

# Multi-Phase Multi-Modal Haptic Teleoperation

Maximilian Mühlbauer<sup>1,2</sup>, Franz Steinmetz<sup>2</sup>, Freck Stulp<sup>2</sup>, Thomas Hulin<sup>2</sup>, Alin Albu-Schäffer<sup>1,2</sup>

**Abstract**— Virtual Fixtures facilitate teleoperation, for instance by guiding the human operator. Developing these Virtual Fixtures in tasks with tight tolerances remains challenging. Fixtures with a high stiffness allow for more precise guidance, whereas a lower stiffness is required to allow for corrections. We observed that many assembly operations can be split into different phases – approaching, positioning, in-contact manipulation – each with different accuracy requirements. Therefore, we propose to use multi-modal fixtures, satisfying the different requirements of these phases: i.e. a position-based Trajectory Fixture for approaching and a more accurate Visual Servoing Fixture for the positioning phase. A state estimation and arbitration component ensures smooth transitions between the fixtures to provide optimal support for the operator and to achieve global availability paired with local precision at the same time. It also allows a high stiffness to be used throughout, thus achieving good guidance for all phases. The approach is validated in an application from a space scenario, consisting of the assembly of a CubeSat subsystem. The empirical results from a pilot study on this task show that our approach is faster and requires less interaction force from the operator than the baseline method.

## I. INTRODUCTION

Virtual Fixtures (VFs) enhance teleoperation by haptically guiding the human operator, or keeping the robot out of forbidden regions [1]–[3]. Most fixtures are based on a geometric definition and are statically linked to e.g. some perceptual input. Designing VFs is challenging, especially when high precision is required, or when the target pose is not known a priori with sufficient accuracy.

Take for instance in-orbit microsatellite assembly [4], [5] which is our use case throughout the paper. This task requires precise teleoperation to assemble CubeSat subsystems. The subsystem is a printed circuit board (PCB) which is inserted into another PCB called backplane, as illustrated in Fig. 1. The poses of the PCB connectors which are mated in the process are supplied by a digital twin. Their estimates may deviate several millimeters from the actual connector poses, but a successful mating of the connectors requires an accuracy of less than 0.7 mm.

In such scenarios, the design of VFs is challenging due to two contrasting requirements. If the stiffness is set too high, it might be impossible for the operator to compensate for incorrectly estimated target positions to complete the task. With a low fixture stiffness on the other hand, the operator faces an increased workload to achieve the required precision, even when the fixture’s target pose is correct, as

<sup>1</sup> Department of Informatics, Sensor Based Robotic Systems and Intelligent Assistance Systems, Technical University Munich, Garching, Germany maximilian.muehlbauer@tum.de

<sup>2</sup> German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchner Str. 20, 82234 Weßling, Germany

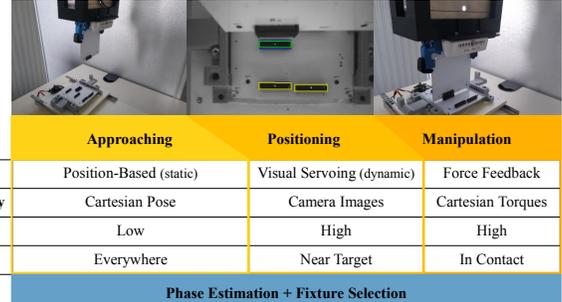


Fig. 1. Overview of the proposed approach from a user perspective. Our multi-modal virtual fixture approach is designed to optimally support the user during the “Approaching” and “Positioning” phases. No special fixture is implemented yet for the “Manipulation” phase but force feedback helps the user during this phase.

the guiding forces are low. Methods for live updates of virtual fixtures based on vision data have been proposed [6] to alleviate this issue. Vision has also been used directly for a VF based on visual servoing [7]. In many cases however – our CubeSat assembly being one of them – the target is not visible at the beginning of a manipulation task. A VF based on visual servoing can thus only be used in a later phase of the assembly task, when the target is in the field of view.

A key insight in this paper is that many assembly processes consist of multiple phases with different requirements on accuracy and stiffness of the virtual fixtures, as illustrated in Fig. 1. Furthermore, these different phases can be based on different modalities. As summarized in Fig. 1, static position-based fixtures work throughout the workspace and allow the target position to be approached, which is a phase with coarse precision requirements. Fixtures based on visual servoing achieve the high accuracy required for the positioning phase. However, they require the target to be visible by the in-hand camera and thus only work near the target in a later phase.

In this paper, we propose a novel phase-dependent combination of position- and vision-based fixtures, which combines the advantages of both fixture types, and allows for stiff VF control throughout all phases. The main contributions are the following: 1) Combining position and vision-based fixtures in one framework; 2) Developing an arbitration scheme that ensures smooth transitions between them; 3) Implementing the approach on a teleoperation system based on light-weight robots; 4) Empirically demonstrating the efficacy of the approach in a pilot study on the subsystem assembly of a CubeSat [4], [5].

## II. RELATED WORK

Virtual Fixtures have first been presented in the seminal work by Rosenberg [1] comparing them to the real-world

analogon of a ruler. Similarly, virtual fixtures can be designed in a way to help the user achieve goals by either restricting admissible motions (forbidden region virtual fixtures) or by guiding the user, e.g. along a trajectory (guiding virtual fixtures) [8]. Their usefulness in the case of time-delay has also been studied [9].

### A. Vision-based Virtual Fixtures

Traditionally, virtual fixtures are defined using fixed 3D-coordinates. In changing environments or when higher precision is needed, vision can help to better position virtual fixtures. In [6], visual information is used to detect limit switches. A trajectory virtual fixture is then generated guiding the operator from the current pose to this target. Users are able to switch between different fixtures with hard assignments and a controller ensuring passivity of the approach.

This vision-based virtual fixture generation approach is generalized in [10] by allowing a human operator to select visual features and generate useful tool constraints. Using this toolbox one can e.g. align the axis of a wrench with the axis of the nut to be manipulated.

In this work, we focus on using vision-based methods to continuously update the estimated target, thus allowing a higher precision. An early work using visual servoing techniques for 2D line following is presented in [11], [12]. At each camera frame, the difference between the target pose or reference curve and the current end-effector pose is estimated. Based on this error, a force field is generated which pulls the user towards the target or curve while allowing motion in direction of the curve.

In [7], a model-free visual servoing algorithm is used to guide the robot towards a target position. Two operational modi – called “Teleoperation Before Visual Servoing” and “Teleoperation During Visual Servoing” – are used. The former allows for estimating the Jacobian needed for visual servoing during the free-space teleoperated motion and completing the task using pure visual servoing, while the latter uses a virtual spring between current joint angles and those calculated by the visual servoing algorithm.

### B. Multi-Phase Manipulation

Traditionally, the problem of increased accuracy requirements during parts of a manipulation task has been addressed by mounting micro-manipulators at the end of robots [13], [14]. This approach combines the larger workspace of a bigger robot with the increased precision of a specifically designed micro-manipulating robot.

More recently, methods for switching between different visual servoing strategies have been proposed. The methods learn different CNNs for both the coarse approaching phase as for precise motion [15], [16]. In [17], this strategy is simplified by using inverse kinematics to move to a detected “bottleneck” pose from where a learned controller uses vision data to perform precise manipulation. This combination of a position-based first phase and a vision-based precise manipulation phase comes closest to our idea of a multi-phase

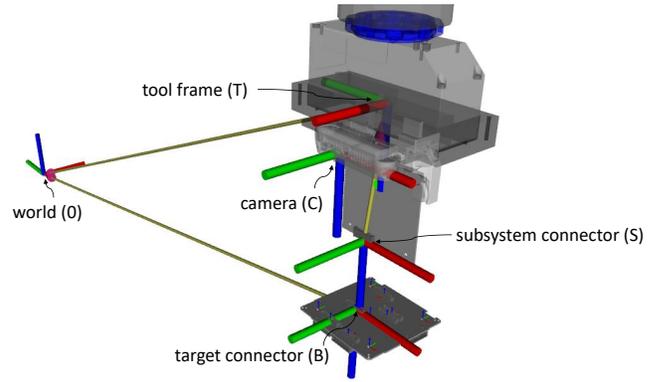


Fig. 2. Relevant coordinate systems for our method ( $x$  in red,  $y$  in green,  $z$  in blue). The tool frame has its origin inside the gripper body. We use a classic camera coordinate system with forward-facing  $z$  axis. The subsystem connector, target connector and world coordinate systems are also shown.

manipulation, however, to the best of our knowledge we are not aware of methods combining these different phases and modalities in telemanipulation.

## III. FUNDAMENTALS

First we introduce the coordinate systems and the haptic coupling of the remote robot with the input device needed for our approach.

### A. Coordinate Systems

The relevant coordinate systems for our method are shown in Fig. 2. For the best performance of haptic teleoperation, a careful selection of the coordinate system in which forces and torques are applied is required. In our case, this is the so-called tool frame  $T$ , which is defined with an offset along the  $z$  axis from the last joint of the robot arm combined with a rotation around the same axis to account for the tool rotation. Its pose in world coordinates  ${}^0_T\mathbf{H}$  is updated at each timestamp using the forward kinematics of the robot. On the remote robot, the tool frame is placed inside the center of mass of the tool. On the input device, we use the position of the handle grasped by the user (Fig. 7).

As we use an in-hand camera configuration, the transformation  ${}^T_C\mathbf{H}$  from the camera frame  $C$  to the tool frame  $T$  is static and calibrated beforehand. Chaining this transformation with  ${}^0_T\mathbf{H}$  can then be used to determine the world coordinates of detected connectors at each timestamp.

### B. Teleoperation Controller

A prerequisite for haptic teleoperation is the teleoperation controller that applies the commands given by the human on the input device to the remote side and provides force feedback. We assume impedance controlled robots that are coupled by a virtual spring-damper mechanism. Wrench notation with  $\mathbf{w} \in \mathbb{R}^6$  allows us to combine force  $\mathbf{f} \in \mathbb{R}^3$  and torque  $\boldsymbol{\tau} \in \mathbb{R}^3$  in one vector [18]

$$\mathbf{w} = \begin{bmatrix} \mathbf{f} \\ \boldsymbol{\tau} \end{bmatrix}. \quad (1)$$

The wrench of the virtual fixtures  $\mathbf{w}_{VF}$  is added to the wrench commanded to the remote robot, becoming

$$\mathbf{w}_{\text{remote}} = \alpha (\mathbf{K}\Delta\mathbf{x} + \mathbf{D}\Delta\dot{\mathbf{x}}) + \mathbf{w}_{VF} \quad (2)$$

where  $\Delta\mathbf{x}$  and  $\Delta\dot{\mathbf{x}}$  denote the current relative displacement and velocity of both robots,  $\mathbf{K}$  the stiffness and  $\mathbf{D}$  the damping. The wrench commanded to the input device results from a transformation with the adjoint of the transformation matrix  $\mathbf{Ad}_{ir}$  that transforms remote robot wrenches into the coordinate system of the haptic input device

$$\mathbf{w}_{\text{input}} = -\alpha \mathbf{Ad}_{ir} \mathbf{w}_{\text{remote}}. \quad (3)$$

This transformation can be updated by the user with a so-called indexing operation, where the remote device is held stationary and only the input device moves. The scaling factor  $\alpha = 0.7$  is used to scale the movements (translations and rotations) of the remote robot to 70% of those of the input device. This allows for a more precise manipulation. To ensure stability, the feedback forces are scaled accordingly.

#### IV. APPROACH: MULTI-PHASE MULTI-MODAL HAPTIC TELEOPERATION

The core of our approach are the multi-modal virtual fixtures and the Arbitrator that adaptively selects these fixtures (Fig. 1). These components are presented in the next sections.

##### A. Position-based Trajectory Fixture

For the ‘‘approaching’’ phase of our method (Fig. 1), we use a static position-based *Trajectory Fixture*. Static means that the fixture is, once being created in world coordinates, not updated anymore during the movement. This fixture does not depend on vision information and is thus useful from the beginning, even when the target connector has not been seen yet. The fixture is represented as a trajectory with associated orientation and stiffness values (Fig. 3). We use a spline from the start pose to the target pose as obtained from the digital twin [5]. In more complex scenarios motion planning could be used to find a collision-free path.

We use an increased stiffness at the beginning and the end of the trajectory to account for higher precision requirements while grasping or plugging the PCB. In between, a reduced stiffness gives the user more motion freedom when less precision is required.

To generate forces, we first calculate the closest point on the trajectory based on the current tool pose. A virtual spring-damper mechanism generates the wrench  $\mathbf{w}_{PB}$  pulling the user towards the trajectory, as used in Section III-B for the teleoperation. As all calculations are performed at each robot control cycle with 1 kHz, the applied forces are orthogonal to the trajectory, thus the user does not feel any force in the direction of the trajectory.

##### B. Visual Servoing Fixture

Once the target connector is visible by the in-hand camera (Fig. 5), the *Visual Servoing Fixture* generates the corrective wrench  $\mathbf{w}_{VS}$  in the  $XY$  plane and around the  $Z$  axis for guiding the operator. This fixture creates guiding

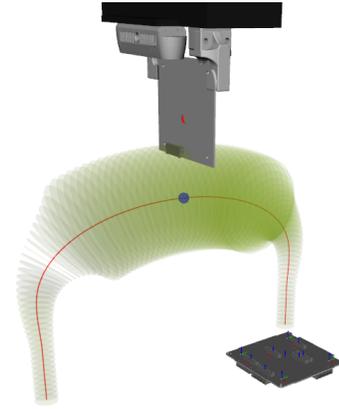


Fig. 3. The position-based Trajectory Fixture. The red line marks the center of the trajectory while the varying translational stiffness is shown by the green circles. The fixture force is pulling the Robot (red dot) orthogonally towards the trajectory (blue dot). Forces are acting on the tool frame, the red and blue dot are however displayed at an offset for visualization purposes.

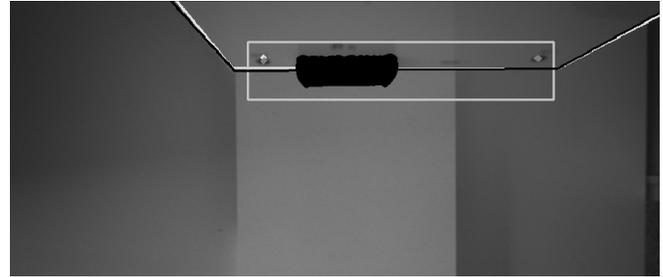


Fig. 4. Image of the subsystem in the in-hand camera. Overlay information shows the mask for the in-hand connector (black filled are), the projected PCB outline with information from the digital twin (white) as well as the corrected PCB outline (black).

forces based on the continuous estimation of the relative transformation  ${}^S_B\mathbf{H}$  between the connector on the grasped PCB and the target connector. This transformation is used to create an attractive virtual fixture correcting the position-based fixture. Our camera allows an accuracy of  $0.2 \text{ mm px}^{-1}$  at the working distance of 10 cm given that the camera is properly calibrated, which is sufficient for our application. Three steps are performed by this fixture:

1) *Grasped Subsystem Detection*: The first step of the detection pipeline, the detection of the grasped subsystem as shown in Fig. 4, already occurs when only the position-based Virtual Fixture is active. Furthermore, a mask is created for the subsystem connector. Grayscale value based thresholding is used to separate the white PCB from the black connectors and the PCB holes. The white rectangle in Fig. 4 visualizes the region of interest for these detections which are highlighted in gray.

Knowing that the centering gripper jaws only allow a displacement in  $Z$  direction as well as a rotation around the  $Y$  axis in camera coordinates, we can calculate the displacement of the grasped subsystem from its expected pose  ${}^S_B\mathbf{H}$  using the detected subsystem holes. This displacement is stored and later used to correct the target connector pose.

Using the same thresholded image, we also extract the connector of the grasped subsystem to create a mask for

target connector detection. This mask is essential, as the target connector is partially hidden during the plugging operation and it is not possible to separate the subsystem connector from the backplane connector, thus degrading detections. This mask is also shown in Fig. 4.

2) *Target Connector Detection*: When the user moves closer to the target, the connectors on the backplane become visible for the in-hand camera (fig. 5). Using OpenCV [19] rectangle detection on the image thresholded as before for the subsystem detection (the subsystem connector mask being applied), we can detect connector candidates. Using the expected and approximately known poses of the target connector, which we obtain from the target pose of the position-based Trajectory Fixture, we can filter valid detections as shown in Fig. 5.

From the width of those detected connectors in the image together with the knowledge about their real world size, we can calculate their coordinates in camera coordinates to

$$c_Z = f \frac{D}{d} \quad c_X = c_Z \frac{x - c_x}{f} \quad c_Y = c_Z \frac{y - c_y}{f}, \quad (4)$$

where  $D$  and  $d$  are the connector width in world units respectively pixels.  $f$  denotes the focal length of the camera,  $x$  and  $y$  the center of the detection in pixels. The rectangle rotation  $\theta$  corresponds to the target connector rotation around the  $Z$  axis of the camera frame. Rotations around the  $X$  and  $Y$  axis are assumed to be zero as the position-based virtual fixture holds the gripper approximately parallel to the target PCB. The target connector pose is then transformed in world coordinates  ${}^0_B\mathbf{H}$  using the tool frame  ${}^0_T\mathbf{H}$  of the image timestamp.

To ensure that the detection also works when the target connector is partially hidden by the subsystem connector, we use the known connector width-to-height-ratio to estimate its center based on the part of the connector which is still visible. As the Visual Servoing Fixture corrects rotations around the  $Z$ -axis, the connector gets gradually hidden along its short axis, which allows the detection to continue

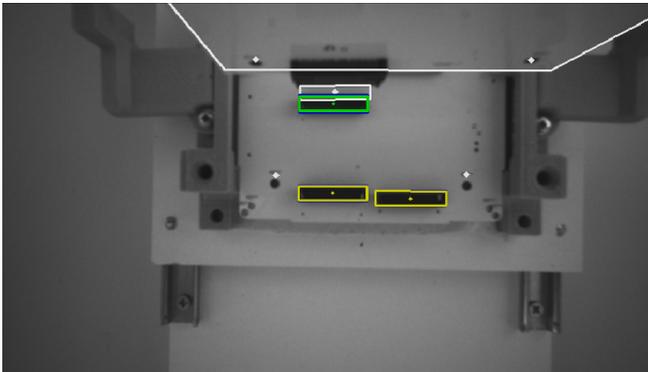


Fig. 5. Image of tracked detections on the backplane viewed from the in-hand camera. The green rectangle marks the tracked detection which is currently used as visual servoing target. The yellow rectangles mark tracked detections. The expected poses of the the in-hand PCB, the connector on the backplane as well as the PCB holes are displayed in white.

almost until the connectors come into physical contact. For increased robustness during the pilot study, we stop updating the estimated target pose once less than 65 % of the target connector are visible.

3) *Target Connector Tracking*: For each raw detection of a 3D connector pose, possible matches with already tracked targets are evaluated. We use a threshold of a maximum translational difference of 5 mm and a rotational difference of 0.04 rad. If a matching tracked target is found, the pose of this target is added to the tracked target. We use exponential smoothing with the factor  $\alpha = \frac{1}{7}$  to filter the detections. For rotations, we are using quaternion slerp for the smoothing operation. As final step, the target pose closest to the expected target pose is being sent to the real-time controller for impedance-based visual servoing.

4) *Impedance-Based Servoing*: The implementation of impedance-based visual servoing closely follows [7]. A virtual spring wrench  $\mathbf{w}_{VS}$  acting on the transformation

$${}_{B'}^{S'}\mathbf{H} = {}_{S'}^S\mathbf{H} \cdot {}_S^0\mathbf{H}^{-1} \cdot {}_B^0\mathbf{H} \quad (5)$$

pulls the subsystem connector towards the target connector.  ${}_{S'}^S\mathbf{H}$  is the stored subsystem offset,  ${}_S^0\mathbf{H}$  the nominal subsystem pose obtained using the robot kinematics and  ${}_B^0\mathbf{H}$  the tracked target. We only output forces in the  $XY$  plane of the tool frame as well as the torque around the  $Z$  axis. Rotations around the  $X$  and  $Y$  axis cannot be observed by our servoing method and are thus controlled by the position-based Trajectory Fixture. Motion along the  $Z$  axis is, as for the position-based Trajectory Fixture, entirely controlled by the operator. This wrench is being applied in coordinates of the subsystem connector to allow for a more precise positioning.

We calculate the wrench inside the real-time controller to be able to generate force feedback at the controller rate of 1 kHz for a good haptic experience. As the vision algorithm is running at only 30 Hz, we again apply a low pass filter to the target pose inside the realtime model to ensure smooth updates. This rate transition is possible as the target connector pose is estimated in world coordinates and assumed to be only slowly moving<sup>1</sup>. A good time-synchronization of robot and camera measurements is however still necessary to be able to achieve consistent position measurements and thus be able to implement a spring with high stiffness. We observe a latency of 15 ms to 30 ms which would in the uncorrected case lead to positional deviations high enough to cause unstable vibrations of the spring-damper system.

### C. Arbitrating Position- and Vision-based Fixtures

The *Arbitrator* component is required to estimate the current state and select a virtual fixture appropriately. Depending on the manipulation phase and the availability of the Visual Servoing Fixture, we assign weights to the different fixtures. The resulting virtual fixture wrench is then calculated to

$$\mathbf{w}_{VF} = \sum_i \beta_i \mathbf{w}_i \quad (6)$$

<sup>1</sup>Please see the supplementary video for how the fixture works with a moving target.

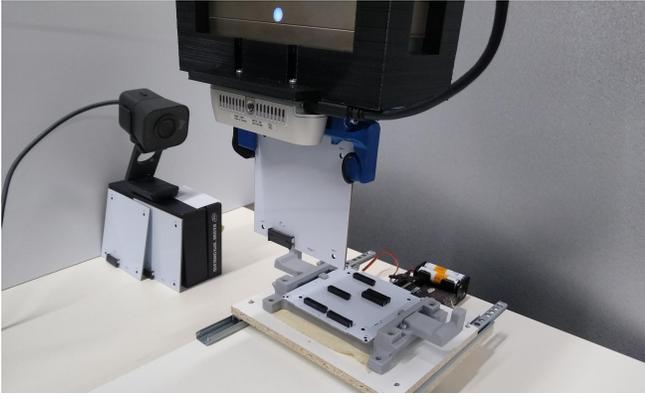


Fig. 6. Experimental scenario for the subsystem assembly. The gripper holds a PCB to be inserted into the backplane which is placed on the table. The backplane can be moved on a set of rails to simulate position inaccuracies.

where  $\beta_i$  denotes the weight of fixture  $i$  and  $w_i$  is the wrench calculated by the respective fixture. In our case with one position-based and one Visual Servoing Fixture, this results in

$$w_{VF} = \beta_{PB} w_{PB} + \beta_{VS} w_{VS}. \quad (7)$$

We ensure that  $\beta_{PB} + \beta_{VS} = 1$  to allow for a smooth transition between the position-based and the Visual Servoing Fixture, which both share the same Cartesian stiffness values.

For the problem of CubeSat subsystem assembly, we found that a determination of the weights to

$$\beta_{PB} = \begin{cases} 1 & d > 0.08 \text{ m or target not found} \\ \frac{d-0.03}{0.05} & 0.08 \text{ m} \geq d > 0.03 \text{ m} \\ 0 & d \leq 0.03 \text{ m} \end{cases} \quad (8)$$

$$\beta_{VS} = 1 - \beta_{PB} \quad (9)$$

with the distance  $d$  to the plugging position is sufficient. Note that we always set the rotational components around the  $X$  and  $Y$  axis to 1 for  $\beta_{PB}$  and to 0 for  $\beta_{VS}$  as the vision system is not able to reliably determine those rotations.

## V. EVALUATION

To evaluate the proposed approach, we perform a pilot study with 10 participants<sup>2</sup> on the task of assembling of a CubeSat subsystem [5] as seen in Fig. 6. We hypothesize that human operators are able to perform this assembly task both quicker and with less required force compared to only a position-based fixture, even when there is a translational or rotational perturbation of the known plugging position, which should also lead to a higher satisfaction of the operator.

### A. Teleoperation Setup

We use one arm of the bimanual haptic device HUG [20] as haptic input device with force feedback while the other arm is used as remote robot. On the remote side, a gripper

<sup>2</sup>All participants are DLR employees with experience on haptic interaction with LWR robots so no special ethic permission is required. Permission for the questionnaires was obtained from the works council and data protection of DLR, participants gave their informed consent.



Fig. 7. Workplace for the human teleoperator. A LWR arm in mirrored joint configuration is used as haptic input device (top right section of the image). Different camera views (right screen: front view, middle screen: top view and visualization, laptop: side view) are available for the operator. The left screen displays the vision detection algorithm and is hidden during the user study.

with centering jaws holds the subsystem to be plugged into the backplane. A separator prevents the operator from viewing the remote side directly. Instead, for this purpose, cameras are installed at an angle of  $90^\circ$  to allow for a precise positioning along both the  $X$  and  $Y$  axis. Fig. 7 shows the workplace of the human operator with both camera views, a view of the 3D robot model as well as the in-hand camera and the arm used as input device.

With an allowed positional offset of only 0.7 mm as well as a maximum angular deviation of  $4^\circ$  (short axis) respectively  $2^\circ$  (long axis) the operator needs to be very precise. Additionally, a force of 25 N is required to successfully mate the connectors.

To be able to test the behavior of our fixtures with small positional ( $\Delta_{pos}$ ) and rotational ( $\Delta_{rot}$ ) perturbations from the known target pose, we mount the backplane on rails which allow to move the PCB by  $\Delta_{pos} = \pm 5$  mm both towards the front and the back. A rotational perturbation of  $\Delta_{rot} = \pm 6^\circ$  is implemented in software by changing the target rotation of the fixture. We program start and target poses into the digital twin with the backplane being at the center of the possible motion range on the rails.

### B. Experimental Design

Introduction	Scenario 1	Scenario 2	Scenario "Rotation"
<ul style="list-style-type: none"> <li>• Introductory Questionnaire</li> <li>• Test trials for familiarization</li> </ul>	<ul style="list-style-type: none"> <li>• 5 plugging trials</li> <li>• System Usability Scale</li> <li>• Open Questions</li> </ul>	<ul style="list-style-type: none"> <li>• 5 plugging trials</li> <li>• System Usability Scale</li> <li>• Choice: Fixture 1 or 2</li> <li>• Open Questions</li> </ul>	<ul style="list-style-type: none"> <li>• 5 plugging trials</li> <li>• System Usability Scale</li> <li>• Open Questions</li> </ul>

Fig. 8. Experimental procedure for evaluating the proposed method. For "Scenario 1/2" we alternate between *Vision-Corrected Position* and *Static Position*. The order of the 5 plugging trials inside each block is randomized.

The full experimental procedure is shown in Fig. 8. Participants are first given an introduction to the system and camera setup, and we let them test the teleoperation task using the Trajectory Fixture with no positional or rotational perturbations until they feel sufficiently comfortable with the system and task. This reduces the otherwise observable learning curve over the trials.

In the actual experiments, we use three different assistive scenarios, with different combinations of the fixtures introduced in section IV. *Vision-Corrected Position*: the novel multi-modal approach proposed in Section IV. *Static*

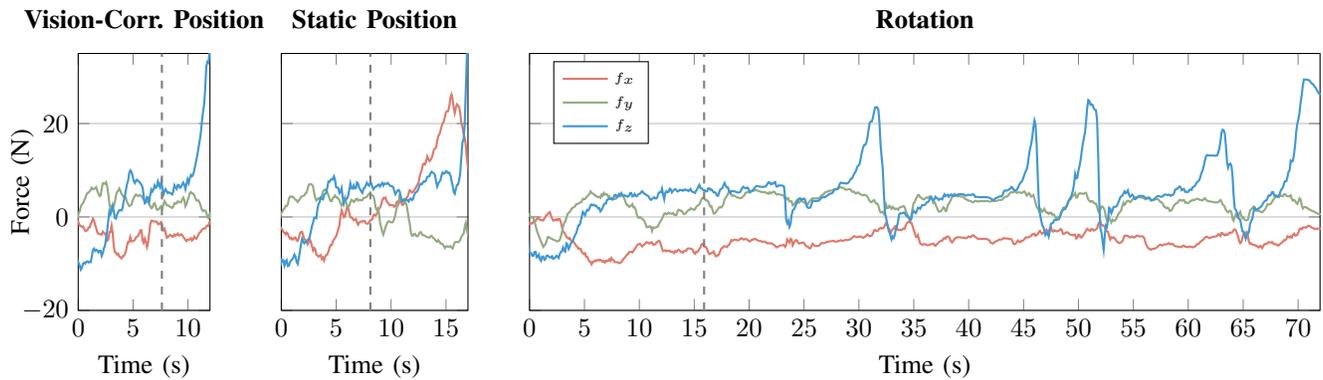


Fig. 9. Teleoperation forces of one subject for all VFs from the start until successful plugging for a perturbation  $\Delta_{\text{pos}} = 5$  mm. The dashed gray line symbolizes the start of the manipulation phase at 2 cm above the plugging position.

*Position:* using only the Trajectory Fixture. *Rotational:* using the Trajectory Fixture with only guidance for the rotational degrees of freedom enabled, the translation being purely controlled by teleoperation. Comparing *Vision-Corrected Position* with *Static Position* allows to single out effects from the Visual Servoing Fixture and the arbitration. *Rotational* serves as a further baseline for both methods. As pure teleoperation without any VF is too challenging for the user, this scenario provides a fixture with rotational guidance.

Every assistive scenario is assigned to one *block*, each of which consists of performing the plugging operation five times, once with no positional or rotational perturbation and then two times each with these perturbations. The order of those perturbations is chosen randomly for each block and each participant. In the first block, we use either *Vision-Corrected Position* or *Static Position*, and for the second block the other. This avoids learning effects that could bias in favor of either the full or the trajectory-only fixture. The last block always uses *Rotational*, because we found that even with experienced teleoperators a lot more effort is needed to perform the task successfully. We would thus expect distortions in the subjective evaluation of the method tested directly afterwards, which would shadow the differences between the *Vision-Corrected Position* and the *Static Position* scenarios.

After each plugging operation, we ask the participants to rate their workload using the raw NASA TLX questionnaire [21], after each block we examine the usability of the current scenario using the System Usability Scale (SUS) [22] as well as asking for open comments about the method. After the subject has tested the second scenario, we additionally ask them which they prefer and to justify their choice.

### C. Objective Results: Overview

As objective data, we record the *teleoperation force* which is rendered to the haptic input device and the *completion time* which we define as the time it takes from 2 cm above the connector until the subsystem is successfully plugged.

Fig. 9 shows representative trials of one subject for each scenario with the position-based Trajectory Fixture having a perturbation  $\Delta_{\text{pos}} = 5$  mm. After the user comes close to the

target (symbolized by a dashed gray line), the main effect of the different fixtures can be seen. For the *Vision-Corrected Position* scenario, users could mostly plug the connector without issues. The start of the plugging procedure can be seen by an increasing teleoperation force in  $z$  direction, which is the insertion force required by the connector.

For the *Static Position* scenario, a teleoperation force of 20 N is needed to overcome the position inaccuracy of 5 mm. The user needs to apply this force on the input device to correct against the perturbed virtual fixture. Users also mostly needed more time to complete in this scenario.

The trial of the baseline *Rotation* scenario represents an extreme case where multiple attempts are needed to find the correct plugging position, which can be seen by multiple peaks of force in  $z$  direction. However also in general more time is needed to finish the task with this VF.

One trial of one subject with the *Rotation* scenario was unsuccessful, this trial is thus removed from the quantitative and subjective analysis. With all other trials, the plugging succeeded. We exclude one subject from the subsequent statistical analysis because of their much higher completion time and standard deviation for the *Vision-Corrected Position* scenario which can be explained by insufficient training before the experiment combined with a much more ergonomic arm position for the *Static Position* scenario.

### D. Objective Results: Quantitative Evaluation

For a quantitative evaluation, we perform a repeated measures ANOVA for the dependent variables *completion time*, *teleoperation force* and *teleoperation torque* comparing the *Vision-Corrected Position* and the *Static Position* scenarios with the within factors “scenario” (*Vision-Corrected Position*, *Static Position*) and “perturbation” (*center*, *front*, *back*, *left rot.*, *right rot.*). Please see fig. 10 for an overview and Table I for the statistical analysis.

For *completion time*, we find a medium effect (Cohen effect size  $d = 0.62$ ). For *teleoperation force* ( $d = 1.3$ ) and *teleoperation torque* ( $d = 1.1$ ) we find large effects.

### E. Subjective Results: Questionnaires

Fig. 11 shows the raw NASA TLX values and significant differences between the *Vision-Corrected Position* and *Static*

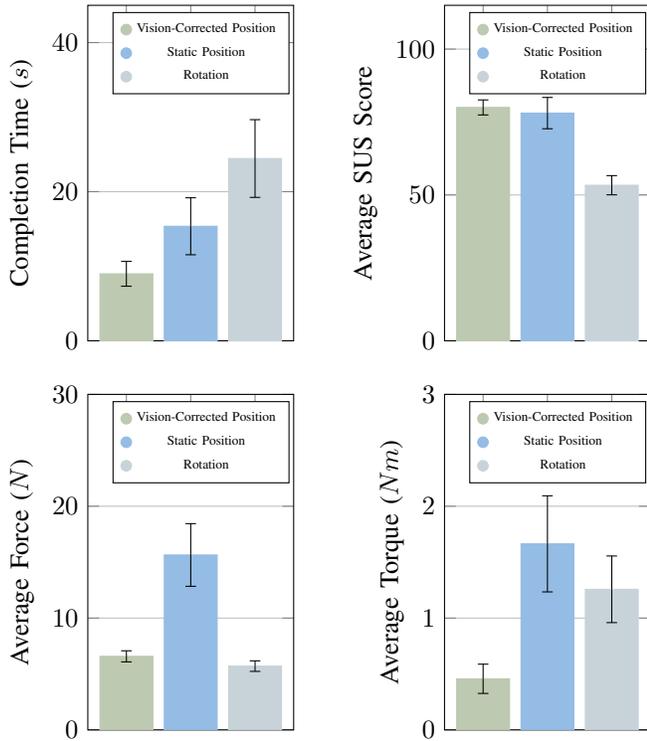


Fig. 10. Mean and 95 % confidence interval for the completion time (left top), SUS (right top), teleoperation force (bottom left) and teleoperation torque (bottom right) averaged over the 10 participants for the different scenarios.

TABLE I

ANOVA RESULTS COMPARING THE VISION-CORRECTED POSITION AND STATIC POSITION SCENARIOS.

Dependent Variable	Factor	df1	df2	F	p
Completion Time	Scenario	1	8	5.33	< .05
	Perturbation	4	32	3.39	< .05
	Fixt. * Pert.	4	32	1.76	> .05
Teleop. Force	Scenario	1	8	223.96	< .001
	Perturbation	4	32	11.23	< .05
	Fixt. * Pert.	4	32	13.68	< .001
Teleop. Torque	Scenario	1	8	195.27	< .001
	Perturbation	4	32	33.31	< .001
	Fixt. * Pert.	4	32	68.01	< .001

Position scenarios using a paired t-test. We compare the task load values between the scenarios matching each of the position- and rotation perturbations. For  $p < 0.05$  we get significant results for the mental, physical and effort items as well as for the average score. For the items temporal, performance and frustration no significant differences could be found.

For the SUS, no significant differences between the Vision-Corrected Position and the Static Position scenarios could be found. They get a rating of  $80 \pm 9.9$  (SD) respectively  $78 \pm 20.5$  (SD) (cf. fig. 10), with values above 68 generally being considered above average [23].

Each of the 10 participants selected the proposed Vision-Corrected Position scenario over the Static Position scenario in direct comparison. Most subjects justified this choice by

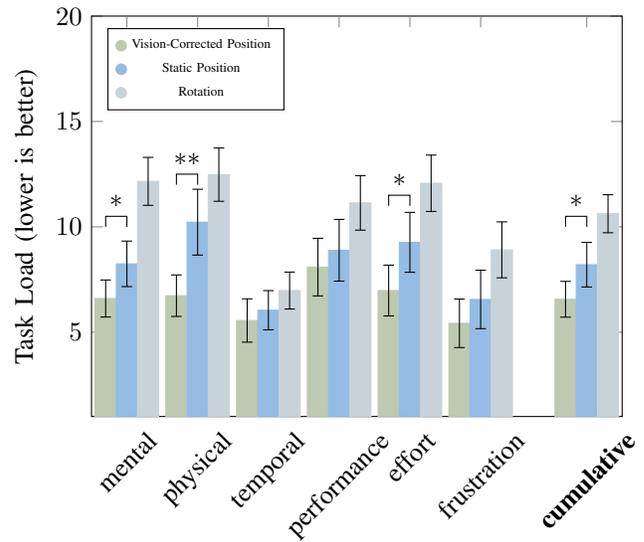


Fig. 11. Raw NASA TLX scores and 95 % confidence interval for the different scenarios. Brackets symbolize significant differences, \* above the bracket denotes  $p < 0.01$  and \*\* denotes  $p < 0.001$ . Values are obtained using a paired t-test for related samples.

the much lower needed forces and torques for this scenario.

#### F. Discussion

In the pilot study, we compared the proposed Vision-Corrected Position scenario with the Static Position scenario and, as baseline, a Rotation-only guidance.

We found that even though the the Static Position scenario already proves to be very valuable for the task of subsystem assembly for a CubeSat, the proposed Vision-Corrected Position scenario has advantages, which both the objective (Table I) and subjective results (Fig. 11) show. The participants were able to complete the teleoperation tasks faster and with less physical effort because they had to apply less forces against the virtual fixtures (Fig. 10). Strong interaction effects of Scenario \* Perturbation for the dependent variable *teleoperation force* and *teleoperation torque* highlight the effect of perturbations on the *Static Position* scenario. We do not find this interaction for *completion time* which highlights that the user's performance measured in this quantity is not so much influenced by the perturbation even though those perturbations significantly increase the physical demand for the *Static Position* scenario.

Only one participant was able to perceive the transition from the Trajectory Fixture to the Visual Servoing Fixture in some of their trials, the other participants only noticed that the VF brought them closer to the target. Force measurements (Fig. 9) do not show any spikes or discontinuities when the transition from the position-based Trajectory Fixture to the Visual Servoing Fixture is happening. This lets us conclude that the Arbitrator ensures the smoothness we expect from a seamless integration of these multi-modal fixtures.

However, we have found that under challenging conditions for the vision system the method could fail as we do not take vision quality into account. For example, in one case with a

rotational perturbation ( $\Delta_{rot}$ ) applied, a participant of the user study was moving too fast for the vision system to correct the rotation. The performance may also be impaired when users command motions which hide the connector early. We think those shortcomings can be mitigated by learning the Arbitrator function as well as implementing more advanced control strategies for the Visual Servoing Fixture [24].

In our target scenario, the method would be deployed to an in-orbit factory. Time delay for the communication from space to Earth would then be inevitable and also affect the teleoperation. Passivity-based control [25] can then be used to assure stability of the teleoperated system. As our fixtures are implemented on the remote robot, we would still expect a good performance of the system. Nonetheless, approaches for partial automation could prove useful.

## VI. CONCLUSION

This paper proposes a multi-modal virtual fixture approach to aid in teleoperation tasks with high precision requirements, achieving global availability paired with local precision at the same time. A position-based Trajectory Fixture is combined with a Visual Servoing Fixture using an arbitration component to select the optimal virtual fixture support during the current assembly phase and to enable automatic, smooth and seamless transition between these fixtures. A pilot study with 10 participants shows very promising results regarding usability and performance improvement of the human operator in comparison to static fixtures. The approach also proved to work robustly for different users who were not aware of the operation principle or limitations of the vision system.

Our approach is designed for easy generalization to other similar assembly tasks as well by adapting the visual detector. Learning techniques could help creating VFs to support the operator during the “Manipulation Phase”. We are also evaluating more advanced arbitration functions than eq. (8) taking uncertainties of the respective fixture into account.

## ACKNOWLEDGMENTS

The results presented here were achieved within the framework of the AI-In-Orbit-Factory project funded by the Federal Ministry for Economic Affairs and Energy (BMWi) as well as the Helmholtz AI project LearnGraspPhases. We thank João Silvério for feedback on the manuscript.

## REFERENCES

- [1] L. Rosenberg, “Virtual fixtures: Perceptual tools for telerobotic manipulation,” in *Proceedings of IEEE Virtual Reality Annual International Symposium*, 1993, pp. 76–82.
- [2] G. Raiola, S. Sanchez Restrepo, P. Chevalier, P. Rodriguez-Ayerbe, X. Lamy, S. Tliba, and F. Stulp, “Co-manipulation with a Library of Virtual Guiding Fixtures,” *Autonomous Robots*, pp. 1573–7527, Jun. 2018.
- [3] K. Hagmann, A. Hellings-Kuß, J. Klodmann, R. Richter, F. Stulp, and D. Leidner, “A digital twin approach for contextual assistance for surgeons during surgical robotics training,” *Frontiers in Robotics and AI*, vol. 8, 2021.
- [4] T. Weber Martins, A. Pereira, T. Hulin, O. Ruf, S. Kugler, A. Giordano, R. Balachandran, F. Benedikt, J. Lewis, R. Anderl *et al.*, “Space factory 4.0—new processes for the robotic assembly of modular satellites on an in-orbit platform based on “industrie 4.0” approach,” in *Proceedings of the International Astronautical Congress, IAC*, 2018.
- [5] F. Kempf, M. S. Mühlbauer, T. Dasbach, F. Leutert, T. Hulin, R. Radhakrishna Balachandran, M. Wende, R. Anderl, K. Schilling, and A. O. Albu-Schäffer, “AI-In-Orbit-Factory - AI approaches for adaptive robotic in-orbit manufacturing of modular satellites,” in *Proceedings of the International Astronautical Congress, IAC*, 2021.
- [6] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, “Enhancing bilateral teleoperation using camera-based online virtual fixtures generation,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1483–1488.
- [7] L. Wu, K. Wu, and H. Ren, “Towards hybrid control of a flexible curvilinear surgical robot with visual/haptic guidance,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 501–507.
- [8] S. A. Bowyer, B. L. Davies, and F. R. y Baena, “Active constraints/virtual fixtures: A survey,” *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2013.
- [9] L. B. Rosenberg, “The use of virtual fixtures to enhance operator performance in time delayed teleoperation,” Air Force Material Command, Wright-Patterson Air Force Base, OH., Tech. Rep., 1993.
- [10] V. Pruks and J.-H. Ryu, “A framework for interactive virtual fixture generation for shared teleoperation in unstructured environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 234–10 241.
- [11] G. D. Hager, “Human-machine cooperative manipulation with vision-based motion constraints,” in *Visual Servoing via Advanced Numerical Methods*, G. Chesi and K. Hashimoto, Eds. London: Springer London, 2010, pp. 55–70.
- [12] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, “Vision-assisted control for manipulation using virtual fixtures,” *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 953–966, 2004.
- [13] A. Sharon and D. Hardt, “Enhancement of robot accuracy using endpoint feedback and a macro-micro manipulator system,” in *1984 American Control Conference*, 1984, pp. 1836–1845.
- [14] S. Salcudean and C. An, “On the control of redundant coarse-fine manipulators,” in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 1834–1840 vol.3.
- [15] Z. Zhuang, J. Leitner, and R. Mahony, “Learning real-time closed loop robotic reaching from monocular vision by exploiting a control lyapunov function structure,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4752–4759.
- [16] P. Raj, V. P. Nambodiri, and L. Behera, “Learning to switch CNNs with model agnostic meta learning for fine precision visual servoing,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 210–10 217.
- [17] E. Valassakis, N. Di Palo, and E. Johns, “Coarse-to-fine for sim-to-real: Sub-millimetre precision across wide task spaces,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5989–5996.
- [18] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [19] G. Bradski, “The OpenCV Library,” *Dr. Dobbs’s Journal of Software Tools*, 2000.
- [20] T. Hulin, K. Hertkorn, P. Kremer, S. Schätzle, J. Artigas, M. Sagardia, F. Zacharias, and C. Preusche, “The DLR bimanual haptic device with optimized workspace,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3441–3442.
- [21] S. G. Hart and L. E. Staveland, “Development of nasa-tlx (task load index): Results of empirical and theoretical research,” in *Advances in psychology*. Elsevier, 1988, vol. 52, pp. 139–183.
- [22] J. Brooke, “SUS: a “quick and dirty” usability scale,” *Usability evaluation in industry*, vol. 189, no. 3, 1996.
- [23] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *Intl. Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.
- [24] N. R. Gans and S. A. Hutchinson, “Stable visual servoing through hybrid switched-system control,” *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 530–540, 2007.
- [25] R. Balachandran, J.-H. Ryu, M. Jorda, C. Ott, and A. Albu-Schaeffer, “Closing the force loop to enhance transparency in time-delayed teleoperation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 198–10 204.