Exploring the Application of Particle Filters to Grasping Acquisition with Visual and Tactile Occlusion

L. Zhang and J.C. Trinkle

Abstract—The G-SL(AM)$^2$ (or grasp-SLAM) problem is the analog of the well studied Simultaneous Localization and Mapping (SLAM) problem in the domain of grasping. The G-SL(AM)$^2$ problem is to localize the object relative to the hand while simultaneously updating the system model and manipulating the object. Stick-slip friction and intermittent contact are fundamental aspects of grasping that lead to nonsmoothness of the mathematical model. For this reason, a particle filter is designed and its effectiveness is explored through a simplified instance of the G-SL(AM)$^2$ problem. Experiments from a planar grasping testbed are used to test the ability of the particle filter to track the object and adjust friction model parameters with or without visual occlusion and with or without binary tactile sensor information. The results show that the particle filter outperforms a simple Kalman filter especially during periods of visual occlusion.

I. INTRODUCTION

Why is autonomous grasping and manipulation in unstructured environments still so hard for robots after 30+ years of research? For a robot to perform skilled grasping and manipulation, it has to have information that is hard to obtain with sufficient accuracy: a geometric model of the object, estimates of important physical properties (such as weight and friction), and the ability to track the pose of the object and contacts in real-time.

The best scenario for a robot is that it can get an accurate physical model from a database, has a vision system that can track the object, and tactile sensors that can aid tracking when the fingers occlude the visual tracking features. Even in this case, the localization errors of the perception system and positioning errors allowed by the control systems are not small enough to ignore during the process of grasp acquisition or subsequent manipulation. These errors can cause the hand to bump the object accidentally when reaching or curling the fingers, possibly causing the object to slide or tumble out of the grasp. The problems are magnified when the physical properties of the object are known only roughly.

This paper is motivated by the idea that we can dramatically advance the state of the art in autonomous grasping and manipulation by designing algorithms that can incrementally improve a physical model$^1$ of a grasping system (composed of hand, object, and environment), while accurately tracking the object during grasping and manipulation. This is analogous to the Simultaneous Localization and Mapping (SLAM) problem developed that allows a mobile robot to autonomously build a map of its environment, while simultaneously navigating and tracking its own position.

This paper represents the authors’ first small steps toward solving the G-SL(AM)$^2$ problem.

The G-SL(AM)$^2$ Problem:
The G-SL(AM)$^2$ problem is to autonomous robotic grasping, what the SLAM problem is to autonomous robotic mobility. The G stands for Grasping. SL(AM)$^2$ stands for: Simultaneous Localization, and Modeling, and Manipulation. The word “Modeling” implies that the robot will use its sensor systems (tactile, visual, and kinesthetic) to build and improve a model of the object. “Manipulation” implies that the robot will physically manipulate the object to help accomplish the modeling task. “Localization” implies that the robot will track the pose of the grasped object during manipulation. “Simultaneous” implies that localization, modeling, and manipulation will all occur together, in real time.

The particular scaled-down version of the G-SL(AM)$^2$ problem studied here, assumes an accurate geometric model of the hand, object, and environment. The main objective is to marry particle filtering methods [6] with the complementarity-based dynamic model [3] to estimate the pose of the object and four parameters of the friction model of the system.

A. Related Research

In [9], Yanbin Jia et al. investigated the problem of blindly determining the pose and motion of a planar object with known geometry from pushing the object. Their method used tactile data and geometric models during the pushing process to infer the location of the contact points on the object and the object pose. In [7], Haidacher et al. presented an approach to locally estimate the pose of an object during grasp acquisition in 3D when visual servoing is obstructed by the gripper. The approach first refined the object description offline by characteristic relations between planar facets of the geometric model and stored those values in a description database. After receiving tactile measurements from the robotic hand, this database was searched for possible matching facet combinations to determine the position of the object relative to the hand. Both of these research efforts relied on tactile sensors to infer object pose when object geometry was given and when there was no visual information.

$^1$The physical model includes the shapes of the bodies and other quantities such as friction coefficients and the mass of the object.

This work was supported by NSF.

Zhang is a PhD candidate in the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA zhangl15@rpi.edu

Trinkle is a professor in the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA trink@cs.rpi.edu
Dieter Fox et al. proposed Monte Carlo Localization (MCL) algorithm in [4], [12], which solved the global localization and kidnapped robot problem robustly and efficiently.

The rest of the paper is organized as follows. Section II introduces the experiment setup as well as the planar dynamic model, the accuracy and robustness of which has been shown in previous work [1], [3]. Section III brings up some specific problems that arise when applying particle filter to $G-SL (AM)^2$ problem and describes a proposed solution. In Section IV, we show the effectiveness of proposed algorithm and its capability of fusing multiple sensory information by comparing filtering results on sets of experiments. Section V summarizes the work and points out future directions.

II. EXPERIMENT AND DYNAMIC MODEL

A. Experimental Setup and Assumptions

Our experimental environment is a planar grasping testbed as introduced in [14]. As shown in Fig. 1, one linear pusher (or thumb) and three fixels (or fixed fingers) are mounted on an aluminum plate with a positioning hole array to create simple a one-degree-of-freedom “hand.” The pusher pushes the object toward the fixels to achieve a grasp. As this process proceeds, frames are captured by an overhead camera at 30Hz and are post-processed to extract the configuration trajectory of the system. This is the observed data of the testbed. After calibration, the camera provides positions and orientations with a resolution of about 1mm in displacement and 1° in rotation.

Since gravity acts perpendicular to the support plane and the pusher moves parallel to the plane, we can reasonably assume that the object will maintain in face-face contact with the support plane. In addition, all bodies are assumed rigid and all contacts are assumed to be discrete points where Coulomb’s Friction Law applies. The face-face contact between the object and the support plane is represented by three points of support rigidly fixed to the object. The points form an equilateral triangle whose center is directly below the object’s center of mass. This model greatly simplifies a highly complex friction process that actually depends on the object’s pose with a backward Euler rule. As such our problem satisfies the conditions of the 2.5D dynamic model available in our dVC2d simulator as described in [3]. The “half” degree of freedom is included, because the friction forces acting between the object and the support surface are determined by the gravity force, which acts perpendicular to the plane of motion, so out-of-plane effects cannot be entirely ignored as one normally does in 2D problems.

Below, the mixed NCP implementing the 2.5D, discrete-time, dynamic model is presented (equations (1) and (2)) (see [1] for a complete derivation). Let $\Delta t$ be the (constant) simulation time-step and let $\ell = \{0, 1, \ldots, N\}$ denote the index of the current time step. In the NCP formulation, the unknowns are the contact impulses $p^{\ell+1}$, the object velocity $v^{\ell+1}$, and contact slip indicators $\sigma^{\ell+1}$, where the superscript is not an exponent, but rather indicates time $t = \Delta t(\ell + 1)$. All the other terms in the equations are constructed from information available at the current time: $p^\ell$, $v^\ell$, $\sigma^\ell$, input forcing functions, collision detection algorithms, and physical properties of the system. To simulate the grasping process, this NCP is formulated and solved at each discrete time point, $\ell = 1, \ldots, N$. The velocity is used to update the object’s pose with a backward Euler rule.

Remark: Notice that the first four equations of system (1) are linear in the unknowns. The last three are nonlinear (the $\circ$ operator is the Hadamard product of two vectors). Also, note that the $p$’s and $s$’s on the left sides of equations (1) are simply names for the expressions on the right-hand sides. These are used to express equation (2) in a compact form.
0 = -M\dot{v}^t + M\dot{v}^t + W_{\text{ts}}p_{\text{ts}} + W_{\text{ts}}f_{\text{ts}} + W_{\text{ts}}\rho_{\text{ts}} + p_{\text{app}}(t) \\
p_{\text{ts}} = (W_{\text{ts}}W_{\text{ts}} + \frac{\partial W_{\text{ts}}}{\partial \dot{v}^t}) \\
p_{\text{ts}}f_{\text{ts}} = (W_{\text{ts}}W_{\text{ts}} + E\sigma_{\text{ts}}) \\
f_{\text{ts}} = U_{\text{ts}}p_{\text{ts}} + E\sigma_{\text{ts}} \\
\sigma_{\text{ts}} = \frac{\partial W_{\text{ts}}}{\partial \dot{v}^t} + \sigma_{\text{ts}} \\
0 \leq \begin{bmatrix} p_{\text{ts}} \\ f_{\text{ts}} \\ \rho_{\text{ts}} \\ \sigma_{\text{ts}} \\ s_{\text{ts}} \end{bmatrix} \geq 0 (2)

In equation (1) above, the first equation is the Newton-Euler equation. The second imposes linearized non-penetration constraints, the third and fourth are the linear friction laws for the object contacts with the pusher and fixels, and the last three equations encode the friction laws for the object contacts with the support surface. In addition, the subscripts \( n \) and \( f \) indicate that a quantity is related to the normal and frictional components of the contact impulse, respectively. The subscript \( s \) indicates relationship to the support surface, and the subscripts \( ts \) and \( os \) refer to the two perpendicular components of the support friction impulse in the plane. The other quantities in the formulation are the object’s inertia matrix \( M \), the contact constraint Jacobian matrix \( W \), the diagonal matrix \( U \) of friction coefficients, the selection matrix \( E \), and the impulse applied to the body from all non-contact sources \( p_{\text{app}} \).

To form the NCP problem defined by equations (1) and (2) at the current time, we need estimates of the state \( X^t \), the unknown physical parameters \( \beta \), and the input \( u^t \). Solving the NCP yields the state at the end of the current time step. For the planar grasp testbed, the mass and moment of inertia of the object and the shapes of the pusher, object, and fixels may be considered constant and known exactly. However, the friction model parameters vary in space and time. These parameters are the radius of the support tripod \( d_{\text{ts}} \) and the coefficients of friction between the object and pusher \( \mu_f \), the object and fixels \( \mu_f \), and the object and support \( \mu_s \). These four parameters are the elements of \( \beta \) for the testbed. For the remainder of this paper, we will denote the mixed NCP as \( \Gamma \), and write the time-stepping subproblem in the following compact form:

\[ X^{t+1} = \Gamma \left( X^t, u^t, \beta \right). \] (3)

III. A PARTICLE FILTER FOR GRASPING

Some robotic grasping and manipulation tasks have challenging requirements on the accuracy of state estimation of the hand and object. Kinesthetic sensors typically provide sufficient accuracy for the positions of the finger links, but not the object. The object can be tracked visually, but accuracy suffers when the fingers occlude the view. In this case, it is desirable to improve the accuracy with the help of a dynamic model and tactile sensor data. In the event that contacts are not on active tactile sensor sites, only the dynamic model provides a means of approximating the future motion of the system.

A recursive Bayesian filter that uses a parametric dynamic model of manipulation and data from all available sources (e.g., visual, tactile, and kinesthetic sensors) can help us to track the object while also estimating unknown model parameters \( \beta \). With this approach, if all the sensors fail, the model will be as accurate as possible, so error accrual during a sensor black-out will be minimized. Because the underlying dynamic model of grasping systems is nonlinear and nonsmooth, a particle filter is the best type of Bayesian filter to apply.

A particle filter is a simulation-based Bayesian filter that aims to sequentially estimate the distribution of the state vector \( X^{t+1} \) given an observed sequence of output vectors \( Y^{1:t+1} \) on-line\(^2\). The estimation process works by iteratively applying a model to predict the state one time step in the future, and then using observation data to improve the prediction. A general discrete-time Bayesian state-space model is given by the following two conditional probability density functions:

\[ X^{t+1} \sim P_{X^{t+1}}(X^t, u^t) \]
\[ Y^t \sim P_{Y|X^t}(X^t, u^t) \] (4)

where \( Y^t \) denotes the observation at \( t = \Delta t \), \( P_{X^{t+1}|X^t}(\cdot) \) is the state transition model and \( P_{Y|X^t}(\cdot) \) the observation model, or sensor model. Particle filters assume that the system dynamics is a first-order Markov process, which means that past and future data are independent if one knows the current state \( X^t \). Also the observation data is conditionally independent if the state \( X^t \) is given.

The condensation algorithm, a.k.a. Sequential Importance Resampling, is a popular particle filtering method. It approximates the state distribution by a weighted set of particles \( \langle (i)^j, g^j \rangle \). Each \( (i)^j \) is a possible system state, or particle, and \( g^j \) is the importance weight. The sum of the weights over all the particles must be one. The weighted particles can be used to approximate expectations and higher-order moments of various functions with respect to the approximated state distribution.

In particle filtering, when approximating a state distribution by weighted particles, it would be ideal to draw those particles from the actual distribution. However, this distribution is not available, so particles are drawn from a “proposal distribution,” which represents our best guess of the actual distribution at the current time. In the condensation algorithm, a common choice for the proposal distribution is

\(^2\)A tutorial on particle filtering methods and a mathematical derivation can be found in [11].
the state distribution from the current time step. This choice makes sampling and weight update computations very easy.

A. State Space Definition

To track the object’s pose (x and y coordinate for position and θ for orientation) and estimate unknown model parameters, one may define the filter’s state as the concatenation of the object’s state and the unknown model parameters. In our problem, the filter state is the 10-dimensional vector defined as follows:

\[ X = [x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, \mu_s, \mu_p, \mu_t, d_{\text{fix}}]^T. \]

When incorporating tactile sensors into the testbed, the filter’s state space is extended with components describing the contact state. Depending on the type of the sensor, it could be binary variables indicating contact with a particular fixel or (discretized) continuous variables approximating the contact forces.

It is worth noting here that in the dynamic model, the pusher in our testbed is viewed as a position source and not a dynamic object. Therefore its position and velocity are not part of the state in equation (3). Rather they are included in the system input \( u \). The only time when the pusher does not behave as a position source is when the object touches the fixels in a way that causes the pusher to jam (i.e., when the grasp achieves frictional form closure [8], [10]). A final point here is that in our experimental study, we use the actual motion of the physical pusher to drive the simulation model. Since the pusher is not a perfect position source, its motion was tracked using a simple Kalman filter.

B. Issues in Applying Particle Filters to Grasping

To apply a particle filter to the grasping problem, we’re faced with some specific challenges. A commonly used model in robot localization is:

\[
\begin{align*}
X^{t+1} &= f(X^t, u^t) + w^t & \text{System} \\
y^t &= h(X^t, u^t) + v^t & \text{Observation} 
\end{align*}
\]

In this model, \( w \) denotes the process noise and \( v \) the observation noise. Here both \( w \) and \( v \) are mutually independent and identically distributed sequences with known probability density functions. The functions, \( f(\cdot) \) and \( h(\cdot) \), are known deterministic state transition and observation functions. If we use this form of the system model (by replacing \( f \) with Gamma), noise will be added to the solution of the state predicted by the dynamic model of the grasping system. This noise will be obtained by choosing random rigid body displacements in the ambient state space. However, when contacts are present, those sample should come from a lower-dimensional manifold implied by the contact constraints. Physically, this means that in configurations with contact, essentially all of the particles will correspond to configurations in which bodies are overlapping or are separated. As a result, the particles and weights found will be a very poor approximation of the state distribution.

Previous work in [13] introduced a way of estimating parameter by composing a parameter dynamics model with the system dynamic model, thereby forming an expanded state transition model. The main complications here are that there are no physically-motivated dynamics for the friction parameters, and the system model cannot generate useful estimates of parameters that are not currently impacting the system. An example of the latter problem arises before the object touches the fixels. Before contact, the object-fixel friction coefficient \( \mu_t \) cannot be estimated, so adding noise to its part of the expanded dynamic model is computationally wasteful.

C. Solution

To attack the challenges explained above, we designed a specific particle filtering scheme based on the condensation algorithm. The process noise is broken into two components, external force noise \( w_{\text{app}} \) and parameter noise \( w_p \). The force noise is added to the external force in the dynamic model, so that at the end of the time step, the nonpenetration and friction constraints are satisfied. The parameter noise is added to the current value of the parameter vector to allow its estimate to evolve as the state estimates to evolve. In particular, we generate an intermediate value \( \hat{\beta}^{t+1} \) as follows: \( \hat{\beta}^{t+1} = \beta^t + w_p \cdot \tau \), where \( \tau \) is chosen to ensure that the parameters remain within physically reasonable limits. Now, the expanded state transition model can be written as follows:

\[
\begin{align*}
\mathbf{v}^{t+1} &= \Gamma([q^t, \mathbf{v}^t], u^t, \hat{\beta}^{t+1}, w_{\text{app}}) \\
\beta^{t+1} &= \Phi(\hat{\beta}^t, \hat{\beta}^{t+1}) \\
q^{t+1} &= q^t + \Delta t \cdot \mathbf{v}^{t+1}
\end{align*}
\]

where \( \Phi(\cdot, \cdot) \) encodes rules that update only those parameters which are actively involved in the evolution of the system state (see Table I). Note that \( \Gamma^{t} \) differs from the dynamic model described in equation (1) only in the last term of the first equation: \( M_{\mathbf{v}} = M_{\mathbf{v}} + W_{\mathbf{f}} p_{n}^{t+1} + W_{f} p_{f}^{t+1} + W_{\text{ts}} p_{\text{os}}^{t+1} + W_{\text{os}} p_{\text{os}}^{t+1} + p_{\text{app}} + w_{\text{app}} \).

D. Algorithm

In this section, the algorithm flow will be described. To start, we assume the initial distribution of \( q \) is a normal distribution centered about the first observation data, all of which are contact-free. Initial velocities are zero. All particles share common initial parameter values, namely a nominal parameter set \( \beta_0 = [\mu_{\text{os}}, \mu_{\text{p}}, \mu_{\text{i}}, d_{\text{q0}}] \). It could either be obtained from offline calibration or from relevant databases, for example, the friction coefficients can be looked up based on the material used. However, the accuracy of this parameter is not strictly required. The weights of all particles are initially equal.
Algorithm: Grasp Acquisition Particle Filter

For each time step $\ell = 1,\ldots,N$:
If satisfies resample condition
   Resample all particles
For each particle $i = 1,\ldots,N_p$:
   Run the system transition model defined in eq (6)
   Run observation model
   if sensory data is available
      Update particle weight $(i)g^{\ell}$
   else
      Particle weight is unchanged
   end
Calculate estimated mean $\hat{x}_{\ell+1}$ using $\{(i)g^{\ell}\}$

IV. APPLICATION TO REAL EXPERIMENT

In this section, to demonstrate the effectiveness of the proposed scheme, we will present the results from using our particle filter to postprocess the tracking data gathered in two experiments. The tracking outputs of the filter will be plotted beside the results from deterministic simulations using fixed nominal values of the unknown parameters. Since it is impossible to know object’s true trajectory to verify the our estimates, we calculate certain statistics for comparison. In addition, the filter estimates will be projected back onto the camera images to allow visual verification. In the first experiment, the performance of our particle filter is compared to that of a simple Kalman filter that had no knowledge of contact information.

The first experiment is a typical 2D grasp from among a set of similar experiments, in which the contacts achieved frictional form closure soon after the first fixel contact, causing the pusher to jam. We processed the data with both our particle filter and a simple Kalman filter to compare the tracking performances. We repeated the processing after removing the visual data during the critical time of grasp formation. Both filters were compared again. The second experiment is from a set of experiments characterized by extended sliding against the fixels, ending with frictional form closure with contacts against all three fixels. Visual information was again taken away just before grasp achievement. In addition, we compared filter performance with and without simulated tactile data.

A. Comparison to Kalman Filter

In the first experiment, we compare our particle filter to a Kalman filter. Kalman filters are a very widely used Bayesian estimation technique for linear dynamic systems subject to Gaussian noise. To apply a Kalman filter to our problem, we had to linearize the model and remove the sources of discontinuities (the inequality constraints). This meant treating all contact forces as disturbance forces. In addition, since the support surface was horizontal (i.e., $p_{app} = 0$), so the dynamic model became a quasistatic. As a consequence, there was no way for the Kalman filter to estimate the friction parameters, so we only compared the object trajectories, see Fig. 5a)-c) and Fig. 2a). Fig. 2a) shows the trajectory errors of $x$ coordinate of the nominal simulation and two filters taking the unfiltered observed trajectories as exact. All error curves started to increase around $\ell = 130$ when the pusher first touched the object, but the error of the Kalman filter grew the most, due to the quasistatic assumption. After the contact event, the Kalman filter error slowly decayed to the levels of the simulation and particle filter. The same poor performance of the Kalman filter is again visible around the time object contacts a fixel at about $\ell = 270$.

Another interesting feature of the trajectories is apparent near the time of the object-fixel contact. Figs. 5a)-c) show the final grasp configurations predicted by simulation with nominal parameter values, the Kalman filter, and the particle filter, respectively. The error of the nominal simulation is very large. To see why, notice that the simulation Fig. 5a) ends in a frictional form closure grasp with contact with the bottom fixel, but the real system contacts the middle fixel. On the other hand, the particle filter estimates a final configuration with the correct fixel. The Kalman filter appears to succeed in this way too, but its lack of contact force information allows the object to overlap the fixel. The simulation fails, because it gets no updates to the parameters of the dynamic model, so errors accumulate through the entire trajectory. Note that the error plot for $y$ and $\theta$ revealed similar trends as shown in Fig. 3 and Fig. 4.

It is still not clear which filter is best. Therefore we introduce a weighted statistical value $WMSE$(weighted mean square error) to indicate the accuracy of the result.

$$WMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 + \eta(\hat{\theta}_i - \theta_i)^2}$$

$N$ is the number of time steps of the experiment, $\hat{}$ denotes the estimated or simulated trajectory component, and $\hat{}$ denotes the observed trajectory component. $\eta$ is the so-called “virtual radius” which acts as a weighting coefficient to balance the orientation component and the two translational components of the object’s configuration. The WMSE value from three sources are listed in Table II in the first row. Two filtered result are about the same magnitude with particle filtered result slightly smaller. Both are much smaller than the result from the nominal simulation.

B. Visual Occlusion

We reran the same filtering experiment with partial visual occlusion from $\ell = 200$ to the end of the experiment. As in real robotic manipulation, occlusion caused by a robot hand is common when acquiring a grasp. During the visual signal blackout, the particle filter updated all particles using the system transition model without either updating the
particle weights or introducing parameter noise. As a result, the friction parameters did not change after this time. The estimated results and nominal simulation are presented in Figs. 2b) and 5d)-f). The colored markers on the object are covered to indicate the withholding of visual data from the filters. However, the visual data was still used to calculate the error. WMSE values are listed in the second row of Table II.

The plots in Fig. 2b) are identical to those in Fig. 2a) up to the time of the blackout (note that the vertical scales on the plots are different). A big difference only comes from Kalman filter’s result after $\ell = 270$. After the grasp is formed and the system stops moving, the Kalman filter’s quasistatic, contact oblivious model allows the estimates to continue on...
with constant velocity. This is clearly evident in Fig. 5f) as the projected object configuration significantly overlaps the middle fixel. On the other hand, particle filter still delivers good estimates through the visual occlusion. This implies that the parameter estimates were accurate just before the blackout. More importantly, only the particle filter captured the correct grasp, including contact with the middle fixel.

Fig. 6 shows the trajectories estimated by the particle filter with the visual data. The case with visual occlusion is not shown, since the only difference is that after the occlusion, the estimated parameter values do not change. In Fig. 6, notice that $\mu_b$, $\mu_p$ and $d_{tri}$ started to vary as soon as the pusher touched the object. However, the filter could not gain any knowledge of $\mu_f$ until object touched a fixel. Except $\mu_f$, all three parameters vary within a relatively smaller range after some initial transients. One problem to notice is the spikes with $\mu_p$, $\mu_f$ and $d_{tri}$, especially $\mu_p$. This is not an indication that the real parameters underwent drastic sudden changes. Actually, the spikes are caused by the fixel contact. At this point in the filtering process, most particles generated corresponded to frictional form closure.

With respectively different parameter samples, big amount of particles enter frictional form closure, which reduces the sample size in effect and lead to a poor estimate of filtering distribution. Such problem will be addressed in future work.

C. Using Simulated Tactile Sensor

Can we infer from the second experiment that it is safe to rely on a particle filter’s model inference ability completely when occlusion is present? No, not necessarily.

The green plots in Fig. 8b), Fig. 7 showed the results from applying particle filter to another experiment with occlusion starting from $\ell = 240$. From Fig. 8b), we can see that estimated object pose differs from the actual object quite a lot at time $\ell = 300$. The estimated pose only has contact with one fixel, while real object pose has contacts with all three fixels. One can tell from object’s real position at steps $\ell = 257$ and $\ell = 300$ that the object actually slides both clockwise and upward after it first touches one fixel until it reaches frictional form closure with all four fingers. However, with nominal simulation and the particle filter, the object stops upon first contact with the fixel. This result does not necessarily suggest that all our parameter estimates till the time visual occlusion happened were wrong. It could be because we have incorrect prior knowledge of $\mu_f$ (bigger than the real value which causes the sliding) and particle filter has no way of refining this parameter before the occlusion happens (no contact with the fixel happened yet).

To verify our assumption, we added simulated tactile sensors to the experiment. We assumes that binary tactile sensors were available on all three fixels. As suggested in section III, the state space was expanded to include three
boolean contact variables. During the time when no visual data is available, the particle filter updated all particles’ weights using the simulated tactile sensory information. The blue plots in Figs. 8c) and 7 show the new results. The y error plot in Fig. 7b) reveals an opposite trend of the new estimate from the green trajectory. It turned out that the particle filter without tactile data estimated that the object slides faster than real object’s movement. Since no visual sensory information would be fed to the particle filter during this time, it would be possible that such an error can happen. One interesting thing to notice in Fig. 8a), the spikes in the first half of the error trajectories. This was caused by a problem with our frame grabber not keeping up with the 30Hz refresh rate, and thus occasionally having the same data in two successive frames.

From the error plot and the comparisons of the images, we can easily see that with the aid of tactile sensor information, the particle filter was able to follow the correct trajectory again, even without visual information. The WMSE value in Table III confirms the result. Fig. 9 shows the estimated parameters from both cases. Estimated \( \mu \) with the simulated tactile sensor is smaller than the nominal value since the contact happened, which justifies our assumption. Actually, the nominal simulated trajectory partially justifies this guess too.

D. Computational Issue

In current work, we found 500 particles was sufficient for good filtering results. Each particle runs collision detection, forms NCP problems, and calls the PATH solver [5] to solve it independently of the other particles. Due to this parallel nature, we adopted parallelism when computing state transitions for the particles. However, processing a pre-collected 20-second experiment currently requires 10 minutes of cpu time on an i7-core desktop PC. This is far slower than real time, and if we expand the problem to include geometric model parameters of the object in the filter state, then we will require even more particles and cpu time. Nonetheless, we have reason to be optimistic. Over 90% of the computational effort was spent on solving the dynamic model. New solvers based on new formulations of the dynamic model have the potential to be hundreds time of faster and suitable for deployment on modern GPUs. These are topics for our future work.

V. CONCLUSIONS AND FUTURE WORKS

In this paper we tried to apply a particle filter to a simplified grasping problem as a first step in exploring the G-SL (AM)\(^2\) problem. After analyzing the specific problems of applying a particle filter, a specialized filtering scheme was proposed. The scheme makes sure that non-penetration constraints of the dynamic model are always satisfied with all samples and also provides a reasonable model to refine the physical parameters.

From the particular grasping experiments studied, one can conclude that the proposed scheme is effective in improving the system’s physical parameters through the use of kinesthetic, visual, and tactile data, which (at least in our example), enabled accurate tracking even with visual occlusion.

For future work, as mentioned in the end of section IV, we need to solve the “sample degeneracy” problem when object has a low dimensional free space. A possible idea is to investigate hybrid sampling by sampling from the sensory model. Furthermore, we would like to incorporate geometric model parameters and extend the filtering scheme to G-SL (AM)\(^2\) problem in three-dimensional space. Computational problems must be addressed to facilitate achieving these goals.

VI. ACKNOWLEDGMENTS

REFERENCES