The Application of Particle Filtering to Grasping Acquisition with Visual Occlusion and Tactile Sensing

Li (Emma) Zhang and Jeffrey C. Trinkle

***Abstract*— Advanced grasp control algorithms could benefit greatly from accurate tracking of the object as well as an accurate all-around knowledge of the system when the robot attempts a grasp. This motivates our study of the G-SL(AM)^2 problem, in which two goals are simultaneously pursued: object tracking relative to the hand and estimation of parameters of the dynamic model. We view G-SL(AM)^2 problem as a filtering problem. Because of stick-slip friction and collisions between the object and hand, suitable dynamic models exhibit strong nonlinearities and jump discontinuities. This fact makes Kalman filters (which assume linearity) and extended Kalman filters (which assume differentiability) inapplicable, and leads us develop a particle filter. An important practical problem that arises during grasping is occlusion of the view of the object by the robot's hand. To combat the resulting loss of visual tracking fidelity, we designed a particle filter that incorporates tactile sensor data. The filter is evaluated off-line with data gathered in advance from grasp acquisition experiments conducted with a planar test rig. The results show that our particle filter performs quite well, especially during periods of visual occlusion, in which it is much better than the same filter without tactile data.**

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 quantities (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 system may not be 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 slip and tumble out of the grasp. The problems are magnified when the physical quantities are known only roughly, or vary over space and time as friction parameters are known to do.

This paper is motivated by the idea that one 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. We will refer to this problem as 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 object during grasp acquisition and manipulation. "Simultaneous" implies that localization, modeling, and manipulation will all occur together – in real time.

The G-SL(AM)^2 problem is potentially a high-dimensional filtering problem that presents special challenges peculiar to dynamic systems with intermittent contact: a nonsmooth dynamic model, a very high-dimensional state-space, and unknown contact friction parameters that vary unpredictably. To complicate things further, the dimension of the state-space varies in time as contacts form and break, and each such event effectively changes the structure of the dynamic model.

Our approach to the G-SL(AM)^2 problem is based on Bayesian filtering, in particular, particle filtering [5]. A straight forward application would be to directly sample over the whole state space, which will be bound to *fail* in our problem, because in grasping, large portions of the state space are invalid. Specifically, particles should not be chosen that correspond to overlap between the geometric models of the bodies, nor should they correspond to contact forces that are not within their respective friction cones. Some may argue that a trial-and-error approach will help to recognize the correct samples, while we insist that such a method would lead to very few effective samples when contact is present, thereby causing poor estimation.

To shed light on these issues, we present a case study of a particular scaled-down version of the G-SL(AM)^2 problem, using data from our planar grasp acquisition testbed shown in Fig. 1. The main advantages of using this testbed for our

This work was supported by NSF CCF-0729161.

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

Trinkle is Professor of Computer Science, Rensselaer Polytechnic Institute, Troy, NY 12180, USA. trink@cs.rpi.edu

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

initial study are the lower-dimensional state space and the smaller number of unknown model parameters. In this paper, it is assumed that the geometric models and most parameters of the physical model are constant and known; only four friction parameters are assumed unknown.

A. Related Research

In [8], 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 locations of the contact points on the object and the object pose. In [6], 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. M. Krainin et al. developed an approach to build 3D surface model of unknown object using data from a depth camera. Their approach doesn't rely on either highly accurate depth sensor or highly precise manipulator as they use a Kalman filter to help track both the arm and object's positions in [9]. These research work all focused on dealing with object geometries, while we interest in building a complete model of the system which includes geometry as well as physical parameters. It appears though there's no direct approach to incorporating these information to their work.

Yuval Tassa et al. formulated a Stochastic Linear Complementarity Problem (SLCP) when trying to apply local methods to nonlinear control problems involving contact and friction by claiming that SLCP dynamics is differentiable in [13]. However, as the authors pointed out, the actual distributions propagated by the true dynamics become multimodal upon contact, requiring either nontrivial parameterization or mixture of samples where the dynamics still be non-differentiable.

The rest of the paper is organized as follows. Section II introduces our planar grasp testbed and its discrete-time dynamic model in the form of a nonlinear complementarity problem [10]. The accuracy of this model and the robustness of physics engine `dVC2d` in simulating the grasping process has been shown in previous work [15]. Section III develops a particle filter algorithm that effectively deals with the issues associated with the contact and friction constraints mentioned earlier. These modifications are not specific to grasping, but are appropriate for any dynamic system composed of nominally rigid bodies which undergo intermittent contact with Coulomb friction, such as assembly problems. In Section IV, we apply our particle filter to two planar grasp acquisition problems to demonstrate its ability to track the state of the grasped object and friction parameters, with and without visual and tactile signals. Section V summarizes the work and points out future directions.

II. EXPERIMENT AND DYNAMIC MODEL

A. Experimental Set-Up and Assumptions

In our experimental environment shown in Fig. 1, one linear pusher (or thumb) and three fixels (or fixed fingers) are mounted on an aluminum plate to act as a simple one-degree-of-freedom "hand." The object is shown in configuration $q = (x, y, \theta)$ defined by the x- and y-coordinates of the origin of the object-fixed frame, and the relative angle of the x-axes of the base and object frames. During grasp acquisition, the object is pushed toward the fixels, until fixel contacts halt the motion or the pusher reaches its travel limit.

As the system moves, frames action are captured by an overhead camera at $30Hz$ and are post-processed to extract the configuration trajectory of the object and pusher. After calibration, the camera provides positions and orientations with resolutions of about $1mm$ in displacement and 1° in rotation. The thumb's actuator is very stiff with settable "constant" speeds, so its motion is tracked using visual data and a simple Kalman filter.

As this paper is being written, the testbed has no tactile sensors. However, in Section IV, synthetic tactile data is generated to study the potential impact of using tactile data when, in 3D systems, the robot's view of the object is occluded by the hand. In our testbed, the camera's view is never occluded. This is a benefit for this study, since we can choose the periods of visual occlusion.

Our main modeling assumptions are as follows. 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 face-to-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-to-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 details of the geometries of the object and support surface at the contact points, material properties, dirt, relative humidity, and more.

B. Dynamic Model

The dynamic model is composed of the Newton-Euler equation (restricted to the plane), non-penetration constraints (restricted to the object, pusher, and fixels), and Coulomb's Friction Law. Frictional contact occurs between the object and fixels, the object and pusher, and the object and the support surface. For the former two, the contact normals lie in the plane of motion, so the friction cones reduce to 2D linear cones. For the support contacts, the contact normals are perpendicular to the plane, so the full 3D quadratic cone is retained. The resulting dynamic model can be formulated

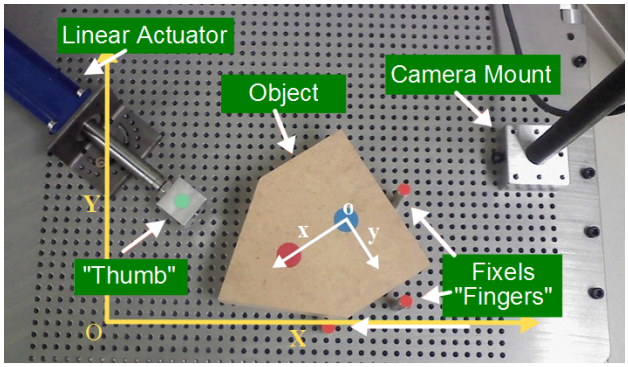

Fig. 1. Planar Grasping Testbed

as a mixed nonlinear complementarity problem (mixed NCP) and solved by the `path` solver [4]. This mixed NCP could be linearized by approximating the quadratic cones with polyhedral cones, but previous results show that the NCP model is more accurate and can be solved just as quickly as the linearized model [2].

Our decision to use a tripod to represent the support contacts allows us to solve uniquely for the normal forces at the vertices of the tripod. The support contact points each bear one-third of the object's weight. This partial solution reduces the size of the NCP that must be solved for every simulation time-step. However, we *cannot* solve for the friction forces at the tripod vertices without considering the complete dynamics of the system, because they depend on the motion of the object. As such our problem satisfies the conditions of the 2.5D dynamic model available in dVC2d simulator as described in [3]. The "half" degree of freedom is added to represent the fact that, 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.

Below, the mixed NCP implementing the 2.5D, discrete-time, dynamic model is presented in equations (1) and (2) (see [1] for a complete derivation). Let Δt be the (constant) simulation time-step and let $\ell = \{0,1,...,N\}$ 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^{ℓ}, v^{ℓ}, σ^{ℓ}, 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, $\ell = 1,...,N$. The new velocity $v^{\ell+1}$ is used to update the object's pose with a backward Euler time-stepping rule.

$$
\begin{aligned}
0 &= -Mv^{\ell+1} + Mv^{\ell} + W_{\mathrm{n}}^{\ell} p_{\mathrm{n}}^{\ell+1} + W_{\mathrm{f}}^{\ell} p_{\mathrm{f}}^{\ell+1} \\
&\quad + W_{\mathrm{ts}}^{\ell} p_{\mathrm{ts}}^{\ell+1} + W_{\mathrm{os}}^{\ell} p_{\mathrm{os}}^{\ell+1} + p_{\mathrm{app}}(t) \\
\rho_{\mathrm{n}}^{\ell+1} &= (W_{\mathrm{n}}^{\ell})^T v^{\ell+1} + \tfrac{\psi_{\mathrm{n}}^{\ell}}{h} + \tfrac{\partial \psi_{\mathrm{n}}^{\ell}}{\partial t} \\
\rho_{\mathrm{f}}^{\ell+1} &= (W_{\mathrm{f}}^{\ell})^T v^{\ell+1} + E\sigma^{\ell+1} \\
s^{\ell+1} &= U p_{\mathrm{n}}^{\ell+1} - E^T p_{\mathrm{f}}^{\ell+1} \\
s_{\mathrm{s}}^{l+1} &= U_{\mathrm{s}}^2 p_{\mathrm{ns}}^{\ell+1} \circ p_{\mathrm{ns}}^{\ell+1} - p_{\mathrm{ts}}^{\ell+1} \circ p_{\mathrm{ts}}^{\ell+1} - p_{\mathrm{os}}^{\ell+1} \circ p_{\mathrm{os}}^{\ell+1} \\
0 &= (U_{\mathrm{s}} p_{\mathrm{ns}}^{\ell+1}) \circ (W_{\mathrm{ts}}^{\ell} v^{\ell+1}) + p_{\mathrm{ts}}^{\ell+1} \circ \sigma_{\mathrm{s}}^{\ell+1} \\
0 &= (U_{\mathrm{s}} p_{\mathrm{ns}}^{\ell+1}) \circ (W_{\mathrm{os}}^{\ell} v^{\ell+1}) + p_{\mathrm{os}}^{\ell+1} \circ \sigma_{\mathrm{s}}^{\ell+1}
\end{aligned}
\tag{1}
$$

$$
0 \leq \begin{bmatrix} \rho_{\mathrm{n}}^{\ell+1} \\ \rho_{\mathrm{f}}^{\ell+1} \\ s^{\ell+1} \\ s_{\mathrm{s}}^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} p_{\mathrm{n}}^{\ell+1} \\ p_{\mathrm{f}}^{\ell+1} \\ \sigma^{\ell+1} \\ \sigma_{\mathrm{s}}^{\ell+1} \end{bmatrix} \geq 0. \tag{2}
$$

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 $_{\mathrm{n}}$ and $_{\mathrm{f}}$ indicate that a quantity is related to the normal and frictional components of the contact impulses, respectively. The subscript $_{\mathrm{s}}$ indicates relationship to the support surface, and the subscripts $_{\mathrm{ts}}$ and $_{\mathrm{os}}$ 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_{app}.

A key point to keep in mind about this model is that it is *nonsmooth* - not just nonlinear with inequality constraints imposed by contacts, but nonsmooth. Clearly Kalman filters (which assume a linear dynamic model) do not apply. Model nonsmoothness manifests as impulsive contact forces and discontinuous velocities. These are a result of transitions in and out of contact and between sticking and slipping at sustained contacts. At these times, derivatives with respect to the state variables do not exist. This fact makes the application of extended Kalman filters (which assume model differentiability) to our model impossible. Strictly speaking, a Kalman filter or extended Kalman filter could be used, but only after removing the contact features of the model. In terms of the dynamic model, equations (1) and (2), this would mean eliminating complementarity conditions (2) entirely and all but the first equation in equation (1). This first equation is Newton's second law with all contact forces treated as unknown disturbances in p_{app}.

Notice also that the first four equations of system (1) are linear in the unknowns. The last three are bi-linear or quadratic (the $\circ$ operator is the Hadamard product of

two vectors). The derivatives of these expressions with respect to the state variables exist everywhere, so it might appear as though one could use an EKF for the G−SL(AM)[2] problem. However, the combination of system (1) with the complementarity conditions expressed in equation (2) causes nonexistence of the needed derivatives.

Remark: To avert a potential minor point of confusion, the ρ'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.

The NCP problem defined by equations (1) and (2) is needed within our filter. To form the NCP at the current time, we need estimates of the state of the object $X^\ell = [q, v]$ (where q is the object's configuration and v is its time derivative), the unknown physical parameters β^ℓ, and the input u^ℓ. Solving the NCP yields the system state $X^{\ell+1}$ 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_{tri} and the coefficients of friction between the object and pusher μ_{p}, the object and fixels μ_{f}, and the object and support μ_{s}. These four parameters are the elements of β for the testbed. For the remainder of this paper, we will denote the mixed NCP as Γ, and write the time-stepping subproblem in the following compact form:

$$X^{\ell+1} = \Gamma\left(X^\ell, u^\ell, \beta\right). \tag{3}$$

Note that this form is generally applicable to systems of bodies with intermittent frictional contact, not just 2.5D grasping problems.

III. A PARTICLE FILTER FOR GRASPING

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 β. With this approach, if all the sensors fail, the model will be as accurate as possible, so error accrual during a sensor blackout will be minimized. As argued above, we have chosen to apply a particle filter to the G−SL(AM)[2] problem.

A particle filter is a simulation-based Bayesian filter that aims to sequentially estimate the distribution of the state vector $X^{\ell+1}$ given an observed sequence of output vectors $Y^{1:\ell+1}$ 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:

$$\begin{aligned} X^{\ell+1} &\sim P_{X^{\ell+1}|X^\ell}\left(X^\ell, u^\ell\right) \\ Y^\ell &\sim P_{Y^\ell|X^\ell}\left(X^\ell, u^\ell\right) \end{aligned} \tag{4}$$

[2]A tutorial on particle filtering methods and a mathematical derivation can be found in [12].

where Y^ℓ denotes the observation at $t = \ell\Delta t$. $P_{X^{\ell+1}|X^\ell}(\cdot)$ is the state transition model and $P_{Y^\ell|X^\ell}(\cdot)$ 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^ℓ. Also the observation data is conditionally independent if the state X^ℓ is given. Most multibody dynamic models satisfy these conditions.

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 $\langle {}^{(i)}X^\ell, {}^{(i)}g^\ell \rangle$. Each ${}^{(i)}X^\ell$ is a possible system state, or particle, and ${}^{(i)}g^\ell$ 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, the proposal distribution is chosen as the state distribution from the current time step. This choice makes sampling and weight update computations very easy, but accuracy is reduced.

A. State Space Definition

To track the object's pose and estimate the unknown model parameters, one may define the filter's state vector as the object's state, the model parameters, and tactile sensor values. In our problem, the filter state is defined as follows:

$$X = [x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}, \mu_{\text{s}}, \mu_{\text{p}}, \mu_{\text{f}}, d_{\text{tri}}, c_1, c_2, c_3]^T,$$

where c_1, c_2, and c_3 are the binary outputs of the tactile sensors on each of the three fixels (Note that it is also possible to incorporate contact force data if the tactile sensors can provide it.). Two experiments are studied in the next section. One of them ignores the tactile data and so the filter state is 10-dimensional. The other experiment uses tactile data, so the filter is 13-dimensional.

It is worth noting here that the control of the pusher is very stiff. Therefore, in the dynamic model, the pusher is viewed as a position source and *not* a dynamic object. As such, the pusher's 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 [7], [11]). A final point here is that in our experimental study, we use the actual motion of the physical pusher to drive the simulation model.

B. Issues in Applying Particle Filters to Grasping

A commonly used model in tracking problems is:

$$\begin{aligned} X^{\ell+1} &= f(X^{\ell},u^{\ell})+w^{\ell} \quad \text{System} \\ Y^{\ell} &= h(X^{\ell},u^{\ell})+v^{\ell} \quad \text{Observation} \end{aligned} \tag{5}$$

In this model, w denotes the process noise and v the observation noise. Here both w and v are mutually independent distributed sequences with known probability density functions. The functions, $f(.)$ and $h(.)$, are known deterministic state transition and observation functions. If we use this form of the system model (by replacing f with Γ), 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 possible, those samples should come from a collision-free subset of the state space or from a lower-dimensional manifold defined by the contact constraints. Physically, this means that in configurations with contact, many of the particles will correspond to configurations in which bodies are overlapping or are separated when they should be touching. As a result, the particles and weights found will be a very poor approximation of the state distribution. A trial-and-error approach will be able to tell which particles are in bad configuration. However, we argue that such an approach still cannot avoid the problem of lacking effective particles and thereby causing poor state representation.

Previous work in [14] introduced a way of estimating unknown parameters by composing a parameter dynamic model in addition to 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. As an example, before the object contacts a fixel, the object-fixel friction coefficient μ_{f} cannot be estimated, so computations associated with its estimation would be wasteful. Our approach to handling this problem is described in the next two subsections.

C. Solution

To attack the challenges explained above, we designed a specific particle filter based on the condensation algorithm. The process noise is broken into two components, external force noise w_{app} and parameter noise w_{β}. The force noise enters the dynamic model as follows:

$$M\nu^{\ell+1} = M\nu^{\ell} + W^{\ell}p^{\ell+1} + p_{\text{app}} + w_{\text{app}}.$$

$W^{\ell}p^{\ell+1}$ represents the sum of all contact and friction impulse terms from the first equation of equation (1) and we denote the new dynamic model as Γ'. Adding noise in this way guarantees that the nonpenetration and friction constraints will be satisfied at the end of the time-step. The parameter noise is added to the current value of the parameter vector to allow its estimate to evolve as the state estimate evolves. In particular, we generate an intermediate value $\hat{\beta}^{\ell+1}$ as:

TABLE I

SELECTIVE STRATEGY OF ϕ

contact between the pusher and object exists	choose $\hat{\mu}_{\text{p}}^{\ell+1}$
contact between the pusher and fixel exists	choose $\hat{\mu}_{\text{f}}^{\ell+1}$
$\|\nu^{\ell}\| > 0$ or $\|\nu^{\ell+1}\| > 0$	choose $\hat{\mu}_{\text{s}}^{\ell+1}$ and $\hat{d}_{\text{tri}}^{\ell+1}$

$\hat{\beta}^{\ell+1} = \beta^{\ell} + w_{\beta}\cdot\tau(\beta^{\ell})$, where τ is chosen to ensure that the parameters remain within physically reasonable limits. Now, the expanded state transition model can be written as follows:

$$\begin{aligned} \nu^{\ell+1} &= \Gamma'([q^{\ell},\nu^{\ell}],u^{\ell},\hat{\beta}^{\ell+1},w_{\text{app}}) \\ \beta^{\ell+1} &= \Phi(\beta^{\ell},\hat{\beta}^{\ell+1}) \\ q^{\ell+1} &= q^{\ell}+\Delta t\cdot\nu^{\ell+1} \end{aligned} \tag{6}$$

where $\Phi(\cdot,\cdot)$ encodes rules that update only those parameters which are actively involved in the evolution of the system state according to the rules in Table I. When the state is expanded with c_1, c_2, and c_3, their values will be obtained from collision detection at every time step.

D. Algorithm

We assume the initial distribution of q is normal with mean equal to the first observation data. The variance was carefully chosen to balance granularity and diversity of the particles. The system is initially at rest and the object does not contact the fixels. All initial particles generated share common initial parameter values, namely a nominal parameter set $\beta_0 = [\mu_{\text{s0}},\mu_{\text{p0}},\mu_{\text{f0}},d_{\text{tri0}}]$, and are assigned equal weights. The parameter values could be obtained via offline calibration or from relevant databases. Fortunately, the accuracy of these parameters is not strictly required for good filter performance. The general algorithm framework is described as followed:

Algorithm: Grasp Acquisition Particle Filter

For each time step $\ell = 1,...,N$:
  If satisfies resample condition
    Resample all particles
  For each particle $i = 1,...,N_{\text{p}}$:
    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 EXPERIMENTS

In this section, to demonstrate the effectiveness of the proposed approach, we will present the results from using our particle filter to postprocess the visual data gathered in two representative 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 will calculate certain statistics for comparison. In addition, the state estimates will be mapped back onto the camera images to allow visual verification.

The first experiment was a typical one from among a set of similar experiments, in which the system jammed very soon after the object first made contact with a fixel. In this experiment, the performance of our particle filter was compared to that of a Kalman filter for which the dynamic model treats all contact forces as disturbances, as discussed above. We repeated the comparison after removing the visual data a few time steps before grasp formation.

The second experiment was a typical one from a set of experiments characterized by extended sliding against the fixels, ultimately ending with contact with all three fixels. Visual information was again taken away just before the first fixel contact, but in this experiment, we introduced synthetic binary tactile data for the fixels. Because the Kalman filter cannot use tactile data, the comparison here was between our particle filter with and without tactile data.

A. Comparison to Kalman Filter

Since the Kalman filter did not contain terms representing the contact forces, it could not estimate friction parameters, so in the first experiment, we compared the object trajectories for two scenarios with different sensory data availability. In the first scenario, the visual data was available throughout the entire grasping process. In the second, visual data was blacked out before the first fixel contact.

Fig. 2a) shows the trajectory errors of the x-coordinate [3] of the nominal simulation (yellow curve), the particle filter (green curve), and the Kalman filter (red curve), taking the camera tracking data as exact. Figs. 3 a), b), and c) show the final grasp achieved by the real system overlaid by the final configurations of the object predicted by the nominal simulation, the particle filter, and the Kalman filter (drawn in yellow (Fig. 3a)), green (Fig. 3b)), and red (Fig. 3c))), respectively.

The performance of the Kalman filter is poor only right after the object contacts the pusher (at about $\ell = 130$) and a fixel ($\ell = 270$). At these times, the red curve exhibits transients caused by the contact impulses that the Kalman filter treats as disturbances. Note that by the time the second transient subsides, the Kalman filter's estimate of the object's final state matches the that of the real system (again look at (Fig. 3c))).

Another interesting feature of the trajectories is apparent near the time of the object-fixel contact. The error of the nominal simulation becomes very large, because it uses constant nominal values of the friction parameters. By contrast, the particle filter updates the parameters over time to keep the model as accurate as possible. Even though both the nominal simulation and the particle filter predict that the final grasp has force closure with one fixel contact, they predict contact with different fixels. The particle filter, with its more accurate model, predicts the correct fixel contact.

[3]The error plots for y and θ revealed similar trends and hence are not presented.

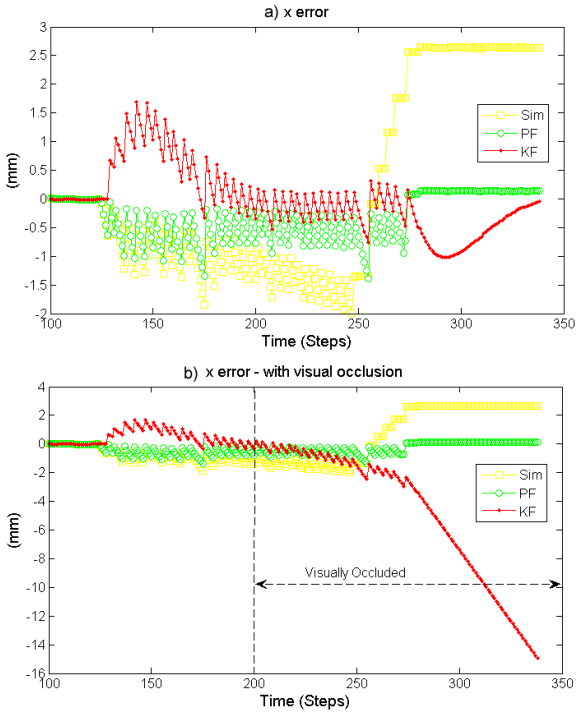

Fig. 2. Estimated Error of Object x-trajectory : a) no visual occlusion b) visual occlusion

TABLE II
WMSE (MM)

Experiment	Nominal Simulation	PF	KF
No Visual Occlusion	4.8078	0.1950	0.2549
Visual Occlusion	4.8078	0.1905	17.7117

While the particle filter is the clear winner in the particular comparison above, we introduce a weighted statistical value WMSE (weighted mean square error) to quantify the accuracy of the results:

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

where $\hat{}$ denotes the estimated or simulated trajectory component, $\bar{}$ denotes the observed trajectory component, and η is the so-called "virtual radius," which acts as a weighting coefficient to balance the orientation component and the two translational components. The WMSE values for the simulation and two filters with visual data always available are listed in the first row of Table II. The large WMSE value for the simulation and small similar values for the filters agrees with visual inspection of Figs. 2 and 3.

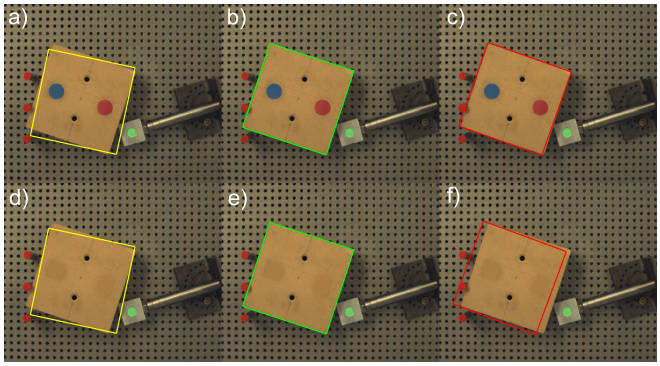

Fig. 3. Estimated Object Pose at $\ell = 300$: Top row: no visual occlusion a) nominal simulation b) particle filter estimate c) Kalman filter estimate Bottom row: visual occlusion d) nominal simulation e) particle filter estimate f) Kalman filter estimate

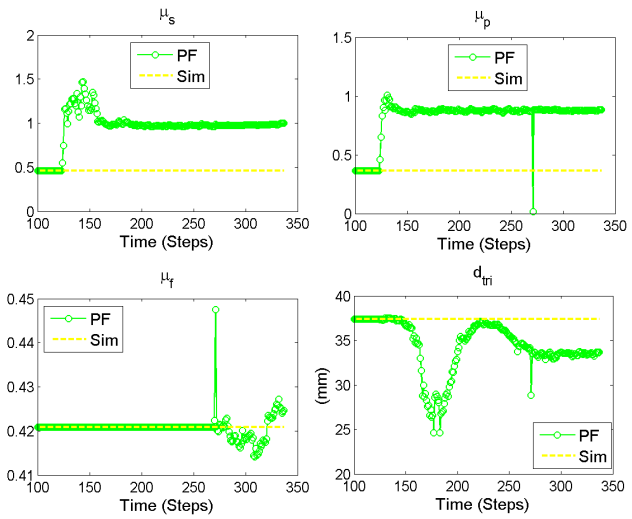

Fig. 4. Estimated Parameters: no visual occlusion

B. Visual Occlusion

We reran the same filtering experiment with visual occlusion from $\ell = 200$ to the end of the experiment. 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 resulting x-coordinate trajectories of the nominal simulation and the two filters are presented in Fig. 2b) with a compressed vertical scale to accommodate the large error of the Kalman filter during visual occlusion. The second row of images in Fig. 3 show the final predicted grasp configurations overlaid on an image of the real system. The plots in Fig. 2b) are identical to those in Fig. 2a) up to the time of the occlusion. After that, the errors of the Kalman filter accrue slowly until the grasp is formed at $\ell = 270$. At this point, the system stops moving, but the Kalman filter predicts that the object continues to move with the velocity it last estimated. This is clearly evident at the right end of the red curve in Fig. 2 and also in Fig. 3f), which shows overlap of the object pose predicted by the Kalman filter with the fixels. On the other hand, the particle filter still delivers good estimates through the period of visual occlusion.

Fig. 4 shows the friction parameter 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. 4, notice that the pusher friction coefficient μ_p and the support friction parameters, μ_s and d_{tri}, start to vary as soon as the pusher touches the object. However, the filter cannot gain any knowledge of μ_f until object touches a fixel. Except for μ_f, all three parameters vary within a relatively smaller range after some initial transients.

One should also observe the spikes in the plots, especially for μ_p. This is not an indication that the real parameters underwent drastic sudden changes. Actually, the spikes are caused by the fixel contact. Specifically, since the grasp sticks to the two contacts, the dimension of the effective state space of valid particles is reduced. As a result our algorithm produced too few particles to accurately represent the state density function.

C. Using Synthetic Tactile Sensor Data

In the second experiment, our particle filter was tested on an experiment in which the acquired grasp had contact with all three fixels after a significant period of sliding on one fixel. We assumed visual occlusion occurred just before the first fixel contact.

Three error plots are shown in Fig. 5: simulation with nominal parameters (yellow), the particle filter without tactile data (green), and the particle filter with tactile data (blue). The pusher contacts the object at time-step $\ell = 130$, the first fixel contact occurs at $\ell = 257$, and the grasp is completed at $\ell = 270$. Visual occlusion is chosen to occur at $\ell = 240$. Fig. 6 shows predicted object poses on camera images at times $\ell = 257$ (top row of images) and $\ell = 300$ (bottom row). The colors of the projected object correspond to the filter error colors.

Consider the green plots in Fig. 5 and the corresponding images in Fig. 6b). We can see from the upper image in column b) of Fig. 6 that at the time of the first fixel contact, the filter without tactile data estimates the object's pose accurately. However, by the end of the grasping process, less than two seconds later, the error has become significant. The filter's estimate shows the object in contact with only one fixel and overlapping the pusher. Interestingly, the orientation error of the filter with tactile data is slightly larger than that of the filter without. This is due to the fact that the filter with tactile data has different dynamics.

Comparing images in the center and right of the bottom row, the filter with tactile data clearly outperforms the filter without by the end of the grasping process. The error plots in Fig. 5 also support this conclusion, which might not be so obvious with the y-error plot. Notice that this plot ends with nearly $2mm$ of error. This is because the binary tactile sensor data contains no information about the displacement when sliding along all three contacts.

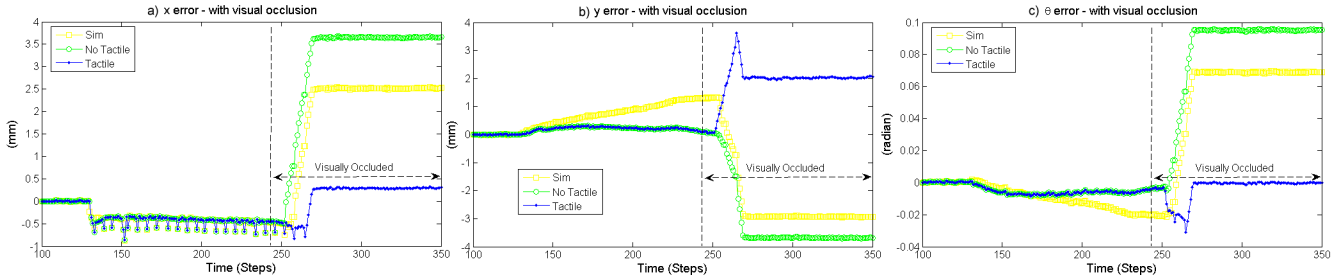

Fig. 5. Estimated Object Trajectory Error: a) x error b) y error c) θ error

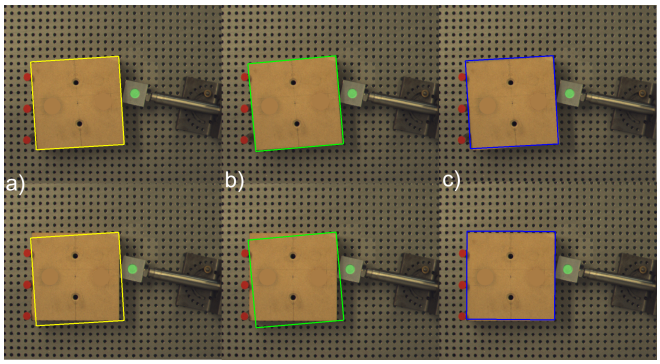

Fig. 6. Estimated Object Pose with Visual Occlusion: Top row: at $\ell =$ 257 Bottom row: at $\ell = 300$ a) nominal simulation b) no tactile sensor c) simulated tactile sensor

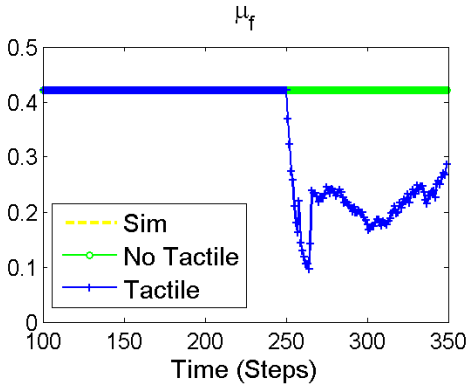

Fig. 7. Estimated Parameters with Visual Occlusion

TABLE III
WMSE (MM)

Nominal Simulation	No Tactile	Tactile
3.9990	6.7502	1.3301

One curious characteristic to notice in Fig. 6a) is the set of fairly regular small spikes in the first half of the error trajectories. These were caused by a problem with our frame grabber not keeping up with the $30Hz$ refresh rate, and thus having the same data in several successive frames. This problem is also responsible for the rapid oscillations in the error curves in Fig. 2. Nonetheless, the particle filter performed reasonably well on our problems. The WMSE value in Table III quantifies this fact.

D. Several More Experimental Results

Table IV gives the results from processing four other experiments with the Kalman filter and with our particle filter with and without tactile and visual information. Column 2 contains the WMSE errors for the nominal simulation using the best available constant friction parameter values. The data in columns 3, 4, and 5 contain the WMSE values for the three filters discussed above, and assuming that visual occlusion occurred just before the first fixel contact. The last column was generated using our particle filter with tactile data and assuming visual data was always available.

Generally, from the results we can see that when visual occlusion occurs, the Kalman filter can have very big errors. The particle filter without tactile data performs marginally better, but the particle filter with tactile data performs much better – nearly as well as the case with all data sources available all the time.

E. Computational Issue

In the current work, we found 500 particles to be sufficient for good filtering results. Each particle runs collision detection, forms NCP problems, and calls the `path` solver [4] to solve the dynamic model 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 a core-i7 desktop PC. This is far slower than real time, and if we expand the filter state 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 times of faster when depolyed on modern multi-threaded CPUs and GPUs.

V. CONCLUSIONS AND FUTURE WORKS

In this paper we developed an approach to apply particle filters to the G-SL(AM)^2 problem. The approach ensures that non-penetration and stick-slip friction constraints of the dynamic model are always satisfied with all particles and

TABLE IV
WMSE (MM)

Experiment	Nominal Simulation	Kalman Filter	No Tactile	Tactile	No Occlusion
1(Contact with a fixel and slide)	9.9906	40.5436	14.2866	0.3754	0.4326
2(Contact with a fixel and slide)	32.8811	56.0046	39.6202	2.2959	0.2572
3(Contact with fixel and stop)	1.4412	4.5666	0.6282	1.3973	0.3876
4(Contact with fixel and stop)	0.584	21.8527	0.4534	0.336	0.3739

also provides a reasonable model to refine the physical parameters. From the particular grasping experiments studied, one can conclude that the proposed approach can be effective in improving our knowledge of the system's physical parameters while simultaneously tracking the object during visual occlusion. While the paper focused on the $\texttt{G-SL(AM)}^2$ problem, it is important to note that the approach taken is applicable to a wide variety of dynamic multibody system with intermittent contact.

For future work, we need to solve the "sample degeneracy" problem when the valid subset of state space is tightly constrained. 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 $\texttt{G-SL(AM)}^2$ problem in three-dimensional space. Computational problems must be addressed to facilitate achieving these goals.

REFERENCES

[1] S. Berard. *Using Simulation for Planning and Design of Robotic Systems with Intermittent Contact*. PhD thesis, Rensselaer Polytechnic Institute Department of Computer Science, 2009.

[2] S. Berard, B. Nguyen, K. Anderson, and J.C. Trinkle. Sources of error in a simulation of rigid parts on a vibrating rigid plate. *ASME Journal of Computational and Nonlinear Dynamics*, 5(4):041003–1–041003–14, 2010.

[3] S. Berard, J.C. Trinkle, B. Nguyen, B. Roghani, V. Kumar, and J. Fink. daVinci code: A multi-model simulation and analysis tool for multibody systems. In *Proceedings, IEEE International Conference on Robots and Automation*, pages 2588–2593, April 2007.

[4] M.C. Ferris and T.S. Munson. Interfaces to path 3.0: Design, implementation, and usage. *Journal of Computational Optimization and Applications*, 12(1-3):207–227, January 1999.

[5] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107 –113, April 1993.

[6] S. Haidacher and G. Hirzinger. Estimating finger contact location and object pose from contact measurements in 3d grasping. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, volume 2, pages 1805 – 1810 vol.2, 2003.

[7] L. Han, J.C. Trinkle, and Z. Li. Grasp analysis as linear matrix inequality problems. *IEEE Transactions on Robotics and Automation*, 16(6):663–674, 2000.

[8] Yan-Bin Jia and Michael Erdmann. Pose and motion from contact. *International Journal of Robotics Research*, 18(1):466–490, May 1999.

[9] M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in-hand model acquisition. In *Proceedings, IEEE International Conference on Robots and Automation*, 2010. Workshop on Mobile Manipulation and Best Practices in Robotics.

[10] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Helderman-Verlag, 1988.

[11] D. Prattichizzo and J.C. Trinkle. Grasping. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*. Springer-Verlag, 2008.

[12] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[13] Y. Tassa and E. Todorov. Stochastic complementarity for local control of discontinuous dynamics. In *Proceedings of Robotics: Science and Systems (RSS)*, 2010.

[14] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158, August 2002.

[15] L. Zhang, J. Betz, and J.C. Trinkle. Comparison of simulated and experimental grasping actions in the plane. In *First International Multibody Dynamics Symposium*, June 2010.