Real-Time Finger Gaits Planning for Dexterous Manipulation ⋆⋆

Yongxiang Fan * **Wei Gao** ** **Wenjie Chen** ***
Masayoshi Tomizuka *

* *Department of Mechanical Engineering, University of California, Berkeley, CA, 94720, USA (e-mail: {yongxiang_fan, tomizuka}@berkeley.edu).*
** *School of Aerospace, Tsinghua University, Beijing, 100084, P. R. China (e-mail: gaow13@mails.tsinghua.edu.cn)*
*** *FANUC Corporation, Yamanashi Prefecture, 401-0597, Japan (e-mail: wjchen@berkeley.edu)*

Abstract:
Dexterous manipulation has broad potential applications in assembly lines, warehouses and agriculture, and so on. To perform large-scale, complicated manipulation tasks, a multi-fingered robotic hand sometimes has to sequentially adjust its grasping status, i.e. the finger gaits, to deal with the workspace limits and object stability. However, realizing finger gaits planning in dexterous manipulation is challenging due to the involved hybrid dynamics, complicated grasp quality metrics, and uncertainties during the finger gaiting. In this paper, a dual-stage optimization based controller is proposed to handle these challenges. First, a velocity-level finger gaits planner is introduced by combining object grasp quality with hand kinematic limitations. The proposed finger gaits planner is computationally efficient and can be solved in real-time. Second, a manipulation controller using force optimization is presented. To deal with mass uncertainties and external disturbances, a modified impedance control is integrated into the manipulation controller. The dual-stage controller does not require the shape of the object, nor does it rely on expensive 3D/6D tactile sensors. Simulation results verify the efficacy of the proposed dual-stage controller.

Keywords: dexterous manipulation, finger gaits planning, multi-fingered hands, autonomous robotic systems, robotics technology, modeling for control optimization, real-time control.

1. INTRODUCTION

Dexterous manipulation is essential for manipulators to execute complicated tasks, such as circuit assembly, commodity organizing, and fruit harvesting. To perform large-scale complex manipulations, a robotic hand may have to change its grasping status by relocating fingers during the manipulation, which gives the hand more dexterity and robustness. Such strategy is called finger gaits planning. However, finger gaits planning in dexterous manipulation is difficult due to factors such as high dimensionality, intermittent contact dynamics, and complicated grasp quality metrics.

As a result, problems related to dexterous manipulation and finger gaits planning have received significant attention. Bicchi (2000) summarized the challenges of dexterous manipulation and finger gaiting, namely, the analysis and control of hybrid systems during gaits planning, and the optimization of the plans. Khalil and Payeur (2010) reviewed modeling and control techniques during the robot-environment interaction. Kumar et al. (2016) learned a local model for dexterous manipulation by reinforcement learning. Erez et al. (2014) achieved a lifting task robustly by applying model predictive control. Ciocarlie and Allen (2009) searched hand posture subspaces by simulated annealing method to realize pre-grasping in interactive grasping tasks. Liu (2009) achieved dexterous manipulation by using contact force planning and hand motion synthesis. However, none of them handled objects with large-scale motions in constrained workspaces.

An effective solution to the limited hand workspaces is finger gaits planning. Andrews and Kry (2013) trained task-specific finger gaiting policies by the covariance matrix adaptation method, given the goal states of objects. However, the learned policies cannot be adapted to other objects and tasks. Furukawa et al. (2006) used a high-speed hand and a high-speed vision system to perform dynamic re-grasping. However, the object model should be precisely known, and the presented success rate (35%) is not suitable for many applications. Li et al. (2014) learned impedance parameters from human demonstration for robust grasping and dexterous manipulation. Vinayavekhin et al. (2011) used a tangle topology to reproduce object pose from learned human demo. However, the object gravity is not

⋆ The order of the first two authors was determined by a coin toss.
⋆⋆This project was supported by FANUC Corporation.

considered during their gaits changing process. Platt Jr (2006) proposed a set of controllers to realize unknown object grasping by sliding on the surface to maximize grasp stability. Platt et al. (2004); Mordatch et al. (2012) explored the unknown surface of the object by designing a global re-grasping planner and searching local optimal contact points. However, predefined finger gaits are used in these approaches, and the exploration of local optima does not incorporate necessary constraints, such as joint velocity and acceleration limitations. As a result, these approaches tend to be slow in re-grasping and manipulation, and the predefined finger gaits might be inapplicable to other objects and robotic hands. Xu et al. (2010) used a sampling-based method to plan finger gaits. Mordatch et al. (2012) used contact-invariant optimization method to compute the states of the hand and the object, given the high-level goals. These approaches are not computationally efficient for real-time finger gaits planning.

In this paper, a dual-stage optimization based controller is developed, which consists of a finger gaits planner and a manipulation controller. Instead of formulating the finger gaits planner in position-level, which results in a complicated non-linear constrained optimization problem, the finger gaits planner is formulated in the velocity level, i.e., the joint velocity $\dot{q}$ is selected to be the decision variable. At each time step, an optimal joint velocity is computed to improve the hand manipulability as well as the object grasp quality, and the calculated joint velocity is converted to joint torque by a velocity-force controller. The proposed manipulation controller consists of a contact force optimizer and a joint-level torque controller. An impedance control scheme is incorporated into the manipulation controller to deal with uncertainties such as imprecise dynamic parameters and external disturbances.

The proposed dual-stage optimization based controller achieves real-time finger gaits planning in dexterous manipulation, under the object and environment uncertainties. To be more specific, the velocity-level finger gaits planner is cast into a linear programming (LP) problem, which is computationally efficient and can be solved in real-time. Furthermore, the velocity-level gaits planning incorporates joint kinematic constraints, which makes the generated motions feasible. The proposed manipulation controller can handle a certain amount of mass uncertainties and external disturbances. The proposed dual-stage optimization based controller does not rely on exact shapes of the objects, nor does it require expensive 3D/6D tactile sensors. The efficacy of the proposed controller is verified by simulations. The video demo is available at Fan et al. (2016).

This paper is organized as follows. Section 2 shows the overall dual-stage optimization based controller framework. Section 3 explains the grasp quality analysis. The velocity-level gaits planner and manipulation controller are presented in Section 4 and Section 5, respectively. Section 6 shows simulation results on a robotic hand with four fingers and twelve degrees of freedom (DOFs). Section 7 concludes the paper.

2. OPTIMIZATION BASED CONTROLLER FRAMEWORK

Figure 1 shows the proposed optimization-based controller framework. First, a grasp quality analysis is conducted by computing hand manipulability and object grasp quality, and the weakest finger is chosen to break contact and change gait once the overall quality drops below a predefined threshold. The index of the breaking finger is denoted as I_b. A velocity-level gaits planner is evoked by this event, and the planner generates torque command τ_{I_b} to drive the selected breaking finger towards the better quality region.The remaining fingers are controlled by a manipulation controller to manipulate the object stably, as shown in Fig. 1. If the overall quality is above the threshold, all fingers will be controlled by the manipulation controller.

A joint level torque tracking controller is also designed to track the torque command. The torque tracking controller uses a PID scheme and runs at a higher frequency, in comparison with the gaits planner and the manipulation controller.

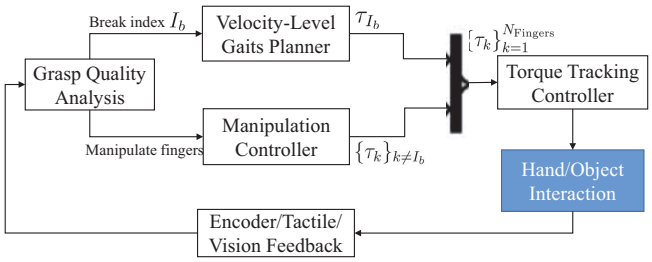


Fig. 1. The general framework of the proposed optimization based controller.

The following sections will focus on the grasp quality analysis, the velocity level gaits planner, and the manipulation controller, respectively.

3. GRASP QUALITY ANALYSIS

Grasp quality has been well explored in Roa and Suárez (2015); Kim et al. (2001); Murray et al. (1994). During finger gaits planning, both hand manipulability and object grasp quality should be considered. The hand manipulability here means the ability for a hand to manipulate the object to realize arbitrary motions. The object grasp quality means the capacity to resist external disturbances given a group of contact points on the object. In this paper, we adopt a quality metric in Liegeois (1977) to represent hand manipulability Q_h:

$$Q_h = -\frac{1}{2}\sum_{j=1}^{N_{\text{finger}}}\sum_{i=1}^{N_{\text{joint}}}\left(\frac{q_j^i - \bar{q}_j^i}{q_{\text{max,j}}^i - q_{\text{min,j}}^i}\right)^2 \tag{1}$$

where q_j^i is the joint angle of the i-th joint of the j-th finger, $q_{\text{min,j}}^i$ and $q_{\text{max,j}}^i$ are the limits of q_j^i, $\bar{q}_j^i = (q_{\text{max,j}}^i + q_{\text{min,j}}^i)/2$ is the middle position of the corresponding joint, N_{finger} and N_{joint} are the number of fingers, the number of joints per finger, respectively.

The object grasp quality Q_o is adopted from Supuk et al. (2005), which can be represented as:

$$Q_o = 2\text{Area}\left(p_1, p_2, p_3, \text{proj}(p_{I_b})\right) \tag{2}$$

where p_j means the contact position in Cartesian space for the j-th fingertip, and p_{I_b} indicates the position of the free fingertip. proj(p_{I_b}) means the projection operation of p_{I_b} onto the plane specified by p_1, p_2, p_3. An illustration of Q_o is shown in Fig. 2.

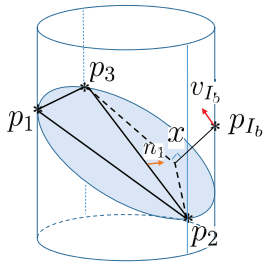


Fig. 2. Object grasp quality illustration. We indicate the case where the 4-th finger is relocating. p_1, p_2, p_3 are the static contact points, while p_{I_b} represents the position of the 4-th fingertip. The shaded ellipse indicates the plane specified by p_1, p_2, p_3. The object grasp quality used in this paper is the area of convex hull formed by p_1, p_2, p_3, x, where x is the projection point of p_{I_b} onto the plane.

The overall quality Q can be obtained by combining (1) and (2):

$$Q = w_1 Q_o + w_2 Q_h \tag{3}$$

where $w_i > 0$ is the weight for the corresponding term.

Once the overall grasp quality Q drops below a certain threshold, the hand gaits should be replanned to adjust contact points on the object. It is observed that humans tend to relocate their fingers one by one during the finger gaits planning. We adopt this philosophy and sequentially plan the finger gaits. Thus, our algorithm will compare all the fingers and choose one of them to initialize finger gaits planning, if all fingertips are in static contacts and $Q < \delta_Q$, where δ_Q is a threshold.

Our selection is based on the finger manipulability and the grasp quality of the remaining fingers. To be more specific, the finger manipulability for the I_b-th finger is:

$$-\sum_{i=1}^{N_{\text{joint}}} \left((q^i_{I_b} - \bar{q}^i_{I_b})/(q^i_{\max,I_b} - q^i_{\min,I_b})\right)^2$$

The grasp quality of remaining fingers is the area of convex hull spanned by remaining fingertips, for instance, the triangle $\Delta p_1 p_2 p_3$ in Fig. 2. The preferred finger for gaiting is the one with small finger manipulability and large remaining grasp quality. If there is already one free finger, that finger will continue its gaiting.

4. VELOCITY-LEVEL FINGER GAITS PLANNER

The task of the finger gaits planner is to generate commands to change the contact location of a finger, to achieve better object grasp quality and hand manipulability. In this section, the related position-level gaits planner is first presented; then the velocity-level gaits planner is proposed to resolve the problems in a position-based planner.

4.1 Position-Level Finger Gaits Planner

Position-level finger gaits planner consists of contact optimization and trajectory planning. The contact optimization finds optimal contact points under the overall grasp quality (3), and the trajectory planning generates trajectories to relocate fingers to optimal contact points.

The contact optimization can be represented in the following form:

$$\max_{s, q_{I_b}} \quad Q \tag{4a}$$
$$s.t. \quad p_{I_b} \in \partial O \tag{4b}$$
$$\|p_{I_b} - p_{I_b,o}\| \le \epsilon \tag{4c}$$
$$p_{I_b} = \text{FK}(q_{I_b}) \tag{4d}$$
$$q_{\min,I_b} \le q_{I_b} \le q_{\max,I_b} \tag{4e}$$

Constraint (4b) indicates that the fingertip position of free finger p_{I_b} should be on the surface of object ∂O. Constraint (4c) means that the searching region should be constrained in certain region ϵ from original position $p_{I_b,o}$ to roughly keep the stability of the object. Constraint (4d) is the forward kinematics of the robotic hand. Constraint (4e) means that the joint space searching should be in the feasible region.

After finding the optimal contact points, the trajectory planning algorithm is required to generate a feasible trajectory. Generally speaking, the trajectory planning should be able to consider the kinematic constraints (e.g. joint limits, velocity and acceleration constraints), the time optimality to reach optimal contact points, and collision avoidance with the object.

The position-level finger gaits planning has the following drawbacks. First, the problem has non-linear equality constraints. Therefore, it is difficult for real-time computing. Besides, this method requires a complicated trajectory planning to reach the planed optimal point. Furthermore, the equality constraint (4b) corresponding to object surface is usually unknown in advance. Lastly, the contact optimization (4) uses current p_1, p_2, and p_3 to find the optima, while p_1, p_2, and p_3 actually keep moving during the contact optimization, trajectory planning and execution.

With the all listed issues above, an efficient velocity-level finger gaits planner is proposed below.

4.2 Velocity-Level Finger Gaits Planner

The position-level finger gaits planner divides finger gaits planning into a contact optimization and a trajectory planning. Both of them are complicated and difficult to solve in real-time. Therefore, an efficient approximation of the position-level gaits planner is proposed, which is called the velocity-level gaits planner. In this planner, the contact optimization is modified into a short-term optimization. To be more specific, rather than finding an optimal contact point, an optimal moving velocity is calculated at each time step, and the finger is actuated to achieve that velocity. Formally, instead of optimizing Q in (3), $\dot{Q}$ is optimized with joint velocity of the I_b-th finger $\dot{q}_{I_b}$ as the decision variable. The solution $\dot{q}_{\text{des},I_b}$ is used to control the robotic hand in each time step.

The intuition behind it is the Taylor series expansion. Q is a function of states $\{q_{I_b}, p_{I_b}\}$, and the states are the functions of time t. Therefore, Q is a function of t. By this interpretation, Q in time instant $t+T_s$ can be written as:

$$Q(t+T_s) \approx Q(t) + \dot{Q}(t)T_s \tag{5}$$

where T_s is the time step. In this equation, higher order terms have been omitted, because T_s is usually a small period. Therefore, designing control policy to maximize $Q(t+T_s)$ is equivalent to maximizing $\dot{Q}(t)$.

With the short term approximation, $\dot{Q}$ becomes:

$$\begin{aligned} \dot{Q} &= w_1\dot{Q}_o + w_2\dot{Q}_h \\ \dot{Q}_h &= \sum_{i=1}^{N_{\text{joint}}} \left(\frac{\bar{q}^i_{I_b} - q^i_{I_b}}{(q^i_{\max,I_b} - q^i_{\min,I_b})^2} \dot{q}^i_{I_b} \right) \\ \dot{Q}_o &= ||p_2 - p_3||_2 n_1^T v_{I_b} \end{aligned} \tag{6}$$

where $\dot{q}^i_{I_b}$ is joint velocity of the i-th joint of the I_b-th finger. n_1 is a normal vector of line segment $\overline{p_2p_3}$ in the plane specified by p_1, p_2, p_3, as shown in Fig. 2, and v_{I_b} is the velocity of p_{I_b}. In this optimization, the states $\dot{q}_{I_b}$ and v_{I_b} in $\dot{Q}_h$ and $\dot{Q}_o$ are coupled linearly by $v_{I_b} = J(q_{I_b})\dot{q}_{I_b}$, where $J(q_{I_b})$ is Jacobian of the finger I_b. By plugging in the coupled term, $\dot{Q}$ becomes:

$$\dot{Q} = w_1\|p_2 - p_3\|_2 n_1^T J\dot{q}_{I_b} + w_2 \sum_{i=1}^{N_{\text{joint}}} c^i_{I_b} \frac{\bar{q}^i_{I_b} - q^i_{I_b}}{(q^i_{\max,I_b} - q^i_{\min,I_b})^2} \dot{q}^i_{I_b} \tag{7}$$

The weighting function $c^i_{I_b}$ is added to (7) in order to address joint limits:

$$c^i_{I_b} = \begin{cases} \ln\left(\frac{\bar{q}^i_{I_b} - q^i_{\min,I_b} - q^i_{\text{thres}}}{q^i_{I_b} - q^i_{\min,I_b}}\right) + 1, & q^i_{I_b} - \bar{q}^i_{I_b} < -q^i_{\text{thres}} \\ 1, & |q^i_{I_b} - \bar{q}^i_{I_b}| \le q^i_{\text{thres}} \\ \ln\left(\frac{q^i_{\max,I_b} - \bar{q}^i_{I_b} - q^i_{\text{thres}}}{q^i_{\max,I_b} - q^i_{I_b}}\right) + 1, & q^i_{I_b} - \bar{q}^i_{I_b} > q^i_{\text{thres}} \end{cases}$$

where q^i_{thres} is a threshold where the weighting should be increased.

In the meantime, constraints (4b) and (4d) of the contact optimization (4) become $n_{I_b}^T J(q_{I_b})\dot{q}_{I_b} = 0$, where n_{I_b} is the surface normal at p_{I_b}. The surface normal can be inferred by the tactile sensor on the fingertip. The constraint (4c) of (4) is eliminated because we are working on short-term optimization, and the optimization would be solved in each time step.

With above analysis, a new optimization problem can be formulated to approximate (4):

$$\max_{\dot{q}_{I_b}} \quad \dot{Q} \tag{8a}$$

$$s.t. \quad \dot{q}_{\min,I_b} \le \dot{q}_{I_b} \le \dot{q}_{\max,I_b} \tag{8b}$$

$$n_{I_b}^T J(q_{I_b})\dot{q}_{I_b} = 0 \tag{8c}$$

$$\|\dot{q}_{I_b} - \dot{q}_{\text{des,prev}}\|_\infty \le \sigma \tag{8d}$$

The solution of (8) for the I_b-th finger is denoted as $\dot{q}_{\text{des},I_b}$. Constraint (8b) means that the joint velocity $\dot{q}_{I_b}$ should be bounded in $[\dot{q}_{\min,I_b}, \dot{q}_{\max,I_b}]$. Constraint (8c) indicates that p_{I_b} in Fig. 2 must move perpendicular to current surface normal n_{I_b}. Constraint (8d) means that the joint acceleration should be bounded by σ/T_s. $\dot{q}_{\text{des,prev}}$ is the desired joint velocity in previous time step. The optimization (8) is a linear programming, which can be solved in real-time.

After obtaining the desired joint velocity $\dot{q}_{\text{des},I_b}$ by solving (8), a velocity-force controller is implemented to calculate the desired torque of the I_b-th finger:

$$\tau_{I_b} = K_v(\dot{q}_{\text{des},I_b} - \dot{q}_{\text{act},I_b}) + K_f J(q_{I_b})^T(f_{\text{des}} - f_{\text{act},I_b}) \tag{9}$$

where $\dot{q}_{\text{des},I_b}$ and $\dot{q}_{\text{act},I_b}$ are the desired and the actual joint velocities, f_{des} and f_{act,I_b} are the desired and the actual contact forces in *normal* direction. K_v and K_f are two gain matrices for the velocity and force components. The actual normal contact force f_{act} can be measured by 1D tactile sensor. The force component $K_f J(q_{I_b})^T(f_{\text{des}} - f_{\text{act},I_b})$ in (9) attempts to maintain the contact between the fingertip and the surface, which makes the normal vector n_{I_b} measured from the tactile sensor updated.

As the optimized Cartesian velocity of the fingertip is in the tangent space, the finger might break its contact with the object if the curvature at the contact point is excessively large. However, the force component in (9) will drive the fingertip back to the surface. Consequently, the gaiting finger might exhibit the behavior of short-range jumps.

The velocity-level gaits planner that composed of (8) and (9) has several advantages. First, the proposed planner is computationally efficient. The optimization (8) is an LP that can be solved in each time step. Second, the 3D object model is not required. Instead, 1D tactile sensor is applied to detect the contact point on the hand and infer the surface normal by the known fingertip structure, and the sensor update can be accomplished by the force component in the velocity-force controller.

The grasp quality is expected to be improved at the beginning of gaits planning. The velocity-level gaits planner can be terminated when there is little grasp quality improvement (e.g. $\dot{Q} < \delta$, where δ is a small positive number).

4.3 Similarities Between Position-Level and Velocity-Level Planners

In this section, we will demonstrate that the performance of the velocity-level planner (i.e. linear programming (8) and velocity-force controller) is similar to one step of (4) if solved by gradient projection method (Rosen (1961)). For notational simplicity, an abbreviated form of (4) is considered:

$$\max_x Q \quad \text{s.t. } h(x) = 0, g(x) \le 0 \tag{10}$$

where x is a concatenation of $[p_{I_b}^T, q_{\text{des},I_b}^T]^T$, equality constraints (4b) and (4d) are abbreviated as $h(x) = 0$, inequality constraints (4c) and (4e) are represented as $g(x) \le 0$.

In each step of the gradient projection method, the search direction d is found by projecting $\nabla_x Q$ onto the tangent space of equality constraints $T = \{y | \nabla_x h^T y = 0\}$, as shown in Fig. 3. Then, an iterative technique is employed to project the points along d, such as $\hat{y}$, onto $h(x) = 0$, until the projected point x_{k+1} lies in the set $\{x | h(x) = 0, g(x) \le 0\}$.

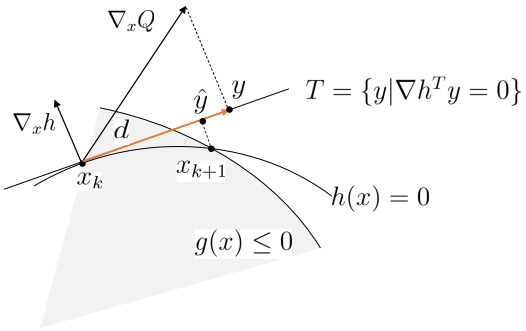


Fig. 3. Illustration of the gradient projection method. The optimal searching direction is chosen by projecting the gradient of the cost function onto the tangent space T, then an iterative technique is employed to find the best feasible projection x_{k+1}.

In the proposed method, the optimal search direction is found by LP (8). The joint-acceleration constraint (8d) and the joint-velocity constraint (8b) are incorporated into (8), instead of projecting the gradient into the tangent space. Moreover, the velocity-force controller (9) attempts to maintain the contact force between the fingertip and the surface, as a physical actualization of projecting $\hat{y}$ onto $h(x) = 0$. As the LP (8) is solved in each time step T_s, the search step $\Delta x_k = x_{k+1} - x_k$ is quite small. Thus, the planned $\dot{q}_{\text{des},I_b}$ in (8) is usually smooth.

5. MANIPULATION CONTROLLER

Given the contact points between the fingertips and the manipulated object, the task of the manipulation controller is to generate torque commands for the hand to drive the object to follow desired motions. The desired motion of the object is transformed into desired forces on the object through impedance control, which is similar to Li et al. (2014); Wimböck et al. (2012). In this section, an additional integral term is added to impedance controller to address object mass uncertainty:

$$\begin{aligned} F_{\text{imp}} =& M^d\ddot{x}_o + B^d(\dot{x}_o - \dot{x}_{\text{des}}) + K^d(x_o - x_{\text{des}}) + \\ & I^d\left(\int_0^t (x_o - x_{\text{des}})dt\right) \end{aligned} \tag{11}$$

where $\ddot{x}_{\text{des}}, \dot{x}_{\text{des}}, x_{\text{des}}$ is the desired motion of the object, $\ddot{x}_o, \dot{x}_o, x_o$ is the actual motion of the object, and M^d, B^d and K^d are the desired inertia, damping and stiffness, respectively. I^d is a gain matrix for additional integral term.

The manipulation controller consists of a force optimizer and a joint level torque controller. The force optimizer finds desired contact force of each contact point, and the joint level torque controller generates appropriate joint torques to produce the contact forces.

The force optimization is formulated into a QP problem:

$$\min_{\lambda} \quad \alpha_1\|f\|_2^2 + \alpha_2\|f - f_{\text{prev}}\|_2^2 + \alpha_3\|\Delta\|_2^2 \tag{12a}$$

$$s.t. \quad M_o(x_o)\ddot{x}_o + g(x_o) = F_{\text{des}} + F_{\text{imp}} \tag{12b}$$

$$\Delta = F_{\text{des}} - G(x_o, q)f \tag{12c}$$

$$f = B\lambda \tag{12d}$$

$$\lambda \geq 0 \tag{12e}$$

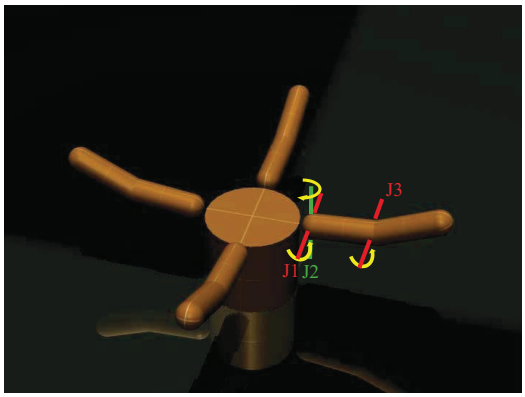


Fig. 4. The hand model that is used in the simulation. The hand has four identical fingers and 12 DOFs. Each finger has three revolute joints J1, J2, and J3. The joint angles of J1, J2 and J3 are constrained in $[-10°, 135°]$, $[-45°, 45°]$ and $[-10°, 170°]$, respectively. The hand wrist is static with respect to the world.

where $f = [f_1^T, ..., f_{N_{\text{fingers}}}^T]^T$ is the contact force at the contact frame, and f_{prev} is the contact force of the previous step. $q = [q_1^T, ..., q_{N_{\text{fingers}}}^T]^T$ is the joint angles of the hand, and x_o is the pose of the object. $M_o(x_o)$ and $g(x_o)$ are the object inertia and gravity of the object, $G(x_o, q)$ is called grasp map (Murray et al. (1994)) and transforms f from contact frame into object frame, $B = \text{diag}\{B_1, ..., B_{N_c}\}$ and B_i is a conservative pyramid approximation of friction cone that is similar to the one in Liu (2009). λ is the linear coefficient of columns of B. The desired inertia M^d is set as M_o in order to remove the inertia term and acceleration measurement in the controller (11), as shown in Li et al. (2014).

A slack variable Δ is introduced to relax the hard constraint $F_{\text{des}} = Gf$, because the location measurements of contact points might be noisy. The entries of G for I_b are set to zero, since the breaking finger does not have contribution to F_{des}. The constraints (12d) and (12e) together ensure that the contact forces remain within B. The wights α_1, α_2 and α_3 are used to balance between different cost terms.

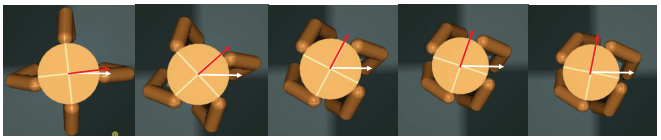
The joint level torque control takes the optimal contact force f^* from the force optimization as input, and yields the control torque as:

$$\tau = J_h(q)^T f^* + N(q)$$

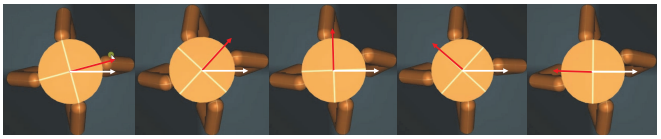
where $\tau = [\tau_1, ..., \tau_{N_{\text{finger}}}]$, $J_h(q)^T$ is the concatenated hand Jacobian transpose that maps the force on each fingertip to the torque of the joints, and $N(q)$ is the gravitational force of the hand. $\{\tau_k\}_{k \neq I_b}$ is extracted and combined with (9) for torque tracking control.

6. SIMULATIONS

In this section, simulation results are presented to verify the effectiveness of the proposed finger gaits planner and manipulation controller. The simulation video is available at Fan et al. (2016).



(a) The rotation is limited by joint position bounds without the proposed finger gaits planner.



(b) The cylinder achieves continuous rotation with the proposed finger gaits planner.

Fig. 5. The hand rotates a cylinder with/without the proposed finger gaits planner. Snapshots are from left to right.

6.1 Simulation Setup

The controller is implemented in Mujoco physical engine, which is introduced by Todorov et al. (2012). The simulation time step is set to 2 ms. Our platform is a desktop with 4.0 GHz Intel Core Quad CPU, 32GB RAM, running Windows10 operating system.

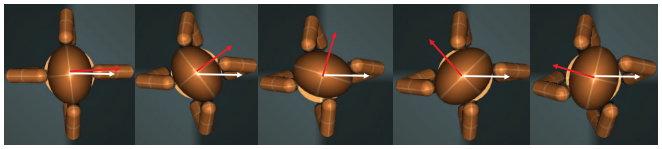
The hand is set up with four identical fingers and twelve DOFs, as shown in Fig. 4. Each finger has three revolute joints J1, J2, and J3. The joint angles of J1, J2 and J3 are constrained in $[-10°, 135°]$, $[-45°, 45°]$ and $[-10°, 170°]$, respectively. The hand is equipped with high-resolution position sensors for joint position/velocity measurements, motor torque sensors for motor torque feedback, one-dimensional distributive tactile sensors to measure normal force and infer surface normal. The manipulated objects are approximately 0.5 kg. The 3D mesh models of objects are unknown to the controller. Rather, a vision system can be employed to obtain the motion of the object by tracking the features on it. Currently, the object motion is obtained from the simulator. In future real world experiments, methods in Horn and Schunck (1981) and Lim et al. (2010) will be adopted to obtain the motion of the object.

6.2 Parameter Lists

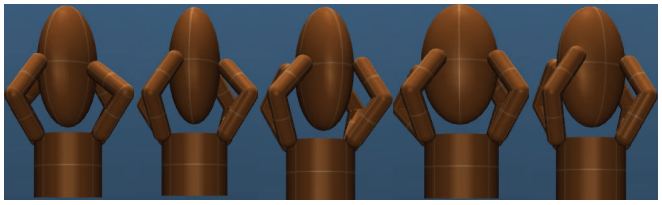
The parameter values for the velocity-level finger gaits planner are: $w_1 = 0.99, w_2 = 0.01$. $q^i_{\text{thres}} = 0.25(q^i_{\max,k} - q^i_{\min,k})$. $\dot{q}_{\min,k} = -1$ rad/s, $\dot{q}_{\max,k} = 1$ rad/s, $\sigma = 0.002$ rad/s, $\delta = 10^{-5}$. The parameter values for velocity-force controller are: $K_v = \text{diag}([0.1, 0.1, 0.1]), K_f = \text{diag}([1, 1, 1])$. For manipulation controller, $\alpha_1 = 0.01$, $\alpha_2 = 0.01$ and $\alpha_3 = 1000$. The desired parameter values for impedance controller are: $K^d = 50 \times \text{diag}([1, 1, 1, 0.2, 0.2, 0.05])$, $B^d = 5 \times \text{diag}([1, 1, 1, 0.02, 0.02, 0.005])$, and $I^d = 50 \times \text{diag}([1, 1, 1, 0.02, 0.02, 0.005])$.

6.3 Simulation Results

A lifting and rotating task is employed to verify the efficacy of the proposed gaits planner and manipulation controller. The desired object motion is to move along z axis by 11mm, and rotate around z axis with 0.2 rad/s.



(a) Top view of the hand rotating an ellipsoid



(b) Lateral view of the hand rotating an ellipsoid

Fig. 6. The hand rotates an ellipsoid with the proposed finger gaits planner. Snapshots are from left to right. (a) is the top view, (b) is the lateral view.

Figure 5 illustrates the performance with/without the proposed velocity-level gaits planner. The white and red arrows indicate the initial and current rotational poses of the object. The object can not be rotated over 90° without the proposed gaits planner because of the hand manipulability limitation, as shown in Fig. 5a. The object can be continuously rotated with the proposed planner, which improves both object grasp quality and hand manipulability, as shown in Fig. 5b. The average computation time for solving the dual-stage optimization based controller is less than one millisecond for each time step.

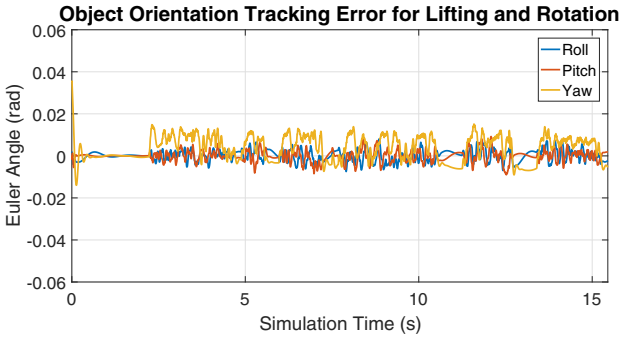
The robustness of the proposed gaits planner to different shapes is demonstrated by lifting and rotating an ellipsoid, as shown in Fig. 6. An exactly same controller is employed for ellipsoid manipulation. Figure 6a shows the top view and Fig. 6b is the corresponding lateral view. Compared with cylinder, the ellipsoid is less symmetric and the surface normal has more change. The performance is similar, except that the gaiting steps are smaller. This is influenced by the velocity-force controller gain and the curvature of the object. Despite the difference, the object still be able to achieve continuous large-scale motion, as shown in Fig. 6 and video Fan et al. (2016).

Figure 7 shows the pose errors of the object during the lifting and rotating process by the proposed gaits planner. The finger gaits planning starts around 2 seconds, due to the decreasing hand manipulability. The pose errors do not attenuate to zero since the changing of contacts in finger gaits planning can cause disturbance to the object. The maximum position error during the gait changing process is 0.005 m (in gravity direction), and the maximum orientation error is less than 0.02 rad (1.15°).

Figure 8 shows that the $\dot{Q}$ in (8) is positive during a typical finger gaits planning period, which means that the proposed velocity-level planner is able to improve the grasp quality, i.e. Q in (5) is continuously increased in the finger gaits planning.

Figure 9 shows the performance of manipulation controller under object dynamics uncertainty and external disturbances. The inertia of the object is not necessary, since the inertia term in (12) is cancelled by impedance control (11).

(a) Position error during gaits planning



(b) Orientation error during gaits planning

Fig. 7. The position and orientation error of the object during lifting and rotation task using the proposed velocity-level gaits planner.

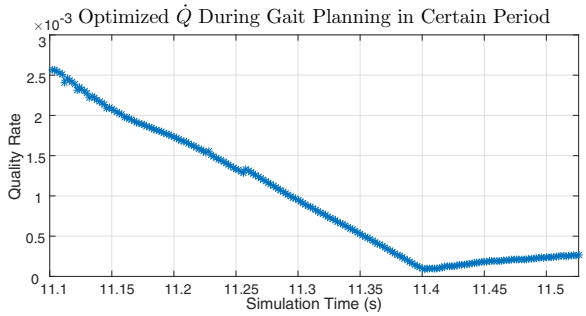


Fig. 8. Optimized quality rate by our proposed linear programming (8), in a typical finger relocation process. The quality rate is positive during the execution of (8), which means the grasp quality is improving.

Besides, the object is subject to 25% of mass uncertainty and 2N external disturbance. The manipulation controller is capable of driving the object to the desired pose.

7. CONCLUSION

This paper has proposed a dual-stage optimization based controller, which includes a velocity-level finger gaits planner and a manipulation controller, to achieve real-time finger gaiting and manipulation. In the finger gaits planner, we searched an optimal velocity to improve the grasp quality and hand manipulability, rather than directly finding optimal contact points by nonlinear programming methods. The proposed planner is computationally efficient and can be solved in real-time. Besides, the planner does not rely on the exact shape of the objects, nor does it require expensive 3D/6D tactile sensors. The presented

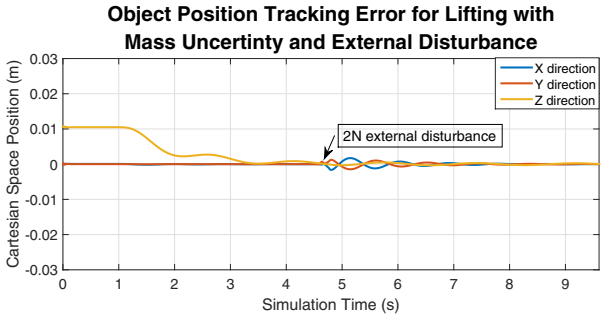


Fig. 9. Illustration of modified impedance controller. The manipulated object is subject to 25% mass uncertainty and 2N external disturbance

.

manipulation controller can handle a certain amount of uncertainties such as imprecise dynamic parameters and external disturbances. Simulations showed that our method can achieve real-time finger gaiting, and realize large-scale object motions that are infeasible without the proposed gaits planner.

Currently, our method is limited to objects with smooth surfaces. In the future, we plan to extend our method to objects without smooth surfaces. Also, we would like to incorporate contact point uncertainties caused by the noise of 1D tactile sensors, remove the requirement of velocity measurements, and perform experiments on a real robotic hand.

ACKNOWLEDGEMENTS

This project was supported by FANUC Corporation. The authors would like to thank Prof. Shmuel S. Oren for his constructive comments, and Prof. Emanuel Todorov for his help on Mujoco, and Hsien-Chung Lin and Yu Zhao for their review and discussion.

REFERENCES

Andrews, S. and Kry, P.G. (2013). Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics*, 37(7), 830–839.

Bicchi, A. (2000). Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. *IEEE Transactions on robotics and automation*, 16(6), 652–662.

Ciocarlie, M.T. and Allen, P.K. (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7), 851–867.

Erez, T., Kolev, S., and Todorov, E. (2014). Receding-horizon online optimization for dexterous object manipulation. *preprint available onliine*.

Fan, Y., Gao, W., Chen, W., and Tomizuka, M. (2016). Real-time finger gaits planning for dexterous manipulation. URL `https://youtu.be/X0yrPLv5rXk`.

Furukawa, N., Namiki, A., Taku, S., and Ishikawa, M. (2006). Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 181–187. IEEE.

Horn, B.K. and Schunck, B.G. (1981). Determining optical flow. *Artificial intelligence*, 17(1-3), 185–203.

Khalil, F.F. and Payeur, P. (2010). *Dexterous robotic manipulation of deformable objects with multi-sensory feedback-a review*. INTECH Open Access Publisher.

Kim, B.H., Oh, S.R., Yi, B.J., and Suh, I.H. (2001). Optimal grasping based on non-dimensionalized performance indices. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 2, 949–956. IEEE.

Kumar, V., Todorov, E., and Levine, S. (2016). Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 378–383. IEEE.

Li, M., Yin, H., Tahara, K., and Billard, A. (2014). Learning object-level impedance control for robust grasping and dexterous manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 6784–6791. IEEE.

Liegeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE transactions on systems, man, and cybernetics*, 7(12), 868–871.

Lim, Y.C., Lee, M., Lee, C.H., Kwon, S., and Lee, J.h. (2010). Improvement of stereo vision-based position and velocity estimation and tracking using a stripe-based disparity estimation and inverse perspective map-based extended kalman filter. *Optics and Lasers in Engineering*, 48(9), 859–868.

Liu, C.K. (2009). Dextrous manipulation from a grasping pose. In *ACM Transactions on Graphics (TOG)*, volume 28, 59. ACM.

Mordatch, I., Popović, Z., and Todorov, E. (2012). Contact-invariant optimization for hand manipulation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 137–144. Eurographics Association.

Murray, R.M., Li, Z., Sastry, S.S., and Sastry, S.S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

Platt, R., Fagg, A.H., and Grupen, R.A. (2004). Manipulation gaits: Sequences of grasp control tasks. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, 801–806. IEEE.

Platt Jr, R.J. (2006). *Learning and generalizing control-based grasping and manipulation skills*. Ph.D. thesis, Citeseer.

Roa, M.A. and Suárez, R. (2015). Grasp quality measures: review and performance. *Autonomous robots*, 38(1), 65–88.

Rosen, J. (1961). The gradient projection method for nonlinear programming. part ii. nonlinear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 9(4), 514–532. URL `http://epubs.siam.org/doi/pdf/10.1137/0109044`.

Supuk, T., Kodek, T., and Bajd, T. (2005). Estimation of hand preshaping during human grasping. *Medical engineering & physics*, 27(9), 790–797.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Vinayavekhin, P., Kudoh, S., and Ikeuchi, K. (2011). Towards an automatic robot regrasping movement based on human demonstration using tangle topology. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 3332–3339. IEEE.

Wimböck, T., Ott, C., Albu-Schäffer, A., and Hirzinger, G. (2012). Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 31(1), 3–23.

Xu, J., Koo, T.K.J., and Li, Z. (2010). Sampling-based finger gaits planning for multifingered robotic hand. *Autonomous Robots*, 28(4), 385–402.