Real-Time Robust Finger Gaits Planning under Object Shape and Dynamics Uncertainties

Yongxiang Fan, Te Tang, Hsien-Chung Lin, Yu Zhao, Masayoshi Tomizuka

Abstract—Dexterous manipulation has broad applications in assembly lines, warehouses and agriculture. To perform large-scale manipulation tasks for various objects, a multi-fingered robotic hand sometimes has to sequentially adjust its grasping gestures, i.e., the finger gaits, to address the workspace limits and guarantee the object stability. However, realizing finger gaits planning in dexterous manipulation is challenging due to the complicated grasp quality metrics, uncertainties on object shapes and dynamics (mass and moment of inertia), and unexpected slippage under uncertain contact dynamics. In this paper, a dual-stage optimization based planner is proposed to handle these challenges. In the first stage, a velocity-level finger gaits planner is introduced by combining object grasp quality with hand manipulability. The proposed finger gaits planner is computationally efficient and realizes finger gaiting without 3D model of the object. In the second stage, a robust manipulation controller using robust control and force optimization is proposed to address object dynamics uncertainties and external disturbances. The dual-stage planner is able to guarantee stability under unexpected slippage caused by uncertain contact dynamics. Moreover, it does not require velocity measurement or expensive 3D/6D tactile sensors. The proposed dual-stage optimization based planner is verified by simulations on Mujoco. The simulation video is available at [1].

I. INTRODUCTION

Dexterous manipulation is essential for manipulators to execute complicated tasks. To perform large-scale complex manipulations, the 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 challenging. First, the high-level optimization of finger gaiting under complicated grasp quality metrics is computationally expensive [2]. The optimization searches optimal contact points on a nonlinear object surface by maximizing object stability and hand manipulability metrics [3]. These two metrics are represented in different spaces, and associated by a high degree-of-freedom (DOF) nonlinear forward kinematics. The optimization becomes more challenging when the object 3D surface model is not available. Second, the low-level manipulation controller should be robust to dynamical parameter uncertainties such as mass and moment of inertia (MoI) uncertainties caused by object variations and sensing limitations. Third, the object is not directly controlled by actuators. Alternatively, energy is transferred from the fingertips to the object through contacts, which are complex to model because of various surface properties. Therefore, the dual-stage planner should be able to address the uncertain contact dynamics and the unexpected slippage arising from them.

As a result, problems related to finger gaits planning and robust manipulation have received significant attention. A task-specific finger gaiting policy was trained by covariance matrix adaptation method [4]. However, the learned policies cannot be adapted to different objects/tasks. A sampling-based method was used to plan finger gaits in [5]. Contact-invariant optimization method is used to compute the states of the hand and the object in [6]. These approaches are not computationally efficient for real-time finger gaits planning. By designing a global re-grasping planner and searching local optimal contact points, the unknown surface of object was explored [7]. However, predefined finger gaits are used in this approach, and the exploration of local optima does not incorporate necessary constraints, such as joint velocity and acceleration limitations. As a result, the approach tends to be slow in re-grasping and manipulation, and the predefined finger gaits might be inapplicable to other objects and robotic hands.

In terms of addressing dynamics uncertainties of nonlinear systems, a disturbance observer (DOB) was proposed for tracking control [8]. The nonlinearities and parameter uncertainties are lumped into a disturbance term. Moreover, it assumes full state feedback, while in dexterous hand, the velocity feedback is difficult due to the size constraints, backlash error and cost issue. To employ the dynamics of the system, synthesis with descriptor form has been broadly applied in different applications [9], [10], [11]. The synthesis in these applications work on linearized systems without considering the stability under the influence of nonlinearities. To consider parameter variations caused by nonlinearities, a linear parameter-varying (LPV) control with smooth scheduling was applied in [12], with an assumption that the nonlinearities can be approximated through linear varying parameters. This approximation, however, cannot work well for general nonlinear systems.

In this paper, a dual-stage optimization based planner is developed for real-time robust finger gaits planning under the object shape and dynamics uncertainties. The dual-stage optimization based planner consists of a finger gaits planner and a robust manipulation controller. To achieve real-time computation, the finger gaits planner is formulated in the velocity level, instead of the position level; formulation in the position level results in a complicated nonlinear constrained optimization problem. At each time step, the optimal joint
velocities are computed to improve the hand manipulability as well as the object grasp quality, and the computed joint velocities are fed into motors by a velocity-force controller. The proposed robust manipulation controller is formulated as a robust control and a contact force optimization. Feedback linearization and \( \mu \)-synthesis are combined for the robust controller design to deal with nonlinearities, dynamics uncertainties and external disturbances.

The contributions of this paper include: 1) The velocity-level finger gaits planner is cast into a linear programming (LP), 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. 2) The dual-stage optimization based planner is robust to object shape and dynamics uncertainties. The velocity-level finger gaits planner utilizes velocity-force control to detect the object surface and searches motions in tangent space. Moreover, the robust manipulation controller can handle at least 40% mass and 50% MoI uncertainties of the object. 3) The proposed dual-stage optimization based planner reduces cost of the dexterous hand. To be more specific, it does not require precise 3D reconstruction for exact object surface model, high resolution encoder or accelerometer for velocity measurement, or expensive 3D/6D tactile sensors for friction feedback. The efficacy of the proposed controller is verified by simulations. The video demo is available at [1].

The remainder of this paper is organized as follows. Section II shows the overall dual-stage optimization based planner framework. Section III introduces the finger gaits planner. The design of robust manipulation controller is presented in Section IV. Section V shows simulation results on a robotic hand with four fingers and twelve DOFs. Section VI concludes the paper.

II. DUAL-STAGE OPTIMIZATION BASED PLANNER FRAMEWORK

Figure 1 shows the proposed dual-stage optimization based planner framework. First, grasp quality analysis is conducted by combining hand manipulability and object grasp quality, and the free finger index \( k \) is chosen to change gait once the overall quality drops below a predefined threshold. A velocity-level finger gaits planner [13] is evoked by this event, and the planner generates the torque command \( \tau_{\text{des,k}} \) to drive the selected finger towards the better quality region. The remaining fingers are controlled by a robust manipulation controller [14] to generate desired torque \( \{ \tau_{\text{des,j}} \}_{j \neq k} \) to manipulate the object stably and track the reference motion of the object, as shown in Fig. 1. If the overall quality is above the threshold, all fingers will be controlled by the robust manipulation controller.

A joint level torque tracking controller is used to track the desired torque command \( \{ \tau_{\text{des,j}} \}_{j=1}^{N_{\text{finger}}} \), where \( N_{\text{finger}} \) denotes the number of fingers. The torque tracking controller uses a PID scheme and runs at a higher frequency, in comparison with the finger gaits planner and the robust manipulation controller (500 Hz).

III. FINGER GAITS PLANNING

A. Grasp Quality Analysis

Grasp quality has been well explored in [3], [15]. It is desired that both hand manipulability and object grasp quality are considered during the finger gaits planning. The hand manipulability describes the ability for a hand to manipulate the object to realize arbitrary object motions. The object grasp quality describes the capacity to resist external disturbances given a group of contact points on the object. This paper adopts a quality metric in [16] to represent the hand manipulability \( Q_h \): \( Q_h = 0.5 \sum_{i=1}^{N_{\text{inf}}} \sum_{j=1}^{N_{\text{join}}} \left( q_j^i - q_j^o \right)^2 / (q_{\text{max},i} - q_{\text{min},i})^2 \), where \( q_j^i \) is the \( i \)-th joint angle of the \( j \)-th finger, \( q_{\text{max},i} \) and \( q_{\text{min},i} \) are the limits of \( q_j^i \) and \( q_j^o = (q_{\text{max},i} + q_{\text{min},i})/2 \) is the middle position of the corresponding joint, \( N_{\text{joint}} \) is the number of joints per finger. The object grasp quality \( Q_o \) can be represented as: \( Q_o = 2 \text{Area}(\{ p_j \}_{j \in I_c}, \text{proj}(p_k)) \) [17], where \( I_c \) is the set of indices of all fingertips that are in contact with the object. \( p_j \) is the contact position in Cartesian space for the \( j \)-th fingertip. \text{proj}(p_k) denotes the projection operation of \( p_k \) onto the plane specified by \( \{ p_j \}_{j \in I_c} \).

The overall quality \( Q \) can be obtained by combining \( Q_o \) and \( Q_h \):

\[
Q = w_1 Q_o + w_2 Q_h
\]

where \( w_i > 0 \) is the weight for the corresponding term.

Once the overall grasp quality \( Q \) drops below a threshold, the finger 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 gaeting. This idea is adopted and the finger gaits are sequentially planned. Thus, the proposed 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 \( \delta_Q \) is a threshold. The free finger is selected based on the finger manipulability of itself and the grasp quality of the remaining fingers to the object. To be more specific, the finger manipulability for the \( k \)-th finger is \( -0.5 \sum_{i=1}^{N_{\text{join}}} \left( q_k^i - q_o^i \right)^2 / (q_{\text{max},i} - q_{\text{min},i})^2 \). The grasp quality of remaining fingers to the object is the area of convex hull spanned by the remaining fingertips. The candidate free finger for gaeting is the one with small finger manipulability and large remaining grasp quality. If there is already one free finger that are not in contact with the object, that finger will continue its gaeting.
B. Velocity-Level Finger Gaits Planning

The task of the finger gaits planner is to generate commands to change the contact location of the free finger, to achieve better object grasp quality and finger manipulability in real time. However, searching contact position by maximizing the quality \( q \) is challenging. First, the search of \( p_k \) should be conducted on the surface of the object, and the formulation of the object surface requires 3D reconstruction and surface modeling. Second, the searching of \( q_k \) should be constrained within the joint limits, and \( q_k \) is coupled with \( p_k \) by forward kinematics. Third, after finding the optimal contact point, a trajectory planning algorithm is required to generate a feasible trajectory. In our previous work [13], a velocity level finger gaits planner is proposed to overcome the aforementioned challenges.

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 of the fingertip of the free finger is calculated at each time step, and the finger is actuated by a velocity-force controller to achieve that velocity. Formally, instead of optimizing \( Q \) in (1), we optimize \( Q \) with joint velocity of the \( k \)-th finger \( q_k \) as the decision variable. The solution \( q_{\text{des},k} \) is used to control the robotic hand in each time step.

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

\[
\dot{Q} = w_1 \dot{Q}_o + w_2 \dot{Q}_h
\]

\[
\dot{Q}_o = \| p_{j_2} - p_{j_3} \|_2 n_{j_1}^T v_{p_k}
\]

\[
\dot{Q}_h = \sum_{i=1}^{N_{\text{con}}} \left( \frac{q_{i,k} - q_{i,k}^\text{th}}{(q_{i,k}^\text{max,k} - q_{i,k}^\text{min,k})^2} \dot{q}_i \right)
\]

We assume that \( \{ p_j \}_{j \in I} = \{ p_{j_1}, p_{j_2}, p_{j_3} \} \), \( n_{j_1} \) is a normal vector of line segment \( p_{j_1}p_{j_3} \) in the plane specified by \( \{ p_j \}_{j \in I} \), and \( v_{p_k} \) is the velocity of contact point \( p_k \). \( \dot{q}_k \) is joint velocity of the \( k \)-th joint for the \( k \)-th finger. In this optimization, the states \( v_{p_k} \) and \( \dot{q}_k \) in \( \dot{Q}_o \) and \( \dot{Q}_h \) are coupled linearly by \( v_{p_k} = J(q_k) \dot{q}_k \), where \( J(q_k) \) is the Jacobian matrix of the \( k \)-th finger. By plugging in the coupled term, \( \dot{Q} \) becomes:

\[
\dot{Q} = w_1 \| p_{j_2} - p_{j_3} \|_2 n_{j_1}^T J q_k + w_2 \sum_{i=1}^{N_{\text{con}}} c_k(q_{i,k} - q_{i,k}^\text{th})^2 \dot{q}_i
\]

\[
c_k = \begin{cases} 
\ln(\frac{q_{i,k}^\text{th} - q_{i,k}}{q_{i,k}^\text{th} - q_{i,k}^\text{min,k}}) + 1, & q_{i,k} - q_{i,k}^\text{th} < -q_{i,k}^\text{thres} \\
1, & |q_{i,k} - q_{i,k}^\text{th}| \leq q_{i,k}^\text{thres} \\
\ln(\frac{q_{i,k}^\text{th} - q_{i,k}}{q_{i,k}^\text{min,k} - q_{i,k}^\text{th}}) + 1, & q_{i,k}^\text{th} - q_{i,k} > q_{i,k}^\text{thres} 
\end{cases}
\]

where \( q_{i,k}^\text{thres} \) is a threshold where the weighting should start to increase.

With above analysis, a new optimization can be formulated to approximate the original contact optimization:

\[
\dot{q}_{\text{des},k} = \arg \max_{\dot{q}_k} \dot{Q}
\]

s.t. \( \dot{q}_{\text{min},k} \leq \dot{q}_k \leq \dot{q}_{\text{max},k} \)

\[
q_{\text{min},k}^T J(q_k) \dot{q}_k = 0
\]

\[
\| \ddot{q}_k - \dot{q}_{\text{des},\text{prev}} \|_\infty \leq \sigma
\]

where constraint (5) means that the desired joint velocity \( \dot{q}_k \) should be bounded in \( [\dot{q}_{\text{min},k}, \dot{q}_{\text{max},k}] \). Constraint (6) indicates that \( p_k \) must move perpendicular to current surface normal \( n_{p_k} \). Constraint (7) limits the joint acceleration by \( \sigma/T_s \), where \( T_s \) denotes the sampling time of the system. \( \dot{q}_{\text{des},\text{prev}} \) is the desired joint velocity in previous time step. The optimization (4) is a linear programming, which can be solved in real-time.

After obtaining the desired joint velocity \( \dot{q}_{\text{des},k} \) by solving (4), a velocity-force controller is implemented to calculate the desired torque for the \( k \)-th finger:

\[
\tau_{\text{des},k} = K_v \dot{q}_{\text{des},k} + K_f J(q_k)^T (f_{\text{des}}^n - f_{\text{act}}^n)
\]

where \( \tau_{\text{des},k} \) and \( f_{\text{des}}^n \) are the desired torque and desired contact force in the normal direction. \( f_{\text{act}}^n \) is set to be a constant small force during finger gaiting. \( f_{\text{act}}^n \) is the actual contact force in normal direction and can be measured by 1D tactile sensor. The force component \( K_f J(q_k)^T (f_{\text{des}}^n - f_{\text{act}}^n) \) in (5) attempts to maintain the contact between the fingertip and the surface, which makes the normal vector \( n_{p_k} \) measured from the tactile sensor updated.

The velocity-level gaits planner that composed of (4) and (5) has several advantages. First, the proposed planner is computationally efficient. The optimization (4) is an LP that can be solved in each time step. Second, the 3D object model is not required. Instead, 1D tactile sensor is employed 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 (i.e. \( \dot{Q} < \delta \), where \( \delta \) is a small positive number), or when the grasp quality is above the threshold (i.e. \( Q > \delta Q \)).

IV. ROBUST MANIPULATION CONTROLLER

In our previous work [13], a modified impedance control is used to compute the desired Cartesian force on the object. This method cannot guarantee robust stability and robust performance under dynamics uncertainties and disturbances. Moreover, it has to use object velocity measurement as feedback. In this paper, our previous work on robust manipulation controller [14] is combined with finger gaits planner to realize large-scale object motion.

The design goal of the robust manipulation controller is to: 1) track the desired motion of the object, 2) be robust to object dynamics uncertainties and external disturbances, and 3) satisfy friction cone constraints. The robust manipulation controller consists of a robust controller and a manipulation
controller. The robust controller takes the pose tracking errors of the object as input, and generates desired Cartesian force for the object. The desired Cartesian force is converted into the torque command to the hand by the manipulation controller.

A. Robust Controller Design

The goal of the robust controller is to obtain the desired Cartesian force \( \mathbf{F}_{\text{des}} \) of the object for motion tracking with guaranteed stability and performance robustness. The generalized plant \( P_{\text{general}} \) that the robust controller will work on is shown in Fig. 2. \( G_{NL} \) denotes the nominal model of augmented nonlinear plant. \( \Delta = \text{diag}(\delta_m I_{3 \times 3}, \delta_x, \delta_x, \delta_x) \) with \( \| \Delta \|_\infty \leq 1 \) represents the structure of uncertainties after decomposing into descriptor form. Refer [14] for more details. \( G_{NL} \) and \( \Delta \) define an upper linear fractional transformation (LFT, see [18]) w.r.t. \( \Delta \) (denoted as \( F_\delta(G_{NL}, \Delta) \)) to represent the nonlinear uncertain dynamics, as shown in red dash box. The feedback linearization described by \( \alpha(x) + \beta(x)u \) is connected with the nonlinear uncertain plant to linearize the nominal model and compensate nominal gravity, as shown in blue dash-dot box.

The inputs to the generalized plant \( P_{\text{general}} \) are \( \{r, \mathbf{u}_{\text{dis}}, n, u\} \), where \( r, \mathbf{u}_{\text{dis}}, n \) and \( u \) denote the reference pose of the object, input disturbance, noise and the control input from robust controller, respectively. The outputs of the plant are \( \{w_{\text{perf}}, w_n, e\} \), with \( w_{\text{perf}}, w_n \) and \( e \) denoting the tracking performance, the action magnitude and the pose tracking errors. \( w_{\text{perf}} \) is to suppress tracking errors at different frequencies. \( w_n \) is to regulate the control input. \( \mathbf{w}_{\text{dis}} \) is to shape the input disturbance. \( \mathbf{w}_n \) is to shape the measurement noise.

The connection between the generalized plant \( P_{\text{general}} \) and the robust controller \( K \) is described by Fig. 3. \( P_{\text{general}} \) and \( K \) define a lower LFT w.r.t. \( K \) as \( F_\delta(\mathbf{G}_{\text{general}}, \mathbf{K}) \), to denote the closed-loop system. The closed-loop system concatenates the inputs \( \{r, \mathbf{u}_{\text{dis}}, n\} \) as \( \mathbf{d} \) and the outputs \( \{w_{\text{perf}}, w_n\} \) as \( \mathbf{e} \). The objective of the robust controller design is to synthesize \( K \) to keep \( \mathbf{e} \) small for all reasonable \( \mathbf{d} \). The small is in the sense of infinity norm, i.e.

\[
K = \arg\min_{\mathbf{K}} \| F_L(\mathbf{G}_{\text{general}}, \mathbf{K}) \|_\infty
\]

with:

\[
\mathbf{e} = F_L(\mathbf{G}_{\text{general}}, \mathbf{K})\mathbf{d}
\]

\[
\| F_L \|_\infty := \max_{\omega \in \mathbb{R}} \sigma(F_L(j\omega))
\]

The D-K iteration is applied to solve (6):

\[
\min \inf_K \| F_L(\mathbf{G}_{\text{general}}, \mathbf{K})D^{-1} \|_\infty < 1 \quad (7)
\]

Readers can refer [18] for more details.

The designed robust controller \( K \) will be used to calculate \( u \) based on the pose errors \( e \). Then the output of the controller is combined with feedback linearization to obtain the desired Cartesian space force \( \mathbf{F}_{\text{des}} \) for the object.

B. Manipulation Controller Design

Given the contacts between the fingertips and the object, the task of the manipulation controller is to generate desired torque commands \( \tau_{\text{des}, \text{f}} \) for non-breaking fingers to drive the object and achieve the desired force \( \mathbf{F}^* \) on the object. The manipulation controller consists of a force optimizer, which computes an optimized contact force vector \( \mathbf{f}^* \) on fingertips from the desired force \( \mathbf{F}_{\text{des}} \) on the object, and a joint-level torque controller, which generates an appropriate joint torque vector \( \tau_{\text{des}, \text{f}} \) to reproduce \( \mathbf{f}_{\text{des}} \).

The force optimization is formulated into a quadratic programming (QP):

\[
\begin{align}
\min_{\mathbf{f},\mathbf{\Psi}} & \quad \alpha_1 \| \mathbf{f} \|_2^2 + \alpha_2 \| \mathbf{f} - \mathbf{f}_{\text{prev}} \|_2^2 + \alpha_3 \| \mathbf{\Psi} \|_2^2 \\
\text{s.t.} & \quad \mathbf{\Psi} = \mathbf{F}_{\text{des}} - \mathbf{G}(\mathbf{q}, \mathbf{x}_o)\mathbf{f} \\
& \quad \mathbf{f} = \mathbf{B}\mathbf{\beta} \\
& \quad \mathbf{\beta} \geq 0 \\
& \quad \tau_{\text{min}} \leq \mathbf{J}_h^T(q, \mathbf{x}_o)f \leq \tau_{\text{max}}
\end{align}
\]

where \( \mathbf{f} = [\mathbf{f}_1^T, \ldots, \mathbf{f}_{N_{\text{finger}}}^T]^T \) is a concatenated contact force vector in contact frame. \( \mathbf{f}_{\text{prev}} \) is the contact force of the previous time step. \( \mathbf{G}(\mathbf{q}, \mathbf{x}_o) \in \mathbb{R}^{6 \times N_{\text{finger}}N_{\text{joint}}} \) is called grasp map, where \( \mathbf{q} \) is a concatenated joint angle vector of \( \{\mathbf{q}_j\}_{j \in L} \), and \( \mathbf{x}_o \) is the pose of the object. The entries of \( G \) for the free finger are set to zero. \( \mathbf{J}_h^T(q, \mathbf{x}_o) \) is transpose of hand Jacobian and maps the contact force on fingertips to the joint torque vector \([3]. \mathbf{B} = \text{diag}\{\mathbf{B}_1, \ldots, \mathbf{B}_{N_{\text{joint}}}\} \) and \( \mathbf{B}_i \) is a conservative pyramid approximation of friction cone [19]. \( \beta \geq 0 \) is the non-negative linear coefficients of columns of \( \mathbf{B} \). The weights \( \alpha_1, \alpha_2, \alpha_3 \) are used to balance different cost terms.

A slack variable \( \mathbf{\Psi} \) is introduced to relax the hard constraint \( \mathbf{F}_{\text{des}} = \mathbf{G}\mathbf{f} \), since \( \mathbf{F}_{\text{des}} = \mathbf{G}\mathbf{f} \) might result in an infeasible solution, and the location measurements of contact points might be noisy. The constraints \( (8c) \) and \( (8d) \) together ensure that the contact force remains within positive \( \text{colspan}(\mathbf{B}) \) (i.e. friction cone). Constraint \( (8e) \) guarantees that the contact force \( f \) is realizable.
The joint level torque control takes the optimized contact force \( f^* \) from the force optimization as input, and yields the torque \( \tau = J_h^T(q, x_o)f^* \), where \( \tau = [\tau_{\text{des},1}, \ldots, \tau_{\text{des},N_{\text{imp}}}]^T \).

The desired torques for non-breaking fingers are: \( \tau_{\text{des},j} = (\tau_{\text{des},k} = 0 \text{ from this optimization due to the structure of } G \text{ and the minimization of } \| \tilde{c} \|) \).

V. SIMULATIONS

In this section, simulation results are presented to verify the effectiveness of the proposed dual-stage optimization based planner. The simulation video is available at [1].

A. Simulation Setup

The controller is implemented in the Mujoco physical engine [20]. The simulation time step is set to 2 ms. Our platform is a desktop with 4.0 GHz Intel Quad Core 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 \([-10^\circ, 135^\circ], [-45^\circ, 45^\circ] \) and \([-10^\circ, 170^\circ] \), respectively. The hand is equipped with joint encoders, motor torque sensors, one-dimensional distributive tactile sensors. The manipulated objects are approximately 0.5 kg. The 3D mesh models of objects are unknown to the controller. The manipulated object is subject to 20% mass and 50% MoI uncertainties. The object pose is obtained from the simulator. In future real world experiments, methods in [21] will be adopted to obtain the pose of the object.

B. Parameter Lists

1) Velocity-Level Finger Gaits Planner: The parameter values for the linear programming \( \begin{bmatrix} w_1 \end{bmatrix} = [0.99, 0.99, 0.01, 0.01, 0.05, 0.05, 0.05] \). The parameter values for velocity-force controller are: \( \begin{bmatrix} K_v \end{bmatrix} = \begin{bmatrix} 0.5, 0.5, 0.5 \end{bmatrix} \).

2) Robust Manipulation Controller: The weighting function is specified by DC gain \( G_I \), crossover frequency \( \omega_c \), high-frequency gain \( G_h \), and order \( n \). Readers can refer [14] for more details. The parameters of the weighting functions are shown in Table I.

| Weightings | \( \omega_c \) | \( G_I \) | \( G_h \) | \( \{a_1, a_2, a_3\} \) | \( n \) |
|-----------|---------------|-----------|-----------|----------------|-------|
| \( W_{\text{perf}} \) | \( 2\pi \) | 1100 | 0.9 | \( (1,1,2) \) | 2 |
| \( W_u \) | N/A | 0.0001 | 0.0001 | \( (1,1,0.5) \) | 1 |
| \( W_{\text{des}} \) | 200\( \pi \) | 80 | 0.1 | (1,1,10) | 2 |
| \( W_n \) | 20\( \pi \) | 0.1 | 10 | (1,1,1) | 1 |

For manipulation controller, \( \alpha_1 = 0.01, \alpha_2 = 0.01 \) and \( \alpha_3 = 1000 \). \( \tau_{\text{min}} = -0.5 \text{ Nm and } \tau_{\text{max}} = 0.5 \text{ Nm.} \)

C. Simulation Results

1) Performance of the Robust Manipulation Controller: A general 3D manipulation task is executed to verify the robust manipulation controller. The desired object motion is to move (4, 4, 10) mm along X, Y, Z-axis, and rotate 0.5 rad around Z-axis. The finger gaits planner is not activated. The manipulated object is subject to 20% mass and 50% MoI uncertainties. A robust controller that can resist 40% mass and 50% MoI uncertainties is implemented.

Figure 5 shows the snapshots of the tracking process from 0 - 1.2 seconds. Both the front view and top view are presented at each time. By employing the structures of uncertainties, the proposed robust manipulation controller achieves fast object motion tracking and guarantees the robust stability under uncertainties. It can be seen that the gravity mismatch caused by mass uncertainty can be efficiently compensated within 0.5 seconds. Moreover, the feedback linearization reduces the nonlinearities of the system, so that the robust controller is able to work well on this highly nonlinear hand. It is worth mentioning that the Coriolis and Centrifugal term is treated as disturbance. Therefore, the velocity measurement is not required.

The maximum settling time of all channels are 1.1064 seconds for 20% mass and 50% MoI uncertainties, and are 2.2138 seconds for 40% mass and 50% MoI uncertainties, as shown in Fig. 6. The non-smoothness of the error curves at the beginning of the simulation are caused by unexpected contact between the palm and the object.

2) Performance of the Dual-Stage Planner: A lifting and rotating task is employed to verify the efficacy of the proposed finger gaits planner and the robust manipulation controller. The desired object motion is to move along Z-axis by 11 mm, rotate continuously around Z-axis with 0.2 rad/s, and rotate sinusoidally around Y-axis with 0.4 rad/s.

\( ^1\)5% threshold is used for all settling time calculations.
The object has 20% mass and 50% MoI uncertainties. Figure 7 illustrates the performance without/with the proposed velocity-level finger gaits planner. Snapshots are from left to right. The white and red arrows indicate the initial and current rotational poses of the object. The object cannot be rotated over 90° without the proposed gaits planner because of the decreasing finger manipulability (Fig. 7 Top).

In comparison, the manipulation result with the proposed finger gaits planner is shown in Fig. 7 (Bottom). The object can be lifted and continuously rotated with the desired velocity. The finger gaits planner guarantees the grasp quality (4) is kept above a threshold, and the robust manipulation controller guarantees tracking performance and robust stability. The average computation time for solving the dual-stage optimization based planner is less than one millisecond for each time step.

The tracking errors for this lifting and rotating task are shown in Fig. 8. It can be seen that from 0 ~ 2 seconds the robust manipulation controller can drive the object to track a moving target pose without any delay. The finger gaits planning (4) starts around 2 seconds, due to the decreasing quality. The pose errors during the finger gaits planning can cause disturbance to the object. The maximum position error is 0.0037 m, and the maximum orientation error is 0.027 rad (1.55°).

The response of the dual-stage optimization based planner to external disturbances are shown in Fig. 9. The force disturbance and the corresponding position errors are shown in Fig. 9 (a)(b), and the torque disturbance and the corresponding orientation errors are shown in Fig. 9 (c)(d). The pose errors introduced by the disturbance can be attenuated within 0.5 seconds (e.g. force disturbance in 1.9s ~ 3.3s). Moreover, the system can resist at least 5 N force and 0.03 Nm torque in different directions without going unstable.

VI. CONCLUSION

This paper has proposed a dual-stage optimization based planner, which includes a velocity-level finger gaits planner and a robust manipulation controller, to achieve real-time finger gaits planning under object shape and dynamics uncertainties. The finger gaits planner searched an optimal velocity to adaptively adjust the friction coefficient based on 3D/6D force feedback. This paper employs velocity-level finger gaits planner to relocate the slipping finger if the quality drops below a threshold, instead of developing complex algorithms to prevent slippage. The robustness of the dual-stage optimization based planner to friction coefficient is shown in video [1].
feedback. The presented robust manipulation controller can handle at least 40% mass and 50% MoI uncertainties of the object. Simulations showed that the proposed method can achieve real-time finger gaiting, and realize large-scale object motions that are infeasible without the proposed finger gaits planner.

Currently, the proposed method is limited to objects with smooth surfaces. In the future, the authors plan to extend the method to objects without smooth surfaces. Also, the authors would like to address the uncertainties on grasp points, test the proposed method on more complex objects such as non-convex ones, and perform experiments on a real world robotic hand.

ACKNOWLEDGMENT

The authors would like to thank Prof. Andy Packard for his advice on robust control, Prof. Shmuel S. Oren for his constructive comments, and Prof. Emanuel Todorov for his help on Mujoco. Thank Zining Wang for his help on dynamics.

REFERENCES

[1] Y. Fan, T. Tang, H.-C. Lin, Y. Zhao, and M. Tomizuka, (2017) Real-time robust finger gaits planning under object shape and dynamics uncertainties. Youtube. [Online]. Available: https://youtu.be/vsFtAyODs
[2] A. Bacchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” IEEE Transactions on robotics and automation, vol. 16, no. 6, pp. 652–662, 2000.
[3] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, A mathematical introduction to robotic manipulation. CRC press, 1994.
[4] S. Andrews and P. G. Kry, “Goal directed multi-finger manipulation: Control policies and analysis,” Computers & Graphics, vol. 37, no. 7, pp. 830–839, 2013.
[5] J. Xu, T.-K. J. Koo, and Z. Li, “Sampling-based finger gaits planning for multifingered robotic hand,” Autonomous Robots, vol. 28, no. 4, pp. 385–402, 2010.
[6] I. Mordatch, Z. Popović, and E. Todorov, “Contact-invariant optimization for hand manipulation,” in Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation. Eurographics Association, 2012, pp. 137–144.
[7] R. Platt, A. H. Fagg, and R. A. Gruppen, “Manipulation gaits: Sequences of grasp control tasks,” in Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on, vol. 1. IEEE, 2004, pp. 801–806.
[8] C.-S. Liu and H. Peng, “Disturbance observer based tracking control,” Journal of Dynamic Systems, Measurement, and Control, vol. 122, no. 2, pp. 332–335, 2000.
[9] A. Pilat and P. Whodarczyk, “The μ-synthesis and analysis of the robust controller for the active magnetic levitation system,” Automatyka/Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, vol. 15, no. 1, pp. 85–98, 2011.
[10] K. Nonami, N. Ide, and H. Ueyama, “Robust control of magnetic bearing systems using μ-synthesis with descriptor form.” JSME International Journal Series C, vol. 40, no. 4, pp. 688–693, 1997.
[11] T. Ohtani, Y. Hamada, T. Nagashio, T. Kida, S. Mitani, I. Yamaguchi, T. Kasai, and H. Igawa, “Robust attitude control using μ-synthesis for the large flexible satellite ets-viii,” The Journal of Space Technology and Science, vol. 25, no. 1, pp. 1,27–1,40, 2009.
[12] H. Koc, D. Knittel, M. de Mathelin, and G. Abba, “Modeling and robust control of winding systems for elastic webs,” IEEE Transactions on control systems technology, vol. 10, no. 2, pp. 197–208, 2002.
[13] Y. Fan, W. Gao, and M. Tomizuka, “Real-time finger gaits planning for dexterous manipulation,” The 20th World Congress of the International Federation of Automatic Control (IFAC), 2017 to appear.
[14] Y. Fan, L. Sun, M. Zheng, W. Gao, and M. Tomizuka, “Robust dexterous manipulation under object dynamics uncertainties,” Advanced Intelligent Mechatronics (AIM), 2017 IEEE International Conference on, 2017 to appear.
[15] M. A. Roa and R. Suárez, “Grasp quality measures: review and performance.” Autonomous robots, vol. 38, no. 1, pp. 65–88, 2015.
[16] A. Liegeois, “Automatic supervisory control of the configuration and behavior of multibody mechanisms,” IEEE transactions on systems, man, and cybernetics, vol. 7, no. 12, pp. 868–871, 1977.
[17] T. Supuk, T. Kodek, and T. Bajd, “Estimation of hand preshaping during human grasping,” Medical engineering & physics, vol. 27, no. 9, pp. 790–797, 2005.
[18] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith, “μ-analysis and synthesis toolbox,” MUSYN Inc. and The MathWorks, Natick MA, 1993.
[19] C. K. Liu, “Dexterous manipulation from a grasping pose,” in ACM Transactions on Graphics (TOG), vol. 28, no. 3. ACM, 2009, p. 59.
[20] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 5026–5033.
[21] Y. Fan, H.-C. Lin, Y. Zhao, C.-Y. Lin, T. Tang, M. Tomizuka, and W. Chen, “Object position and orientation tracking for manipulators considering nonnegligible sensor physics,” in Flexible Automation (ISFA), International Symposium on. IEEE, 2016, pp. 450–457.