2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

An Analysis of Unified Manipulation with Robot Arms and Dexterous Hands via Optimization-based Motion Synthesis

Vatsal V. Patel, Daniel Rakita, and Aaron M. Dollar

***Abstract*—Robot manipulation today generally focuses on motions exclusively with a robot arm or a dexterous hand, but usually not a combination of both. However, complex manipulation tasks can require coordinating arm and hand motions that leverage capabilities of both, much like the coordinated arm and hand motions carried out by humans to perform everyday tasks. In this work, we evaluate unified manipulation with robot arms and dexterous hands, using a motion optimization framework that synthesizes a series of configuration states over the entire manipulation system. We characterize the possible benefits of unifying arm and dexterous hand capabilities within a single model via metrics such as pose accuracy, manipulability, joint-space smoothness, distance to joint-limits, distance to collisions, and more. Several arm-hand combinations are quantitatively compared in simulation on a variety of experiment tasks and performance measures. Our results suggest that combining motions from robot arms and dexterous hands indeed has compelling benefits, highlighting the exciting potential of continued progress in unified arm-hand motion synthesis for robotics applications.**

I. Introduction

Humans regularly coordinate arm and hand motions for everyday activities. For instance, the coarse movements in reaching for a phone in one's pocket are largely executed by the arm, whereas the subsequent finer motions to grasp and pull the phone out are done by the hand in conjunction with the arm. Developing similarly intricate manipulation capabilities has long been a priority for the robotics community [1, 2]. Robot manipulation approaches today generally focus on building algorithms around *whole-arm* motions from 6/7-degrees-of-freedom (DOF) robot arms and simple grippers [3, 4]. In recent years, researchers have been able to also demonstrate interesting *within-hand* manipulation (WIHM) capabilities [5–7]. However, unifying whole-arm and within-hand motions can further enable robots to carry out even more complex manipulation tasks in challenging environments.

One main immediate challenge of equipping 6/7-DOF robot arms with hands capable of WIHM is that they create highly kinematically redundant systems. Moreover, the arm and hand sub-systems each have distinct motion characteristics and desired performance metrics. To effectively manipulate with the entire arm-hand system, novel approaches for redundancy resolution and motion planning are required that coordinate actions of the arm and the hand [8]. Prior work has posited that combining robot arm motions with the strategy of repositioning or reorienting objects with respect to the hand frame could improve motion qualities such as accuracy, energy efficiency, and obstacle avoidance [9], though few works have fully analyzed or characterized these effects in practice.

In this paper, we investigate the benefits of unified arm-hand manipulation by defining the desirable *configuration quality* objectives that maximize performance attributes of the overall system, while achieving the individual arm and hand sub-

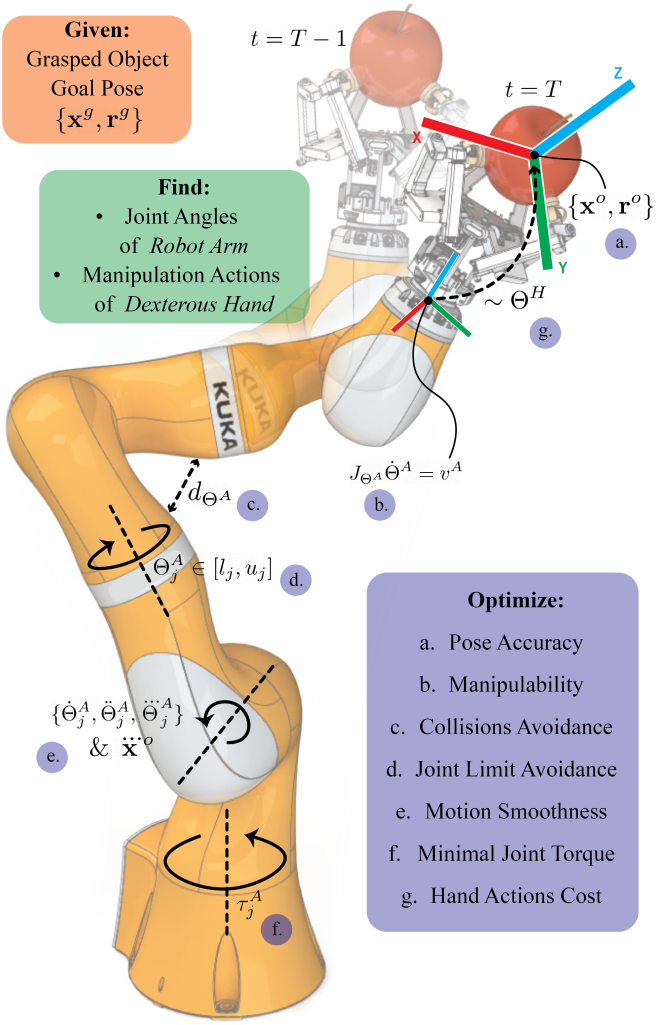


Fig. 1. We evaluate unified manipulation with a framework that synthesizes arm-hand configurations optimized for various quality measures, given a series of goal object poses. An example 7-DOF KUKA arm with a 6-DOF dexterous hand is shown. This method finds joint angles of the redundant robot arm as well as manipulation actions required from the hand for the objectives listed.

This work was supported by the U.S. National Science Foundation (NSF) under grants CMMI-1928448 and IIS-1734190.

V. V. Patel and A. M. Dollar are with the Department of Mechanical Engineering & Materials Science, Yale University, USA, emails:{v.patel, aaron.dollar}@yale.edu, and D. Rakita is with the Department of Computer Science, Yale University, USA, email: daniel.rakita@yale.edu

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10161325

system goals (Fig. 1). This evaluation is done using a modular motion optimization framework that synthesizes a sequence of optimal arm-hand configurations over time for given goal object poses, and is based on per-instant pose optimization methods for generating feasible robot manipulator motion [10]. The utility of unified manipulation is then tested on a wide variety of experiment tasks with two robot arms and five different hand types.

Once the optimal arm-hand poses are obtained along a given task path, the mean cost for each quality objective along the path can be compared, serving as an effective benchmarking tool to analyze an arm-hand system's unified manipulation performance. We can thus examine what benefits (e.g., accuracy, collision avoidance, motion smoothness, and so on) can be obtained by combining WIHM from various hands with whole-arm motions for specific manipulation tasks. The code for our optimization framework and detailed results from the experimental evaluation are made available on the project repository page[1].

A. Related Works

There have been some relevant studies in the area of unified motion with redundant robots such as controlling macro-micro manipulators [11], generating human-like motion [12], resolving redundancy through performance or task-based measures [13], planning for arms with independent wrists [14], manipulation with dual-arms [15], and planning for mobile manipulators [16]. These works can serve as useful inspirations for tackling some of the same challenges arising from the kinematic redundancy of robot arms equipped with dexterous hands. Most robot manipulation approaches today treat the arm and hand as siloed sub-systems, either manually deciding where to place the arm in space for the hand to then carry out its actions, or decoupling the arm and the hand with separate motion plans for each. Our method takes a step towards bridging this gap between manipulation algorithms for robot arms and dexterous hands.

Researchers have also previously studied the arm-hand system as a whole, and presented control methods for grasping [17, 18], attitude manipulation with feedback [19], hands with under-actuation [20], and force control in human-robot interaction [21]. To the authors' knowledge, none of the published works for redundant arm-hand systems have tackled *goal configurations selection*, i.e., finding goal states in the system's configuration space for given goals in the task-space. This is one of the first steps towards building motion planners that operate in the configuration space, and enabling sampling-based planning algorithms to search along optimal arm-hand configurations that account for constraints and capabilities of its constituent sub-systems.

In Ma et al. [9], the authors offer some high-level qualitative discussion on why added dexterity from hands is desirable, and the situations that can benefit from in-hand over arm-based manipulation, but the advantages stated were not formalized or tested.

[1]https://github.com/grablab/unified_manipulation

II. Motion Synthesis Framework

In order to analyze and characterize unified robot arm and dexterous hand motions, we use an optimization-based motion synthesis approach that affords a suite of evaluation benchmark tasks. In this section, we overview the motion generation approach used in our experiments.

A. Technical Overview

We adapt our motion optimization approach from a framework called *RelaxedIK*, proposed by Rakita et al. [10]. This framework optimizes a sequence of robot configurations, Θ_{t_1}, Θ_{t_2}, ..., that, upon interpolation, can be interpreted as a robot motion over time. The approach is able to effectively make trade-offs between many, potentially competing objectives, such as matching end-effector pose goals, avoiding self-collisions, creating smooth joint motion, and so on, thus creating motions that exhibit both accuracy and feasibility [10]. The approach achieves these numerous goals by casting a generalized inverse kinematics problem as a weighted sum non-linear optimization formulation, where each term in the sum can encode a desired motion goal.

In our work, we utilize the modular and flexible nature of the *RelaxedIK* weighted sum objective function to incorporate objective terms that relate to both the arm and hand motion qualities, thus unifying both within a single optimization framework. These objectives collectively contribute to what we refer to as *configuration quality*, and include goals such as avoiding workspace singularities, self-collisions, environment collisions, and joint-limits, as well as achieving smooth motion in both joint-space and task-space. In the following sections, we specify all objective terms used in our approach and detail how they are formulated.

To evaluate arm-hand systems using this motion synthesis approach, we discretize task paths into object poses along a particular trajectory, then optimize over the arm-hand configuration at each goal object pose. The choice and formulation of the criteria for *configuration quality* ensure that the optimal arm-hand poses identified at each discrete point together generate continuous and smooth robot motion along the path. The weights in this multi-objective framework can also be easily tuned to favor some criteria over others, such as minimizing joint jerk over pose accuracy.

B. Optimization Framework

The motion optimization framework used to characterize arm and hand motions in our work is structured as follows:

$$\Theta^* = \arg\min_{\Theta} \ \mathbf{f}(\Theta) \quad : \quad l_i \leq \Theta_i \leq u_i, \forall i \tag{1}$$

Here, $\mathbf{f}$ is the objective function, Θ denotes a robot configuration, and l_i and u_i are the lower and upper bounds of the configuration values. Other constraints can also be added into this formulation, but we embed all our *quality* goals into the objective function.

The objective function, $\mathbf{f}$, is a weighted sum structured as follows:

$$\mathbf{f}(\Theta) = \sum_{i=1}^{k} w_i \cdot \ell(g_i(\Theta), \Omega_i) \tag{2}$$

Here, w_i is a weight associated with the i-th objective g_i, Θ is the arm-hand configuration that the framework is trying to optimize, and Ω_i is the set of parameters for the parametric loss function, ℓ. The structure and normalization for each of the objectives, g_i, are described in detail in §III.

Throughout this work, we consider the configuration Θ is composed of the joint angles of the *arm* (Θ^A), and the manipulation actions of the *hand* (Θ^H). These manipulation actions are represented in the cardinal translation (along X, Y, Z) and rotation (about X, Y, Z) directions in the frame of the hand. The joint angles of the hand are not directly used because almost all hand architectures have distinctly unique kinematic structures [5], and the shape and size of the grasped object also affect the required joint motions. Using the more abstract cardinal manipulation actions of the hand makes the framework more modular to test a wide variety of hand architectures.

C. Parametric Loss Function

We wrap the outputs of the objective functions with a parametric loss proposed in previous work by Rakita et al. [10] that rewards minimizing the output value and places a narrow groove around the optimal value. This loss function has been shown to afford successful trade-offs between many, potentially competing objectives [22]. Before this loss is applied, the objective outputs are each normalized to the $[0, 1]$ range (described below) so that the loss function has a uniform effect on all the objectives.

$$\ell(g_i(\Theta), \Omega_i) = -\exp\left(\frac{-g_i(\Theta)^2}{2c^2}\right) + r \cdot g_i(\Theta)^4 \tag{3}$$

We choose the parameters, Ω_i, of this loss function empirically ($c = 0.1, r = 7$), but other values of these parameters were also observed to work in task evaluations after minimal tweaking of optimization options like maximum allowable iterations and step tolerance values. The formulations for the various configuration objectives are detailed in the following section.

III. Configuration Objectives

The multi-objective optimization framework returns an optimal arm-hand configuration at each time-point based on a weighted sum of objective functions that define its *quality*. The formulations and normalization steps for each of these objectives are detailed in this section.

A. Object Pose Error

The baseline goal of this framework is to generate arm-hand configurations that achieve object poses along a given space-time trajectory. Thus, we compose the first objective function for object position error as the Euclidean distance between the goal position ($\mathbf{x}^g$) and the actual object position ($\mathbf{x}^o$). The second objective function is intended to match the actual object orientation ($\mathbf{r}^o$) to the goal orientation ($\mathbf{r}^g$), both represented in XYZ Euler angles. The actual object position and orientation are obtained from the forward kinematics model of the arm-hand system.

$$g_1(\Theta) = \|\mathbf{x}^g - \mathbf{x}^o\|_2 \tag{4}$$

$$g_2(\Theta) = \|\mathbf{r}^g - \mathbf{r}^o\|_2 \tag{5}$$

B. Singularity Avoidance

A good *quality* arm-hand configuration should be as far away as possible from kinematic singularities. While these singularities may arise within the hand, we focus on the arm in this work due to the wide variety of hand kinematic structures [5]. Several manipulability measures have been proposed in literature for quantifying a manipulator's distance from points of singularity in its workspace [23, 24]. We use the Jacobian-based measure of *inverse condition number* in this formulation to signify the closeness of an arm pose to a singularity. This metric has range of $[0, 1]$, where 0 indicates that the Jacobian (J_{Θ^A}) is ill-conditioned. We incorporate the inverse condition number into an objective function that needs to be minimized as follows:

$$\widetilde{g_3}(\Theta) = 1 - \frac{1}{\text{cond}(J_{\Theta^A})} \tag{6}$$

Some robot arms may have higher manipulability across their workspace compared to others. To normalize the outputs of this objective function for a specific arm's workspace, we randomly sample over 800,000 poses for each robot arm tested in an offline compute step, and record the mean (μ_3) and standard deviation (σ_3) of the objective function values. In the optimization, the function outputs are then normalized by mapping the range $[\mu_3 \pm 1.6 \cdot \sigma_3]$ to $[0, 1]$ in order to cover 90% of outputs assuming a normal distribution. Any values below (or above) this range are assigned 0 (or 1, respectively).

$$g_3(\Theta) = \widetilde{g_3}(\Theta)\Big|_{\mu_3, \sigma_3}$$

C. Joint-Limits Penalty

Although the joints of a robot arm often have a large range of motion, it is not desirable to use that entire range and operate in joint-space close to the mechanical hard limits. We adapt a cost function from work by Tsai that penalizes joint values as they get near to the limits [25]. The proximity of an arm joint j to its joint-limits $[l_j, u_j]$ is calculated as follows:

$$p_j = \frac{(\Theta_j^A - l_j) \cdot (u_j - \Theta_j^A)}{(u_j - l_j)^2}$$

Assuming there are n joints in the robot arm, the objective function to be minimized can be written as:

$$\widetilde{g_4}(\Theta) = \exp\left(-k \cdot \prod_{j=1}^{n} p_j\right) \; ; \; g_4(\Theta) = \widetilde{g_4}(\Theta)\Big|_{\mu_4, \sigma_4} \tag{7}$$

Similar to the manipulability objective function, we normalize the penalty values for each robot arm by sampling

configurations in its workspace, and noting the mean (μ_4) and standard deviation (σ_4) of the distribution in an offline compute step. In our empirical testing, $k = 8e6$ generated a suitable penalizing behavior around joint-limits.

D. Collision Avoidance

One of the main stated benefits of using dexterous hands on robot arms is improving the safety of the overall system by maintaining a distance from obstacles, and work in cluttered environments where arm motions alone would not be feasible to execute a desired object trajectory without causing a collision [9]. We include collision avoidance as an objective function in optimizing the arm-hand *configuration quality*. In this formulation, the closest distance, d_{Θ^A}, between any two solid meshes in the workspace is considered. Both robot and environment bodies are examined, to account for self and world collisions, respectively. The objective function using the smallest distance of an arm link, d_{Θ^A}, to any other obstacle or link can be written as follows:

$$\widetilde{g_5}(\Theta) = \exp(-a \cdot d_{\Theta^A}) \; ; \; g_5(\Theta) = \widetilde{g_5}(\Theta)\Big|_{\mu_5,\sigma_5} \tag{8}$$

We empirically found $a = 8$ to sufficiently avoid collisions. The outputs for this objective function are also normalized to an output range of $[0, 1]$ following a similar process as for the singularity and joint-limit avoidance objectives.

E. Joint and Object Motion Smoothness

If only the configuration at each discrete path pose is optimized, the resulting joint-space trajectory for the robot arm may have discontinuities, or large and erratic joint motions may be required. To enforce continuous and gradual changes at the arm joint angles, we implement 3 motion smoothness objectives that minimize the joint velocity, acceleration, and jerk. These derivatives are calculated using backward finite difference with a horizon over the past 4 optimal configurations on the task path, and normalized by the maximum value that the arm joints can reach. We use a p-norm of 20 for its smooth approximation of the ∞-norm, which is independent of length of the vector, and thus, the number of arm joints.

$$g_6(\Theta) = \left\| \frac{\dot{\Theta}^A_j}{\dot{\Theta}^A_{j,max}} \right\|_p \quad g_7(\Theta) = \left\| \frac{\ddot{\Theta}^A_j}{\ddot{\Theta}^A_{j,max}} \right\|_p$$

$$g_8(\Theta) = \left\| \frac{\dddot{\Theta}^A_j}{\dddot{\Theta}^A_{j,max}} \right\|_p \tag{9}$$

While minimizing higher derivatives of *joint* velocity have been shown to improve robot motion [26], human arm movements have been observed to minimize jerk in the *task-space* [27]. So, another objective function is added to minimize the norm of the jerk in object movements, normalized by the maximum value expected for smooth motion [28].

$$g_9(\Theta) = \left\| \frac{\dddot{\mathbf{x}}^o}{\dddot{\mathbf{x}}^o_{max}} \right\|_2 \tag{10}$$

F. Joint Torque Against Gravity

The kinematic redundancy in robot manipulators has often been utilized to choose an arm configuration that requires minimum joint torques [29]. Minimizing joint torques required on a trajectory is also a strategy used to generate human-like arm motion [12]. In this implementation, we consider the joint torques required to compensate gravitational forces on the robot arm, and write an objective function that minimizes the norm of these values. Each of the values are normalized by the maximum torque that the arm joint can apply, and similar to joint motion objectives, a p-norm of 20 is used.

$$g_{10}(\Theta) = \left\| \frac{\tau^A_j}{\tau^A_{j,max}} \right\|_p \tag{11}$$

G. Hand Manipulation Cost

The objective functions formulated so far add costs either at the arm sub-system level (singularity, collision, joint-limits, torques, and smoothness), or at the overall system level (object pose error and jerk). However, the manipulation actions by the hand also need to be included in the weighted sum cost function. If there was no loss associated with within-hand manipulation, then the optimal configuration would maximize the actions possible by the hand before requiring any arm motion. In our framework, we choose an objective function that, in effect, linearly increases the cost of manipulation actions (Θ^H_i) as they reach closer to the hand's workspace range ($\Theta^H_{i,max}$) along that action axis.

$$g_{11}(\Theta) = \left\| \frac{\Theta^H_i}{\Theta^H_{i,max}} \right\|_p \tag{12}$$

This cost profile could be more carefully constructed for specific hands based on the hand's workspace attributes, such as accuracy characterizations, known regions of slip/drop, or other empirical data. Dynamic weights could also be added that change based on the current hand configuration, especially if the hand can manipulate more precisely in some directions over others. In this work, however, we use the linear function as stated above, and a static weight that can be tuned depending on a hand's expected robustness or accuracy in manipulation.

IV. Experimental Evaluation

In this section, we describe the experimental tests conducted to both validate our optimization approach, and demonstrate its functionality in assessing various arm-hand systems. Specifically, the motion framework outlined above is used to create a sequence of optimal configurations for arm-hand systems on several discrete task paths. This procedure serves as an effective tool to quantitatively benchmark and evaluate the utility of combining motions from dexterous hands with those from robot arms.

TABLE I
MANIPULATION AXES AND RANGES OF THE HAND TYPES TESTED

Hand	Manip. Axes	Manip. Ranges (mm or °)
Fixed [32]	n/a	n/a
Model O [33]	[Z XR YR]	±[20 20° 20°]
Spherical [34]	[XR YR ZR]	±[30° 30° 30°]
Model Q [35]	[Y Z XR ZR]	±[50 50 90° 55°]
Stewart [36]	[X Y Z XR YR ZR]	±[25 25 25 30° 30° 90°]

A. Experiment Setup

Our prototype solver that implements the motion synthesis approach outlined in §II, as well as the simulation environment used throughout our evaluation, were developed using the MATLAB Robotics System Toolbox. All non-linear optimization routines used the `interior-point` algorithm [30]. The solution time is about 15 minutes for a task path with 100 discrete object poses. A majority of this time is spent calculating collision distances between bodies (implemented with the in-built collision checker [31]). We also tested some of the arm-hand combinations with an `sqp` (sequential quadratic programming) solver and obtained similar results [30].

All experiments were run on a system with an Intel Core i7-10700 2.9 GHz processor and 16GB RAM. The simulation requires a URDF for the robot arm model, and the hand kinematic structures are implemented as simple manipulation actions in the frame of the hand as described previously. For each robot arm tested, an offline processing is conducted to sample around 800,000 different configurations and record the distribution of the values for manipulability, joint-limits penalty, and collision distances. The means and standard deviations of these distributions are then used in normalization steps for the respective objective functions. This computation step takes about 30-45 minutes but only needs to run once for a robot arm model.

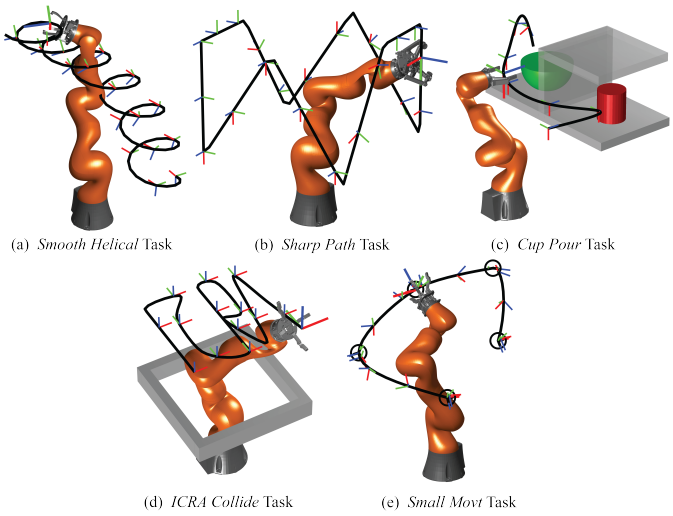
In the results below, we show two versions of the optimization (*Arm-Hand-Config* $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) with different sets of weights on the objective functions. In *Arm-Hand-Config* $\boldsymbol{\alpha}$, the weights used are $w_i = \{40, 30, 2, 1, 0.1, 2, 3, 4, 0.1, 0.1, 0.1\}$, so this version prioritizes matching the object pose to the goal pose. And, in *Arm-Hand-Config* $\boldsymbol{\beta}$, the weights used are $w_i = \{7, 5, 2, 1, 0.1, 6, 5, 4, 1, 0.1, 0.1\}$, placing a higher emphasis on joint and object motion smoothness. Results from other sets of weights tested are not included here for brevity, but are available on the project repository page.

B. Arm-Hand Comparisons

For robot arm types, we chose to test a redundant 7-DOF (KUKA LBR iiwa 14), and a non-redundant 6-DOF robot manipulator (Universal Robots UR10). The kinematic models of both are loaded from URDF files along with collision meshes for each arm link. A variety of different robot hand models were evaluated with these two robot arms: parallel jaw or fixed gripper [32], Model O [33], spherical mechanism-based hand [34], Model Q [35], and Stewart hand [36]. These hands were chosen primarily from the Yale OpenHand project [37] for their distinct kinematic structures, so that a diverse set of hand types could be compared. The baseline fixed gripper condition requires all motion to be conducted with the robot arm. For the remaining dexterous hands, we abstract the manipulation action axes and the corresponding ranges, and outline them in Table I. These hands vary in their physical dimensions, and thus, their actual workspaces are centered at different distances from their mounting points. However, for parity in comparison, we assume that all the hands' manipulation actions start at 0.1m away from the end of the arm. The manipulation actions activated for each hand on the evaluation tasks below are annotated in the supplementary video and can also be seen on the project web page.

C. Evaluation Tasks

The arm-hand combinations described above, each with the two versions of weight options, are evaluated on five different tasks. The goal paths for these tasks are shown in Fig. 2, and aim to test the various benefits of dexterous hands, such as those stated in Ma et al. [9]. Each task path is composed of uniformly discrete goal poses, and some also have additional collision geometries in the environment that the robot must avoid. The first task (*Smooth Helical*) serves as a baseline gradual path absent of any obstacles. The second task (*Sharp Path*) demands the object to be guided on a path with abrupt changes in direction, which may induce larger joint velocity, acceleration, and jerk. The third and fourth tasks (*Cup Pour* and *ICRA Collide*) require navigating collision geometries, even getting close to the arm's workspace limits at some points. In the *Cup Pour* task, the red cup is expected to be grasped and poured into the green bowl around the shelf walls, and in the *ICRA Collide* task, the object traces the letters "I-C-R-A" while the arm is caged in with walls on all four sides. The last path (*Small Movt*) simulates a precision manipulation task, wherein the object is required to be translated and rotated in all six directions by small magnitudes (30mm or 0.3rad) at



(a) *Smooth Helical* Task (b) *Sharp Path* Task (c) *Cup Pour* Task

(d) *ICRA Collide* Task (e) *Small Movt* Task

Fig. 2. Experiment tasks used to evaluate the arm-hand combinations on the various configuration quality objectives, such as ability to avoid collisions, maintain pose accuracy, smooth out motion, and so on.

TABLE II
AGGREGATED OBJECTIVE COSTS ON THE 5 TASKS FOR THE KUKA AND UR10 ARMS WITH THE DIFFERENT HAND TYPES

	Pos. Error (mm)	Ang. Error ($*10^{-3}$)	Singularity Avoid	J. Limits ($*10^{-3}$)	Collision Avoid	Joint Velocity	Joint Acceleration	Joint Jerk ($*10^{-3}$)	Obj. Jerk ($*10^{-3}$)	Joint Torque	Hand Manip.
Arm-Hand-Config $\boldsymbol{\alpha}$ - prioritizes reducing pose error											
Fixed	17.27 ± 9.45	4.54 ± 2.52	0.37 ± 0.13	101 ± 73	0.49 ± 0.28	0.43 ± 0.14	0.13 ± 0.06	31 ± 15	4 ± 1	0.15 ± 0.01	n/a
Model O	5.25 ± 1.5	1.15 ± 0.21	0.26 ± 0.08	3 ± 5	0.48 ± 0.23	0.22 ± 0.08	0.03 ± 0.02	7 ± 5	3 ± 1	0.2 ± 0.02	0.65 ± 0.12
Spherical	4.37 ± 1.99	0.92 ± 0.35	0.22 ± 0.09	4 ± 3	0.47 ± 0.24	0.23 ± 0.07	0.04 ± 0.01	8 ± 3	3 ± 1	0.19 ± 0.03	0.66 ± 0.11
Model Q	3.49 ± 1.67	0.7 ± 0.38	0.16 ± 0.06	20 ± 33	0.33 ± 0.23	0.22 ± 0.07	0.04 ± 0.02	7 ± 4	3 ± 1	0.22 ± 0.03	0.66 ± 0.14
Stewart	3.61 ± 1.11	0.93 ± 0.57	0.19 ± 0.09	7 ± 15	0.46 ± 0.26	0.21 ± 0.06	0.03 ± 0.01	6 ± 3	3 ± 1	0.26 ± 0.04	0.72 ± 0.1
Arm-Hand-Config $\boldsymbol{\beta}$ - prioritizes increasing motion smoothness											
Fixed	59.83 ± 23.66	18.62 ± 6.57	0.29 ± 0.09	58 ± 77	0.49 ± 0.27	0.39 ± 0.15	0.12 ± 0.06	28 ± 15	6 ± 2	0.15 ± 0.01	n/a
Model O	34.02 ± 11.39	11.16 ± 4.52	0.24 ± 0.09	13 ± 13	0.47 ± 0.21	0.21 ± 0.06	0.04 ± 0.02	7 ± 4	4 ± 1	0.19 ± 0.01	0.67 ± 0.1
Spherical	30.57 ± 13.48	7.02 ± 6.6	0.22 ± 0.09	10 ± 23	0.47 ± 0.25	0.19 ± 0.07	0.03 ± 0.02	6 ± 5	4 ± 1	0.19 ± 0.02	0.72 ± 0.09
Model Q	25.09 ± 12.29	8.71 ± 8.07	0.2 ± 0.05	3 ± 4	0.49 ± 0.25	0.17 ± 0.06	0.03 ± 0.01	5 ± 2	4 ± 1	0.21 ± 0.01	0.74 ± 0.09
Stewart	26.98 ± 15.17	5.17 ± 10.07	0.18 ± 0.08	3 ± 7	0.46 ± 0.2	0.17 ± 0.05	0.03 ± 0.01	5 ± 3	4 ± 1	0.26 ± 0.04	0.75 ± 0.08

different points (indicated by the black circles in Fig. 2) in the system's workspace.

D. Performance Measures

The performance of an arm-hand combination on each evaluation task is analyzed using the means of outputs from the optimization objectives ($g_i(\Theta)$). These objective output values allow us to quantitatively assess specific abilities of different arm-hand combinations, such as to circumvent singularity points, avoid collisions, smooth out joint motions, and so on.

E. Results

The aggregated results from the 5 evaluation tasks with the KUKA and UR10 arms, and optimized with 2 sets of quality criteria weights are shown in Table II. More individualized results for each arm and task are available on the project page.

The largest improvement observed in going from a fixed gripper to a more dexterous hand is in pose accuracy. This is observed in the path taken by the robot arm with the fixed gripper, which rounds some of the abrupt and acute bends, possibly in favor of maintaining smoothness in the arm joint motion. There is also a decrease in error as the number of manipulation DOFs in the hand increases. In addition to pose accuracy, the joint velocities, accelerations, and jerks demanded by the various task paths are lower in the cases with the dexterous hands. And this improved smoothness is particularly noticeable in the higher derivatives of joint motion (acceleration and jerk).

Based on our results, we see that adding more DOFs at the end of a robot arm frees up its joint angles to focus on other tasks, such as avoiding zones of low manipulability in the workspace. Arm singularities can also occur at joint-limits, and the measures for both singularity and joint-limits avoidance are reduced by the added within-hand manipulation capabilities. Note that all the hand varieties tested are assumed to be at the same distance from the end of the robot arm, i.e., the manipulator length stays the same in all comparisons.

Our experimental results also indicate some improvement in collision avoidance for conditions with dexterous hands, but only small weights for the collision objective were tested in both versions. The added ability from dexterous hands to evade obstacles is larger in magnitude for the redundant 7-DOF KUKA arm than for the 6-DOF UR10. Especially in the *ICRA Collide* task, the KUKA arm with a fixed gripper was in collision states on 63% of the path, compared to 43% for the KUKA with dexterous hands. Whereas, all the hands on the UR10 had around 75% collision states on the same path.

A different set of weights (*Arm-Hand-Config* $\boldsymbol{\beta}$) shows a scenario when an objective other than pose error might be important to the task (such as, carrying a carton of eggs). The results show that some gains in motion smoothness can be achieved by sacrificing pose accuracy. The two versions, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, demonstrate that our framework is robust to a wide range of weight choices, and can be intuitively tuned to achieve desired manipulation characteristics. Results for additional sets of weights can be found on the project repository page.

V. CONCLUSIONS AND FUTURE WORK

In this work, we investigated the benefits of unified arm-hand manipulation using a framework to synthesize feasible configurations optimized for various quality objectives. Several robot arms and hands with varying degrees of dexterity were evaluated on a range of task paths, and the utility offered by combining arm and hand manipulation motions was quantitatively benchmarked on performance measures such as pose accuracy, manipulability, joint-limit avoidance, collision avoidance, joint and object motion smoothness, and required joint torques. Overall, our results suggest unified arm-hand motions can substantially improve current robot manipulation capabilities, highlighting the exciting potential for continued progress in this area of research going forward.

In future work, we hope to address a couple of key limitations of this framework. Our motion synthesis approach was only tested in a simulation environment, with synthetic and noise-less data. Subsequent iterations of this work will evaluate arm-hand manipulation with real-world systems and improve cost definitions that incorporate grasp stability, contact and environmental constraints. This will additionally require integrating kinematic models of the hands to map the abstracted manipulation actions to actual joint angles for the hand. The current prototype method in this paper also uses crude models for obstacle avoidance and in-built collision checkers, which can be significantly improved upon with approaches such as *Proxima* [38]. Lastly, our method currently uses discrete task poses as input, but we hope to incorporate motion path planners in the future. This will also prevent the arm-hand system from getting stuck in local minima situations (for example, outside the walls in the *ICRA Collide* task).

REFERENCES

[1] J. Cui and J. Trinkle, "Toward next-generation learned robot manipulation," *Science robotics*, vol. 6, no. 54, eabd9461, 2021.

[2] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, eaat8414, 2019.

[3] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments [grand challenges of robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.

[4] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation," in *Third IEEE Int. Conf. on Robotic Computing (IRC)*, IEEE, 2019, pp. 590–595.

[5] C. Piazza, G. Grioli, M. Catalano, and A. Bicchi, "A century of robotic hands," *Annual Review of Control, Rob., and Autonomous Systems*, vol. 2, no. 1, pp. 1–32, 2019.

[6] O. M. Andrychowicz *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[7] W. G. Bircher, A. S. Morgan, and A. M. Dollar, "Complex manipulation with a simple robotic hand through contact breaking and caging," *Science Robotics*, vol. 6, no. 54, eabd2666, 2021.

[8] M. T. Mason, "Toward robotic manipulation," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, 2018.

[9] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th Int. Conf. on Advanced Rob. (ICAR)*, 2011, pp. 1–7.

[10] D. Rakita, B. Mutlu, and M. Gleicher, "RelaxedIK: Real-time synthesis of accurate and feasible robot arm motion.," in *Robotics: Science and Systems*, Pittsburgh, PA, 2018, pp. 26–30.

[11] A. Sharon, N. Hogan, and D. E. Hardt, "The macro/micro manipulator: An improved architecture for robot control," *Robotics and computer-integrated manufacturing*, vol. 10, no. 3, pp. 209–222, 1993.

[12] G. Gulletta, W. Erlhagen, and E. Bicho, "Human-like arm motion generation: A review," *Robotics*, vol. 9, no. 4, p. 102, 2020.

[13] S. Chiaverini, G. Oriolo, and A. A. Maciejewski, "Redundant robots," in *Springer Handbook of Robotics*, Springer, 2016, pp. 221–242.

[14] K. Gochev, V. Narayanan, B. Cohen, A. Safonova, and M. Likhachev, "Motion planning for robotic manipulators with independent wrist joints," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 461–468.

[15] C. Smith *et al.*, "Dual arm manipulation—a survey," *Robotics and Autonomous systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[16] T. Sandakalum and M. H. Ang Jr, "Motion planning for mobile manipulators—a systematic review," *Machines*, vol. 10, no. 2, p. 97, 2022.

[17] K. Nagai and T. Yoshikawa, "Grasping and manipulation by arm/multifingered-hand mechanisms," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, vol. 1, 1995, pp. 1040–1047.

[18] J.-H. Bae, S. Arimoto, R. Ozawa, M. Sekimoto, and M. Yoshida, "A unified control scheme for a whole robotic arm-fingers system in grasping and manipulation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2006, pp. 2131–2136.

[19] A. Kawamura, K. Tahara, R. Kurazume, and T. Hasegawa, "Sensory feedback attitude control for a grasped object by a multi-fingered hand-arm system," in *IEEE Int. Conf. on Robotics and Biomimetics*, IEEE, 2010, pp. 1542–1548.

[20] V. Ruiz Garate and A. Ajoudani, "An approach to object-level stiffness regulation of hand-arm systems subject to under-actuation constraints," *Autonomous Robots*, vol. 44, no. 8, pp. 1505–1517, 2020.

[21] F. Ficuciello and L. Villani, "Compliant hand-arm control with soft fingers and force sensing for human-robot interaction," in *IEEE RAS & EMBS Int. Conf. on Biomedical Robotics and Biomechatronics (BioRob)*, IEEE, 2012, pp. 1961–1966.

[22] D. Rakita, B. Mutlu, and M. Gleicher, "An analysis of RelaxedIK: An optimization-based framework for generating accurate and feasible robot arm motions," *Autonomous Robots*, vol. 44, no. 7, pp. 1341–1358, 2020.

[23] J.-O. Kim and K. Khosla, "Dexterity measures for design and control of manipulators," in *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems '91*, IEEE, 1991, pp. 758–763.

[24] J. Lee, "A study on the manipulability measures for robot manipulators," in *IEEE/RSJ Int. Conf. on Intelligent Robot and Systems (IROS)*, IEEE, vol. 3, 1997, pp. 1458–1465.

[25] M.-J. Tsai, *Workspace Geometric Characterization and Manipulability of Industrial Robots (Kinematics)*. The Ohio State University, 1986.

[26] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 1988, pp. 364–369.

[27] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.

[28] H. Hayati, D. Eager, A.-M. Pendrill, and H. Alberg, "Jerk within the context of science and engineering—a systematic review," *Vibration*, vol. 3, no. 4, pp. 371–409, 2020.

[29] J. Hollerbach and K. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 308–316, 1987.

[30] "Constrained nonlinear optimization algorithms." (2023), [Online]. Available: https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html.

[31] "Collision geometry meshes, collision avoidance and clearance." (2023), [Online]. Available: https://www.mathworks.com/help/robotics/collision-detection.html.

[32] V. V. Patel, A. S. Morgan, and A. M. Dollar, "Highly underactuated radial gripper for automated planar grasping and part fixturing," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 9910–9916.

[33] L. U. Odhner *et al.*, "A compliant, underactuated hand for robust manipulation," *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.

[34] V. V. Patel and A. M. Dollar, "Robot hand based on a spherical parallel mechanism for within-hand rotations about a fixed point," in *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2021, pp. 709–716.

[35] R. R. Ma and A. M. Dollar, "An underactuated hand for efficient finger-gaiting-based dexterous manipulation," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, IEEE, 2014, pp. 2214–2219.

[36] C. McCann, V. Patel, and A. Dollar, "The stewart hand: A highly dexterous, six-degrees-of-freedom manipulator based on the stewart-gough platform," *IEEE Robotics & Automation Magazine*, vol. 28, no. 2, pp. 23–36, 2021.

[37] R. Ma and A. Dollar, "Yale openhand project: Optimizing open-source hand designs for ease of fabrication and adoption," *IEEE Robotics & Automation Magazine*, vol. 24, no. 1, pp. 32–40, 2017.

[38] D. Rakita, B. Mutlu, and M. Gleicher, "Proxima: An approach for time or accuracy budgeted collision proximity queries," in *Proceedings of Robotics: Science and Systems (RSS)*, 2022.