

Western University

Scholarship@Western

Electrical and Computer Engineering
Publications

Electrical and Computer Engineering
Department

Summer 5-30-2021

Arm-Hand Systems As Hybrid Parallel-Serial Systems: A Novel Inverse Kinematics Solution

Shuwei Qiu
sjiu47@uwo.ca

Mehrdad Kermani Ph.D., P.Eng.
Western University, mkermani@eng.uwo.ca

Follow this and additional works at: <https://ir.lib.uwo.ca/electricalpub>



Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

Citation of this paper:

Qiu, Shuwei and Kermani, Mehrdad Ph.D., P.Eng., "Arm-Hand Systems As Hybrid Parallel-Serial Systems: A Novel Inverse Kinematics Solution" (2021). *Electrical and Computer Engineering Publications*. 554.
<https://ir.lib.uwo.ca/electricalpub/554>

Arm-Hand Systems As Hybrid Parallel-Serial Systems: A Novel Inverse Kinematics Solution

Shuwei Qiu and Mehrdad R. Kermani

Abstract—In this paper, we aim to solve inverse kinematics of the integrated robotic arm-hand systems to achieve precision grasping, provided the desired grasp configuration (contact points + contact normals). The key insights of our approach are three-fold. First, we propose a human-inspired thumb-first strategy and consider one finger of the robotic hand as the “thumb” to narrow down the search space and increase the success rate of the algorithm. Second, we formulate the arm-thumb serial chain as a closed chain while other fingers are still as serial chains such that the entire arm-hand system is controlled as a hybrid parallel-serial system. The closed-chain formulation truncates and simplifies the task hierarchy of the entire arm-hand system. Third, we attach a virtual revolute joint to the thumb’s tip with its rotation axis aligning with the thumb’s contact normal to allow this virtual joint to act as the embodiment of the thumb’s functional redundancy. By selecting the thumb’s joints including the virtual revolute joint as the active joints of the arm-thumb closed chain, the arm-thumb system’s self-motion (i.e., the palm pose) and the thumb’s functional redundancy can be directly controlled without using the null space projection. This provides a new possibility to control the self-motion of robot manipulators. Simulation results will demonstrate the advantages and superb performance of the proposed approach for solving the problem of inverse kinematics of achieving precision grasps compared to other classical approaches based on the Damped Least-Squares method [1] in terms of the average success rate (96% v.s. 12%).

I. INTRODUCTION

Robots with multiple end-effectors have attracted significant attention in the past two decades. The common characteristics of such systems are complex kinematic structure and a large number of degrees of freedom (DOFs). The large number of DOFs endow such robots with the ability to achieve different tasks. Meanwhile, large DOFs also contribute to the complexity of handling the ultrahigh kinematic redundancies of such systems [2].

The null space projection is one of the most effective methods of dealing with the kinematic redundancies due to high DOFs. It is frequently implemented along with a hierarchical organization of tasks, where each task is assigned with a specific priority. The highest-priority task is achieved by exploiting all DOFs in the system while lower-priority tasks are executed within the null space of the higher-priority tasks. Consequently, all tasks except the top-priority task can only be achieved sub-optimally without

This research was supported in part through funds received from the Natural Sciences and Engineering Research Council of Canada, Canada Foundation for Innovation, and Ontario Centres of Excellence.

Shuwei Qiu and Mehrdad R. Kermani are with Department of Electrical and Computer Engineering, Western University, London, Ontario, Canada sqiu47@uwo.ca, mkerman2@uwo.ca

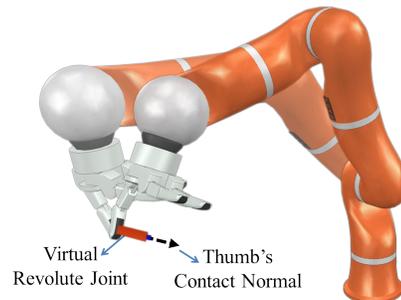


Fig. 1. The arm-thumb closed chain with the virtual revolute joint on the thumb’s tip

interfering with higher-level tasks. The null space projector (N) for a task is calculated as $N = I - J^\dagger J$ where J is the task-related Jacobian matrix and J^\dagger is the Moore–Penrose inverse of J . The projector N projects a velocity vector (\dot{q}) into the null space of J [3]. When multiple tasks are involved, two projection strategies are possible, successive and augmented projection. N is calculated successively in successive projection approaches while N is calculated using the augmented Jacobian in augmented ones.

The successive projection is computationally more efficient than its augmented alternative thanks to the decoupled calculation of the null space projector N . Both projection methods have been applied to highly redundant robots. Dietrich *et al.* [4] proposed a continuous null space projection technique for robots with large DOFs to handle multiple tasks. They considered unilateral constraints and singular Jacobian matrices within the task hierarchy. They designed a diagonal activation matrix to selectively shape the null space projector so that only the selected direction was affected while other directions remained unchanged. In successive projection methods, the task hierarchy is not strictly maintained as the hierarchy’s depth increases. In this projection method, the velocity related to a low-priority task has to be multiplied by the null space projectors of all higher-priority tasks. Each multiplication only assures compliance to the corresponding task but not to all other preceding tasks. Eventually, the execution of lower-priority tasks will interfere with higher-priority tasks [5]. Since the task hierarchy is often deep and complex for highly redundant robots, the successive projection approach is not the first option for these systems.

The augmented projection is more effective in maintaining a strict task hierarchy. Lee *et al.* [6] constructed a relative Jacobian matrix to control a dual-arm system as a single-arm system. Multiple, conflicting tasks were handled by assigning

different priorities. Hu *et al.* [2] also utilized the relative Jacobian and augmented null space projection to design a control framework for dual-arm robots to handle potential task conflicts and robot joint constraints. Although the augmented projection approach maintains strict compliance with the task hierarchy, it cannot avoid algorithmic singularities. Algorithmic singularities appear along with conflicting tasks, which result in adverse effects on the accuracy of conflicting tasks. The consideration and treatment for algorithmic singularities are of high importance for arm-hand systems. To achieve precision grasps, the fingers' contact points (i.e., fingertip locations) and contact normals (i.e., the outward direction of fingertip normals) should be appropriately planned to balance contact forces and gravity when lifting an object [7]. In an arm-hand system, fingers only have limited DOFs and their movements rely on the common arm's motion. Consequently, algorithmic singularities almost always exist in precision grasps. To handle algorithmic singularities, task priority exchange and null space shaping have been applied [8] [9]. But they are only partial solutions since they are essentially trade-offs between different tasks' accuracy and thus they are essentially compromised treatments of the problem. In this paper, we study the inverse kinematics (IK) of arm-hand systems for achieving precision grasps. Rather than focusing on task priority management and null space shaping, our approach emphasizes task hierarchy simplification. To this end, we propose a human-inspired thumb-first strategy combined with a hybrid parallel-serial formulation.

Humans almost always place the thumb on an object to counterbalance other fingers' forces to grasp it. The ability to use the opposing thumb is unique to humans and some primates, which is infrequent in nature [10][11]. The significance of the thumb in grasping has been highlighted in many studies [12][13]. We regard one finger in a robotic hand as the opposing "thumb" and use the thumb-first strategy to locate the thumb at its desired pose first, to enforce meaningful constraints on other fingers and significantly narrow down the IK search space. After achieving the thumb's desired pose, we start to formulate the arm-thumb serial chain as a closed chain while other fingers are still as serial chains (see Fig. 1). To this end, the entire arm-hand system is controlled as a hybrid parallel-serial system. In the arm-thumb closed chain, the arm and the thumb are regarded as the supporting legs and the hand palm is regarded as the end-effector. By doing so, the thumb's tip is fixated and the palm's pose is controlled in the thumb's null space without the null space projection, which allows to truncate and simplify the task hierarchy for the entire system.

Except for the redundancies caused by extra DOFs, the redundancies due to task definitions also exist. Some tasks do not require all six DOFs in the Cartesian space to be controlled. As a result, some extra degrees of redundancy appear in the robotic system, which is often referred to as functional redundancy [14]. In the case of precision grasps, while there are requirements for contact points and contact normals, the rotation around the contact normals is redundant. This undemanded rotation is an example of the

fingers' functional redundancy. The functional redundancy is commonly exploited by null space shaping and projection [14]. But this only provides an indirect and passive approach to control the functional redundancy and it may affect the performance of the tasks whose execution relies on the functional redundancy. In this paper, we also propose a new method to directly and actively utilize the thumb's functional redundancy. To this end, we attach a virtual revolute joint to the thumb's tip with its rotation axis aligning with the thumb's contact normal (see Fig. 1). This virtual joint embodies the thumb's functional redundancy. We can directly control the thumb's functional redundancy by selecting this virtual joint as one of the active joints in the arm-thumb closed chain.

The contributions of this paper are listed below:

- We implement a human-inspired thumb-first strategy to dramatically narrow down the IK search space.
- We formulate the arm-hand system as a parallel-serial hybrid system by formulating the arm-thumb serial chain as a closed chain.
- We propose to attach a virtual revolute joint to the thumb's tip to incorporate the thumb's functional redundancy in our algorithm.
- We provide a new possibility to explicitly utilize the self-motion and the functional redundancy of the arm-thumb system by selecting the thumb's joints including the virtual revolute joint as the active joints of the arm-thumb closed chain.

The rest of this paper is structured as follows. Section II provides the details of the approach. Section III evaluates the performance of the proposed approach and validates its effectiveness through comprehensive simulation results. Finally, Section IV concludes the paper.

II. PROPOSED APPROACH

A. Overview

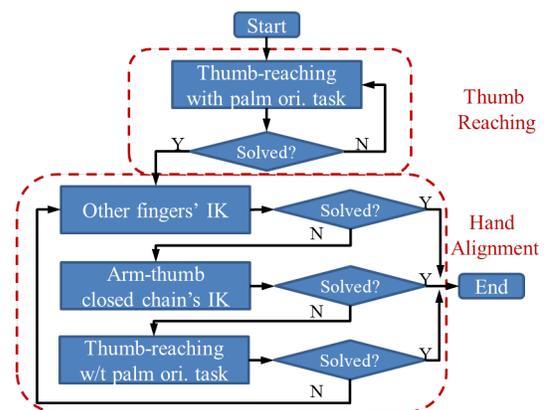


Fig. 2. The pipeline of the proposed algorithm

In this section, we provide the details of the proposed approach. Given a desired precision grasp (contact points + contact normals), the proposed approach serves to find the configuration with which the entire arm-hand system can achieve this grasp. The grasp planning problem is beyond the scope of this paper and will be achieved in subsequent work.

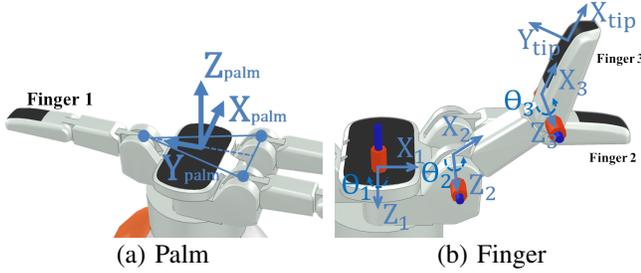


Fig. 3. Axes assignment of the Barrett hand

We use the KUKA LWR 4+ manipulator and the Barrett hand as an example of arm-hand systems to explain the working principle of the proposed approach. We assume all joints are controllable and the KUKA-Barrett system has 14 DOFs in total. The fingers are labelled as Finger 1, 2 and 3 as depicted in Fig. 3 and their contact normal directions on the fingertips are denoted as \vec{n}_i , ($i = 1,2,3$). In this work, Finger 1 is regarded as the thumb. The pipeline of the proposed algorithm is shown in Fig. 2. As noticed, the proposed algorithm consists of two phases: the thumb reaching phase and the hand alignment phase. In the following, the details of these two phases are presented in sequence.

B. The Thumb Reaching Phase

To implement the thumb-first strategy, a typical hierarchical inverse kinematics (HIK) procedure with successive null space projection is conducted. The thumb reaching phase contains three tasks (from the highest priority to the lowest priority): (1) the achievement of hand palm's desired orientation (named as "palm orientation IK" task), (2) the achievement of the thumb's desired contact point (named as "thumb position IK" task), and (3) the achievement of the thumb's desired contact normal (named as "thumb orientation IK" task). The inclusion of the palm orientation task is to prevent the arm-thumb closed-chain formulation from collapse in the following hand alignment phase. The workspace of a parallel robot is generally quite limited. Without the palm orientation task, Finger 2 and 3 are unlikely to close to their desired poses when the thumb-related tasks are achieved. Consequently, a big motion of the arm-thumb closed chain would be required to minimize these fingers' errors. In this case, the arm-thumb closed chain has a great chance to move beyond its workspace boundary such that it would collapse. To avoid this situation, not only the presence of the palm orientation task but also its precise conduction is required in this phase. We need to estimate the desired palm orientation as it is unavailable from the task requirements.

1) *Estimate the desired palm orientation:* The desired palm orientation is estimated based on the hand structure. In the case of Barrett hand, Fig. 3(a) shows the coordinate frame attached to the palm. The palm's frame is defined such that the z -axis (Z_{palm}) aligns with the palm's outward normal, the x -axis (X_{palm}) is parallel to the base of the isosceles triangle formed by the fingers' base points, and the y -axis (Y_{palm}) points toward the thumb's base point along the median of the isosceles triangle. Based on their definitions, the axes of

the palm coordinate frame can be represented as,

$$X_{palm} = p_{2b} - p_{3b}, \quad Y_{palm} = p_{1b} - \frac{p_{2b} + p_{3b}}{2} \quad (1)$$

$$Z_{palm} = X_{palm} \times Y_{palm}, \quad X_{palm} = Y_{palm} \times Z_{palm} \quad (2)$$

where p_{i_b} is the base point of Finger i ($i = 1,2,3$), and (2) is to guarantee the axes' orthogonality. However, fingers' base points are also unavailable from task requirements. To bridge this gap, the fingers' desired contact points can be used in (1) and (2) as the reflection of fingers' base points. Although the desired palm orientation may be ill-estimated by this simple method, it would still significantly reduce the motion of the arm-thumb closed chain and the ill-estimated palm orientation would be corrected in the hand alignment phase. In the following, we delineate how to solve the HIK in the thumb reaching phase.

2) *HIK solution in the thumb reaching phase:* In the thumb reaching phase, the arm-thumb system is controlled as a serial chain and its Jacobian is obtained using the Denavit-Hartenberg (D-H) method. The joint movements for achieving the three tasks (i.e., "palm orientation IK", "thumb position IK", and "thumb orientation IK" task) are solved independently by damped least-squares methods. The resultant joint movements are combined to obtain the total joint movement by successive null space projection.

For the "palm orientation IK" task, the error (denoted as \vec{e}_{hp_o}) is defined as the difference between the desired and the current palm orientation expressed by Euler angles, $\vec{e}_{hp_o} \triangleq [\alpha_d, \beta_d, \gamma_d]^T - [\alpha_c, \beta_c, \gamma_c]^T$. The corresponding null space projector is computed by $N_{hp} = I - J_{hp}^\dagger J_{hp}$ where J_{hp} is the analytical Jacobian of the palm orientation.

As for the thumb-related tasks, the thumb's position and orientation error (denoted as \vec{e}_{t_p} and \vec{e}_{t_o} , respectively) are defined as,

$$\vec{e}_{t_p} \triangleq \vec{p}_{t_d} - \vec{p}_{t_c}, \quad \vec{e}_{t_o} \triangleq \varphi \frac{\vec{n}_{t_c} \times \vec{n}_{t_d}}{\|\vec{n}_{t_c} \times \vec{n}_{t_d}\|} \quad (3)$$

where \vec{p}_{t_c} and \vec{p}_{t_d} are the current and the desired contact point of the thumb, \vec{n}_{t_c} and \vec{n}_{t_d} are the current and the desired contact normal of the thumb, and $\varphi = 2\arctan2(\|U\|, \|V\|)$ is the angle between \vec{n}_{t_c} and \vec{n}_{t_d} , in that $U = \|\vec{n}_{t_d}\|\vec{n}_{t_c} - \|\vec{n}_{t_c}\|\vec{n}_{t_d}$ and $V = \|\vec{n}_{t_d}\|\vec{n}_{t_c} + \|\vec{n}_{t_c}\|\vec{n}_{t_d}$ [14][15]. The Jacobian matrices related to the "thumb position IK" and the "thumb orientation IK" task are denoted as J_{t_p} and J_{t_o} , respectively. The total joint movement is obtained by $\Delta\vec{q} = \Delta\vec{q}_{hp} + N_{hp}\Delta\vec{q}_{t_p} + N_{hp}N_{t_p}\Delta\vec{q}_{t_o}$, where N_{t_p} is the null space of the "thumb position IK" task and $\Delta\vec{q}_{hp}$, \vec{q}_{t_p} , and \vec{q}_{t_o} are the independent joint movements to achieve the three tasks.

C. The Hand Alignment Phase

With the termination of the thumb reaching phase, the hand alignment phase begins to achieve the tasks related to other fingers (i.e., Finger 2 and 3) while maintaining the thumb's pose. In this phase, the arm-thumb serial chain is formulated and controlled as a closed chain while the other fingers are controlled as serial chains. In this respect, the

arm-hand system is controlled as a hybrid parallel-serial mechanism.

As shown in Fig. 2, three steps are executed in sequence in each iteration of this phase: (1) solving other fingers' IK, (2) solving the arm-thumb closed chain's IK, and (3) thumb reaching without the "palm orientation IK" task. Step 1 is to minimize other fingers' errors under the current palm's pose. If other fingers' errors are still unacceptable after Step 1, Step 2 is conducted to adjust the palm's pose to mitigate other fingers' errors. Step 3 is simply the repeat of the thumb reaching phase without the "palm orientation IK" task to keep the thumb at its achieved pose. The details of Step 1 and 2 are explained in the rest of this section.

1) *IK of other fingers*: Let us consider one finger as an example for further explanation. The axes assignment of a finger is shown in Fig. 3(b). The direction of its contact normal (i.e., \vec{Y}_{tip}) with respect to the hand palm can be calculated by the forward kinematics as,

$$\vec{Y}_{tip} = [-s_1 s_{23}, c_1 s_{23}, c_{23}]^T \quad (4)$$

where $s_1 = \sin(\theta_1)$, $s_{23} = \sin(\theta_2 + \theta_3)$, $c_1 = \cos(\theta_1)$, and $c_{23} = \cos(\theta_2 + \theta_3)$ in that θ_i ($i = 1, 2, 3$) is the joint variable of this finger. As noticed, the value of θ_1 and $\theta_2 + \theta_3$ can be solved from the desired contact normal expressed in the current palm's frame. The position IK of this finger expressed in the current palm's frame is then solved as a problem of constrained nonlinear minimization with the constraints being the resultant value of θ_1 and $\theta_2 + \theta_3$ (denoted as $\theta_{1,d}$ and $\theta_{23,d}$, respectively) as well as their joint limits,

$$\begin{aligned} \min_{\theta_2, \theta_3} & \| {}^{hp} \vec{p}_d - {}^{hp} \vec{p}_c \|^2 \\ \text{s.t. } & \theta_1 = \theta_{1,d}, \quad \theta_2 + \theta_3 = \theta_{23,d} \\ & \theta_2, \theta_3 \in [\theta_{i,min}, \theta_{i,max}] \quad i = 2, 3 \end{aligned} \quad (5)$$

where ${}^{hp} \vec{p}_d$ and ${}^{hp} \vec{p}_c$ are the desired and the current contact point of this finger expressed in the current palm's frame. We employed the MATLAB built-in function "fmincon" to solve (5) in this work.

2) *The arm-thumb closed chain*: If other fingers' errors are still unacceptable after the previous step, this means the palm's orientation was ill-estimated in the thumb reaching phase, which hampers the achievement of other fingers' tasks. To adjust the palm's pose without interfering with the achieved thumb-related tasks, we formulate the arm-thumb system as a closed chain (see Fig. 1) with the palm being the end-effector and the thumb and the arm being the two supporting "legs". A virtual revolute joint is attached to the thumb's tip with the rotation axis aligning with the thumb's contact normal such that it acts as the embodiment of the thumb's functional redundancy. In this closed-chain formulation, the thumb is controlled reversely (i.e., the thumb's tip and the virtual revolute joint are the base and the first joint of this "leg", respectively). Otherwise, the thumb's joints including the virtual joint would not influence the palm's pose. By selecting this virtual joint as well as other thumb's joints as active joints of the arm-thumb closed

chain, we can directly exploit the thumb's self-motion and functional redundancy without using null space projection.

3) *Jacobian formulation of the arm-thumb closed chain*: Following the procedure presented in [16], we formulate the screw-based body Jacobian of the arm-thumb closed chain as follows. The kinematic constraint in this closed chain is

$$J_r \vec{q}_t = J_{arm} \vec{q}_{arm} \Leftrightarrow [J_r \quad -J_{arm}] [\vec{q}_t^T \quad \vec{q}_{arm}^T]^T = \vec{0} \quad (6)$$

where J_r and J_{arm} are the screw-based body Jacobian of the reversed thumb and the arm, respectively, and \vec{q}_t and \vec{q}_{arm} are the corresponding joint variables. In the case of the KUKA-Barrett system, since the mobility of the arm-thumb closed chain is four calculated by Grübler's formula [17], each "leg" is required to have at least four DOFs otherwise the end-effector singularity would appear. But the thumb only has three DOFs (including the virtual joint). To prevent potential end-effector singularities and maintain the constraint (6), we mathematically consider the 7th joint of KUKA (denoted as q_{arm7}) as part of \vec{q}_t rather than \vec{q}_{arm} .

Since the mobility of the arm-thumb closed chain is four, four active joints are required. Although any joints can be chosen as active joints in general, it is suggested to select the joints whose rotation axes are not coplanar to avoid possible singularities caused by the selection of active joints [16]. In the current work, we chose the following joints as the active joints: the virtual revolute joint (q_v), the thumb's 2nd joint (q_{t2}), and the arm's 7th and 6th joint (q_{arm7} and q_{arm6}). After rearranging the joints into the groups of active joints and passive joints, (6) can be written as

$$[H_a \quad H_p] [\vec{q}_a^T \quad \vec{q}_p^T]^T = \vec{0} \Leftrightarrow \vec{q}_p = -H_p^{-1} H_a \vec{q}_a = H \vec{q}_a \quad (7)$$

where $\vec{q}_a = [q_v, q_{t2}, q_{arm7}, q_{arm6}]^T$ and $\vec{q}_p = [q_{t1}, q_{arm1}, \dots, q_{arm5}]^T$ are the active and passive joint variables of the arm-thumb closed chain, respectively, and q_{t_i} ($i = 1, 2$) and q_{arm_i} ($i = 1, \dots, 7$) are the joint variables of the thumb and the arm, respectively. The arm-thumb closed chain's Jacobian (J_{cc}) is then obtained from the arm's Jacobian (excluding the 7th joint) as,

$$J_{cc} = J_{arm} [\vec{h}_1^T, \dots, \vec{h}_5^T, \vec{e}_6^T]^T \quad (8)$$

where \vec{h}_i ($i = 1, \dots, 5$) is the i -th row of the H matrix in (7) and $\vec{e}_6 = [0, 0, 0, 1]$. After constructing the arm-thumb closed chain's Jacobian matrix, we need to find the palm's desired motion to minimize other fingers' errors. We describe the palm's desired motion by the difference between its desired and current position and orientation (denoted as \vec{e}_{hp_p} and \vec{e}_{hp_o} , respectively). \vec{e}_{hp_p} and \vec{e}_{hp_o} are computed from other fingers' errors.

4) *Calculation of the palm's rotational motion \vec{e}_{hp_o}* : Let us denote ${}^c_d R_{hp}$ as the rotation matrix from the palm's current orientation to its desired orientation. Suppose other fingers' orientation errors can be eliminated with the palm's desired orientation, we have the following equations,

$${}^b_{hp} R_c \cdot {}^c_d R_{hp} \cdot {}^{hp}_{F_2} R_c = {}^b_{hp} R_c \cdot {}^{hp}_{F_2} R_c \cdot {}^c_d R_{F_2} \quad (9)$$

$${}^b_{hp} R_c \cdot {}^c_d R_{hp} \cdot {}^{hp}_{F_3} R_c = {}^b_{hp} R_c \cdot {}^{hp}_{F_3} R_c \cdot {}^c_d R_{F_3} \quad (10)$$

where ${}^{hp}R_c$ and ${}^{hp}R_{F_i}$ are the current rotation matrix from the robot base to the hand palm and that from the palm to the tip of Finger i ($i = 2, 3$), respectively. ${}^cR_{F_i}$ is the rotation matrix from the current to the desired contact normal of Finger i (denoted as \vec{n}_{i_c} and \vec{n}_{i_d} , respectively) calculated as,

$${}^cR_{F_i} = I_3 + \hat{w}\sin(\zeta) + \hat{w}^2(1 - \cos(\zeta)) \quad (11)$$

where $\vec{w} = \vec{n}_{i_c} \times \vec{n}_{i_d}$, $\hat{w} = \text{skew}(\frac{\vec{w}}{\|\vec{w}\|})$ in that $\text{skew}(\cdot)$ is the operator for calculating the skew-symmetric matrix from a 3×1 vector, and $\zeta = \arccos(\frac{\vec{n}_{i_c} \cdot \vec{n}_{i_d}}{\|\vec{n}_{i_c}\| \|\vec{n}_{i_d}\|})$, ($i = 2, 3$). From (9) and (10), we can obtain,

$$\underbrace{\begin{bmatrix} I_3 \\ I_3 \end{bmatrix}}_A {}^cR_{hp} = \underbrace{\begin{bmatrix} {}^{hp}R_c \cdot {}^cR_{F_2} \cdot {}^{hp}R_c^T \\ {}^{hp}R_c \cdot {}^cR_{F_3} \cdot {}^{hp}R_c^T \end{bmatrix}}_B \quad (12)$$

The least squares solution of (12) is ${}^cR_{hp} = (A^T A)^{-1} A^T B$. To guarantee ${}^cR_{hp} \in \text{SO}(3)$, singular value decomposition (SVD) is applied. The palm's rotational motion (\vec{e}_{hp_o}) is obtained as $\vec{e}_{hp_o} = \text{unskew}(\log {}^cR_{hp})$, where $\text{unskew}(\cdot)$ is the operator for attaining the corresponding 3×1 vector from a skew-symmetric matrix.

5) *Calculation of the palm's translational motion \vec{e}_{hp_p}* : \vec{e}_{hp_p} is obtained by averaging the palm's translational motions for minimizing the errors of Finger i (denoted as $\vec{e}_{hp_{ip}}$, $i = 2, 3$) as

$$\vec{e}_{hp_p} = \frac{1}{2} \sum_{i=1}^2 \vec{e}_{hp_{ip}}, \quad \vec{e}_{hp_{ip}} = \vec{e}_{F_{ip}} - \vec{e}_{F_{io}} \times {}^{hp}t_{F_i}^T \quad (13)$$

where $\vec{e}_{F_{ip}}$ and $\vec{e}_{F_{io}}$ are the position and orientation errors of Finger i ($i = 2, 3$), and ${}^{hp}t_{F_i}^T$ ($i = 2, 3$) is the translation vector from the palm to the fingertip. $\vec{e}_{F_{ip}}$ and $\vec{e}_{F_{io}}$ ($i = 2, 3$) are calculated by

$$\begin{bmatrix} \vec{e}_{F_{io}} \\ \underline{0} \end{bmatrix} \quad \vec{e}_{F_{ip}} = \log \begin{bmatrix} {}^cR_{F_i} & {}^c t_{F_i} \\ \underline{0} & 1 \end{bmatrix} \quad (14)$$

where $[\vec{e}_{F_{io}}]$ is the skew-symmetric matrix corresponding to $\vec{e}_{F_{io}}$ ($i = 2, 3$), ${}^cR_{F_i}$ is calculated from \vec{n}_{i_c} and \vec{n}_{i_d} as per (11), and ${}^c t_{F_i} = \vec{p}_{i_d} - \vec{p}_{i_c}$ in that \vec{p}_{i_d} and \vec{p}_{i_c} are the desired and current contact point of Finger i ($i = 2, 3$). Note that \vec{n}_{i_c} , \vec{n}_{i_d} , \vec{p}_{i_c} and \vec{p}_{i_d} are expressed in the palm's frame as screw-based body Jacobian is used for the arm-thumb closed chain.

6) *IK of the arm-thumb closed chain*: After obtaining the palm's desired motions (\vec{e}_{hp_p} and \vec{e}_{hp_o}), we now need to find the joint movements of the arm-thumb closed chain to conduct these motions. For the active joints' movement ($\Delta \vec{q}_a$), the weighted damped least-squares method [18] with null space projection is employed. The position and orientation IK of the arm-thumb closed chain are solved independently and the resulting joint movements are combined by using null space projection to obtain $\Delta \vec{q}_a$ as $\Delta \vec{q}_a = \Delta \vec{q}_{a_p} + N_{hp_p} \Delta \vec{q}_{a_o}$, where $N_{hp_p} = I - J_{cc_p}^\dagger J_{cc_p}$ and J_{cc_p} are the null space projector of the palm's position and the arm-thumb closed chain's position Jacobian, respectively, and $\Delta \vec{q}_{a_p}$ and $\Delta \vec{q}_{a_o}$ are the active joints' movement to minimize \vec{e}_{hp_p} and \vec{e}_{hp_o} , respectively. After obtaining $\Delta \vec{q}_a$, we calculate the passive

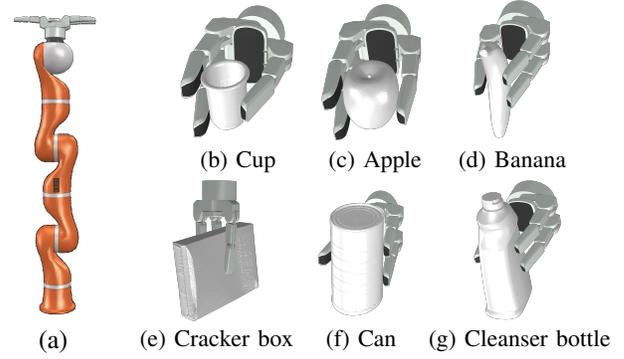


Fig. 4. (a) The initial arm-hand configuration (b)-(g) Barrett hand configurations for grasping different objects.

joints' movement ($\Delta \vec{q}_p$) as per (7). It is noteworthy that the calculated joint movements related to the thumb (including the virtual joint and the arm's 7th joint) in $\Delta \vec{q}_a$ and $\Delta \vec{q}_p$ have to take the opposite values of the calculated ones since the thumb is reversely controlled in the arm-thumb closed chain.

To prevent the arm-thumb closed-chain formulation from collapse, we took two precautions in the weighted damped least-squares method [18]. First, we selected a considerably large damping factor to limit the joint motions of the arm-thumb closed chain. Second, a large weighting factor is assigned to the virtual revolute joint since we observed that this virtual joint affects more on the palm's motion than other active joints.

III. SIMULATIONS

To test the performance of the proposed approach, a series of offline numerical tests were conducted. All the tests were conducted in MATLAB and visualized in V-REP [19]. The kinematic model of the KUKA-Barrett system was used during these experiments. The steps of the simulations are listed below,

- 1) The initial configuration of the KUKA-Barrett system for each run was set as shown in Fig. 4(a).
- 2) The hand configurations for grasping 12 different objects from the well-known YCB dataset [20] were found as shown in Fig. 4(b)-(g).
- 3) For each hand configuration, randomly generate 10,000 arm configurations that are uniformly distributed within the arm's joint limits.
- 4) For each hand configuration with its corresponding 10,000 arm configurations, calculate the forward kinematics (FK) to generate 10,000 different sets of feasible contact points and contact normals.
- 5) Solve the IK of the integrated KUKA-Barrett system to achieve the contact points and contact normals calculated in the previous step.
- 6) Collect and analyze the data.

To solve the IK of each independent task, the classic damped least-squares (DLS) method [1] and its variant, selectively damped least-squares (SDLS) [21] were employed in the proposed approach. In what follows, we term the proposed approach with DLS and SDLS as "Enhanced DLS" and "Enhanced SDLS", respectively. For each run,

TABLE I
COMPARISON BETWEEN THE METHODS WITH AND WITHOUT THE PROPOSED APPROACH

Objects	Cup				Apple				Banana			
Methods	Enhanced SDLS	Original SDLS	Enhanced DLS	Original DLS	Enhanced SDLS	Original SDLS	Enhanced DLS	Original DLS	Enhanced SDLS	Original SDLS	Enhanced DLS	Original DLS
Success Rate (%)	97.74	2.79	96.96	13.41	95.20	3.52	95.33	10.35	95.79	1.89	96.77	12.93
Avg Time (s)	0.2791	1.254	0.2116	0.139	0.5537	1.663	0.2527	0.126	0.6497	1.623	0.3346	0.127
Ave Iterations	184.34	2460.6	170.87	219.8	219.10	3249.1	169.70	198.2	212.25	3177.7	165.77	211.8
Objects	Cracker box				Can				Cleanser bottle			
Methods	Enhanced SDLS	Original SDLS	Enhanced DLS	Original DLS	Enhanced SDLS	Original SDLS	Enhanced DLS	Original DLS	Enhanced SDLS	Original SDLS	Enhanced DLS	Original DLS
Success Rate (%)	88.28	3.00	94.49	9.98	93.03	3.55	94.72	9.60	95.93	2.84	96.90	11.06
Avg Time (s)	1.003	2.071	0.3204	0.139	1.076	1.963	215.26	0.153	0.6614	1.908	0.3288	0.146
Avg Iterations	285.98	4027.1	184.45	219.2	291.27	3728.5	0.6122	254.1	226.56	3497.5	178.82	255.0

the maximum number of iterations was set as 10,000. The error tolerance for the position and orientation error of each finger was set to 5mm and 3°, respectively. The parameters in the “Enhanced DLS” and the “Enhanced SDLS” were set as follows. In the thumb reaching phase (with and without the “palm orientation IK” task), the damping factor (λ) in “Enhanced DLS” was set to be 1. In “Enhanced SDLS”, the maximum joint change (γ_{max}) for dampening the joint motion was set to be $\frac{\pi}{9}$, and the maximum distance and maximum angle for clamping the position and orientation errors were set to be 200mm and $\frac{\pi}{6}$, respectively. The same values were also used in the methods without the enhancement of the proposed approach (termed as “Original DLS” and “Original SDLS”, respectively). For solving the IK of the arm-thumb closed chain, λ in “Enhanced DLS” and γ_{max} in “Enhanced SDLS” were set to be 150 and $\frac{\pi}{500}$, respectively. The weighting factor for the virtual revolute joint was set to be 10 while the factors for other active joints of the arm-thumb closed chain were set to be 1.

To show the superb performance of the proposed approach, the results of “Enhanced DLS” and “Enhanced SDLS” were compared with the results of “Original DLS” and “Original SDLS”. The results are summarized in Table I. All data were acquired using MATLAB r2019b on a personal computer powered by an i5-6400 CPU @ 2.70GHz and 16GB RAM. In this table, “Success Rate” is defined to be the ratio between the number of solved cases and the total number of cases (i.e., 10,000). For each run, one case is considered to be solved if the magnitudes of all the fingers’ position and orientation errors are less than or equal to the error tolerances. The “Average Iterations” and “Average Time” are the average number of iterations and the average computation time consumed over all solved cases, respectively.

As observed in Table I, the enhanced methods significantly outperformed the original methods thanks to the thumb-first strategy and the arm-thumb closed-chain formulation. These two techniques allow our approach to handle the intrinsic conflicts of the arm-hand system effectively and accomplish multiple tasks simultaneously and accurately. During the IK solving process, not considering the conflicting requirements will lead to system oscillation caused by algorithmic singularities and affect the convergence speed, or even non-convergence with non-negligible errors. These consequences are manifested in the unsatisfactory performance of “Original

DLS” and “Original SDLS”.

Numerical tests also revealed the limitations of the proposed solution. The arm-thumb closed-chain formulation confronts three challenges: (1) the narrow workspace, (2) the unknown range of the virtual revolute joint, and (3) the numerous singularities. In this work, the treatments for these challenges were relegated to the regularization with large damping and weighting factors and the compensation of extra thumb-related tasks, respectively. But these treatments cannot guarantee the stability of the arm-thumb closed chain. Furthermore, although large damping and weighting factors refrained the arm-thumb closed chain from unexpected big motions, it lowered the convergence speed as well. We believe the success rate and the accuracy of the proposed algorithm can be further improved after we thoroughly study the workspace, the virtual revolute joint, and the singularities of the arm-thumb closed chain.

IV. CONCLUSIONS

In this paper, a novel inverse kinematics (IK) solution was proposed to achieve precision grasps using an integrated arm-hand system. In the proposed approach, we apply the human-inspired thumb-first strategy to narrow down the search space. More importantly, we propose to formulate the arm-thumb serial chain as a closed chain and attach a virtual revolute joint to the thumb tip as the embodiment of the thumb’s functional redundancy. By selecting the thumb’s joints including this virtual revolute joint as the active joints in the arm-thumb closed chain, we can directly control the arm-thumb system’s self-motion and the thumb’s functional redundancy without using null space projection. This provides a new possibility to control the self-motion of a robot manipulator. Simulation results manifest the superb performance of the proposed approach.

To further improve the proposed approach, our future works include the following aspects. The workspace and the singularities of the arm-thumb closed chain are going to be thoroughly studied in the future. The range of the virtual revolute joint on the thumb’s tip with different arm configurations is going to be investigated as well.

REFERENCES

- [1] C. W. Wampler, “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 16, no. 1, pp. 93–101, 1986.

- [2] Y. Hu, B. Huang, and G.-Z. Yang, "Task-priority redundancy resolution for co-operative control under task conflicts and joint constraints," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2398–2405.
- [3] G. Antonelli, "Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems," *IEEE Transactions on Robotics*, vol. 25, no. 5, pp. 985–994, 2009.
- [4] A. Dietrich, A. Albu-Schäffer, and G. Hirzinger, "On continuous null space projections for torque-based, hierarchical, multi-objective manipulation," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2978–2985.
- [5] 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.
- [6] J. Lee, P. H. Chang, and R. S. Jamisola, "Relative task prioritization for dual-arm with multiple, conflicting tasks: Derivation and experiments," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1928–1933.
- [7] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, "More than a feeling: Learning to grasp and regrasp using vision and touch," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, 2018.
- [8] G. Jarquín, A. Escande, G. Arechavaleta, T. Moulard, E. Yoshida, and V. Parra-Vega, "Real-time smooth task transitions for hierarchical inverse kinematics," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 528–533.
- [9] A. Dietrich, T. Wimbock, A. Albu-Schäffer, and G. Hirzinger, "Integration of reactive, torque-based self-collision avoidance into a task hierarchy," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1278–1293, 2012.
- [10] E. POUYDEBAT, M. LAURIN, P. GORCE, and V. BELS, "Evolution of grasping among anthropoids," *Journal of evolutionary biology*, vol. 21, no. 6, pp. 1732–1743, 2008.
- [11] R. W. Young, "Evolution of the human hand: the role of throwing and clubbing," *Journal of Anatomy*, vol. 202, no. 1, pp. 165–174, 2003.
- [12] Y. Lin and Y. Sun, "Robot grasp planning based on demonstrated grasp strategies," *The International Journal of Robotics Research*, vol. 34, no. 1, pp. 26–42, 2015.
- [13] G. Cotugno, K. Althoefer, and T. Nanayakkara, "The role of the thumb: study of finger motion in grasping and reachability space in human and robotic hands," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1061–1070, 2016.
- [14] L. Žlajpah, "On orientation control of functional redundant robots," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2475–2482.
- [15] W. Kahan, "How futile are mindless assessments of round-off in floating-point computation?" <https://people.eecs.berkeley.edu/~wkahan/Mindless.pdf>, 2006, [Online].
- [16] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.
- [17] K. H. Hunt, *Kinematic geometry of mechanisms*. Oxford University Press, USA, 1978, vol. 7.
- [18] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.
- [19] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321–1326.
- [20] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.
- [21] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics tools*, vol. 10, no. 3, pp. 37–49, 2005.