Mixture of experts on Riemannian manifolds for visual-servoing fixtures

Maximilian Mühlbauer[1,2], Freek Stulp[2], Alin Albu-Schäffer[1,2], João Silvério[2]

***Abstract*— Adaptive Virtual Fixtures (VFs) for teleoperation often rely on visual inputs for online adaptation. State estimation from visual detections is never perfect, and thus affects the quality and robustness of adaptation. It is therefore important to be able to quantify how uncertain an estimation from vision is. This can, for example, inform on how to modulate a fixture's stiffness to decrease the physical force a human operator has to apply. Furthermore, the target of a manipulation operation might not be known from the beginning of the task, which creates the need for a principled way to add and remove fixtures when possible targets appear in the robot workspace. In this paper we propose an on-manifold Mixture of Experts (MoE) model that synthesizes visual-servoing fixtures while elegantly handling full pose detection uncertainties and 6D teleoperation goals in a unified framework. An arbitration function allocating the authority between multiple vision-based fixtures arises naturally from the MoE formulation. We show that this approach allows a teleoperator to insert multiple printed circuit boards (PCBs) with high precision without requiring the manual design of VFs to guide the robot motion. An exemplary video visualizing the probability distribution resulting from our model is available at: `https://youtu.be/GKMQvbJ5OzA`**

I. Introduction

Virtual Fixtures play an important role in shared control as haptic aids by providing force feedback to the teleoperator [1], [2]. For VFs defined in relation to an object to be manipulated, it is important to track the pose of that object to be robust to changes in the environment or to non-static objects. In this regard, visual-servoing fixtures are especially well suited as they enable online adaptation while not requiring changes of the workpiece. Using vision, it is possible to actively compensate for pose uncertainties that cannot be anticipated or modeled *a priori* [3].

Despite their usefulness, defining VFs can be challenging, particularly when vision is involved. On the one hand, object poses may not be known with sufficient certainty in advance, making it impossible to use constant fixtures that are designed once and rarely change. On the other hand, many external factors, such as lighting, are non-trivial to model and may rapidly degrade teleoperation performance. Indeed, imperfect visual measurements are a reality in robotics – yet, we have to rely on them even though they might be uncertain. This means that teleoperation systems that have access to uncertainty measurements, and are able to assign authority based on them, are better equipped to succeed in challenging environments.

[1] Department of Informatics, Sensor Based Robotic Systems and Intelligent Assistance Systems, Technical University Munich, Garching, Germany `maximilian.muehlbauer@tum.de`

[2] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchner Str. 20, 82234 Weßling, Germany

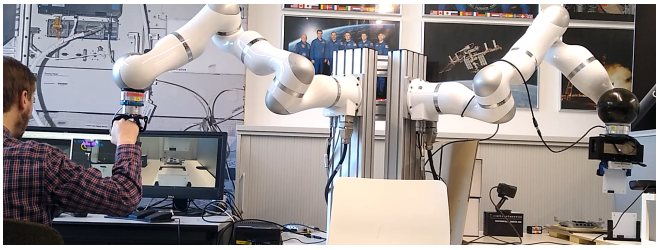


Fig. 1: The teleoperation setup with haptic input device on the left and remote device on the right side.

In this work we introduce a probabilistic *Mixture of Experts* (MoE) [4], [2] approach to automate the arbitration of uncertain visual-servoing fixtures. First, a probability distribution is used to quantify the uncertainty of a detected object (Sections III-A and III-B). The parameters of this distribution define the attractor point and precision of that object's visual-servoing fixture. To take the orientation into account, we use the $\mathbb{R}^3 \times \mathbb{S}^3$ Riemannian manifold (Section II-B).

In parallel, a *gating function* that depends on the robot end-effector and object poses assigns importance to the different objects and their fixtures resulting in a MoE that corresponds to a multi-modal probability distribution (Section III-C). The expectation of the VF under this distribution is used to compute one main fixture that is applied to the robot. Thanks to its probabilistic nature, this fixture can change dynamically as the robot interacts with the environment. On the one hand this approach handles the appearance/disappearance of objects elegantly by weighing their importance against all other objects'. As a consequence, the effort to design fixtures by hand is minimal. On the other hand, it allows for smooth transitions between fixtures, with the teleoperator's freedom of motion being guided by the fixtures' uncertainties. This is achieved by setting end-effector stiffness gains such that higher control authority is given to the teleoperator in directions of high uncertainty (Section III-D).

We present preliminary results on an experimental setup with a PCB connector assembly task (Section IV).

II. Fundamentals

A. Teleoperation System and Virtual Fixtures

We use the teleoperation system from [5] (Fig. 1). In particular, we assume two gravity-compensated, impedance-controlled manipulators where Cartesian wrenches $\mathbf{w}_{\text{ee}} \in \mathbb{R}^6$ are commanded at the end-effector, with joint torques computed as $\boldsymbol{\tau} = \mathbf{J}^\top \mathbf{w}_{\text{ee}}$ [6]. Furthermore, the Cartesian wrenches of remote and input robots are computed with

$$\mathbf{w}_{\text{ee,remote}} = \alpha \left(\mathbf{K}\Delta\mathbf{x} + \mathbf{D}\Delta\dot{\mathbf{x}} \right) + \mathbf{w}_{\text{VF}} \tag{1}$$

$$\mathbf{w}_{\text{ee,input}} = -\alpha \mathbf{Ad}_{\text{ir}} \mathbf{w}_{\text{ee,remote}} \tag{2}$$

where the adjoint $\mathbf{Ad}_{\mathrm{ir}}$ transforms wrenches from the remote robot to the haptic input device which, in our case, is a torque-controlled manipulator. The factor α scales motions between both robots, $\Delta\mathbf{x}$ and $\Delta\dot{\mathbf{x}}$ corresponds to their relative displacement and $\mathbf{K}$, $\mathbf{D}$, are positive definite constant stiffness and damping gain matrices. The term $\mathbf{w}_{\mathrm{VF}}$ is the wrench computed by the VF from[1]

$$\mathbf{w}_{\mathrm{VF}} = \mathbf{K}_{\mathrm{VF}}(\mathbf{x}_{\mathrm{VF}} - \mathbf{x}_{\mathrm{ee}}), \tag{3}$$

with the scaled stiffness matrix $\mathbf{K}_{\mathrm{VF}}$ (Section III-D). The resulting wrench is applied to the end-effector of the remote robot only while also being perceived by the user through the coupling introduced by α. This formulation allows to apply visual-servoing based VFs on the remote robot while providing their effect as force feedback to the haptic input device.

B. On-Manifold Probabilities

Object pose uncertainties appear both at position and orientation levels. To be able to model both, we use an on-manifold approach with Gaussian distributions [7], [8]. We use a pose defined as the product of the 3-dimensional Euclidean space and the quaternion manifold, $\mathbf{x} \in \mathbb{R}^3 \times \mathcal{S}^3$, whose distribution is parameterized by a mean $\boldsymbol{\mu} \in \mathbb{R}^3 \times \mathcal{S}^3$ and a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{6\times 6}$ in the tangent space of $\boldsymbol{\mu}$. Given a set of N pose samples, Maximum Likelihood Estimation (MLE) [9] is achieved by computing the mean iteratively with [7], [8]

$$\Delta = \frac{1}{N}\sum_{i=1}^{N} \mathrm{Log}_{\boldsymbol{\mu}} \mathbf{x}_i \tag{4}$$

$$\boldsymbol{\mu} \leftarrow \mathrm{Exp}_{\boldsymbol{\mu}} \Delta, \tag{5}$$

and, upon convergence of (4)–(5), the covariance matrix

$$\boldsymbol{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}_i)^{\top} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}_i). \tag{6}$$

The logarithm function $\mathrm{Log}_{\boldsymbol{\mu}}(.)$ maps points from the manifold to the tangent space at $\boldsymbol{\mu}$. The exponential map $\mathrm{Exp}_{\boldsymbol{\mu}}(.)$ maps a vector from the tangent space at $\boldsymbol{\mu}$ onto the manifold. Vectors in tangent space can be moved from one linearization point to another using *parallel transport* which compensates for different base vector orientations at different points $\boldsymbol{\mu}$. Similarly to previous works, we rely on this property to transport covariance matrices between different tangent spaces.

III. Approach

Formally we assume that, at any moment, a number of $K \geq 0$ visual-servoing fixtures may be active, each trying to bring the robot towards an object in its field of view with different $\mathbf{x}_{\mathrm{VF}}$. As the field of view changes with the end-effector position, the number of active fixtures and their parameters depend on $\mathbf{x}_{\mathrm{ee}}$. Hence we treat each fixture as a conditional distribution $p_k(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}})$ with $k = 1, \ldots, K$ that is computed from the uncertainty of the predicted poses (Section III-A). To ensure smooth operation despite visual noise Kalman filtering is used (Section III-B). When $K > 1$, several fixtures pull the end-effector simultaneously. We mitigate this issue by using a MoE to compute one single distribution $p(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}})$ from the K candidates (Section III-C). With this distribution we are able to compute not only an attractor point that drives the remote robot pose, but also stiffness gains that regulate the required precision while tracking it (Section III-D).

[1]For the sake of the explanation we show the error term $\mathbf{x}_{\mathrm{VF}} - \mathbf{x}_{\mathrm{ee}}$ using the Euclidean difference although, when $\mathbf{x}$ describes a full pose, the computation is slightly different for the orientation part.

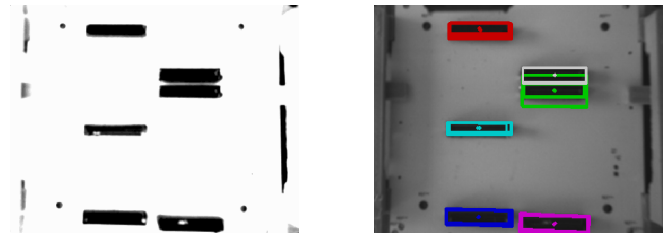


Fig. 2: Probabilistic target connector extraction by using multiple grayscale thresholding values. Different thresholding values lead to "soft" borders (left side, intermediate gray values) while the core of the connectors and the outside, where all thresholding values give the same result, are uniformly black respectively white. This results in different rectangles (right side). Converted to 6DoF poses, we treat the grouped detections as samples from a Gaussian distribution.

A. Probabilistic Target Connector Detection

Although the proposed approach remains general to any type of object, in this work we focus on the use case of CubeSat subsystem assembly [10]. In our previous work [5], a fixed grayscale threshold was used to binarize the intensity image I and extract targets using OpenCV [11] rectangle extraction. Depending on illumination conditions and camera settings, the optimal threshold differs. Furthermore, shadows cast by the connectors make it very difficult to find a single value for optimally extracting the target connector.

Using the idea of soft grayscale thresholds [12], [13] we propose to extract potential target connectors on a symmetric range of different grayscale values (Fig. 2)

$$T \in T_{nom} \pm \{\Delta T_0, \Delta T_1, \ldots\} \tag{7}$$

around the nominal threshold T_{nom} with threshold increments ΔT_i. As shown in Algorithm 1, we group these extractions using their 2D coordinates (*groupByXy*), assigning exactly one matching rectangle per grayscale threshold value. Then, we convert them to 6D poses (*convertTo6D*) and, for each connector k, treat them as set of N_k individual samples drawn from a noisy measurement of the target. Using (4)–(6), the MLE estimate of the samples is computed to approximate $p_k(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}}) = \mathcal{N}(\mathbf{x}_{\mathrm{VF}}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{\Sigma}_k$ provides a measure of the uncertainty associated with a connector. As final step of the detection, we associate new measurements with already existing, tracked connectors based on their distance. If no existing tracked connector is found, a new tracking instance is created using the mean and covariance of the measurement as initial state. In case a tracked connector exists, we employ Kalman filtering for data fusion.

Algorithm 1 Probabilistic target connector detection on grayscale image $I(\boldsymbol{H}_{\mathrm{ee}})$ with threshold values T_i.

```
rects ← empty list
for i in len(T) do
    B ← I > T_i
    rects ← rects + minAreaRects(B)          ▷ list append
end for
sorted_rects ← groupByXy(rects)              ▷ one rect per T_i
for k in len(sorted_rects) do
    6d_det ← H_ee · convertTo6d(sorted_rects[k])
    μ_k ← mean(6d_det)                       ▷ Eqs. (4) and (5)
    Σ_k ← cov_{μ_k}(6d_det)                  ▷ Eq. (6)
end for
```

B. On-manifold Kalman Filtering

For fusing incoming measurements with the existing pose estimate $p_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ of a connector, we employ a Kalman filter. As we assume static targets, the update equations simplify to

$$\boldsymbol{\mu}_{k,t|t-1} = \boldsymbol{\mu}_{k,t-1} \tag{8}$$

$$\boldsymbol{\Sigma}_{k,t|t-1} = \boldsymbol{\Sigma}_{k,t-1} + \boldsymbol{Q} \tag{9}$$

with the on-manifold state mean $\boldsymbol{\mu} \in \mathbb{R}^3 \times \mathbb{S}^3$ and the covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{6\times 6}$ as well as the process noise $\boldsymbol{Q} \in \mathbb{R}^{6\times 6}$ in tangent space. Subscripts t denote the time step, k the connector. Because our measurement and its covariance are expressed in the same coordinates, the measurement equation simplifies to the identity matrix. We thus arrive at a Gaussian product [14], [15] which we solve iteratively using parallel-transported covariance matrices [7] until convergence

$$\boldsymbol{\Sigma}_{k,t,i} = \left(\boldsymbol{\Sigma}_{k,t-1}^{-1} + \boldsymbol{\Sigma}_{k,\mathrm{m}}^{-1}\right)^{-1} \tag{10}$$

$$\boldsymbol{\mu}_{k,t,i} = \boldsymbol{\mu}_{k,t,i-1}\mathrm{Exp}_{\boldsymbol{\mu}_{k,t,i-1}}\big(\boldsymbol{\Sigma}_{k,t,i}\big(\boldsymbol{\Sigma}_{k,t-1}^{-1}\mathrm{Log}_{\boldsymbol{\mu}_{k,t,i-1}}\boldsymbol{\mu}_{k,t-1} + \boldsymbol{\Sigma}_{k,\mathrm{m}}^{-1}\mathrm{Log}_{\boldsymbol{\mu}_{k,t,i-1}}\boldsymbol{\mu}_{k,\mathrm{m}}\big)\big). \tag{11}$$

$(.)_{k,t,i}$ denotes quantities of the current state and next step, $(.)_{k,t,i-1}$ those of the previous iteration step. $(.)_{k,t-1}$ is the state of the previous step and $(.)_{k,\mathrm{m}}$ the measurement.

C. On-Manifold Mixture of Experts

Having represented the uncertainty of candidate VFs in the robot workspace with $p_k(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}})$, we express $p(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}})$ in a unified manner using a MoE model [4], [9], [2]

$$p(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}}) = \sum_{k=1}^{K} \hat{h}_k(\mathbf{x}_{\mathrm{ee}}, \boldsymbol{\mu}_k) p_k(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}}). \tag{12}$$

Our proposed *gating function* h_k takes into account the robot end-effector pose and the predicted expert locations $\boldsymbol{\mu}_k$ to compute an on-manifold, distance-based metric that determines the influence of each expert through

$$h_k(\mathbf{x}_{\mathrm{ee}}, \boldsymbol{\mu}_k) = \exp\left(-\frac{1}{2}\lambda \mathrm{Log}_{\mathbf{x}_{ee}}(\boldsymbol{\mu}_k)^\top \mathrm{Log}_{\mathbf{x}_{ee}}(\boldsymbol{\mu}_k)\right) + \gamma \tag{13}$$

where λ is a hyperparameter that regulates the influence of nearby points and γ is a regularization factor that stabilizes (12) numerically when far from the objects. We further normalize $\hat{h}_k(\mathbf{x}_{\mathrm{ee}}, \boldsymbol{\mu}_k) = h_k(\mathbf{x}_{\mathrm{ee}}, \boldsymbol{\mu}_k)/\sum_j^K h_j(\mathbf{x}_{\mathrm{ee}}, \boldsymbol{\mu}_k)$ to ensure that the value of the gating functions sums to 1. Our chosen gating function shares connections with kernel methods, where it can be seen as a linear combination of a RBF kernel and a constant kernel [16], [17]. Equation (13) ensures a peaked assignment when close to one connector while assigning very similar weights when far from all connectors. The factors λ and γ can be used to adjust the gating function to the scale of the problem. A smaller λ increases the peak while a smaller γ increases the distance required to assign similar weights to all targets.

Despite unifying predictions from different experts, (12) is by design multi-modal, which is not well-suited to our VF implementation requiring a single attractor point. To mitigate this issue we rely on the expectation and covariance of $\mathbf{x}_{\mathrm{VF}}$ under $p(\mathbf{x}_{\mathrm{VF}}|\mathbf{x}_{\mathrm{ee}})$. Since the experts are Gaussian, the resulting distribution will be a uni-modal Gaussian. This approximation is often referred to as *moment matching*, see [18], [19] for derivations. Similarly to II-B, the mean is computed iteratively, this time using the means of each expert $\boldsymbol{\mu}_k$ and their importance $\hat{h}_k$

$$\Delta = \sum_{k=1}^{K} \hat{h}_k \mathrm{Log}_{\boldsymbol{\mu}_{\mathrm{VF}}}\boldsymbol{\mu}_k, \quad \boldsymbol{\mu}_{\mathrm{VF}} \leftarrow \mathrm{Exp}_{\boldsymbol{\mu}_{\mathrm{VF}}}\Delta. \tag{14}$$

Subsequently, the covariance computation in Euclidean space is modified to the manifold by using

$$\boldsymbol{\Sigma}_{\mathrm{VF}} = \sum_{k=1}^{K} h_k \left(\boldsymbol{\Sigma}_{||\boldsymbol{\mu}_k}^{\boldsymbol{\mu}_{\mathrm{VF}}} + \mathrm{Log}_{\boldsymbol{\mu}_{\mathrm{VF}}}(\boldsymbol{\mu}_k)^T \mathrm{Log}_{\boldsymbol{\mu}_{\mathrm{VF}}}(\boldsymbol{\mu}_k)\right), \tag{15}$$

where $\boldsymbol{\Sigma}_{||\boldsymbol{\mu}_k}^{\boldsymbol{\mu}_{\mathrm{VF}}}$ denotes $\boldsymbol{\Sigma}_k$ mapped from the tangent space of $\boldsymbol{\mu}_k$ to that of $\boldsymbol{\mu}_{\mathrm{VF}}$ using parallel transport. Under (14), we use $\boldsymbol{\mu}_{\mathrm{VF}}$ as the attractor point in (3). Moreover, due to our choice of h_k, (15) matches $\boldsymbol{\Sigma}_k$ in the vicinity of connector k, increasing as the end-effector moves away. For this reason, we use $\boldsymbol{\Sigma}_{\mathrm{VF}}$ to design stiffness gains that regulate the control precision associated with each fixture.

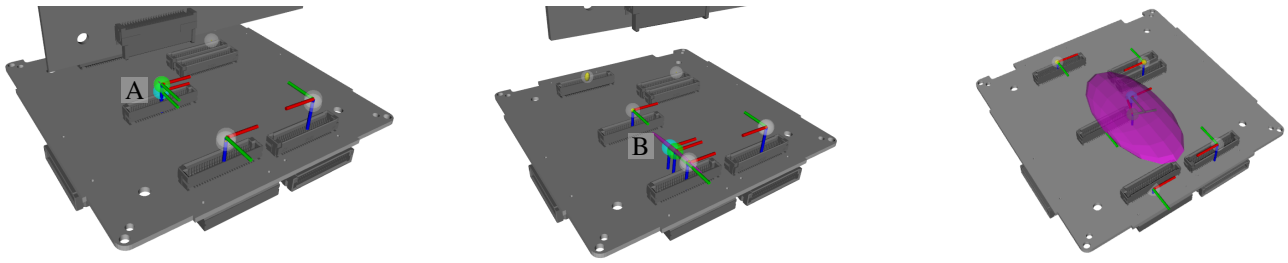
D. Variable Stiffness Control

We use the precision matrix $\boldsymbol{P}_{\mathrm{VF}} = \boldsymbol{\Sigma}_{\mathrm{VF}}^{-1}$ to scale the stiffness of the resulting visual servoing fixture. For this, the elements of $\mathbf{K}_{\mathrm{VF}}$ are set elementwise

$$K_{\mathrm{VF},ij} = K_{ij}\min\left(\max\left(\eta\left(P_{ij} - \kappa\right), 0\right), 1\right) \tag{16}$$

where Pij is the entry of $\boldsymbol{P}_{\mathrm{VF}}$ at indices i, j. Precision entries $< \kappa$ result in zero fixture stiffness and precision entries $> \frac{1}{\eta} + \kappa$ in full stiffness. Values in between are linearly scaled. With this gain design we ensure that directions that have larger variance allow for more freedom to the teleoperator, while directions with low variance are more strict in enforcing the visual-servoing fixture.

Using human prior knowledge that the insertion is performed in z-direction, we do not generate forces along the



(a) End-effector close to one connector. (b) End-effector in-between two connectors. (c) End-effector far from the board.

Fig. 3: Probabilistic visual-servoing fixture estimation. White spheres represent the position of detected connectors with their orientation shown as coordinate frames. The small yellow ellipsoids are the Gaussian distributions associated with each connector, which represent their uncertainty, acting as individual candidate fixtures to drive the robot end-effector (the so-called *experts* in our approach). The purple ellipsoid depicts the 3D Gaussian distribution that represents the main VF in position space. The mean of this distribution is highlighted by the green sphere, making the location of the end-effector attractor point clear. Finally, the blue sphere is added for clarity and it shows the 3D end-effector position projected on the horizontal plane.

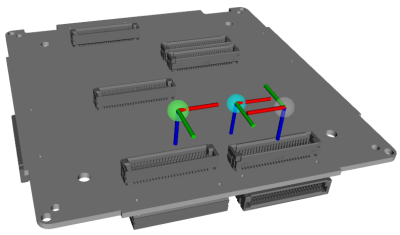


Fig. 4: As the orientation also influences the distance function, the probability of the left connector is higher, despite the end-effector being closer to the right one.

z axis. When there are no tracked detections ($K = 0$), we do not output a force. Both could also be represented in the experts themselves respectively by an additional expert; for simplicity reasons we however resort to explicitly programming these properties.

IV. Evaluation and Discussion

In this section we report preliminary results on the proposed approach. We evaluate our method on the use case of CubeSat subsystem assembly [10]. We empirically set $\lambda = 1 \times 10^4$, $\gamma = 1 \times 10^{-20}$, $\kappa = 3 \times 10^3$, $\eta = 1 \times 10^{-6}$.

Fig. 3 shows different end-effector poses and the resulting estimated Gaussian, according to (14)–(15), given the visualized detections. The obtained results show that the proposed probabilistic fixture gives a strong positional and rotational guidance to the user when close to one target as illustrated by the small purple ellipsoid $\boxed{\text{A}}$ in Fig. 3a. Despite the strong guidance the user is able to 'escape' the fixture and switch to a different connector as seen in Fig. 3b. Notice the large variance of the purple Gaussian in the direction of the connectors $\boxed{\text{B}}$. This is because our choice of gating function influences the MoE model to combine the experts based on distance. Thanks to our variable stiffness formulation (Section III-D), this results in a decreased stiffness along that direction, facilitating the transition. When far away (e.g. above the backplane PCB, Fig. 3c), the user can not only displace the end effector in the xy plane but also rotate the end effector freely around the z axis. This provides a flexible way for the teleoperator to prepare for new insertions.

Fig. 4 shows the effect of including the orientation in the distance function. As seen from the position of the blue sphere, the closest connector is the right-most one. However, due to the large difference in orientation, our model knows that the most likely target is the left-most connector, thus pulling the robot towards it (green sphere as attractor).

TABLE I: Manipulation time and force / torque value comparison, quantities in brackets denote the variance.

Method	Manipulation time [s]	Force [N]	Torque [Nm]
Previous [5]	8.92 (5.45)	11.03 (5.25)	0.144 (0.01)
MoE approach	11.73 (16.01)	7.93 (4.62)	0.172 (0.01)

In order to obtain a preliminary comparison with our previous method [5], an experienced user performed five plugging trials for three of the connectors with both the previous method and the approach described in this paper. The completion times and forces are summarized in Section IV. The results suggest that with the new method, we do also achieve good support for an experienced teleoperator, which can be seen by similar manipulation time and force/torque values. The advantage of the proposed approach, however, is in the fact that there is no need to program fixtures to guide the robot to the connectors manually as our model allows the user to select the target automatically.

V. Conclusion

We proposed an approach based on a *mixture of experts* model to automatically detect and arbitrate visual-servoing fixtures in shared control. Our results show that with our method we could obtain a natural arbitration of multiple targets and extract a meaningful covariance that was used to modulate the end-effector stiffness further facilitating teleoperation. The experimental evaluation shows that the method supports the plugging of CubeSat subsystems into multiple possible target connectors, providing guidance as well as giving the user the choice of different possible targets.

In future work, we plan to extend our method to incorporate position-based VFs to profit from the multi-phase guidance capabilities of our previous approach [5].

Acknowledgment

The results presented here were achieved within the framework of the ACOR project funded by the Federal Ministry for Economic Affairs and Energy (BMWi).

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, 2018.

[3] J. Silvério, G. Clivaz, and S. Calinon, "A laser-based dual-arm system for precise control of collaborative robots," in *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, 2021, pp. 9183–9189.

[4] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991. doi: 10.1162/neco.1991.3.1.79

[5] M. Mühlbauer, F. Steinmetz, F. Stulp, T. Hulin, and A. Albu-Schäffer, "Multi-phase multi-modal haptic teleoperation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[6] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[7] M. J. A. Zeestraten, I. Havoutis, J. Silvério, S. Calinon, and D. G. Caldwell, "An approach for imitation learning on riemannian manifolds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1240–1247, 2017. doi: 10.1109/LRA.2017.2657001

[8] S. Calinon, "Gaussians on riemannian manifolds: Applications for robot learning and adaptive control," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 33–45, 2020. doi: 10.1109/MRA.2020.2980548

[9] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2006.

[10] 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.

[11] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[12] C. Gillmann, P. Arbelaez, J. T. Hernandez, H. Hagen, and T. Wischgoll, "An uncertainty-aware visual system for image pre-processing," *Journal of Imaging*, vol. 4, no. 9, 2018. doi: 10.3390/jimaging4090109

[13] C. Dalitz and H. Niederrhein, "Soft thresholding for visual image enhancement," Hochschule Niederrhein, Fachbereich Elektrotechnik und Informatik, Tech. Rep., 2014.

[14] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014. doi: 10.1109/TRO.2014.2298059

[15] E. Eade, "Lie groups for 2d and 3d transformations," Tech. Rep., 2013.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[17] D. Duvenaud, "Automatic model construction with Gaussian processes," Ph.D. dissertation, 2014.

[18] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intelligent service robotics*, vol. 9, no. 1, pp. 1–29, 2016.

[19] H. G. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, 2004.