based control for bilateral teleoperation: A tutorial. Automatica **47**(3), 485–495 (2011)

[52] Chen, J.Y., Haas, E.C., Barnes, M.J.: Human performance issues and user interface design for teleoperated robots. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **37**(6), 1231–1245 (2007)

Statements and Declarations

Competing interests. The authors have no relevant financial or non-financial interests to disclose.

Authors' contributions. Fabio Stroppa designed and developed the interaction paradigms, designed and developed the overall system (motion tracking, computer vision, network communication, interfaces), designed and developed the experimental setup, designed and built the circuit board for the soft robot, assembled the soft robot, designed and implemented the inverse kinematics for the soft robot, supervised the real-scenario demonstrations, analyzed the data, and wrote the manuscript. Mario Selvaggio designed and implemented the virtual model of the soft robot for simulation, ran the statistical analysis for the parameter tuning for haptic rendering, supervised the simulated-scenario demonstrations, and partially wrote the manuscript. Nathaniel Agharese and Laura H. Blumenschein contributed to implementing the closed-loop control of the system and reviewed the manuscript. Ming Luo built the prototype of the soft robot and reviewed the manuscript. Elliot W. Hawkes reviewed the manuscript. Allison M. Okamura supervised the project and reviewed the manuscript.

Ethics approval. This is an observational study. The Institutional Review Board of Stanford University has confirmed that no ethical approval is required.

Consent to participate. Informed consent was obtained from all individual participants included in the study

Consent for publication. No individual person's data in any form is disclosed in the paper.

Code or data availability. The simulator scene and its scripts, as well as the C++ software implementing the interaction modules described in the text, are publicly released at https://github.com/mrslvg/sgm-sim/.

Soft-Growing Robot Manipulator Kinematics and Control

This document briefly illustrates the strategies used to control the soft-growing robot, including details on the inverse kinematics and the closed-loop control.

1 Inverse Kinematics

The controller includes inverse kinematics that transform a point from Cartesian space into motor space, from three coordinates $\langle x, y, z \rangle$ to four $\langle s_1, s_2, s_3, e \rangle$ as depicted in Fig. 1. In motor space, the dimensions s_1, s_2, and s_3 define the direction of the three motors used for steering in the plane, whereas e is used for eversion.

The procedure is divided into mapping for steering and eversion, which are described in the following sections. We solve the inverse kinematics with a geometric rather than differential approach because the inherent difference in steering and eversion dynamics required unique gains for the motors controlling the different DoFs.
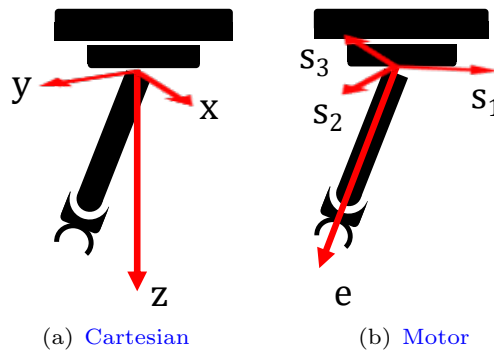

(a) Cartesian   (b) Motor

Fig. 1 The inverse kinematics transform a point from the (a) Cartesian space into the (b) manipulator motor space, defined by three dimensions for steering in the plane and a fourth for eversion. Eversion is oriented with the body of the robot.

2 Steering Mapping

The mapping for steering transforms a point in a plane from a 2D representation in Cartesian space ($\langle x, y \rangle$ as shown in Fig. 2(a)) to a 3D representation in Motor space ($\langle s_1, s_2, s_3 \rangle$ as shown in Fig. 2(b)). The same plane composed of four quadrants in the Cartesian space is divided into three sectors by the axes of the motor space, having an angle of $\frac{2}{3}\pi$ between each pair. These axes are aligned with the direction of the manipulator's motors: the positive direction of the axes indicates a rotation of the motor to retract the cable and pull the manipulator, whereas the negative direction indicates cable release. A coordinated movement along these axes allows the manipulator to steer in a desired direction. The mapping is defined by the following equations, which are given by expressing the half lines of the motor space axes in Cartesian space.

$$
x = s_1 \cdot \cos\left(-\frac{\pi}{3}\right) + s_2 \cdot \cos\left(\frac{\pi}{3}\right) - s_3
$$
$$
y = s_1 \cdot \sin\left(-\frac{\pi}{3}\right) + s_2 \cdot \sin\left(\frac{\pi}{3}\right) \tag{1}
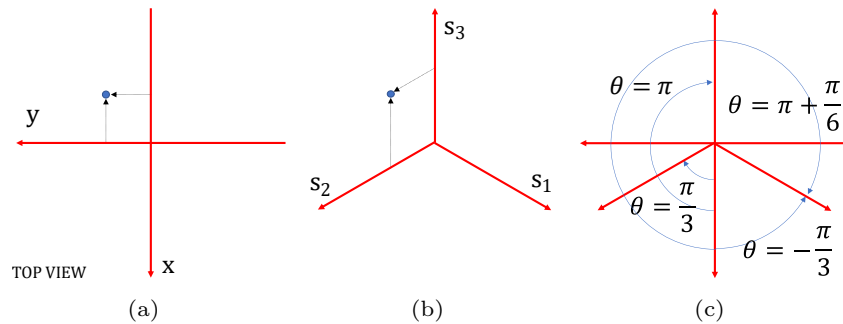$$

Fig. 2 (a) A point in Cartesian space. (b) The same point expressed in motor space. (c) One representation of the angles θ between the x-axis of the Cartesian space and the axes of the Motor space. These angles are used to identify in which sector the point lies, then solve the inverse kinematics for those two dimensions defining that sector by setting the third dimension to zero.

Having a system of two equations in three variables leads to infinite solutions; therefore, we fix one motor to be zero such that only two motors are actuated at a time. The motor set to zero is the one along the axis that does not define the sector in which the point currently lies: by defining θ as the angle between the positive x-axis and the ray to the point $\langle x, y \rangle$ (Fig. 2(c)), the solution to (1) is given by the following:

- When the point lies in the sector defined by s_1 and s_2, the coordinate s_3 is set to zero, and the solution is:

$$\text{if } \theta \in \left[-\frac{\pi}{3}, \frac{\pi}{3}\right] \begin{cases} s_1 = x \cdot \frac{1}{2\cos\left(\frac{\pi}{3}\right)} - y \cdot \frac{1}{2\sin\left(\frac{\pi}{3}\right)} \\ s_2 = x \cdot \frac{1}{2\cos\left(\frac{\pi}{3}\right)} + y \cdot \frac{1}{2\sin\left(\frac{\pi}{3}\right)} \\ s_3 = 0 \end{cases} \tag{2}$$

- When the point lies in the sector defined by s_2 and s_3, the coordinate s_1 is set to zero, and the solution is:

$$\text{if } \theta \in \left[-\frac{\pi}{3}, \pi\right] \begin{cases} s_1 = 0 \\ s_2 = y \cdot \frac{1}{2\sin\left(\frac{\pi}{3}\right)} \\ s_3 = y \cdot \frac{1}{2\tan\left(\frac{\pi}{3}\right)} - x \end{cases} \tag{3}$$

- When the point lies in the convex sector defined by s_1 and s_3, the coordinate s_2 is set to zero, and the solution is:

$$\text{if } \theta \in \left[\pi, \pi + \frac{\pi}{6}\right] \begin{cases} s_1 = -y \cdot \frac{1}{2\sin\left(\frac{\pi}{3}\right)} \\ s_2 = 0 \\ s_3 = -x - y \cdot \frac{1}{2\tan\left(\frac{\pi}{3}\right)} \end{cases} \tag{4}$$
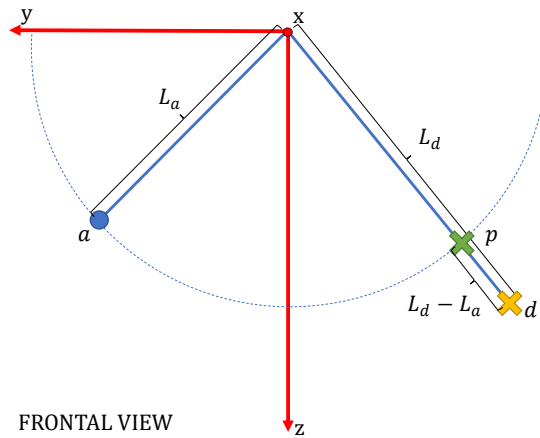

Fig. 3 The inverse kinematics calculates the steering commands on a proxy p rather than the real desired position d, to decouple steering from eversion.

3 Eversion Mapping

Starting from a point in the Cartesian space, the eversion coordinate e is simply given by the length of the robot calculated as in (5).

$$e = \sqrt{x^2 + y^2 + z^2} \tag{5}$$

4 Error Evaluation

We calculate the error between the the desired position, $d = \langle x, y, z \rangle \in \mathbb{R}^3$, and the actual position, $a = \langle x, y, z \rangle \in \mathbb{R}^3$, of the robot end-effector for closed-loop control. The output error is $\psi = \langle s_1, s_2, s_3, e \rangle \in \mathbb{R}^4$. The inputs d and a are defined in Cartesian space, whereas the output ψ is in motor space. To solve the system in (1), we constrain one of the steering dimensions to be zero to ensure balance among cable actuation. In the following steps, we reference the same point (a or t) in the two different spaces: when this is accompanied by the superscript m, then we refer to motor space; if no superscript is indicated, then we refer to Cartesian space.

   We first calculate the length of the robot, for both the actual (L_a) and the desired (L_d) configuration. Next, in order to decouple steering and eversion, the commands in the plane are evaluated on a proxy rather than directly on the desired position d. This proxy p is the position the manipulator would reach without eversion, which is achieved by steering from a to d following the surface of a semi sphere, as shown in Fig. 3. Once p has been given as a goal for steering, the position d is reached by everting the manipulator of the required length $L_d - L_a$.

   The proxy p is obtained by multiplying a unit vector $\hat{u}_d$ pointing in the direction of d and the displacement in length required to keep the manipulator on the same semi sphere, which is the actual length of the robot L_a.

$$\hat{u}_d = \frac{d}{|d|} = \frac{d}{L_d} = \frac{d}{\sqrt{d_x^2 + d_y^2 + d_z^2}} \tag{6}$$

$$p = \hat{u}_d \cdot \sqrt{a_x^2 + a_y^2 + a_z^2} = \hat{u}_d \cdot L_a = d \cdot \frac{L_a}{L_d} \tag{7}$$

   The steering actuation is primarily achieved by pulling cables; releasing a cable will still actuate the robot, but the effect is reduced as no force is exerted against the friction of the cable routing except for gravity. Because cable release does not respect the inverse kinematics, we perform all steering as cable pulling as follows. First, p and a are translated such that a coincides with the origin of the reference frame, generating a point t. Next, the point t is transformed into motor space t^m, which will only have positive commands (cable pulling) to ensure actuation.

   Unfortunately, performing only cable pulling results in wrinkling of the fabric of the robot at the base where the steering occurs. Too much wrinkling

leads to the robot being unable to steer. Thus, when the robot is not required to steer in the positive direction of a given motor, that motor rotates in the opposite direction avoiding wrinkling. In our kinematics, this translates into transferring part of the most positive value from one steering axis to the other two in the opposite direction; or in other words, finding a different route in the space for reaching the same desired point from a starting one. Finding a different route is always possible, as the system in (1) has infinite solutions.

The most positive value $t_{s_{max}}^m$ is defined by (8), and is it always possible to retrieve it as we chose the solutions (2), (3), and (4) to be always positive.

$$\forall j \in [1,3] \;:\; 0 \leq t_{s_j}^m \leq t_{s_{max}}^m = \max t_{s_j}^m \tag{8}$$

The redistribution adds part of $t_{s_{max}}^m$ as cable release to all motors, which means that all positive commands should pull less and all negative commands should release more. We define $\epsilon \in [0,1]$ as a coefficient that removes part of $t_{s_{max}}^m$.

$$\forall j \in [1,3] \;:\; t_{s_j}^m \leftarrow t_{s_j}^m - t_{s_{max}}^m \cdot \epsilon \tag{9}$$

Due to the geometry of the Motor space, this operation is trivial: any triangle having a and p as vertices and all sides parallel to the space axes is equilateral; therefore, adding or subtracting the same value to all the dimensions will provide a valid solution for (1), as shown in the example on Fig. 4.
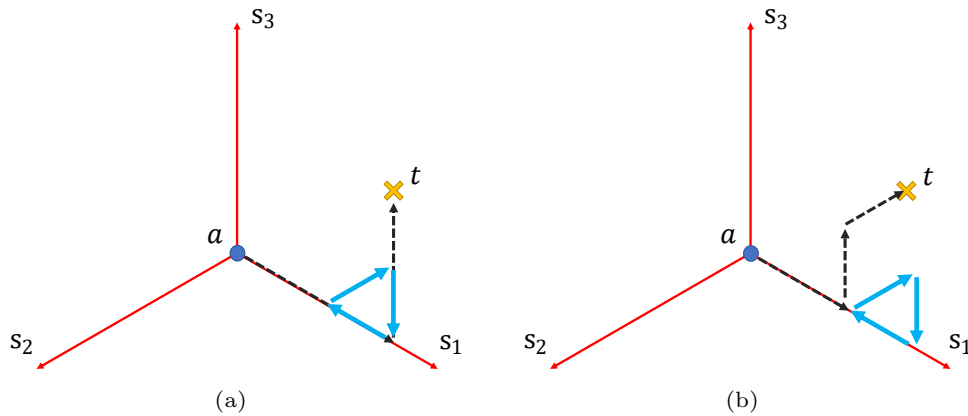

(a)   (b)

Fig. 4 Alternative route to redistribute motor actuation. (a) Based on the inverse kinematics, going from a to t by only performing cable pulling would generate two positive commands on s_1 and s_3 (with $s_1 > s_3$), leaving s_2 to be zero. (b) Alternatively, a part s_1 can be transferred to the other two axes, resulting in two positive values on s_1 and s_3. The final actuation would be two motors pulling (s_1 and s_3) and one slightly releasing (s_2). The triangle represented in the figure has a side equal to the part of s_1 we intend to redistribute; this triangle is also equilateral, as the space is divided into three sectors having angles of $\frac{2}{3}\pi$ between each axis. Therefore, redistributing a value on opposite directions can be simply performed by addition or subtraction.

However, this procedure is not required when the direction of the robot remains the same in two consecutive iterations. Therefore, actuation redistribution is only performed when (10) is true. Given the motor space coordinates

of the actual end-effector position a, which represents the steering commands to go from the origin to the previous desired position, we compared it to a new position t. If both t^m and a^m share the same null coordinate, then the robot is moving in the same direction as in the previous iteration, and there is no need for cable release.

$$\nexists i \in [1,3] \quad : \quad t^m_{s_i} = 0 \ \wedge \ a^m_{s_i} = 0 \tag{10}$$

In our setup we set $\epsilon = 0.3$, as we found that higher values caused slack in the cables and lower values produced excessive wrinkling.

Finally, we transform the steering coordinates from distances to angles. Sending angles rather than distances to the closed loop control is preferable, as the commands given to steer depend on the manipulator's length. Angles are calculated over the actual length of the robot L_a as both a and p share this value. The error ψ is then returned as output, where the steering commands are defined in t^m and the eversion command is $L_d - L_a$.

5 Closed-Loop Control

Separate controllers are used for the eversion and steering DoFs because they have different dynamics. For eversion, we use feed-forward augmentation on a proportional controller. This augmentation is required to counter the passive growth that occurs from pressurizing the system. The proportional term, which is applied to an error of length in millimeters, proved sufficient for good tracking of growth, so we did not include an integral or derivative term. For steering, we use a scaled proportional-derivative controller on the error in rotation of the three motors; our gains are scaled proportionally with the length of the robot. This scaling is necessary because the steering dynamics are a function of the manipulator's length. We did not use an integral term because this is detrimental to telepresence during teleoperation, and we wanted to use the same low-level control strategy for all of the paradigms. All gains are hand tuned.