A Pilot Study in Vision-Based Augmented Telemanipulation for Remote Assembly over High-Latency Networks

Jonathan Bohren,
Chavdar Papazov, Darius Burschka,
Kai Krieger, Sven Parusel, Sami Haddadin,
William L. Shepherdson, Gregory D. Hager, Louis L. Whitcomb

***Abstract*— In this paper we present an approach to extending the capabilities of telemanipulation systems by intelligently augmenting a human operator's motion commands based on quantitative three-dimensional scene perception at the remote telemanipulation site. This framework is the first prototype of the *Augmented Shared-Control for Efficient, Natural Telemanipulation* (ASCENT) System. ASCENT aims to enable new robotic applications in environments where task complexity precludes autonomous execution or where low-bandwidth and/or high-latency communication channels exist between the nearest human operator and the application site. These constraints can constrain the domain of telemanipulation to simple or static environments, reduce the effectiveness of telemanipulation, and even preclude remote intervention entirely.**

ASCENT is a semi-autonomous framework that increases the speed and accuracy of a human operator's actions via seamless transitions between one-to-one teleoperation and autonomous interventions. We report the promising results of a pilot study validating ASCENT in a transatlantic telemanipulation experiment between The Johns Hopkins University in Baltimore, MD, USA and the German Aerospace Center (DLR) in Oberpfaffenhofen, Germany. In these experiments, we observed average telemetry delays of 200ms, and average video delays of 2s with peaks of up to 6s for all data. We also observed 75% frame loss for video streams due to bandwidth limits, giving 4fps video.

I. Introduction

There have been tremendous advances in the sensing and manipulation capabilities of robotic platforms over the past decade. Newly available platforms such as DLR's Justin [3], the Willow Garage PR2 [2], and NASA's Robonaut 2 [8] now have sufficient dexterity to accomplish to a broad spectrum of complex manipulation tasks. However, the development of algorithms for perception, reasoning, planning, and sensor-based task execution have not advanced with commensurate rapidity. As a result, there is a fundamental gap between what *could* be done with today's robotic platforms, and what *can* be done by a fully autonomous system.

J. Bohren, W. L. Shepherdson, G. D. Hager, and L. L. Whitcomb are with the Laboratory for Computational Sensing and Robotics, The Johns Hopkins University, Baltimore, Maryland, USA.

C. Papazov and D. Burschka are with the Department of Computer Science, Technische Universität München. Germany.

K. Krieger, S. Parusel, and S. Haddadin are with the Robotics and Mechatronics Center of the German Aerospace Center (DLR).

Authors email addresses: jbo@jhu.edu, papazov@in.tum.de, burschka@tum.de, kai.krieger@dlr.de, sven.parusel@dlr.de, sami.haddadin@dlr.de, wls@jhu.edu, hager@cs.jhu.edu, llw@jhu.edu.

The authors gratefully acknowledge the support of the National Science Foundation (NSF) under Grant OISE-1065092. This work has been partially funded by the European Commission's Sixth Framework Programme as part of the projects SAPHARI under grant no. 287513.

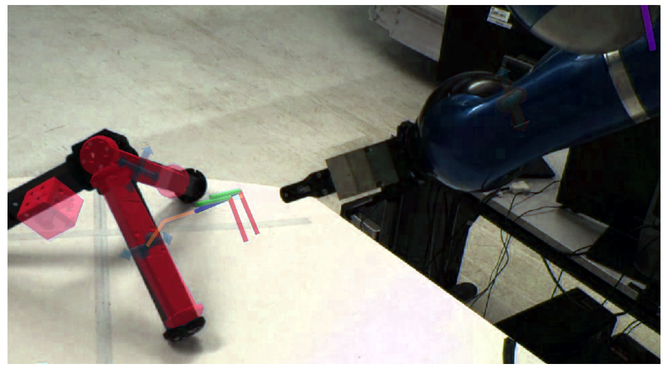


Fig. 2: A screen capture of a representative view of the operator visualization in the prototype system over a low-performance network during a sensor-based control augmentation. The GRASP-Beams (left) have been recognized by the 3D perception system and the commanded trajectory of the DLR Lightweight Robot (right) has been interpreted to dispatch a trajectory to grasp one of the parts (center).

An alternative is to place a human in the control loop. Traditionally, this has taken the form of direct teleoperation for manipulation (henceforth telemanipulation) where the operator views images from the remote site via a video feed and commands the motions remote robot with joysticks, keyboards, and high-level instructions. For example, the master control console of the da Vinci® surgical robot provides a stereoscopic view of the laparoscopic robotic tools inside a patient's body, and these tools are controlled by a pair of 7-degree-of-freedom (DoF) manipulators with pinch grips for tool actuation. While quite effective in many situations, this approach has a number of well-known limitations, namely:

- Controlling high-dexterity end-effectors (e.g. multi-finger hands) is difficult. Not only does this necessitate an input device capable of efficiently controlling many degrees of freedom, but even with such a device the human operator is often in a situation where the haptic cues usually critical for manipulation are missing. As a result, all manipulation must be performed from visual cues which is challenging at best.

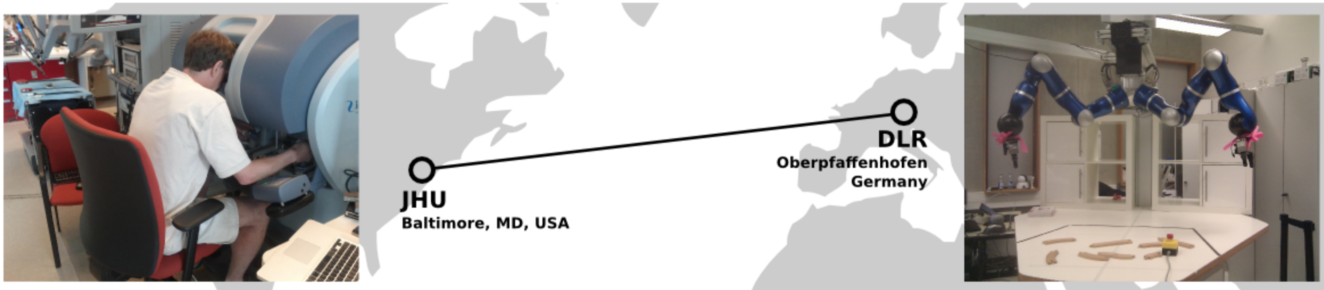


Fig. 1: The experiments were conducted with a human operator at The Johns Hopkins University (JHU) Homewood Campus in Baltimore, MD, USA, utilizing a da Vinci® Master Console (left) commanding a DLR LWR as part of the SAPHARI platform at the German Aerospace Center (DLR) in Oberpfaffenhofen, Germany (right).

- Many remote telerobotic applications have limitations on bandwidth, creating a situation where the fidelity of the imaging is compromised. The availability of stereoscopic imaging, image resolution and frame rates may be limited, leading to a limited ability to resolve necessary detail for manipulation. This is particularly challenging given the absence of haptic cues noted above *increases* the reliance on visual perception.
- Some environments impose additional communication latency (time-delay) on telemetry as well. For example, telemanipulation from Earth to low-earth orbit typically imposes delays that exceed half a second for direct line-of-sight communications and 2-7 seconds when using larger-coverage on-orbit communications networks. The limitations of human performance in telemanipulation are well-studied, and the threshold at which human performance begins to suffer is far below that [12].

As a result of these limitations, the performance of direct telemanipulation is often poor in the presence of bandwidth limitations or delayed telemetry. It limits the maximum rate at which safe manipulations can be made, increases mission completion times, and requires substantial skill and training on the part of the operator to be effective.

The need to improve telemanipulation is well-known. For example, two recent studies by the National Research Council studies have identified the development of human and robotic on-orbit servicing capabilities as a national priority for the United States [7], [6]. Furthermore, recent on-orbit spacecraft servicing missions and on-orbit spacecraft engineering experiments have demonstrated the feasibility on-orbit repair and servicing of spacecraft by humans [19], telerobotics [13], [18], [14], [31], and combined human-telerobotic operations [16]. However, we can expect most such operations will require a human "in the loop" since they were not originally designed for robotic servicing. In addition, future robotic assembly missions will include tasks which are far more complex than those which today's stat-of-the-art autonomous systems are capable of completing robustly.

The challenge therefore is to combine the judgment and adaptability of human reasoning with the precision and repeatability of autonomous execution in appropriately constrained circumstances. ASCENT takes a *collaborative systems* approach that transcends the limitations of either purely autonomous or purely teleoperated control modes by combining task-specific sensor-based feedback with input from an operator. As a result, the operator is able to provide gross motion guidance to the system, and the remote manipulator is able to adapt that motion based on environmental information. We have implemented this approach with a DLR lightweight arm driven by a da Vinci® S master console separated by over 4000 miles. We demonstrate that ASCENT greatly improves manipulation performance, particularly when subtle motions are necessary in order to correctly perform the task.

II. Background

Presently, robots that are deployed to perform high-value tasks usually fall into two broad categories:

1) *The task can be automated*. These tasks tend involve structured environments, simple geometry, slow dynamics, little or no sensing, and actions with easily detected and characterized failure modes from which the robot can easily recover.
2) *The task can be teleoperated*. These tasks are usually too complex, too diverse, or too small in number to be automated. Instead, human perception and cognition are used to provide the "intelligence" necessary to perform these tasks. To be effective, however, the application requires an environment that accommodates high-bandwidth, low-latency telemetry and dexterous human-machine interfaces.

Our goal is to address the broad class of tasks that fall between these two extremes by recognizing automatable sub-tasks, and performing them automatically [24]. Our approach consists of three major components: 1) a direct telemanipulation interface; 2) an intent recognition system that "parses" operator motion and detects automatable sub-tasks; and 3) a perception and control system that performs that associated automation. Here, we briefly review related work in these areas.

There is a long history of efforts to overcome the limitations of autonomous robotic systems with human intervention. Broadly speaking, approaches to *semi-autonomous* *teleoperation* can be categorized as *teleprogramming* [10] or *supervised autonomy* [9]. These model-based systems take many forms, but they all rely on either predictive simulations of the remote environment or a layer of command abstraction between the operator and the robot.

One way to enable more natural teleoperation (with or without haptic feedback) under large time delays is model-based teleoperation, which uses models of the environment acquired from models developed a priori and updated in real time during manipulation [29], [31], [28], [10], [32]. Preliminary work has shown that model-based teleoperation with haptic feedback improves user performance under very simple conditions with delays of up to 4 seconds [23], but manipulation has not yet been accomplished under this paradigm.

Teleprogramming has proven useful in scenarios with simple, imprecise tasks as described above. These are tasks in which tools are inherently robust to imprecise positioning like the extraction of soil samples on a remote world, or actions which can be performed without physical contact with other objects at all. Stein et al., report a teleprogramming interface was used to attempt to cut tape securing a mock-up of thermal satellite insulation at a remote site [30], and Sayers [28] report an undersea application. In both of these cases, however, numerous manipulation errors occurred during telemanipulation due to discrepancies between the predictive display and actual environment.

Another demonstration mission was performed in a collaborative effort between the German Aerospace Center (DLR) and NASA with ROTEX on STS-55 [15]. This experiment used a promising *shared autonomy* paradigm which added a more sophisticated control loop to the slave robot. Its capabilities, however, were at that time limited to a reduced set of demonstration tasks such as grasping fixed and floating objects at the direction of the teleoperator's interaction with a simulation of the remote environment.

1) Teleoperator Intention Recognition: The ASCENT prototype reported here relies on the ability of the system to recognize operator intent within a structured task environment. Prior work on intent recognition has relied predominantly on learning this context from observation of expert performance [24], [33], [20]. However, to a great extent these approaches have not made use of a strong prior task model, nor have they relied on closed-loop execution based on sensing.

2) Visual Scene Parsing: Automated interaction with the world requires the system to identify the objects with which needs to interact and complete the occluded parts of the geometry to perform a manipulation task. Holz et. al. [17] introduce an approach for mobile robots that allows them to create the obstacle map of the environment and classify the graspable objects by analyzing the data acquired from an RGB-D sensor. In [21] a technique is presented in which a hierarchical, multi-view dataset of objects is created based on the RGB-D data. This is then used to identify and register objects. Hager and Wegbreit present a complete system for scene modeling based on range data [11]. A system using RGB-D sensor to manipulate objects in cluttered scenes is presented in [25], and the processing introduced there is an integral part of the perception system used in the ASCENT prototype framework.

III. Approach and Design

ASCENT aims to improve the performance of Cartesian kinesthetic telemanipulation in two ways. The first aspect is to mitigate the effects of large time-delay on the speed at which an operator can accomplish a task. In this context, a task is an atomic action with a discrete success or failure after completion. We can separate the delay effects into two categories: those that take place on the time-scale of a task, and those that take place at the sub-task time-scale. For example, once an operator has adopted a *move-and-wait* execution strategy when maneuvering in the remote environment or grasping an object, he or she must wait for the entire closed-loop delay cycle multiple times while moving. We believe that early on in this stuttered motion, there is enough information to determine the user's intention. If we can properly classify this intention, then, we can eliminate the sub-task scale delays by driving the end-effector of the manipulator to the predicted target automatically.

The second aspect is simply to improve manipulation accuracy. In our target applications, real bilateral force reflection is not feasible, so there is an inherent danger that the teleoperator could damage the objects that he or she is manipulating. While passive or active compliant control methods can be used to manage these risks when manipulating objects with large inertia, this is insufficient if a task requires that an object or surface remain stationary until it is securely grasped. By incorporating non-contacting sensing methods like RGB-D 3D perception, we can accurately dispatch an autonomous trajectory that does not disturb the object prematurely or in an undesirable way.

We developed the ASCENT prototype system with several open-source and proprietary third-party software frameworks. Both the da Vinci® master API and DLR BEASTY controller are proprietary, closed-source frameworks, and are not available to the public. We relied heavily on open-source frameworks, however, to connect these two end-points. We utilized and extended software built with the Robot Operating System (ROS) [27] for distributing computation and the Open Robot Control Software (OROCOS) [5] for controllers that talk to the lower-level APIs and need to run at high loop rates.

A. Operator Interface

The operator interface consists of an Intuitive Surgical da Vinci® S console (Intuitive Surgical, Inc., Sunnyvale, CA, USA). The console incorporates two 7-DoF master manipulator arms with finger pinch-grips that open and close. Through access to the da Vinci API, we have written ROS interface that broadcasts resolved Cartesian positions of the end-effectors. The API also provides a set of events that supports the capture of foot pedal press events in the console and the operator head sensor for safety. Stereo visualization

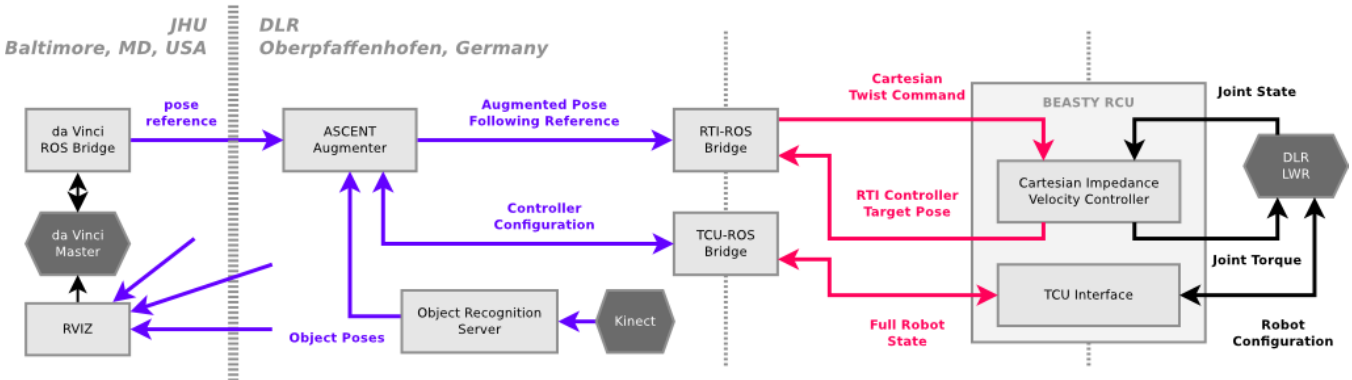


Fig. 3: A high-level overview of the ASCENT prototype. Commands are sent from the human operator at the da Vinci® master on the left over a ROS network (purple) to the remote site. At the remote site, they are intercepted by the ASCENT Augmenter, which classifies user intention based on real-time sensor feedback from the RGB-D perception pipeline. The augmented commands are then forwarded to the DLR BEASTY Real-Time Interface ROS bridge, which commands the Cartesian impedance velocity controller running in the BEASTY RCU over DLR aRDnets (red).

is also provided to the operator by injecting stereo imagery into the da Vinci TilePro® interface with rendered overlays in the standard ROS 3D visualization framework, RViz.

B. Control Flow

The implementation of the ASCENT prototype, shown at a high-level in Fig. 3, differs from a standard telemanipulation architecture with the insertion of the *ASCENT Augmenter* in the control path between the master and target robot. The augmenter's input and output interfaces are identical to the input and output interfaces of the target and master components, respectively, so that the addition of this component is transparent to the system.

This command interface is defined as a hardware-agnostic ROS message, and contains the following generic Cartesian control information:

- ROS Header (reference frame and timestamp),
- Cartesian Position and Orientation $\in$ SE(3),
- Cartesian Twist $\in \mathbb{R}^6$,
- Gripper opening scalar $\in (0.0, 1.0)$,
- Boolean flag designating a dead-man (safety) switch is engaged,
- Boolean flag designating an emergency stop button has been triggered.

This enables us to use a variety of control inputs, in addition to the da Vinci master. We have used other haptic interfaces like the Sensable PHANToM Omni and consumer human input devices like the 3DConnexion SpaceNavigator.

C. Manipulator Platform

The manipulation system (see Fig.4) consists of two DLR Lightweight Robot III (LWR-III) arms equipped with a 2-jaw gripper, but for the purposes of this study, we only controlled one of the arms. The LWR-III is a 1.3 meter-long 7-DoF manipulator instrumented with torque sensors in each joint. We command the robot in Cartesian space and use 6-DoF Cartesian impedance control [1] with a nullspace projector

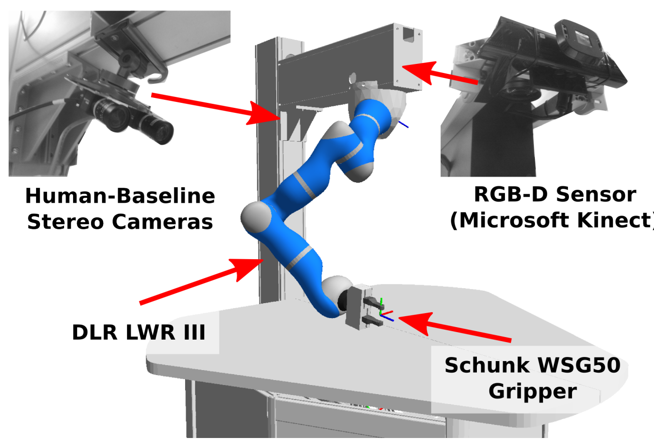


Fig. 4: The remote telerobotic system is comprised of a DLR lightweight robot III (LWR) 7-DoF arm with a Schunk WSG-50 gripper, a Kinect RGB-D sensor for perception supporting semi-autonomous actions, and a human-scale baseline pair of stereo cameras for the human operator's view of the remote site.

for centering the redundant joint motion around its origin of symmetry. The overall control loop reads as[1]:

$$\boldsymbol{\tau}_d = -J(\mathbf{q})^T (K_x \tilde{\mathbf{x}}(\mathbf{q}) + D_x \dot{\mathbf{x}}(\mathbf{q})) + \mathbf{g}(\mathbf{q}) + \mathcal{N}(\mathbf{q}) \boldsymbol{\tau}_{d,ns}, \quad (1)$$

with $K_{\mathbf{x}}, D_{\mathbf{x}} \in \mathbb{R}^{m \times m}$ being the diagonal positive definite desired stiffness and damping matrices, $\mathbf{x}_d \in \mathbb{R}^m$ the desired tip pose in Cartesian coordinates, $\mathbf{x}(\mathbf{q}) = f(\mathbf{q})$ the position and orientation of the tip computed by the direct kinematics map f, and $\tilde{\mathbf{x}}(\mathbf{q}) = \mathbf{x}(\mathbf{q}) - \mathbf{x}_d$ being the position error. $J(\mathbf{q}) = \frac{\partial f(\mathbf{q})}{\partial \mathbf{q}}$ is the Jacobian of the manipulator, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ the gravity compensation. The operational space velocity is $\dot{\mathbf{x}}(\mathbf{q}) = J(\mathbf{q})\dot{\mathbf{q}}$. $\mathcal{N}$ denotes the null-space projector that

[1]In fact the controller is more complex as the LWR has to be treated as a flexible joint robot. However, for sake of clarity, we omit this fact.

maps $\boldsymbol{\tau}_{d,ns}$ into the null-space of the Cartesian control. The nullspace controller aims to keep the joints near their origins if not conflicting with the desired pose.

D. DLR Beasty Control Architecture

Beasty [26], shown on the right in Fig. 3, is the DLR real-time robot control framework. Beasty is composed primarily of the Task Control Unit (TCU) and the Robot Control Unit (RCU). They serve as the general interface to the robot and communicate with each other via asynchronous protocols. The TCU is a general state-based control entity, which runs in non-real-time and provides the nominal robot actions and behaviors to the RCU. The RCU in turn runs at the same clock rate as the robot, assigning control, motion generation, interaction, and safety methods. Furthermore, it interprets and validates the atomic commands from the TCU.

In the ASCENT prototype, we primarily used Beasty's Real-Time Interface (RTI), which facilitates loop-closing for tasks such as visual servoing, velocity tracking, or torque control over the network. More specifically, the RTI supports:

- a closed torque control loop,
- commanded positions/velocities in real-time,
- reading full robot state in real-time, and
- commanding controller parameters in real-time (e.g. impedance parameters).

The signal flow for closing the velocity controlled loop via the Internet is depicted in Fig. 3. In particular, a loop is closed via the commanded master velocity $\mathbf{x}_d^+$ in order to show better tracking and stability. For this, the instantaneous desired position, $\mathbf{x}_d$, is used as the system state on the master side for closing a PD control loop:

$$\dot{\mathbf{x}}_d^+ = K_P(\mathbf{x}^- - \mathbf{x}^+) + K_D(\dot{\mathbf{x}}^- - \dot{\mathbf{x}}^+). \qquad (2)$$

E. Scene Parsing

A semantic description of the environment is essential to identify possible manipulation goals in the scene. In our system, we use the algorithm proposed in [25] that applies a RANSAC-like sampling strategy to match known object geometries to the reconstructed point cloud from an RGB-D sensor like the Microsoft Kinect. The system uses oriented point pairs [34] to align surface patches in the 3D point-cloud with the patches in the object database.

These objects are then associated with pre-determined potential grasp locations to provide semantic information to the semi-autonomous intention classification and motion-planning system.

F. Semi-Autonomous Intervention

Once the semantic percepts have been reported by the scene parsing system, we employ some simple heuristics for triggering autonomous interventions and deciding which grasp pose to acquire.

The augmenter maintains a buffer of the last n telemanipulation commands. At a fixed rate, it passes the command buffer into a series of classifiers. These classifiers return a tuple *(confidence, taskel)* where taskel or "task element" is a callable object that generates some set of commands to be sent to the manipulator. Each taskel has associated with it a list of binary resources to prevent the dispatching of conflicting commands.

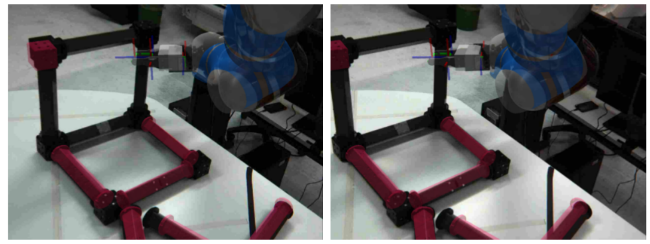


Fig. 5: A view of the operator stereo display showing a real-time overlay of the remote LWR and rendering of the 3-D objects models identified by the scene parsing algorithm.

The augmenter then executes the list of taskels asynchronously until they are complete. At each call, it sends the taskel an updated command buffer as well as a list of semantic percepts and other state data.

For the grasp/approach classifier, we determine a grasp target in the following way. First, we require that the grasp point p meets a few parametrized criterion.

- We are close to the point, but not too close: $0.05 \leq \|p - x_n\| \leq 0.25$ (in *meters*)
- We are moving fast enough: $|\sum_{k=n-5}^{n} \dot{x}/5| > 0.001$ (in *m*eters/s)

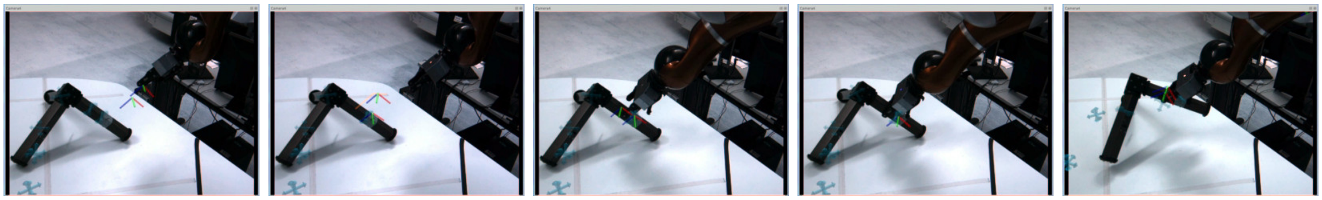
Of the remaining potential grasps, we determine the most likely grasp point. This is simply done by projecting the list of approach velocities onto the normalized vectors pointing from the manipulator to a given grasp point, $\hat{d}_k$. This gives the approach score, σ:

$$\sigma = \sum_{k=0}^{n} \hat{d}_k^{\mathrm{T}} \dot{x_k} \qquad (3)$$

Once the grasp point has been determined from the highest approach score, the grasp orientation is selected based on the error between the potential orientations and the current orientation of the gripper.

Then, a simple linearly-interpolated trajectory is generated from the current manipulator pose to a grasp approach point, and then on to the grasp pose itself. Once the trajectory has been executed, the taskel dispatches a command to grasp the part.

One issue with any sort of autonomous intervention is that once the remote manipulator pose *s*eparates from the command pose, the operator must re-align the frames to continue moving. Since aligning the frame is just as hard as grasping to begin with, we give the operator a single switch (foot pedal) to attract the remote manipulator to the command pose. This sequence of events is shown in Fig. 6. Additionally, this switch could be activated like a clutch at any time to interrupt or disable the semi-autonomous actions provided by the ASCENT prototype.



(a) The operator jogs the manipulator towards a recognized grasp pose (designated by faint blue arrows), implying that he wants to grasp it.

(b) The augmenter recognizes the intention, and dispatches a trajectory (orange) from the current robot pose to the grasp point.

(c) The robot begins to interpolate between the operator command and the trajectory and then follows the trajectory to the grasp point, ignoring the user's input.

(d) The augmenter grasps the object, and waits for the user to re-align the master with the remote pose.

(e) Finally, the user triggers a re-alignment of the target robot with the master command frame, and lifts the assembly.

Fig. 6: The sequence of events involved in a successful semi-autonomous grasp. (Images taken from an experiment trial).

IV. Experiments

We report here on our initial experience with this system. As currently constructed, the system operates on an unenhanced internet connection. A virtual private network (VPN) was used across this connection to facilitate communication and a secure connection between the sites. Under the best of circumstances, the round trip delay for low-bandwidth telemetry was nominally $200ms$ (averaged over a $30s$ window), with occasional peaks of 2–$6s$. High-bandwidth telemetry like the stereo video streams never arrived under $2s$ after their capture at the remote site. Additionally, the bandwidth limitations caused 75% of the video frames to be dropped during normal execution (averaged over a $30s$ window). In order to aid the operator in visualizing the commanded robot motion, a graphical overlay operates in real time with the master console. The object recognition system would update the displayed hypotheses roughly every $2s$. When characterizing the network during the periods before and after the experiments, we observed discrete jumps in the available network bandwidth which would last for hours before settling back to a nominal state. Due to the complexity and number of routing systems between the two sites, the time of day and day of week had a large impact on the network performance.

We performed experiments to explore the usability of the described system by manipulating structures assembled from magnetically-mating trusses developed at The University of Pennsylvania for autonomous assembly experiments with quadrotor helicopters [22]. These trusses are shown in Fig. 5. The ends of each truss contain four magnets that mate to corresponding points in the nodes. As such, the trusses are more easily assembled than a stiffer structure that would be used in a real application. Still, since magnets do not support shear, the trusses must be precisely oriented in order to mate properly. The arrangement of magnets creates several "local minima" that can cause assembly errors if misaligned. When assembled, the sides of the truss form a $45°$ angle with the node, making teleoperated grasping and alignment challenging even without time-delay.

Procedure: The primary experiment involved the basic manipulation of a pyramid of objects shown at the left of Fig. 2. Users were instructed to grasp the pyramid by one of the legs and raise it above the table. The challenge in this task is simple: in order to stably grasp and manipulate the pyramid, the gripper has to be precisely aligned with the sides of any one truss. Any discrepancy will impart a moment on the truss and may cause it to disengage and the pyramid to collapse.

Results: Several users with varying degrees of experience with the system performed remote manipulation with and without automated grasping. While several operators tried the platform, data was only collected for twelve trials of one user who was not on the research team. This user has extensive experience using the da Vinci® robot over a low-latency connection, but had no experience using it to control the robot used in this study nor experience using the ASCENT prototype. We present only this data to remain unbiased, but note, however, that this was still an informal study. Of the twelve trials, the user was asked to grasp and lift the object in one of two configurations, and either with or without the semi-autonomous actions. The first six trials (Table I) were meant to characterize the accuracy and speed differences with and without the ASCENT system (the "Efficient" aspect), and the second six trials were meant to investigate how easily the user could interrupt the ASCENT system under different circumstances (the "Natural" aspect).

Without automated grasping, the task is quite difficult to perform quickly, or even successfully. As noted, it is possible to grasp the truss in a way that causes the structure to

Trial Order	Mode	Time [s]
1	*assisted*	15.60
2	*assisted*	11.92
3	*assisted*	31.00 (14.12)
4	*manual*	37.88
5	*manual*	41.88
6	*manual*	40.84

TABLE I: Times between trial start and grasp contact in pilot study. Note that in Trial 3, the user did not move with sufficient speed to activate the autonomous intervention, and returned to the initial pose, and attempted again. This second time is in parentheses.

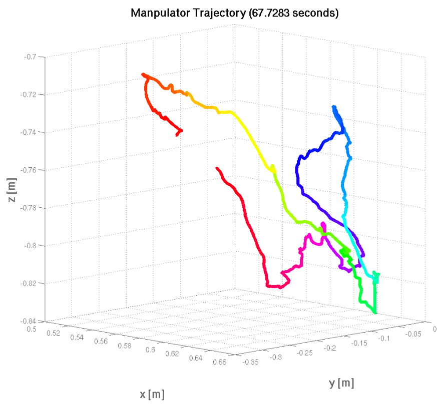


Fig. 7: In this trial, the ASCENT prototype was *disabled*, and the operator was directly controlling the remote manipulator. In this trial, the operator missed the graspable region of the structure, and had to back away and re-grasp it. This can be seen by the two loops (green and purple) in the bottom right of the plot.

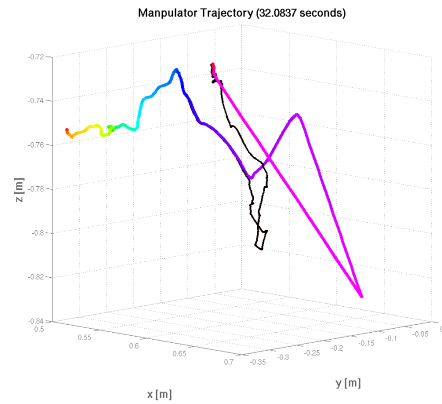


Fig. 8: In this trial, the ASCENT prototype was *enabled*, and assisted the operator to acquire and grasp the object successfully. The first sharp angle in the trajectory designates where the ASCENT augmenter took control, in order to bring the manipulator to the correct approach pose. It is at this point where the actual trajectory (colored) separates from the commanded trajectory (black). The trajectories join when the operator re-takes control.

collapse.

With automated grasping, there was no situation in which the object was incorrectly manipulated, but there were instances where the user successfully grasped the object, and then disengaged the structure after retaking control. In all cases, the user was able to acquire the desired object with a single quick motion when using the ASCENT prototype. This process is relatively straightforward provided the perception process is operating correctly. While we observed numerous false detections during the experiments, these were inconsequential since the operator would only initiate a grasp when he was confident that the detection was correct.

In addition to making the task easier, it also allowed the operators to move with more confidence and complete the grasp acquisition task two to four times faster than when they were unassisted. See Fig. 7 and Fig. 8 for representative trajectories without and with the ASCENT prototype enabled.

Semi-Autonomous Assembly: In addition to the reported trials, the authors also explored the impact of using the semi-autonomous grasping capabilities while assembling and disassembling the structure seen in Fig. 5. While disassembling this structure, an additional piece became loose, and nearly rolled off of the table. The operator then easily "nudged" the structure out of the way to allow the perception system to see the loose truss. Then, the operator initiated an autonomous grasp by reaching for the loose part, and successfully acquired it despite it being precariously balanced on the edge of the table. The capability to accommodate such failure modes and recovery behaviors is what we believe is the true power of our approach.

Qualitative User Experience: Qualitatively, the ease of user experience between the assisted and unassisted modes is dramatic. The da Vinci® master console provides an excellent stereo visualization, and with small time delay, the robot is quite natural to use. However, the time delay is sufficient to force a "move-and-wait" strategy to avoid undesirable contact with the table or manipulables. As a result, grasping, particularly for the inexperienced user, is quite challenging and time consuming. However, with automated grasping, acquiring and manipulating objects becomes relatively easy and allowed the operators to focus more on the task and less on their precise hand movements.

V. Conclusions and Future Work

This paper has demonstrated the use of intent recognition combined with perception-based closed-loop control of grasping primitives. By doing so, we have shown that a relatively simple system, operating on commercially available hardware and communication systems is able to perform reliable transatlantic telemanipulation. Our preliminary experiments have shown that the system makes aspects of assembly and dis-assembly significantly easier than unaugmented manipulation. The impact of effective telemanipulation of complex objects under time delay would be enormous. For example, the on-orbit telemanipulation tasks required by satellite servicing and assembly are too complex to be automated, yet the time-delay and bandwidth limitations inherent in trans-orbital communications greatly impact the effectiveness of teleoperation. At the same time, there are many terrestrial tasks that defy cost-effective automation, and are thus performed by hand. Creating effective systems like the ASCENT prototype for these tasks could provide a way to improve efficiency and reduce potential injury. Our current work is devoted to improving the performance and robustness of the system. Most of the current failure modes relate to the limitations of the perception system, and the use of hand-tuned parameters for the intent recognition.

The latter is easily addressed with improvements to software infrastructure along the lines of [24], while the former will be addressed by using a richer scene parsing system [4]. We also envision the addition of other, higher-level logic and task reasoning to the augmentation executive. Given the promising results reported in this paper, our next steps include polishing the current system and performing a full human subject test to empirically validate the usefulness of the ASCENT prototype.

VI. Acknowledgments

The authors would like to thank our collaborators at DLR and TUM without whom this project would not have been possible. The TUM GSISH exchange program for enabling my visit. Katharina Lang (TUM), Ursula Mühle (TUM), and Rosemary Mountain (JHU) for their help with logistics in Germany. Romain Weitschat, Felix Huber, and the rest of the safe-robots team for their help with system integration. The DLR administrators for allowing access to DLR resources and encouraging this international collaboration.

References

[1] A. Albu-Schäffer, C. Ott, and G. Hirzinger, "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots," *Int. J. of Robotics Research*, vol. 26, pp. 23–39, 2007.

[2] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mosenlechner, W. Meeussen, and S. Holzer, "Towards autonomous robotic butlers: Lessons learned with the pr2," 2011, pp. 5568–5575.

[3] C. Borst, C. Ott, T. Wimböck, B. Brunner, F. Zacharias, B. Bäuml, U. Hillenbrand, S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "A humanoid upper body system for two-handed manipulation," *VIDEO, IEEE Int. Conf. on Robotics and Automation (ICRA2007), Rome, Italy*, pp. 2766–2767, 2007.

[4] M. Brucker, S. Léonard, T. Bodenmuller, and G. Hager, "Sequential scene parsing using range and intensity information," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5417–5424.

[5] H. Bruyninckx, "Open robot control software: the orocos project," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 2523–2528.

[6] N. R. B. Committee on Science Opportunities Enabled by NASA's Constellation System, *Launching Science: Science Opportunities Provided by NASA's Constellation System*. The National Academies Press, 2008. [Online]. Available: http://www.nap.edu/openbook.php?record_id=12554

[7] A. Committee on the Assessment of Options for Extending the Life of the Hubble Space Telescope Space Studies Board and S. E. Board, *Assessment of options for extending the life of the Hubble Space Telescope: Final Report*. The National Academy Press, 2005.

[8] M. Diftler, T. Ahlstrom, R. Ambrose, N. Radford, C. Joyce, N. De La Pena, A. Parsons, and A. Noblitt, "Robonaut 2?initial activities on-board the ISS," in *Aerospace Conference, 2012 IEEE*. IEEE, 2012, pp. 1–12.

[9] W. R. Ferrell and T. B. Sheridan, "Supervisory control of remote manipulation," *Spectrum, IEEE*, vol. 4, no. 10, pp. 81 –88, oct. 1967.

[10] J. Funda and R. Paul, "Teleprogramming: overcoming communication delays in remote manipulation," in *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, Nov 1990, pp. 873 –875.

[11] G. Hager and B. Wegbreit, "Scene parsing using a prior world model," *The International Journal of Robotics Research*, 2011.

[12] R. Held, A. Efstathiou, and M. Greene, "Adaptation to displaced and delayed visual feedback from the hand." *Journal of Experimental Psychology*, vol. 72, no. 6, p. 887, 1966.

[13] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Rotex-the first remotely controlled robot in space," in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 2604–2611.

[14] G. Hirzinger, K. Landzettel, B. Brunner, M. Fischer, C. Preusche, D. Reintsema, A. Albu-Schäffer, G. Schreiber, and B. Steinmetz, "Dlrs robotics technologies for on-orbit servicing," *Advanced Robotics*, vol. 18, no. 2, pp. 139–174, 2004.

[15] G. Hirzinger, K. Landzettel, and C. Fagerer, "Telerobotics with large time delays-the rotex experience," in *Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on*, vol. 1, sep 1994, pp. 571 –578 vol.1.

[16] J. Hoffmana, D. Millerb, and J. Grunsfeld, "Astronaut and robotic maintenance and upgrades of major space assets," in *astro2010: The Astronomy and Astrophysics Decadal Survey*, vol. 2010, 2009, p. 22P.

[17] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *RoboCup*, 2011, pp. 306–317.

[18] T. Imaida, Y. Yokokohji, M. Oda, T. Yoshikwa, *et al.*, "Ground-space bilateral teleoperation experiment using ets-vii robot arm with direct kinesthetic coupling," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 1031–1038.

[19] C. Joppin and D. Hastings, "On-orbit upgrade and repair: The hubble space telescope example," *Journal of spacecraft and rockets*, vol. 43, no. 3, pp. 614–625, 2006.

[20] A. Knoll, C. Mayer, H.and Staub, and R. Bauernschmitt, "Selective automation and skill transfer in medical robotics: a demonstration on surgical knot-tying," *Int. J. Med. Robotics Comput. Assist. Surg.*, 2012.

[21] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *ICRA*, 2011, pp. 1817–1824.

[22] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction with quadrotor teams," *Autonomous Robots*, vol. 33, pp. 323–336, 2012.

[23] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, 2008.

[24] N. Padoy and G. D. Hager, "Human-machine collaborative surgery using learned models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5285–5292.

[25] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3d geometry matching for grasping of known objects in cluttered scenes," *I. J. Robotic Res.*, vol. 31, no. 4, pp. 538–553, 2012.

[26] S. Parusel, S. Haddadin, and A. Albu-Schäffer, "Modular state-based behavior control for safe human-robot interaction: a lightweight control architecture for a lightweight robot," pp. 4298–4305, 2011.

[27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009.

[28] C. Sayers, R. Paul, L. Whitcomb, and D. Yoerger, "Teleprogramming for subsea teleoperation using acoustic communication," *Oceanic Engineering, IEEE Journal of*, vol. 23, no. 1, pp. 60 –71, jan 1998.

[29] T. Sheridan, "Space teleoperation through time delay: Review and prognosis," vol. 9, no. 5. IEEE, 1993, pp. 592–606.

[30] M. Stein, R. Paul, P. Schenker, and E. Paljug, "A cross-country teleprogramming experiment," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 2, aug 1995, pp. 21 –26 vol.2.

[31] E. Stoll, J. Letschnik, U. Walter, J. Artigas, P. Kremer, C. Preusche, and G. Hirzinger, "On-orbit servicing," *Robotics and Automation Magazine, IEEE*, vol. 16, no. 4, pp. 29–33, 2009.

[32] C. Tzafestas, "Teleplanning by human demonstration for vr-based teleoperation of a mobile robotic assistant," in *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*. IEEE, 2001, pp. 462–467.

[33] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2074–2081.

[34] S. Winkelbach, S. Molkenstruck, and F. M. Wahl, "Low-cost laser range scanner and fast surface registration approach," in *DAGM-Symposium*, 2006, pp. 718–728.