

A Dynamic Bayesian Approach to Real-Time Estimation and Filtering in Grasp Acquisition

Li (Emma) Zhang^{*}, Siwei Lyu^{**}, and Jeff Trinkle^{*}

^{*}Department of Computer Science, Rensselaer Polytechnic Institute

^{**}Department of Computer Science, University at Albany, SUNY

Abstract—In this work, we develop a general solution to a broad class of grasping and manipulation problems that we term as *C-SLAM* for contact simultaneous localization and modeling, where the robots need to accurately track the motions of the contacted bodies and the locations of contacts, while simultaneously estimating important system parameters, such as body dimensions, masses and friction coefficients between contacting surfaces. Our solution framework is based on a dynamic Bayesian inference framework, and hence, we refer to it as *Dynamic Bayesian C-SLAM* (DBC-SLAM). DBC-SLAM combines an NCP-based dynamic model with the dynamic Bayesian network, and incorporate model parameter estimation as an intrinsic part of the overall inference procedure. We show two preliminary “proof-of-concept” examples that demonstrate the use of DBC-SLAM in robotic contact tasks.

I. INTRODUCTION

A current weakness of robots is their inability to quickly and reliably perform contact tasks. This is especially problematic in unstructured environments, where the task might be to push jars aside in a refrigerator to reach something behind them or to catch an asteroid or free-floating tool in the International Space Station. To execute contact tasks quickly and reliably, robots need to accurately track the motions of the contacted bodies and the locations of contacts, while simultaneously estimating important system parameters, such as body dimensions, masses and friction coefficients between contacting surfaces.

One can view a broad class of grasping and manipulation problems as analogous to the simultaneous localization and mapping (SLAM) problem that has enjoyed great success in mobile robotics. Here, however, the target application is manipulation rather than mobility, so “localization” refers to tracking the manipulated object relative to the hand. Further, “mapping” is replaced by “modeling,” in the sense that sensor data is used to estimate important uncertain parameters of the system’s dynamic model, rather than constructing a map of the environment. Henceforth, we will refer to such problems as *Contact-SLAM* (*C-SLAM*) problems.

In this work, we develop a general solution to the *C-SLAM* problem that we will refer to as *Dynamic Bayesian C-SLAM* (DBC-SLAM). What distinguishes DBC-SLAM from most previous Bayesian-filtering-based perception methods in robotic contact problems (*e.g.*, [17, 16]) is the incorporation of frictional contact dynamics. This is achieved in DBC-SLAM by combining a *nonlinear complementary program* (NCP)

formulation of a multi-body dynamic model with a hybrid dynamic Bayesian network (DBN) [13] that has continuous and discrete state variables corresponding to the motion and contact states, respectively. The second distinguishing feature of the DBC-SLAM framework is that time varying parameters in the dynamic models (*e.g.*, friction coefficients) are estimated along with motion and contact state tracking.

The paper is organized as following. In Section II, we briefly review relevant backgrounds and previous works. Section III describes in detail of the conceptual framework of DBC-SLAM. In Section IV, we show two preliminary “proof-of-concept” examples that demonstrate the use of DBC-SLAM in robotic contact tasks. Section V concludes the paper with discussion and plans for future work.

II. BACKGROUNDS

A. Multi-body Dynamics

The motions of a dynamic a system of bodies in contact is completely determined by the Newton-Euler differential equation and the kinematic relation:

$$\begin{aligned} M\dot{\mathbf{v}}(t) &= G(t, \mathbf{q}(t); \beta(t), \gamma) \tilde{\lambda}(t) + \tilde{\mathbf{f}}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \tilde{\mathbf{u}}(t) \\ \dot{\mathbf{q}}(t) &= H(\mathbf{q}(t)) \mathbf{v}(t) \end{aligned} \quad (1)$$

where vectors of generalized coordinates $\mathbf{q}(t)$ and $\mathbf{v}(t)$ correspond to the spatial configurations (*e.g.*, position and orientation coordinates of the moveable bodies and coordinates defining body deformations) and velocities, respectively. $\dot{\mathbf{q}}(t)$ and $\dot{\mathbf{v}}(t)$ are their time derivatives. $\tilde{\lambda}(t)$ is the vector of contact forces, $\beta(t)$ corresponds to space/time-varying model parameters, such as friction coefficients, and γ corresponds to space- and time-invariant model parameters, such as body masses. M is the system inertia matrix, G is the Jacobian matrix that maps contact forces into body-fixed frames, and H is the Jacobian matrix that maps the system velocities to the time rates of change of the configuration variables. $\tilde{\mathbf{u}}(t)$ corresponds to the control forces and torques applied by the robot’s actuators, and $\tilde{\mathbf{f}}(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ denotes other external forces in the system such as gravity, Coriolis, and those arising from flexibility.

Since a robot performing manipulation tasks under uncertainty will contact the environment and some of the moveable bodies intermittently, it is important that the dynamic model prevents (virtual) bodies from overlapping and allows contacts to separate. Also, while contact exists between two bodies, the

model must also allow any current contact to switch between sticking and sliding. These features can be enforced through a collection of complementarity conditions [4, 15] that generally take the following form [19]:

$$0 \leq \tilde{\Phi}(t, q(t), \mathbf{v}(t); \beta(t), \gamma) \perp \tilde{\mathbf{z}}(t) \geq 0 \quad (2)$$

where function $\tilde{\Phi}(t, q(t), \mathbf{v}(t); \beta(t), \gamma) \in \mathbb{R}^n$ computes distances between the bodies and relative tangential velocities at the contacts, vector $\tilde{\mathbf{z}} \in \mathbb{R}^n$ contains unknown constraint forces and slide speeds, and the operator \perp implies orthogonality (*i.e.*, $\tilde{\Phi} \cdot \tilde{\mathbf{z}} = 0$). For example, if $\tilde{\Phi}_i$ is the distance between two bodies, then z_i is the normal component of the corresponding contact force. In this case, the complementarity condition $0 \leq \tilde{\Phi}_i \perp \tilde{z}_i \geq 0$ ensures that the bodies do not penetrate ($\tilde{\Phi}_i \geq 0$) and the contact force between them is compressive ($\tilde{z}_i \geq 0$). Through $\tilde{\Phi}_i \cdot \tilde{z}_i = 0$, the following complementary constraints are enforced: when the bodies are separated, the normal component of the contact force must be zero, and conversely, when the contact force is active, the distance between the bodies must be zero.

Let Δt be the sampling time interval and $\ell = 0, 1, \dots, \mathcal{L}$ be the index of the discrete time steps throughout the whole contact task, a discrete-time dynamic model is obtained by replacing $\dot{\mathbf{q}}$ and $\dot{\mathbf{v}}$ with discrete-time approximations (*e.g.*, $\dot{\mathbf{v}}(t) \approx (\mathbf{v}(t + \Delta t) - \mathbf{v}(t))/\Delta t$) in equations (1) and (2), which yields:

$$M^\ell \mathbf{v}^{\ell+1} = G^\ell(\mathbf{q}^\ell; \beta^\ell, \gamma) \lambda^{\ell+1} + \mathbf{f}^\ell + \mathbf{u}^\ell \quad (3)$$

$$\begin{aligned} \mathbf{q}^{\ell+1} &= \mathbf{q}^\ell + H^\ell \mathbf{v}^{\ell+1} \Delta t \\ 0 &\leq \Phi^{\ell+1} \perp \mathbf{z}^{\ell+1} \geq 0. \end{aligned} \quad (4)$$

Equations (3) and (4) represent the discrete-time dynamic model in the form of a *nonlinear complementarity problem* (NCP) [15], the solution of which yields the continuous and discrete states and the contact impulses at time $\ell + 1$. The superscript ℓ corresponds to the discrete time step. \mathbf{q}^ℓ and \mathbf{v}^ℓ correspond to the discrete-time spatial configuration and velocity, and λ^ℓ represents the contact impulse (the integral of the instantaneous contact force $\tilde{\lambda}(t)$ over one time step).

In the following, we will also use the vector \mathbf{x}^ℓ to concisely represent the ensemble of \mathbf{q}^ℓ , \mathbf{v}^ℓ and λ^ℓ in equation (3), which corresponds to all continuous valued states. We will also use a vector σ^ℓ to denote the collection of discrete contact state variables indicating out-of-contact, in-contact sticking, or in-contact sliding. Note that these discrete contact states are implicit in the NCP formulation, as they correspond to the assignment of a zero value to one element in every complementary pair in condition (4). That is, for all $i = 1, \dots, n$, either z_i or Φ_i is set to zero, while the other is constrained to be nonnegative. To simplify notation in later sections, we use vector $\mathbf{i}^\ell = \mathbf{f}^\ell + \mathbf{u}^\ell$ to denote the sum of the input forces to the system in Equation (3). In addition, we use vector \mathbf{o}^ℓ to denote observations from sensors (*e.g.*, joint angle, force, vision, and tactile) that reflect the values of the unobservable motion and contact state variables (\mathbf{x}^ℓ , σ^ℓ). The NCP model parameters include any geometric dimensions and other physical quantities of the robots, moveable objects, and

environment that are relevant to the contact task but not known with sufficient accuracy.

B. Contact Tasks under Uncertainty

To extend the ability of robot hand, the work of Platt *et al.* on planning in belief space incorporates a model of dynamics in a Bayesian estimation process [17]. This method amounts to finding a set of robot control actions that maximizes the probability that a desirable state has been achieved. Belief space planning was used to automatically plan a pushing actions to singulate a box from another and then grasp it. However, the method restricts the model to be continuous, so it does not apply generally to contact tasks.

Several filters have been developed that can estimate the values of continuous and discrete state variables simultaneously. Hebert and Burdick consider the problem of a rectangular block held in a finger-tip grasp [8]. As the arm is moved, an extended Kalman filter estimates the pose of the object relative to the hand and also determines a discrete variable that identifies the pairings of fingertips with box faces, also known as a *contact formation* [20]. The system inputs include stereo images, readings from a wrist force/torque sensor, and finger joint position sensors. However, in this work, no dynamic model was used and the fingertips were assumed to stick to the faces. Also, model parameters (*e.g.*, dimensions of the box) were not estimated. In very similar work in assembly, Bruyninckx *et al.* develop a particle filter for executing a series of compliant motions partitioned by changes in the contact formation [3]. This method simultaneously estimates the pose and velocity of a polyhedral object rigidly held by a robot end effector while it is moved in contact with a known environment. It estimates the gain/loss of contacts, but not the state whether a contact is sticking or sliding. Furthermore, no dynamic model is used, and the filter assumes that contact formations are independent of the object pose and velocity, which is mathematically convenient but physically nonviable.

In a recent work, Zhang and Trinkle [21] design a particle filter to study grasp acquisition in the plane. The filter uses the simulator dVC2d [2] for its dynamic model. It estimates the continuous motion states, several parameters of the dynamic model, and three binary contact states. The main problems with this method are that it uses particles to represent the joint probability distribution over the state space and the filter is designed to use dynamic simulation as a black box. The simulation solves a complementarity problem that grows roughly linearly with the number of contacts and the solution time grows worse than quadratically with the size of the complementarity problem. Therefore the filter is not scalable.

III. METHOD

The central task of the C-SLAM problem is to perform an on-line simultaneous estimation of motion and contact states ($\mathbf{x}^\ell, \sigma^\ell$) and model parameters (β^ℓ, γ) given all observables up to the current time step ($\mathbf{i}^{0:\ell}, \mathbf{o}^{0:\ell}$)¹. In doing so, a solution

¹In the following, for simplicity, we focus on the estimation of the temporally varying parameters and assume that γ is known.

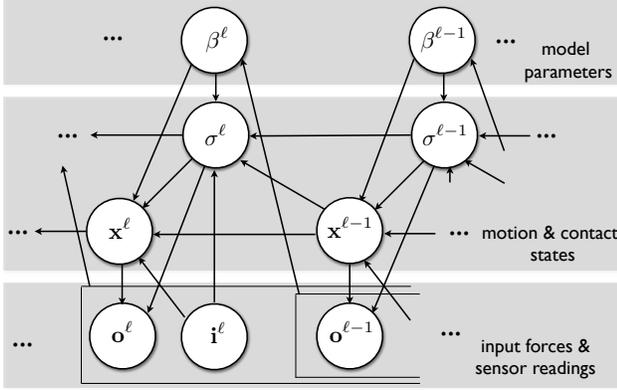


Fig. 1. Graphical structure of DBC-SLAM. For simplicity, we only show models with time/space-varying parameters.

algorithm also has to explicitly handle uncertainties in sensor observations and dynamic model predictions. Physical sensors have limited resolution and measurement inaccuracies due to cost considerations, design deficiencies and manufacturing defects. Similarly, motion and contact states predicted from the discrete-time multi-body dynamic models are imprecise due to modeling errors, and typically viewed as random noise.

In DBC-SLAM, these uncertainties are handled by representing the motion and contact states and the model parameter with their probability distributions given the history of input forces and sensor data, *i.e.*, $p(\mathbf{x}^\ell, \sigma^\ell, \beta^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell})$. These distributions are obtained as conditional posteriors from the DBC-SLAM probabilistic generative model, represented by a joint distribution $p(\mathbf{x}^{0:\ell}, \sigma^{0:\ell}, \beta^{0:\ell} | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell})$, that captures the dynamic relation among all system variables. Represented as a *dynamic Bayesian network* (DBN) [13], this generative model can be factored into a product of several simpler components as:

$$\begin{aligned}
 & p(\mathbf{x}^{0:\ell}, \sigma^{0:\ell}, \beta^{0:\ell} | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell}) \\
 &= p(\beta^0) p(\sigma^0) p(\mathbf{x}^0) \prod_{\ell'=1}^{\ell} p(\sigma^{\ell'} | \mathbf{x}^{\ell'}) p(\beta^{\ell'} | \mathbf{o}^{1:\ell'-1}, \mathbf{i}^{0:\ell'-1}) \\
 &\times p(\sigma^{\ell'} | \mathbf{x}^{\ell'-1}, \sigma^{\ell'-1}, \beta^{\ell'}, \mathbf{i}^{\ell'}) p(\mathbf{x}^{\ell'} | \mathbf{x}^{\ell'-1}, \sigma^{\ell'}, \beta^{\ell'}, \mathbf{i}^{\ell'}),
 \end{aligned} \tag{5}$$

and has a graphical representation as shown in figure 1. $p(\beta^0)$, $p(\sigma^0)$, and $p(\mathbf{x}^0)$ represent the initial uncertainty on the starting motion and contact states and model parameters. The conditional distributions, corresponding to the connectivity patterns in the graphical structure of DBC-SLAM, simplify dependencies among system variables and correspond to various components of DBC-SLAM.

A. Sensor Observations

The conditional distribution $p(\mathbf{o}^\ell | \mathbf{x}^\ell, \sigma^\ell)$ represents the stochastic relation between the actual motion and contact states from their corresponding sensor observations. In an actual system, $p(\mathbf{o}^\ell | \mathbf{x}^\ell, \sigma^\ell)$ may encode direct correspondence between sensor readings to motion states (*e.g.*, measurement of location and velocity, and readings from tactile sensors), or more complex mappings (*e.g.*, collection of image features to identifiable locations on the surface of the manipulated objects). It also indicates that in DBC-SLAM, we assume that

sensor readings across different time steps are independent given the current motion state \mathbf{x}^ℓ .

B. NCP-based Stochastic Dynamic Model

The discrete-time multi-body dynamic model in DBC-SLAM corresponds to the two conditional distributions over motion and contact states, *i.e.*, $p(\sigma^\ell | \mathbf{x}^{\ell-1}, \sigma^{\ell-1}, \beta^\ell, \mathbf{i}^\ell)$ and $p(\mathbf{x}^\ell | \mathbf{x}^{\ell-1}, \sigma^\ell, \beta^\ell, \mathbf{i}^\ell)$. This is equivalent to the assumption that these states are determined from a first-order Markov process, *i.e.*, motion and contact states of the past and the future time steps, $(\mathbf{x}^{\ell-1}, \sigma^{\ell-1})$ and $(\mathbf{x}^{\ell+1}, \sigma^{\ell+1})$, are independent if one knows the current motion and contact states $(\mathbf{x}^\ell, \sigma^\ell)$ and model parameters (β^ℓ, γ) .

However, the introduction of discrete contact states makes a complete specification of the dynamic model challenging, as system with n -dimensional contact state variable leads to $O(3^n)$ individual conditional models for $p(\mathbf{x}^\ell | \mathbf{x}^{\ell-1}, \sigma^\ell, \beta^\ell, \mathbf{i}^\ell)^2$. On the other hand, as mentioned in Section II, the NCP-based dynamic models can encode such exponential number of linear dynamic models using a polynomial number of complimentary conditions. In particular, each simple dynamic model is obtained with a specific assignment of complementary variables to positive or zero values that in turn determines which contact forces and sticking or sliding conditions are included with the Euler-Newton equation.

Nevertheless, one significant drawback of the NCP-based dynamic model is that solving the NCP scales poorly with the number of current and potential contacts, which makes it not suitable for real-time tracking and parameter estimation in the C-SLAM problem [21]. Furthermore, unlike in the situation of planning or simulation, the high prediction accuracy of the NCP solution is not necessary in DBC-SLAM. On the one hand, sensor observations can correct/reduce errors in the predictions of a less detailed dynamic model. On the other hand, the probabilistic nature of DBC-SLAM also means that accurate updates of motion and contact states corresponding to low probability are wasteful since they will have little effect in the subsequent inference.

For these reasons, in DBC-SLAM, we adopt a simpler stochastic dynamic model based on the deterministic NCP-based dynamic model. First, $p(\sigma^\ell | \mathbf{x}^{\ell-1}, \sigma^{\ell-1}, \beta^\ell, \mathbf{i}^\ell)$ is evaluated by softening the hard constraint checking step in the NCP formulation, for instance, the probability of two objects in contact is obtained by converting the output of $\Phi(\mathbf{x}^{\ell-1}, \sigma^{\ell-1}, \beta^\ell, \mathbf{i}^\ell)$, which is defined in the NCP constraints (4), to probability using a sigmoid function. Furthermore, we can also incorporate the spatial and temporal correlations among the contact states in the definition of $p(\sigma^\ell | \mathbf{x}^{\ell-1}, \sigma^{\ell-1}, \beta^\ell, \mathbf{i}^\ell)$. For instance, consider the spatial configuration where a convex polyhedral body with n facets interacts with a planar wall. Even though each vertex can be in any one of the three discrete contact states, they are not independent – if vertex A is in-contact with the wall, then

²This corresponds to the three contact states as “not in-contact”, “in-contact sticking” and “in-contact sliding.”

only vertices sharing a face with A can also be. Furthermore, the contact states at time ℓ have a high probability to be the same as those at time step $\ell - 1$ for most practical scenarios with sufficiently small time step.

The predicted contact states are then applied to the NCP equations, (3) and (4), by specifying strictly positive values in the complementary constraints in the NCP. This yields a simpler dynamic system of linear equations (corresponding to the complementary quantities that *must* now be equal to zero) and inequalities (corresponding to the positive quantities), the solution of which is used to update the continuous motion states. To reflect the inaccurate nature of the predictions made from the reduced system, we inject noise to the updates to construct probability distribution $p(\mathbf{x}^\ell | \mathbf{x}^{\ell-1}, \sigma^\ell, \beta^\ell, \mathbf{i}^\ell)$. Note that the updated motion states may violate constraints in the NCP such as non-penetration between rigid bodies, due to the inaccurate prediction and random effect. However, such violations are normally eliminated over time, as they are usually associated with low prediction probabilities and will further fail to reach sufficient likelihood when validated against the sensor observations.

C. Modeling Temporally Varying Parameters

Conventional dynamic Bayesian modeling usually requires *a priori* knowledge of the model parameters. However, this is not the case for C-SLAM, where a robot could be deployed in a novel environment, and it is essential for the robot to estimate model parameters as it explores.

Many previous works in dynamic Bayesian modeling dealing with the issue of unknown model parameters (*e.g.*, [7, 9, 12, 18]) usually assume the parameters to be invariant temporally, whose estimation can be obtained without requesting the robot to interact with the object. While plausible for parameters such as the geometric dimensions of rigid bodies, there are important types of parameters that defy these assumptions. A case in point would be the effective friction coefficients between contacting bodies.

There exist two simple ways to explicitly estimate temporally varying parameters in a dynamic system. One is to assume parameters of different time to be statistically independent, which simplifies the dynamic estimation of model parameters as they can be performed independently for each time step. However, the complete independence assumption usually leads to sub-optimal estimation, as it ignores dependencies in parameter values of adjacent time steps (*e.g.*, friction coefficients usually change slowly across time, as the underlying materials of contact surfaces tend to be homogeneous within a small spatial area).

Another approach handling temporally varying model parameters introduces an “artificial” dynamic to the parameters across time steps, typically in the form of a simple diffusion process [14] as $\beta^{\ell+1} = \beta^\ell + \eta$, where η corresponds to independent samples from some distribution models (usually Gaussian). The parameter can then be treated similarly as the motion and contact states, where its corresponding artificial dynamics becomes part of the augmented system dynamic

model, and the same Bayesian dynamic inference procedure can be applied to both. The drawback, however, is that the diffusion dynamic usually only accounts for smooth temporal changes in parameters. However, abrupt changes in model parameters abound in cases when there are complex interactions between model parameters and the motion and contact states of the system. Consider again effective friction coefficients, whose temporal values are determined by the physical characteristics of the two contact bodies at their current contact locations, and could change nondeterministically with the spatial locations.

In DBC-SLAM, we take a different approach that the temporally varying model parameters are modeled with the conditional distribution $p(\beta^\ell | \mathbf{o}^{0:\ell-1}, \mathbf{i}^{1:\ell})$ in equation (6), corresponding to the directed edges from observation nodes to the parameter nodes in the graphical representation of DBC-SLAM, figure 1. The underlying justification is that without a precise probabilistic model for these parameters, we only make the minimum assumption that they are *conditionally independent* from each other given the observation and input history $(\mathbf{o}^{1:\ell}, \mathbf{i}^{1:\ell})$. The three different models of parameter dependencies are illustrated in figure 2 for a simplified dynamic system with only motion states \mathbf{x}^ℓ and their corresponding sensor observations \mathbf{o}^ℓ .

Statistical dependencies among parameters in DBC-SLAM are represented by the relation between the consecutive posterior distributions over $p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell})$ and $p(\beta^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell})$. Specifically, if we view $p(\beta^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell})$ and $p(\beta^0)$ as functions of parameter values and denote them as $f^\ell(\beta^\ell)$ and $f^0(\beta^0)$, respectively, $p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell})$ may be specified in one of the following ways:

$$\begin{aligned} p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell}) &= f^\ell(\beta^{\ell+1}), \\ p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell}) &= w^\ell f^\ell(\beta^{\ell+1}) + (1 - w^\ell) f_0(\beta^{\ell+1}), \quad (6) \\ p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell}) &\propto f^0(\beta^{\ell+1})^{w^\ell} f^\ell(\beta^{\ell+1})^{(1-w^\ell)}, \end{aligned}$$

where $w^\ell \in [0, 1]$ is the mixing weight in the last two cases. Specifically, in the first case, the functional form of $p(\beta^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell})$ is reused for $p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{1:\ell})$. In the second and third cases, the prior distribution of the parameters independent of any sensor observations is combined with the current posterior to form the next step predictive distribution. The mixing weight w^ℓ takes values between 0 and 1, which is related with the discrete contact state to reflect whether the parameter distribution should be updated. For instance, for β^ℓ being friction coefficient between two surfaces, w^ℓ will be set to 0 if the two surfaces are not in contact, and subsequently the posterior distribution of $\beta^{\ell+1}$ becomes the “uninformative” prior distribution.

D. Dynamic Bayesian Inference with Particle Filters

With the joint probabilistic generative model in DBC-SLAM, equation (6), dynamic Bayesian inference of motion and contact states and model parameters are formulated as updates of their posterior distributions at each time step given the

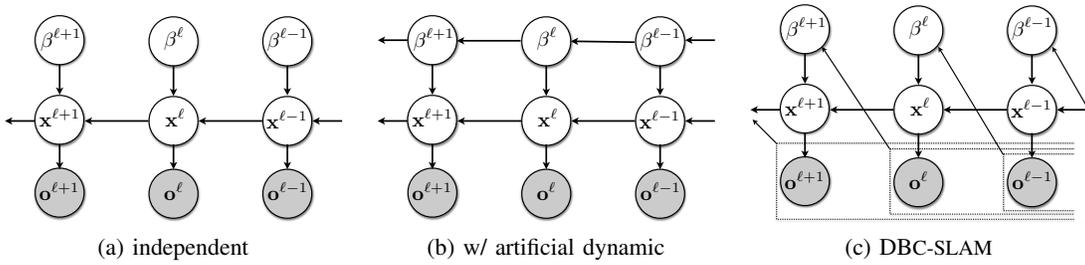


Fig. 2. Illustrations of three different assumptions on the dependencies of parameters for a simplified dynamic system that only has motion states \mathbf{x}^ℓ and the corresponding sensor observations \mathbf{o}^ℓ .

history of input forces and sensor observations, as:

$$p(\mathbf{x}^{\ell+1}, \sigma^{\ell+1} | \mathbf{o}^{0:\ell+1}, \mathbf{i}^{0:\ell+1}) \propto p(\mathbf{o}^{\ell+1} | \mathbf{x}^{\ell+1}) \times \int_{\beta^{\ell+1}} \int_{\mathbf{x}^\ell} p(\beta^{\ell+1} | \mathbf{o}^{0:\ell+1}, \mathbf{i}^{0:\ell+1}) \sum_{\sigma^\ell} p(\sigma^{\ell+1} | \mathbf{x}^\ell, \sigma^\ell, \beta^{\ell+1}, \mathbf{i}^{\ell+1}) p(\mathbf{x}^{\ell+1} | \mathbf{x}^\ell, \sigma^{\ell+1}, \beta^{\ell+1}, \mathbf{i}^{\ell+1}) p(\mathbf{x}^\ell, \sigma^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell}) d\beta^{\ell+1} d\mathbf{x}^\ell.$$

Note that the update step is recursive, in that it relies on the posterior distribution of motion and contact states of the previous time step, $p(\mathbf{x}^\ell, \sigma^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell})^3$. Furthermore, it uses the stochastic dynamic model, the sensor observation model, and the posterior distribution of the model parameters, the last of which is also dynamically updated, as:

$$p(\beta^{\ell+1} | \mathbf{o}^{0:\ell+1}, \mathbf{i}^{0:\ell+1}) \propto p(\mathbf{o}^{\ell+1} | \beta^{\ell+1}) p(\beta^{\ell+1} | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell}). \quad (7)$$

The first term in equation (7), which measures the likelihood of a particular model parameter value generating the sensor observation, is computed using the posterior distributions of the motion and contact states of previous time step, as:

$$p(\mathbf{o}^{\ell+1} | \beta^{\ell+1}) = \sum_{\sigma^{\ell+1}} \int_{\mathbf{x}^{\ell+1}} d\mathbf{x}^{\ell+1} p(\mathbf{o}^{\ell+1} | \mathbf{x}^{\ell+1}) \sum_{\sigma^\ell} p(\sigma^{\ell+1} | \mathbf{x}^\ell, \sigma^\ell, \beta^{\ell+1}, \mathbf{i}^{\ell+1}) \int_{\mathbf{x}^\ell} d\mathbf{x}^\ell p(\mathbf{x}^{\ell+1} | \mathbf{x}^\ell, \sigma^{\ell+1}, \beta^{\ell+1}, \mathbf{i}^{\ell+1}) p(\mathbf{x}^\ell, \sigma^\ell | \mathbf{o}^{0:\ell}, \mathbf{i}^{0:\ell}). \quad (8)$$

The second term in equation (7) can be computed using the approaches represented by equation (6).

Implementing the dynamic Bayesian inference steps in DBC-SLAM presents some special challenges, mainly due to the non-smoothness of the dynamic model, the very high-dimensional state space, and the unknown spatially varying parameters in the dynamic model. In particular, the dynamic model in DBC-SLAM has intrinsic non-smoothness manifested as impulsive contact forces and discontinuous velocities due to gain or loss of contact and transitions between sticking and sliding at sustained contacts. Such a dynamic model precludes the use of Bayesian filtering methods assuming simple linear dynamic models (e.g., Kalman filters [11]) or differentiable nonlinear dynamic models (e.g., extended Kalman filters [10]), and renders particle filters [1] a more effective approach. For a particular DBC-SLAM setting, the dimensionality of joint motion and contact state space could be further reduced by

³Due to the space limit, we omit the detailed derivation of this and the subsequent results, which will be included in an extended version of the current work.

identifying a part of the state space whose distributions can be represented in closed form, and Rao-Blackwellization can be applied to reduce the number of variables requiring particle approximations [6].

Implementation of the dynamic estimation of the temporally varying parameters can usually be further simplified. On the one hand, we can still use particles to represent their posterior distributions. On the other hand, in many practical cases, such model parameters may be well described with a low-dimensional probability model that affords dense sampling. For instance, value ranges of effective friction coefficients corresponding to different materials can be obtained off line to form the prior model $f^0(\beta)$. Such models can then be represented by dense samples in the parameter space, and their updating steps implemented as a grid method – in the updating step, particles corresponding to the parameters are not regenerated each time step, but only their weights are recalculated based on equations (8) and (6) using the particle representations of the motion and contact state variables. Furthermore, in special cases when we only need to choose between individual values of the parameters, the estimation of parameters can be implemented precisely, where the probabilities in equations (8) and (6) are updated directly.

IV. EXPERIMENTS

In this section, we demonstrate two “proof-of-concept” applications of the DBC-SLAM framework.

A. Point-mass Moving on Planar Surface

Imagine a scenario in which a robot is to refinish a tabletop (see Figure 3(a)), where some portions of the surface (corresponding to the smoother regions in the figure) have a low effective friction coefficient ($\mu_2 = 0.3$) and the other portions have a high effective friction coefficient ($\mu_1 = 0.9$). The robot’s main task is to identify rough regions by sliding its hand along the surface to estimate the effective friction coefficients, so that later on it can apply extra sanding only where needed.

We can abstract the robot’s end effector as a point mass driven by a time-varying force \mathbf{u}^ℓ , which motion restricted to a half-space with a linear boundary. The essential task of the robot, which is to determine the effective friction coefficients (choosing between μ_1 and μ_2) at all positions along the linear edge, is a C-SLAM problem. Note that the correspondence between spatial locations and friction coefficients can only be

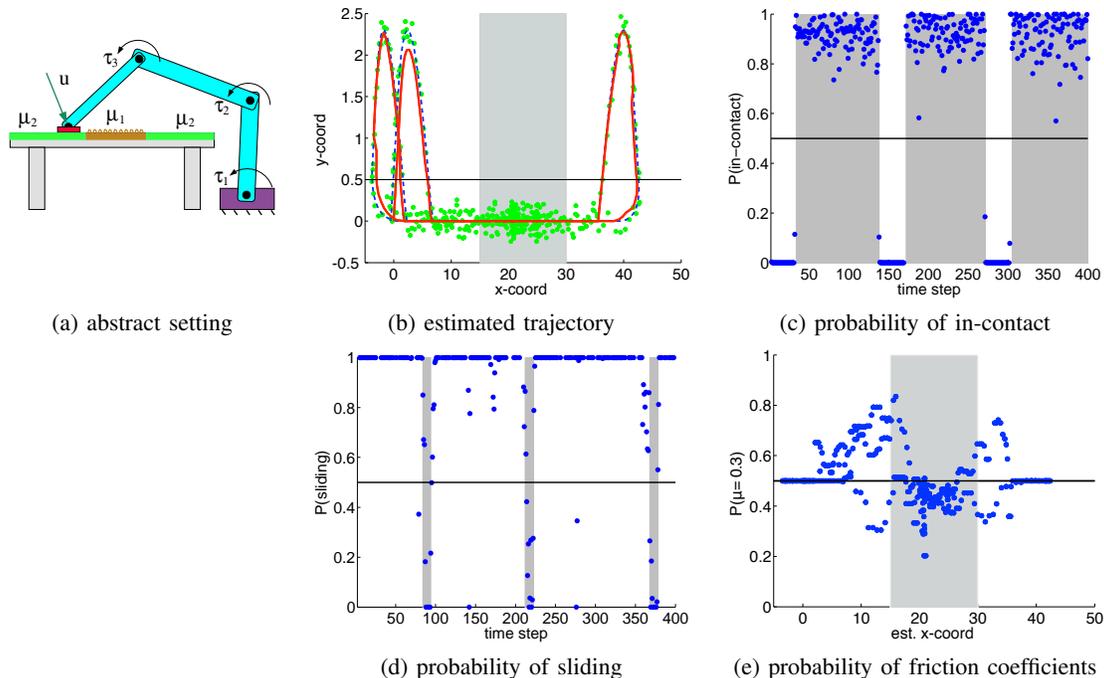


Fig. 3. Experiments of DBC-SLAM for a table polishing task. **(a)** Set-up of the contact problem. **(b)** Estimation of motion trajectories using DBC-SLAM. The red line is the estimated trajectory of the end effector. The shaded area corresponds to the high friction region. **(c)** Estimation of probability of contact with the surface. The shaded area corresponds to when there is contact in the actual trajectory. **(d)** Estimation of probability of the point-mass sliding on the surface. The shaded area corresponds to when the point-mass sticks on the surface in the actual trajectory. **(e)** Estimation of the probability that the effective friction coefficients is 0.3 using DBC-SLAM, indicating contact in the smooth regions. The shaded area corresponds to the rough region, where the friction coefficient is 0.9.

obtained if the robot can estimate both while exploring the environment. Thus this is a C-SLAM problem.

To test DBC-SLAM, we first generated actual trajectories of the point mass using an NCP-based physics simulation engine `dVC2d` [2]. These trajectories represent different modes of the motion of the point mass – the point mass would lose contact near both ends of the edge, and while in contact with the edge, would slide to a halt in the rough region and then, as tangential forces increased, would transition back to sliding. We then simulated the noisy observations from an attached location sensor on the robot by injecting white Gaussian noise to the actual trajectories. One example of the actual trajectory and its corresponding noisy observations are shown in Figure 3(b) as dashed blue curves and green dots, respectively. The shaded area is the vertical projection of the portion of the linear edge with high friction coefficient.

We then applied DBC-SLAM to the observed trajectory to simultaneously estimate the spatial locations (x - and y -coordinates) and the corresponding friction coefficient. In this problem, the dynamic Bayesian inference in DBC-SLAM is implemented with a Rao-Blackwellized particle filter, where the posterior distribution over discrete contact states and the two possible friction coefficients are represented in closed-form and precisely. The posterior distribution over parameters given the history of input forces and sensor observations are updated as a weighted mixture of the prior distribution and the posterior of last time step, corresponding to the second equation in (6).

Figure 3(b) shows the DBC-SLAM estimated point mass trajectory (as the solid red curve) along with the actual trajectory and its noisy sensor observations. Figure 3(c) shows the instantaneous probabilities of the point mass being in the in- or out-of-contact state with the edge, and Figure 3(d) shows the instantaneous probabilities of the point mass being sticking or sliding. Figure 3(e) shows the estimated probabilities of effective friction coefficient taking the value $\mu_2 = 0.3$ at the corresponding estimated x coordinate.

We would like to highlight several points on these results. First, notice that despite large observation noise, DBC-SLAM tracks the simulated motion and the corresponding discrete contact and sliding/sticking states well, though not perfectly due to the noisy sensor data. However, what cannot be deduced from the plots is that tracking performance is quite poor if the filter does not simultaneously estimate the friction coefficient. Indeed, if we assume the friction coefficient is constant over the edge, then soon after the contact moves into a region with different friction coefficient, the filter diverges. Second, as one might expect, estimation of the friction coefficient is more reliable when the point mass is in contact with the surface. In particular, when the point mass slides into the coarse region and switches to a sticking state, the probability that the friction coefficient is the smaller value (*i.e.*, $\mu = 0.3$) is significantly lower than a random guess (0.5), which is predominantly the case in the central portion of the rough region. Last, poor estimation can happen when the momentum of the point mass is large in comparison to the friction impulses acting to stop

it, which is often the case in this example. This is evidenced by the blue points indicating high probability that $\mu = 0.3$ in the left and right ends of the rough region.

B. Parallel-jaw Gripper

In our second set, we use DBC-SLAM to model the task of a parallel-jaw gripper (one may be found in automatic manufacturing or collecting samples for the *International Space Station*) grasping a triangularly shaped object on a supporting horizontal surface (top left image of Figure 4). For the purpose of modeling, we abstract the parallel jaws of the gripper as two position-controlled planar half-spaces. Additionally, a time-varying force and moment are applied to the triangle, so that as the jaws close, the triangle experiences multiple intermittent contacts and switches between sticking and slipping contact. We assume that the object’s mass is known and uniformly distributed so that its center of mass coincides with its geometric center. Furthermore, the dimension of the triangle is also known to the gripper’s controller. There is friction between the object and the gripper, with the coefficient assumed to be time- and space-invariant ($\mu = 0.4$). The gripper’s controller knows only that the friction value is one of the following three ($\mu \in \{0.1, 0.4, 0.7\}$). For simplicity, we assume that there is no friction between the object and the supporting plane.

As in the previous example, we synthesize the actual trajectory of the triangle’s center using `dVC2d` [2], and provide noise-corrupted trajectories of the triangle to our DBC-SLAM filter. As the jaws were closed, time-varying force and moment functions were applied to the triangle. These caused it to approach the lower jaw first, while rotating and moving in the positive x -direction. Contact at a single vertex was formed initially and then the triangle formed edge contact with that jaw while continuing to slide. The applied force and moment were purposely designed to make the triangle “rattle around” between the jaws before the grasp was secure. There were multiple periods of vertex-jaw and edge-jaw contacts and sticking-slip transition before the jaws closed completely on the triangle.

As introduced in Section III-B, DBC-SLAM takes advantage of the compact representation of the NCP formulation to obviate the need for explicit specification of the multitude of dynamic models. Yet, compared with the first experiment, the main challenge here is the enlarged number of contact modes and corresponding dynamic models as a result of the shape of the manipulated object and the two parallel jaws. In particular, when the vertices of the triangle are in contact with the jaws, each vertex may be in four different contact states relative to the two jaws of the gripper: out-of-contact, sticking, sliding left, or sliding right. Each vertex has 4 contact modes on each gripper. Therefore there are $8^3 = 512$ modes. Some of these can be excluded by geometric constraints (*e.g.*, it is not possible for all three vertexes to be in contact with the same jaw at the same time), but there is still a substantial number of contact states and corresponding dynamic models at each time step that the DBC-SLAM filter has to consider. In

implementation, this problem is alleviated by considering only contact states with probability over a preset threshold. Because at each time step, a large fraction of the discrete contact states have small likelihood, in practice, the DBC-SLAM filter needs only to deal with a significantly reduced set of contact modes.

The plots in Figure 4 show the DBC-SLAM tracking results for this task as red solid curves. The left plot shows the motion of the triangle’s center of mass. The center plot shows triangle’s orientation versus time. Overall, as demonstrated in both plots, the DBC-SLAM filter gives reliable estimations of the triangle’s motion despite significant noise in the observations. Also importantly, note that critical contact state transitions (indicated by non-smooth points on the trajectories) are properly predicted by the DBC-SLAM filter. Last, the plot on the right shows the probability that the friction parameter is 0.4. By the end of the simulation, the filter is essentially sure that the value of the parameter is 0.4 (which is the value used in the dynamic model in `dVC2d`). The relatively slow and non-monotonic convergence to the correct value can be partially explained by switches between sticking and sliding. Sliding gives more useful information about the value of the friction coefficient. This can be seen when the triangle formed sliding contact with one jaw, the probability of the correct parameter value gets closer to one near the end of the trajectory. Finally, notice that DBC-SLAM also achieves improved run times compared to a straightforward incorporation of NCP-based models. With the same number of particles, the former is about four times faster than the latter with an unoptimized MATLAB implementation.

V. DISCUSSION

In this paper, we described a general solution to a broad class of grasping and manipulation problems, which we termed as contact simultaneous localization and modeling (*C-SLAM*) problems. In these problems, the robot needs to accurately track the motions of the contacted bodies and the locations of contacts, while simultaneously estimating important system parameters, such as body dimensions, masses and friction coefficients between contacting surfaces. Our solution framework is based on a dynamic Bayesian inference framework and hence, is referred to as *Dynamic Bayesian C-SLAM* (DBC-SLAM). DBC-SLAM combines an NCP-based dynamic model with a dynamic Bayesian network, and incorporate model parameter estimation as an intrinsic part of the overall inference procedure. We showed two preliminary “proof-of-concept” examples that demonstrated the potential value of DBC-SLAM in robotic contact tasks.

There are several directions in which we would like to extend the current work. In particular, we are interested in designing fast reactive grasp acquisition strategies that use the output of DBC-SLAM. These new strategies will be compared to grasp strategies based on less sophisticated task models, for example, Srinivasa’s quasi-static push-grasping [5]. Our hypothesis (to be tested) is that the improved state information will allow faster push-grasping and will also extend the capture envelop for slow quasi-static push-grasps. We would also like

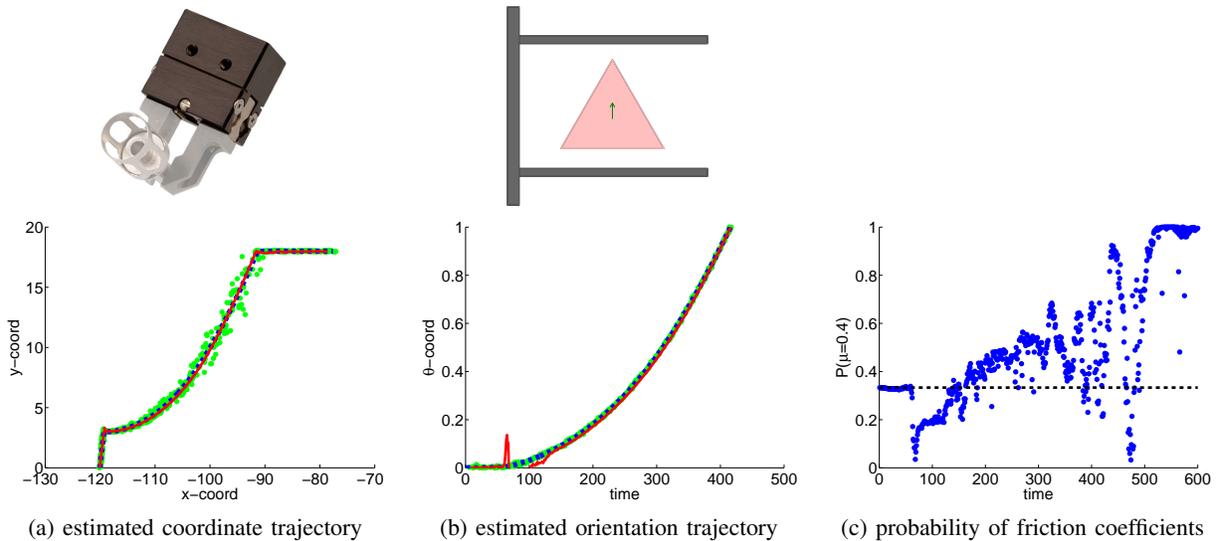


Fig. 4. (top left) A real parallel-jaw gripper and its simplification to the experiment of grasping a triangular object in the plane (top center). (bottom) Estimation results using a prototype implementation of DBC-SLAM. (a) Estimation of motion trajectories of the object's center of mass. (b) Estimation of orientation trajectory. (c) Estimation of the probabilities that the effective friction coefficient takes on its true values of 0.4.

to improve the basic DBC-SLAM framework based on experimental results, and extend it to handle approximate geometric models obtained directly from CAD data or constructed from sensor data (e.g., the tactile sensors of the *Barrett Hand* or a *Kinect*). Last, we would like to analyze human-robot collaborative assembly tasks requiring the use of secure multi-handed grasps of large objects by a robot hand and one or more human hands. This could be done after the task is complete using DBC-SLAM in an off-line “smoothing” configuration. The estimated motion, contact state, and model parameters could be used for improved task models for others to use and for automatic “coding” of the task into qualitative segments such as grasp, pick-up, release, and insert.

REFERENCES

- [1] M.S. Arulampalam, S. Maskell, and T. Gordon, N.; Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [2] S. Berard, J.C. Trinkle, B. Nguyen, B. Roghani, V. Kumar, and J. Fink. daVinci code: A multi-model simulation and analysis tool for multi-body systems. In *Proceedings, IEEE International Conference on Robots and Automation*, pages 2588–2593, April 2007.
- [3] H. Bruyninckx, J. De Schutter, T. Lefebvre, K. Gadeyne, P. Soetens, J. Rutgeerts, P. Slaets, and W. Meeussen. Building blocks for SLAM in autonomous compliant motion. In *International Symposium on Robotics Research*, 2003. URL <http://people.mech.kuleuven.be/~wmeeusse/Papers/slam.pdf>.
- [4] R. W. Cottle, J.S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, 1992.
- [5] M.R. Dogar and S.S. Srinivasa. Push-grasping with dexterous hands: Mechanics and a method. In *Proceedings, IEEE International Conference on Intelligent Robots and Systems*, October 2010. accepted.
- [6] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.
- [7] W. R. Gilks and C. Berzuini. Following a moving target - monte carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society, Series B*, 63(1):127–146, 2001.
- [8] P. Hebert, N. Hudson, J. Ma, and J. Burdick. Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location. In *Proceedings, IEEE International Conference on Robots and Automation*, pages 5935–5941, 2011.
- [9] T. Higuchi. Monte carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, 59(1):1–23, 1997.
- [10] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, 3, 1997.
- [11] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.
- [12] G. Kitagawa. Non-Gaussian state space modeling of nonstationary time series (with discussion). *Journal of the American Statistical Association*, 82(400):1032–1063, 1987.
- [13] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.
- [14] J. Liu and M. West. Combined parameters and state estimation in simulation-based filtering. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 197–223. New York: Springer-Verlag, 2001.
- [15] K.G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Helderman-Verlag, 1988.
- [16] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake. A hypothesis-based algorithm for planning and control in non-Gaussian belief spaces. Technical Report CSAIL Tech Report MIT-CSAIL-TR-2011-039, MIT, 2011. URL http://groups.csail.mit.edu/robotics-center/public_papers/Platt11c.pdfYear=2011.
- [17] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake. Efficient planning in non-Gaussian belief space and its application to robot grasping. In *International Symposium on Robotics Research*, 2011. URL http://www.isrr-2011.org/ISRR-2011/Program_files/Papers/Platt-ISRR-2011.pdf.
- [18] G. Storvik. Particle filters in state space models with the presence of unknown static parameters. *IEEE Transactions on Signal Processing*, 50(2):281–289, 2002.
- [19] J.C. Trinkle, J.S. Pang, S. Sudarsky, and G. Lo. On dynamic multi-rigid-body contact problems with Coulomb friction.

tion. *Zeitschrift für Angewandte Mathematik und Mechanik*, 77 (4):267–279, 1997.

- [20] Jing Xiao and Xuerong Ji. On automatic generation of high-level contact state space. *International Journal of Robotics Research*, 20(7):584–606, 2001.
- [21] L. Zhang and J.C. Trinkle. The application of particle filtering to grasp acquisition with visual occlusion and tactile sensing. In *Proceedings, IEEE International Conference on Robots and Automation*, 2011. accepted.