# Multi-Arm Payload Manipulation via Mixed Reality

Florian Kennel-Maushart[1], Roi Poranne[1,2], Stelian Coros[1]

***Abstract*— Multi-Robot Systems (MRS) present many advantages over single robots, e.g. improved stability and payload capacity. Being able to operate or teleoperate these systems is therefore of high interest in industries such as construction or logistics. However, controlling the collective motion of a MRS can place a significant cognitive burden on the operator. We present a Mixed Reality (MR) control interface, which allows an operator to specify payload target poses for a MRS in real-time, while effectively keeping the system away from unfavorable configurations. To this end, we solve the inverse kinematics problem for each arm individually and leverage redundant degrees of freedom to optimize for a secondary objective. Using the manipulability index as a secondary objective in particular, allows us to significantly improve the tracking and singularity avoidance capabilities of our MRS in comparison to the unoptimized scenario. This enables more secure and intuitive teleoperation. We simulate and test our approach on different setups and over different input trajectories, and analyse the convergence properties of our method. Finally, we show that the method also works well when deployed on to a dual-arm ABB YuMi robot.**

## I. Introduction

Many industries have adopted collaborative robots for the automation of repetitive or strenuous tasks. Their popularity is driven by their ability to co-exist and execute tasks in the same environment as humans. Their improved safety and flexibility compared to traditional, industrial robots make them more accessible and easy to set up. Advances in *Human-Robot Interaction* (HRI) as a means of programming these robots - via intuitive interactions rather than text-based programming [1] - encourage further integration of collaborative robots in a variety of tasks in different industries. One can imagine, that future engineering teams might no longer require members with strong programming skills in order to automate processes, and that entirely new jobs to teach, collaborate with, or supervise robotic systems will be created. A well-known HRI approach to programming robots is kinesthetic teaching, where an operator physically moves the robot arm through a sequence of key positions, which are recorded and replayed by the robot [2]. One of the main challenges is to avoid *singular configurations* during the teaching phase as well as during the replay phase. Singular or close-to-singular configurations can cause the robot to freeze momentarily, or move abruptly. These issues are accentuated in the case of multi-arm cooperative systems. These systems have many advantages, such as increased payload capacity and the ability to re-grasp payloads, which makes them attractive for many applications, e.g. on building sites, robotic fabrication or in warehouses. But operating many robots simultaneously makes it harder for the operators to recognize and correct for close-to-singular configurations, because they can't focus on each individual arm anymore. It is therefore necessary to introduce interaction methods, which lower the cognitive load and make such systems intuitive and reliable. In the single-arm case, several methods to guide the user away from singularities have been proposed. These are mostly based on generating a *force field* when approaching a singularity, which opposes the operator's movement (see e.g. [3], [4]).

[1] The authors are with the Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland. `florian.maushart@inf.ethz.ch; roi.poranne@inf.ethz.ch; scoros@inf.ethz.ch`

[2]Roi Poranne is with the Department of Computer Science, University of Haifa, Haifa, Israel. `roiporanne@cs.haifa.ac.il`

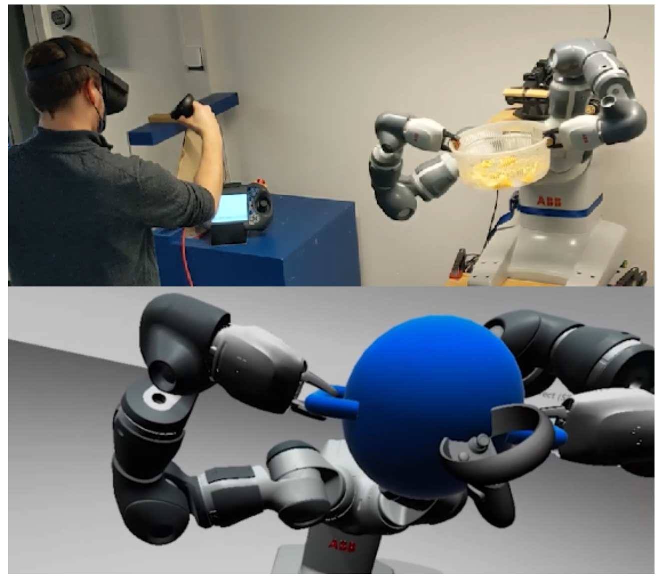


Fig. 1: Controlling a dual-arm YuMi via our MR interface.

These approaches, however, do not work well for immersive teleoperation using Mixed Reality (MR) devices, which generally lack the necessary tactile or force feedback. Such interfaces typically consist of a head mounted display, also known as *headset*, which tracks the user's head movement and displays virtual content. Some interfaces additionally track the user's hands through computer vision, while others rely on wireless controllers as input devices. The use of MR headsets presents many advantages over physical HRI methods, e.g. the ability to *teleoperate* the arm from a remote location, the ability to scale the robot or operate from its point of view, or the option to display virtual clones of objects, visualisations of the robot's internal state and previews of planned robot motion. On the other hand, while hand-tracking or wireless controllers give the operator a lot of freedom, the applicability of aforementioned methods for singularity avoidance becomes limited, as the user's movement cannot physically be restrained. This makes it necessary for the system to autonomously deal with bad input, and adjust its state in a way that maximises tracking fidelity, while effectively avoiding singularities.

We previously introduced a *Virtual Reality* (VR) environment, which allows us to provide real-time input for a multi-arm collaborative manipulation task [5]. We also presented a computationally inexpensive approach to optimize the robots' *manipulability index*. One of the shortcomings of our previous method is, that it only searches along a local, first order approximation of the manipulability, which can lead to tracking errors and sub-optimal improvements. The goal of this paper is to provide a principled approach for real-time optimization, and demonstrate it on real hardware. After an overview over the improved method in Sec. III, we provide simulation results in Sec. IV and finally discuss our implementation on a two-armed ABB YuMi, using an Oculus Quest headset, as well as our implementation on a Microsoft Hololens 2 in Sec. V. We furthermore discuss some implementation considerations and potential extensions of our method to more complex scenarios, like mobile robots, dexterous grippers or systems which greatly differ in scale compared to the human operator.

## II. Related Work

With the advances in VR and MR technology and the availability of relatively affordable headsets, the interest of the HRI community in these interfaces has increased considerably. In this section, we briefly introduce some recent work related to our target application outlined in Section I.

**Single-arm and bi-manual teleoperation** A motion retargeting method for single-arm robot control using wireless controllers has been proposed in [6]. Instead of mapping the operator's motion to the robot arm directly, they introduce a virtual plane as a proxy for interactions with the robot end-effector. The authors conducted a user study, which found that participants were reporting a subjectively better feeling when using the VR controller interface compared to traditional input methods, such as a tablet or a *Geomagic Touch* device. They could also show objective improvements in terms of time required to direct the robot through some key points in 3D space. Rakita et al. proposed another VR interface which allows users to teleoperate a robot arm for pick-and-place tasks [7]. They use a weighted objective, which leads to higher tracking fidelity for precise movements, while avoiding singularities or self-collisions during faster target motions. While this publication focuses on the usecase of teleoperation, they published their approach as a complete framework for weighted Inverse Kinematics in a later paper [8]. An interesting bilateral teleoperation approach has been presented in [9]. A VR interface for control of a robot arm via an interaction proxy is combined with a force feedback control approach via a Geomagic haptic device. This bilateral approach allows two operators to split a task into coarse, intuitive movement, using the VR interface, while being able to receive force feedback for more delicate movements via the haptic interface. Another interesting work by the same author is presented in [10]. Using the same haptic device as in the bilateral approach, a single operator can control a bimanual system for pick and place tasks. The system transmits the remote contact force to the operator and simultaneously uses force and orientation regulation to guarantee a safe grasp during coupled motion.

**Multi-Robot HRI in MR** Although, to the best of our knowledge, no MR interfaces which focus on multi-arm HRI have been presented yet, some work has been done in multi-agent coordination using VR or Augmented Reality (AR) interfaces. In [11], the authors present an immersive interface for multi-robot coordination. An operator is able to select different viewpoints and assign different motion paths to aerial or ground robots using a VR headset and controllers. They report improved situational awareness, reduced workload and better task performance compared to conventional interfaces. They are able to teleoperate a single robot arm to open a valve, but did not include any cooperative tasks in their scenarios. An AR interface for multi-robot task allocation is presented in [12]. The operator is able to assign different tasks to mobile robots, in the form of pushing objects to target locations. While the authors show that they are able to track and interact with multiple robots sequentially, there are no cooperative interactions presented. In [13], the authors present several collaborative multi-arm tasks, which are learned through demonstration by human operators. The operators control a single arm each and are able to communicate with eachother in order to solve problems like lifting two glasses of water while wiping a table. The user interaction happens via a video stream through a web-browser or phone interface. After the demonstrations are collected and used for learning, the tasks can be autonomously executed by the multi-arm system.

**Manipulability optimization as a secondary task** In order to give different tasks different priorities, one can project the gradient of the lower priority task onto the nullspace of the velocity Jacobian. This effectively only makes use of redundant DoFs during the optimization of the lower priority task. An early investigation of this principle can for example be found in [14]. Several applications for this principle have since been investigated, among others singularity avoidance in kinematically redundant arms [15]. With the emergence of collaborative robots, many methods have been developed to apply nullspace projections in the dynamic, torque-controlled domain. For an overview over these methods, see [16].

## III. Method

For cooperative manipulation tasks, there is no need to constantly match an exact target pose with each end-effector. Instead, individual robots can only partially constrain the payload position, and still guarantee the correct position and orientation of the payload. This allows the robot arms to use the redundant DoFs, such as the rotation around a handle, to optimize for secondary objectives, which can improve their ability to react to operator input or avoid obstacles. Real-world examples are the installation of glass panels using suction cups, or the positioning and fastening of pipes and tubes to the ceiling on construction sites. The attachment points in these tasks can be seen as available degrees of freedom (DoFs), which should be leveraged. We propose to utilize these DoFs in order to optimize the manipulability index of the arms. In particular, we demonstrate that the manipulability index can be considerably improved by freeing just one rotational degree of freedom. That is, we allow the end effector to grasp the payload with a specific pose, with respect to the payload's frame of reference, but allow the pose to rotate around one rotational axis.

We suggest to treat the problem as a bi-level optimization problem, which optimizes manipulability in the *null space* of the velocity Jacobian at a valid grasping pose. A similar problem has been proposed in [17], but the solution there requires an algorithm that operates in phases and is too slow for real time interaction. Our approach is based on minimizing our objective value via Newton's method. We elaborate in the following. Additionally, we derive the analytical Hessian for the squared manipulability index here, which to our knowledge has not yet been published in this form.

### A. The optimization problem

We begin by formulating our problem as a bi-level optimization problem for each individual arm

$$\max_{\mathbf{q}} \quad m^2(\mathbf{q}) \tag{1a}$$

$$\text{s.t.} \quad \mathbf{q} = \arg\min_{\hat{\mathbf{q}}} \|\mathcal{K}(\hat{\mathbf{q}}) - \mathbf{x}\|^2 \tag{1b}$$

$$\mathbf{q}_{\text{min}} < \mathbf{q} < \mathbf{q}_{\text{max}}, \tag{1c}$$

where $\mathbf{q}$ are the stacked joint angles, $\mathbf{q}_{\text{min}}$ and $\mathbf{q}_{\text{max}}$ are the joints upper and lower limits, $\mathcal{K}(\mathbf{q})$ is the forward kinematics function for the pose of the robot's end effector, $\mathbf{x}$ is the target end effector pose, and $m$ is the manipulability index.

The squared manipulability index [18] is defined by

$$m^2(\mathbf{J}(\mathbf{q})) = \det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q})) \tag{2}$$

where $\mathbf{J}(\mathbf{q})$ is the velocity Jacobian.

### B. Newton's method

Newton's method iteratively optimizes an objective function by solving the Newton equation $Hdx = -g$ where $g$ and $H$ are the gradient and Hessian of the objective, and $dx$ is the search direction, which is fed into a line search procedure. The gradient and Hessian of the manipulability index are somewhat more involved than the other terms, and we provide them below.

**Derivative of the manipulability** As can be found for example in [19], the derivative of the determinant of a matrix $\mathbf{A}$ can be written as

$$\frac{\partial \det(\mathbf{A})}{\partial x} = \det(\mathbf{A}) \operatorname{tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right) \tag{3}$$

If we take $\mathbf{A} = \mathbf{J}(\mathbf{q})\mathbf{J}(\mathbf{q})^T$, we can see that

$$\frac{\partial m^2}{\partial q_k} = \det(\mathbf{J}\mathbf{J}^T) \operatorname{tr}\left((\mathbf{J}\mathbf{J}^T)^{-1}\frac{\partial (\mathbf{J}\mathbf{J}^T)}{\partial q_k}\right) \tag{4}$$

and finally, using

$$\frac{\partial (\mathbf{J}\mathbf{J}^T)}{\partial q_k} = \frac{\partial \mathbf{J}}{\partial q_k}\mathbf{J}^T + \mathbf{J}\left(\frac{\partial \mathbf{J}}{\partial q_k}\right)^T \tag{5}$$

we obtain

$$\frac{\partial m^2}{\partial q_k} = \det(\mathbf{J}\mathbf{J}^T) \operatorname{tr}\left((\mathbf{J}\mathbf{J}^\mathbf{T})^{-\mathbf{1}}\left(\frac{\partial \mathbf{J}}{\partial q_k}\mathbf{J}^T + \mathbf{J}\left(\frac{\partial \mathbf{J}}{\partial q_k}\right)^T\right)\right) \tag{6}$$

**Hessian of the manipulability** Using the product rule we get

$$\begin{aligned}\frac{\partial^2 \det(\mathbf{A})}{\partial x \partial y} &= \frac{\partial \det(\mathbf{A})}{\partial y} \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial x}\right) + \\ &+ \det(\mathbf{A}) \operatorname{tr}\left(\frac{\partial}{\partial y}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial x}\right)\right)\end{aligned} \tag{7}$$

We can then again use Eq. 3 on the first term of the right-hand side of Eq. 7 and the chain rule on the second to get

$$\begin{aligned}&\frac{\partial^2 \det(\mathbf{A})}{\partial x \partial y} = \\ &\det(\mathbf{A}) \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial y}\right) \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial x}\right) + \\ &+ \det(\mathbf{A}) \left[\operatorname{tr}\left(\frac{\partial \mathbf{A}^{-\mathbf{1}}}{\partial y}\frac{\partial \mathbf{A}}{\partial x}\right) + \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial^2 \mathbf{A}}{\partial y \partial x}\right)\right]\end{aligned} \tag{8}$$

and finally, by using $\frac{\partial \mathbf{A}^{-\mathbf{1}}}{\partial y} = -\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial y}\mathbf{A}^{-\mathbf{1}}$

$$\begin{aligned}&\frac{\partial^2 \det(\mathbf{A})}{\partial x \partial y} = \\ &\det(\mathbf{A}) \left[\operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial y}\right) \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial x}\right) - \right. \\ &\left. - \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial y}\mathbf{A}^{-\mathbf{1}}\frac{\partial \mathbf{A}}{\partial x}\right) + \operatorname{tr}\left(\mathbf{A}^{-\mathbf{1}}\frac{\partial^2 \mathbf{A}}{\partial y \partial x}\right)\right]\end{aligned} \tag{9}$$

If we plug in $\mathbf{A} = \mathbf{J}\mathbf{J}^\mathbf{T}$ again we end up with the following expression for the Hessian

$$\frac{\partial^2 m^2}{\partial q_k \partial q_l} = \det(\mathbf{J}\mathbf{J}^\mathbf{T})\,(t_1 \cdot t_2 - t_3 + t_4) \tag{10}$$

with

$$\begin{aligned}t_1 &= \operatorname{tr}\left[2(\mathbf{J}\mathbf{J}^\mathbf{T})^{-\mathbf{1}}\frac{\partial \mathbf{J}}{\partial q_k}\mathbf{J}^\mathbf{T}\right] \\ t_2 &= \operatorname{tr}\left[2(\mathbf{J}\mathbf{J}^\mathbf{T})^{-\mathbf{1}}\frac{\partial \mathbf{J}}{\partial q_l}\mathbf{J}^\mathbf{T}\right] \\ t_3 &= \operatorname{tr}\left[(\mathbf{J}\mathbf{J}^\mathbf{T})^{-\mathbf{1}}\left(\frac{\partial \mathbf{J}}{\partial q_k}\mathbf{J}^\mathbf{T} + \mathbf{J}\left(\frac{\partial \mathbf{J}}{\partial q_k}\right)^T\right)\right. \\ &\qquad \left.(\mathbf{J}\mathbf{J}^\mathbf{T})^{-\mathbf{1}}\left(\frac{\partial \mathbf{J}}{\partial q_l}\mathbf{J}^\mathbf{T} + \mathbf{J}\left(\frac{\partial \mathbf{J}}{\partial q_l}\right)^T\right)\right] \\ t_4 &= \operatorname{tr}\left[2(\mathbf{J}\mathbf{J}^\mathbf{T})^{-\mathbf{1}}\left(\frac{\partial^2 \mathbf{J}}{\partial q_l \partial q_k}\mathbf{J}^\mathbf{T} + \frac{\partial \mathbf{J}}{\partial q_l}\left(\frac{\partial \mathbf{J}}{\partial q_k}\right)^T\right)\right].\end{aligned} \tag{11}$$

We then combine the Hessian and gradient for the inverse kinematics objective with the projected gradient and Hessian of the manipulability objective, as shown in Algorithm 1.

**Algorithm 1** Assembling gradients of the objective function

**In** Current robot state $\mathbf{q}_k$ **Out** Updated robot state $\mathbf{q}_{k+1}$

Initialise full gradient $g$ as zero vector
Initialise empty Hessian matrix $H$
**for** every term of the objective function **do**
    Calculate gradient
    Calculate Hessian
    **if** current term == manipulability term **then**
        Project gradient into nullspace of the Jacobian
    **end if**
    Add gradient of the current term to the full gradient
    Add Hessian of the current term to the full Hessian
**end for**
Compute Search Direction $d\mathbf{q} = -\mathbf{H}^{-1} \cdot g$
Do bisection line search to find step size $\alpha$
$\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha \cdot d\mathbf{q}$

### C. System

All optimization tasks were run on a Windows 10 machine with an Intel Core i7-9750H CPU @ 2.60GHz, 32GB RAM and an Nvidia Geforce RTX 2080 Max-Q GPU.
For the VR teleoperation interface we used an Oculus Quest 128GB headset and the accompanying wireless controllers. For the MR interface we used a Microsoft Hololens 2 headset. Both interfaces were built in Unity3D and a UDP implementation was used to wirelessly transmit manipulation data to the Windows machine. For the real-world experiments we used an ABB IRB 14000 YuMi with 2 arms and the corresponding ABB smart grippers. The control of the robot was handled via the `abb_libegm` library and the EGM (Externally Guided Motion) interface of the robot controller.

## IV. Results

In order to make our results comparable to our previous implementation in [5], we used the same datasets as previously: A circle trajectory and a square trajectory, with a diameter and side length of 0.4m respectively, as well as a VR trajectory which has been prerecorded from VR input data. For the VR scenario, the maximum distance to the starting point along the trajectory is 0.336m and the trajectory is contained within a 0.34x0.38x0.38m box. The starting point is located at the center of all robots, which are placed on equidistant points on a circle with a diameter of 1.5m for the UR5 setup and 1.2m for the YuMi setup. The mean results and standard deviations for the positional error, joint velocities, accelerations and jerk as well as the manipulability index over the three different trajectories on two YuMi robots and three UR5 arms are summarized in Table I. We furthermore provide some qualitative examples of the resulting manipulability index for the VR trajectory on the UR5 and YuMi robots. For the VR case, we also provide the positional error over the trajectory, which we will discuss in the next section. Finally, we provide comparisons of the computational performance on different setups.

### A. Comparison of the tracking performance

**YuMis** The new method leads to significantly better tracking for the circle and square trajectories regarding the average as well as the standard deviation of the positional error as shown in Table I. In the VR case, the tracking is not significantly improved. This is mostly due to the fact that the handle becomes unreachable for short periods of the trajectory. This can be seen in Figure 3. Consequently, these distance peaks influence the average values. As the standard deviation indicates and as can be seen in the lower half of Figure 3, the number of times where the handle could not be reached while it should be reachable has been reduced significantly with the newer method. While the average joint velocities increase, they stay well below the theoretical limits according to the YuMi's datasheet [20]. In turn, the acceleration values as well as the jerk could be reduced in all cases, indicating an overall smoother motion with the new method.
**UR5 arms** The results paint a similar picture as for the comparisons on the YuMi. In general, the positional error decreases significantly, as do the joint acceleration and jerk, while the average joint velocity increases slightly. Overall, these results indicate good tracking while staying well within the maximum values for joint velocities and accelerations as specified in [21].

### B. Comparison of the manipulability index

**YuMis** Additionally to better tracking, our new method is also able to further improve the manipulability for all three trajectories, with the biggest improvement in the VR case. A qualitative analysis of the data in Figure 3 reflects the results summarized in Table I: The new method is able to reduce variation and overall improve the manipulability. There are, however still two instances where the manipulability value approaches a zero value. As mentioned before, these coincide with brief instances where the handle does become unreachable for an arm due to the user input. For future implementations, an additional constraint to avoid this behaviour should therefore be considered.
**UR5 arms** The manipulability index increases slightly on average for the circle and square trajectories, while significantly improving for the VR trajectory. With the new method, no singularities are encountered during the whole trajectory for all arms. As can be seen in Figure 2 for arm 1, even after

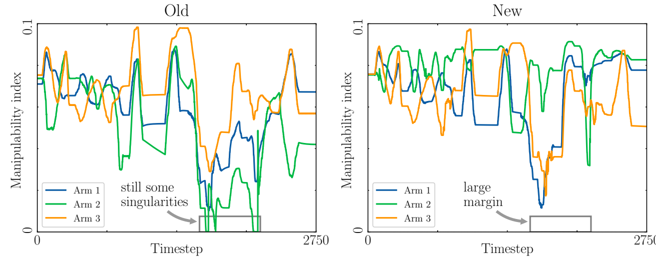



Fig. 2: Comparison between the VR trajectory on 3 UR5 robots with our previous system ("Old") and with enabled manipulability optimization.

TABLE I: Comparisons over different setups and trajectories

| | Pos. Err. [mm] | Vel. [rad/s] | Acc. [rad/$s^2$] | Jerk [$10^{-3}$rad/$s^3$] | m(q) |
|---|---|---|---|---|---|
| YuMi Circle old | $8.454 \pm 16.152$ | $0.030 \pm 0.052$ | $0.001 \pm 0.003$ | $0.066 \pm 0.514$ | $0.026 \pm 0.007$ |
| YuMi Circle new | $\mathbf{0.295 \pm 0.690}$ | $0.154 \pm 0.076$ | $0.000 \pm 0.000$ | $0.002 \pm 0.004$ | $\mathbf{0.031 \pm 0.007}$ |
| YuMi Square old | $9.577 \pm 20.231$ | $0.047 \pm 0.114$ | $0.002 \pm 0.009$ | $0.180 \pm 1.203$ | $0.025 \pm 0.008$ |
| YuMi Square new | $\mathbf{1.240 \pm 5.395}$ | $0.261 \pm 0.150$ | $0.000 \pm 0.001$ | $0.003 \pm 0.021$ | $\mathbf{0.027 \pm 0.010}$ |
| YuMi VR old | $4.614 \pm 15.597$ | $0.016 \pm 0.080$ | $0.001 \pm 0.009$ | $0.191 \pm 1.417$ | $0.026 \pm 0.010$ |
| YuMi VR new | $\mathbf{4.145 \pm 20.339}$ | $0.086 \pm 0.126$ | $0.001 \pm 0.001$ | $0.013 \pm 0.029$ | $\mathbf{0.035 \pm 0.010}$ |
| UR5 Circle old | $0.022 \pm 0.013$ | $0.071 \pm 0.023$ | $0.002 \pm 0.003$ | $0.384 \pm 0.871$ | $0.074 \pm 0.013$ |
| UR5 Circle new | $\mathbf{0.011 \pm 0.019}$ | $0.080 \pm 0.026$ | $0.001 \pm 0.001$ | $0.028 \pm 0.063$ | $\mathbf{0.077 \pm 0.008}$ |
| UR5 Square old | $0.024 \pm 0.011$ | $0.084 \pm 0.024$ | $0.001 \pm 0.002$ | $0.110 \pm 0.592$ | $0.074 \pm 0.014$ |
| UR5 Square new | $\mathbf{0.003 \pm 0.010}$ | $0.095 \pm 0.037$ | $0.000 \pm 0.001$ | $0.009 \pm 0.041$ | $\mathbf{0.077 \pm 0.009}$ |
| UR5 VR old | $0.498 \pm 5.052$ | $0.091 \pm 0.586$ | $0.011 \pm 0.115$ | $2.619 \pm 33.046$ | $0.057 \pm 0.020$ |
| UR5 VR new | $\mathbf{0.172 \pm 0.458}$ | $0.055 \pm 0.110$ | $0.003 \pm 0.006$ | $0.177 \pm 0.514$ | $\mathbf{0.073 \pm 0.016}$ |

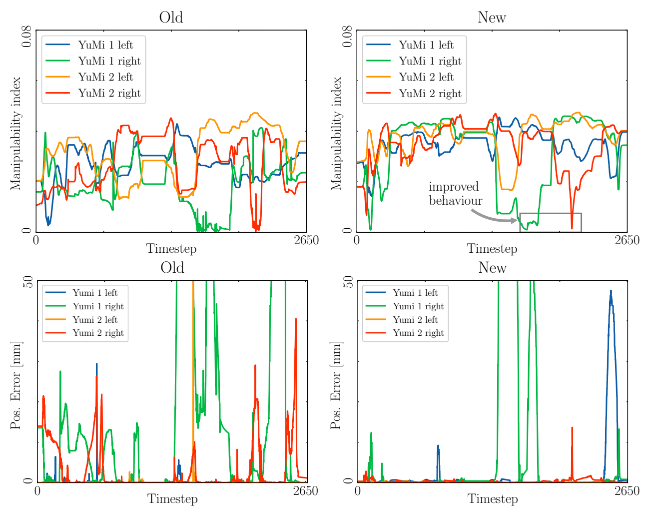
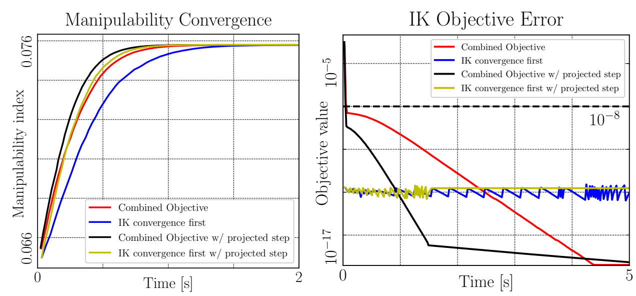



Fig. 3: Comparison between the VR trajectory on two YuMi robots with our previous system ("Old") and with enabled manipulability optimization.




(a) (b)

Fig. 4: Analysis of the convergence properties of two implementations with our combined projection method (in black and red) compared to their more standard counterparts (in yellow and blue). The right graph's y-axis is displayed on a logarithmic scale.

a significant dip in manipulability the current algorithm is able to recover, while in the previous case it still reached very close to singular or singular configurations.

### C. Comparison against different implementations

In order to further test our method, we conducted an additional experiment on the convergence behaviour of different implementations. The four curves in Fig. 4 correspond to the following four implementations:

1) Blue: The IK solution is forced to converge, before manipulability is optimized via nullspace projection of the gradient into the Jacobian nullspace.
2) Red: The same gradient projection, but run at every update step, even if the IK problem has not converged yet. This corresponds to Algorithm 1.
3) Yellow: IK is again enforced first, but afterwards we project the computed Newton step of the manipulability optimization into the nullspace of the Jacobian.
4) Black: Same projection method as for yellow, but convergence of IK is not enforced at every step.

Figure 4(b) shows the violation of the IK objective for each implementation. Note that the y-axis of Fig. 4(b) is in logarithmic scale for improved readability. While our implementation shows a higher error at first, the manipulability improves at a higher rate than for its standard counterparts. The error on the IK objective is very small (below $10^{-8}$), which is in agreement with our other experimental results. Additionally, due to the thresholds on the residuals for the Newton search, the error is further driven down after it has reached the threshold level for the isolated IK objective. We furthermore verified experimentally that the algorithm converges to a point that satisfies the first order optimality conditions of the problem we want to solve. Our projection method therefore generally allows us to improve the manipulability at a higher rate, while keeping the IK solution very close to its goal and converging towards a solution of Equation 1.

### D. Computational performance

As the calculation of the Hessian of the manipulability is quite involved, the computational performance of this algorithm decreases compared to the previous method, which had a largely negligible impact on performance. As can be seen if Fig. 5, we are still able to stay at around 23 frames per second for the VR trajectory with the YuMi on our system.

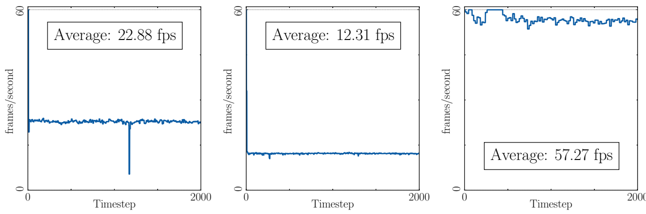



Fig. 5: FPS analysis. Averages left to right: 1 YuMi, 22.88fps; 2 YuMis, 12.31fps; 3 UR5 arms, 57.27fps

This is still sufficient for our real-time teleoperation, as can be seen in the accompanying video material. For the scenario with two YuMis, the performance falls to around 12 frames per second. While still usable for real-time operation, the user can easily notice the drop in frames per second. With an average of 57fps, the setup with 3 UR5 arms stays very close to the capped maximum of 60fps.

## V. Discussion

In this paper, we focused on improving the method for manipulability optimization for our VR/MR teleoperation interface. We introduced a new formulation, providing the gradient and Hessian of the square manipulability index as a sub-objective for our Newton solver. To prioritize the tracking task, we introduced a gradient nullspace projection. At the same time, this removes the need to fine-tune the weight parameters of the two objectives.

The results show a clear improvement of the tracking performance as well as the manipulability over different trajectories. The tracking errors could be reduced in all cases. The good results for the prerecorded VR trajectories could be confirmed on real hardware, as can be seen in the accompanying video material. These improvements came at the cost of a higher computational load, as well as a slightly higher average joint velocity, while the overall smoothness of the joint movements could be improved as can be seen in the generally lower joint acceleration and jerk values.

Allowing the end-effector to rotate around the handle axis enables us to use our method on robots which are non-redundant, which is true for a large part of collaborative robots. Additionally, this method can now be extended to more DoFs, as for example with mobile bases. Some simulations of the UR5 arms on a Clearpath Robotics Ridgeback platform can be found in the accompanying video material and in Fig. 6. Similarly, we can also use our method to only optimize over a subset of the joints of a robot arm. Fig. 7 shows a configuration, where each UR5 arm is equipped with an Allegro hand. While we are able to use the hand as a complex gripper, which can be used as a sort of damping component, we can optimize the manipulability for the arm up to the wrist. This leads to a better ability to follow the user input, while simultaneously not increasing computational complexity unnecessarily. For future work, it might be interesting to see if there are benefits to optimizing the manipulability for each finger as well, although this might require reducing the computational complexity first.

We have also expanded our interface to support a Microsoft Hololens 2 MR headset with hand tracking, such that the user can scale and move the whole setup via hand gestures to accommodate for different working environments. The scaling is useful to make robots of different sizes easier to teleoperate. The robot model shown in Figure 6 and the accompanying video fills a construction hall, but its hologram can be placed on a table for easier teleoperation.

A limitation of our method, which we intend to address in future work, is the independent optimization for each end-effector, which means that we cannot give grasp guarantees. There is also some discussion in the robotics community related to the use of the manipulability index as a metric. Concretely, the need to invert $\mathbf{J}\mathbf{J}^T$ in the gradient can lead to undesired behaviour and can be mitigated through regularization of damping methods, but could be avoided by using different metrics such as the kinematic sensitivity [22].

## VI. Conclusion

We have proposed an optimization method and control interface which improves a user's ability to teleoperate a multi-arm robot system via a MR interface. Optimizing the manipulability index for each arm via a combination of Newton's method and a nullspace projection, we have shown significant improvement in the tracking quality and manipulability over our previous method. We have furthermore implemented our algorithm on a real-world setup with a two-armed ABB YuMi robot. In this setup, the user is able to manipulate a target payload remotely and in real-time. Finally we have provided a discussion of different interfaces as well as an outlook over further possible improvements.

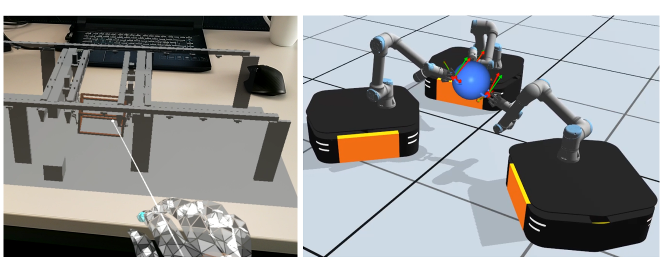


Fig. 6: Left: Interface for the Microsoft Hololens MR headset. Right: Setup with 3 UR5 arms and mobile bases, providing additional DoFs.

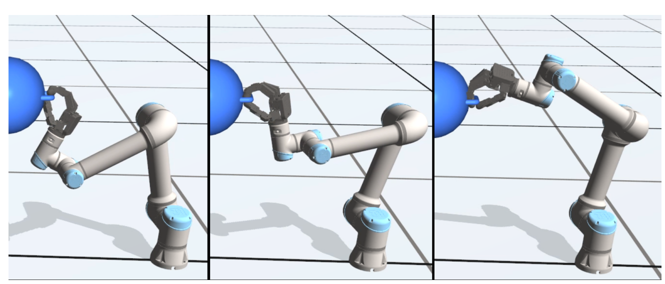


Fig. 7: Nullspace motion for a UR5 arm equipped with an Allegro hand.

# References


[1] F. Sherwani, M. M. Asad, and B. Ibrahim, "Collaborative robots and industrial revolution 4.0 (ir 4.0)," in *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*. IEEE, 2020, pp. 1–5.

[2] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 391–398.

[3] F. Dimeas, V. C. Moulianitis, C. Papakonstantinou, and N. Aspragathos, "Manipulator performance constraints in cartesian admittance control for human-robot cooperation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3049–3054.

[4] M. G. Carmichael, D. Liu, and K. J. Waldron, "A framework for singularity-robust manipulator control during physical human-robot interaction," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 861–876, 2017.

[5] F. Kennel-Maushart, R. Poranne, and S. Coros, "Manipulability optimization for multi-arm teleoperation," *arXiv preprint arXiv:2102.05414*, 2021.

[6] J. H. Lee, Y. Kim, S.-G. An, and S.-H. Bae, "Robot telekinesis: application of a unimanual and bimanual object manipulation technique to robot control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9866–9872.

[7] D. Rakita, B. Mutlu, and M. Gleicher, "A motion retargeting method for effective mimicry-based teleoperation of robot arms," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 361–370.

[8] ——, "Relaxedik: Real-time synthesis of accurate and feasible robot arm motion." in *Robotics: Science and Systems*. Pittsburgh, PA, 2018, pp. 26–30.

[9] D. Sun, Q. Liao, A. Kiselev, T. Stoyanov, and A. Loutfi, "Shared mixed reality-bilateral telerobotic system," *Robotics and Autonomous Systems*, vol. 134, p. 103648, 2020.

[10] D. Sun, Q. Liao, and A. Loutfi, "Single master bimanual teleoperation system with efficient regulation," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1022–1037, 2020.

[11] J. J. Roldán, E. Peña-Tapia, P. Garcia-Aunon, J. Del Cerro, and A. Barrientos, "Bringing adaptive and immersive interfaces to real-world multi-robot scenarios: Application to surveillance and intervention in infrastructures," *Ieee Access*, vol. 7, pp. 86 319–86 335, 2019.

[12] J. A. Frank, S. P. Krishnamoorthy, and V. Kapila, "Toward mobile mixed-reality interaction with multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1901–1908, 2017.

[13] A. Tung, J. Wong, A. Mandlekar, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese, "Learning multi-arm manipulation through collaborative teleoperation," *arXiv preprint arXiv:2012.06738*, 2020.

[14] A. Liegeois *et al.*, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE transactions on systems, man, and cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[15] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 398–410, 1997.

[16] A. Dietrich, C. Ott, and A. Albu-Schäffer, "An overview of null space projections for redundant, torque-controlled robots," *The International Journal of Robotics Research*, vol. 34, no. 11, pp. 1385–1400, 2015.

[17] S. Zimmermann, G. Hakimifard, M. Zamora, R. Poranne, and S. Coros, "A multi-level optimization framework for simultaneous grasping and motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2966–2972, 2020.

[18] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[19] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," November 2012. [Online]. Available: http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html

[20] *Technical data IRB 14000 YuMi (9AKK106354A3254, Rev. H)*, ABB Ltd., May 2019. [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi/irb-14000-yumi-data

[21] *UR5 Technical specifications (110105)*, Universal Robots A/S, 2016. [Online]. Available: https://www.universal-robots.com/media/50588/ur5_en.pdf

[22] H. Shen, Q. Meng, J. Li, J. Deng, and G. Wu, "Kinematic sensitivity, parameter identification and calibration of a non-fully symmetric parallel delta robot," *Mechanism and Machine Theory*, vol. 161, p. 104311, 2021.