daVinci Code: A Multi-Model Simulation and Analysis Tool for Multi-Body Systems

Stephen Berard, Jeff Trinkle, Binh Nguyen, and Ben Roghani
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180
Email: {sberard, trink, nguyeb2, roghab}@cs.rpi.edu

Jonathan Fink and Vijay Kumar
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104
Email: {jonfink, kumar}@grasp.upenn.edu

***Abstract*— This paper discusses the design and current capabilities of a new software tool, dVC, capable of simulating planar systems of bodies experiencing unilateral contacts with friction. Since different problems require different levels of accuracy, dVC provides user-selectable body types (rigid or locally-compliant), motion models (first-order, quasi-static, dynamic), and several state-of-the-art time-stepping methods. One can also choose to include friction between each body and the plane of motion. To support optimal and robust part design, dVC also allows on-the-fly changes to parameters of the geometric and physical models. The results obtained for three representative planar problems are presented: the design of a passive part-orienting device, the planning of a meso-scale assembly operation, and the design of a grasp strategy.**

I. Introduction

This paper discusses a new software tool, dVC, that was designed to facilitate simulation, analysis, and virtual design of multibody systems with intermittent unilateral contacts with dry friction. Understanding the dynamics of such multibody systems is central to many fundamental problems in robotics including grasping, manipulation, walking and assembly. However, these systems are notoriously difficult to simulate accurately due to the nonsmooth nature of the underlying mathematical model. In fact, commercial software for simulating multibody systems with unilateral contacts and dry friction (DADs, Adams, Working Model) deal with the nonsmoothness by *ad hoc* regularization methods (*e.g.,* penalty methods to remove unwanted interpenetration of bodies by local nonlinear spring and damper effects). These methods require tuning of simulation parameters and the algorithms often have no guarantees on stability and convergence.

In contrast, dVC uses state-of-the-art time-stepping methods designed to capture the nonsmooth phenomena (stick-slip transitions and contact loss and formation) without regularization. In these methods, each time-stepping subproblem is formulated as a complementarity problem [1]–[3]. These methods are numerically stable and provably convergent. In particular, it has been proven that when the Stewart-Trinkle (ST) and Song-Pang-Kumar (SPK) methods are applied to dynamic systems, as the step size goes to zero, the trajectories produced by the time-steppers converge to the exact solution of the original instantaneous-time model. This convergence property has been available for solvers for ordinary differential equations and differential algebraic equations for some time, but only recently for some constraint-based time-stepping methods. Further, work in progress shows that under reasonable modeling assumption, it is possible to obtain sensitivity information on the trajectories, which is critical if we want to use the simulator for developing planning algorithms or for design optimization.

dVC allows the user to choose between different motion models. It allows comparison and validation of multibody modeling choices (*e.g.*, one chooses to use a quasistatic model, but should have chosen a dynamic model). Second, it enables a hierarchical and iterative approach to design processes. As an example, consider the design of a part feeder (also discussed later). Geometric and dynamic parameters are critical to the functioning of the device and there are many parameters (e.g., coefficients of friction) that are characterized by uncertainty. We would like to be able to consider only the geometry initially to prune the design space eliminating large sets of geometric parameters that render the design infeasible. This can be done by simple geometric models. We then want to be able to refine the design space with quasi-static models, and then using more expensive dynamic models. Thus we can use low-resolution models to quickly eliminate many design alternatives using more complicated models characterized by many unknown parameters to search over a smaller set of feasible designs.

II. Simulation Overview

dVC consists of several modular components (plugins) that have been written in such a way that they can be extended and replaced at runtime based on the desired configuration. Figure 1 gives a high level view of the components of dVC.

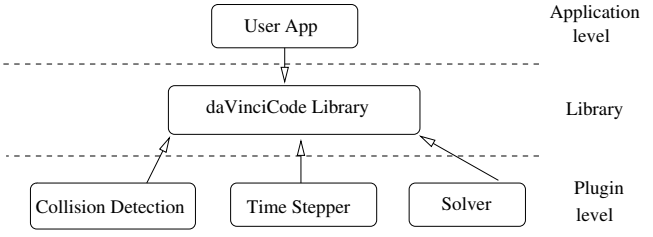


Fig. 1. daVinci Code Architecture.

The main function of the library is to provide the *API* (Application Programming Interface), which is our defined set of calling conventions. The library acts as the central controller organizing calls to the plugins, which in turn do the actual physics simulation work.

The *Collision Detection* module fulfills the collision detection requirements of the currently active time-stepping plugin. In addition, it implicitly defines a geometric model; dVC currently supports both polygonal and implicit body surfaces. The *TimeStepper* plugin formulates the equations of motion of all bodies in the system as a complementarity problem. The *Solver* plugin solves the complementarity problem. During simulation, an example time step consists of the following actions:

1) call collision detection plugin with the current scene
2) call the time-stepper plugin with the results of the collision detection plugin
3) call the solver plugin with the complementarity problem formulated in the time-stepper plugin
4) Update the bodies with the result of the solver plugin

dVC includes two collision detection plugins. The first plugin is a modified version of the PQP collision detection library (Section III-B) and is suitable for polygonal surfaces. The second plugin is appropriated for implicit surfaces and uses a super-ellipse collision detection algorithm [4].

Step 2 includes a *collision response model* that requires the determination of the exact time of impact. There are two ways to determine this "time-of-impact." In the first method, the time-stepper plugin uses the collision detection results, and if penetration has occurred the module can "back-up" the simulation time and decrease the step size. The second technique for implementing a collision response is for the time-stepper plugin to analyze the "active constraint set" (defined in Section III-B) and adaptively change the step-size prior to collision. Once the time-of-impact has been determined, any desired impact law can be applied. This flexibility in allowing various implementations of the plugins is one of the main benefits of dVC as an analysis tool.

Table (I) lists the various time stepping models, body types, and motion models implemented. dVC supports both 2D (Section II-A) and 2.5D (Section II-B) problems. There are several important similarities and differences between these models. The Anitescu-Potra (AP) and Stewart-Trinkle (ST) time-steppers assume rigid bodies and collisions are inelastic, while the Song-Pang-Kumar (SPK) model assumes that the bodies are locally visco-elastic near the contacts. Under the SPK time-stepper, the collision response of colliding bodies is a natural outcome of the integration process, but time steps become extremely small for stiff bodies.

TABLE I

TIME-STEPPING ALGORITHMS AVAILABLE IN DVC

Time-Stepper	*Body Types*	*Motion Models*
Stewart-Trinkle	Rigid	First-Order, Quasi-Static, Dynamic
Anitescu-Potra	Rigid	First-Order, Quasi-Static, Dynamic
Song-Pang-Kumar	Quasi-Rigid	Dynamic

Lastly, since the most robust software available today for solving complementarity problems is PATH [5], it is embedded in our solver plugin (Section III-C).

A. 2D Dynamic Model

The dynamic equations of motion for a planar multi-body system with frictional contacts can be written as:

$$M(q)\dot{\nu} = W_{\mathrm{n}}(q)\lambda_{\mathrm{n}} + W_{\mathrm{t}}(q)\lambda_{\mathrm{t}} + \lambda_{\mathrm{app}}(t,q,\nu) \qquad (1)$$

where $q \in \mathbb{R}^{n_q}$ is the vector of generalized coordinates, $\nu \in \mathbb{R}^{n_\nu}$ is the vector of generalized velocities, $M(q)$ is the inertia matrix, $\lambda_{\mathrm{app}}(t,q,\nu)$ is the vector sum of all generalized non-contact forces, $\lambda_{\mathrm{n,t}}$ are the n_c-dimensional vectors of contact forces in the normal (n) and tangential (t) directions, n_c is the number of (potential) contacts, and W_{n} and W_{t} are the Jacobin matrices that transform the contact forces into equivalent wrench (force and torque) in appropriate frames. Since dVC is intended for simulation of planar systems, $n_q = n_\nu = 3n_b$, where n_b is the number of bodies. Further, we have the following kinematic relationship:

$$\dot{q} = G(q)\nu. \qquad (2)$$

where $G(q)$ is a parametrization matrix relating the system velocity to the system configuration.

Non-penetration at each (potential) contact i of the bodies is enforced by the following complementarity condition:

$$0 \le \lambda_{i\mathrm{n}} \perp \psi_{i\mathrm{n}}(q,t) \ge 0 \qquad (3)$$

where $\psi_{i\mathrm{n}}(q,t)$ is a vector of signed distances between pairs of body features that might come into contact and $\perp$ denotes orthogonality (*i.e.,* $\lambda_{i\mathrm{n}}\ \psi_{i\mathrm{n}}(q,t) = 0$). Note that at least one of $\psi_{i\mathrm{n}}$ and $\lambda_{i\mathrm{n}}$ must be zero for every $i = 1,\ldots,n_c$.

The last part of the model is a friction law that generates forces opposing sliding at contacts and thus dissipates energy. All of our implemented time-stepper plugins assume Coulomb's Law. We denote by $\mathcal{F}(\mu_i\lambda_{i\mathrm{n}})$, the constraints on the frictional force $\lambda_{i\mathrm{t}}$:

$$\mathcal{F}(\mu_i\lambda_{i\mathrm{n}}) \equiv |\lambda_{i\mathrm{t}}| \le \mu_i\lambda_{i\mathrm{n}}. \qquad (4)$$

where μ_i is the coefficient of friction at the i-th contact point. In addition, when sliding occurs, the force must lie on the boundary of the friction cone in the direction opposing sliding:

$$\lambda_{i\mathrm{t}} = \operatorname{argmin}\left\{\dot{\psi}_{i\mathrm{t}}\lambda_{i\mathrm{t}} \ :\ \lambda_{i\mathrm{t}} \in \mathcal{F}(\mu_i\lambda_{i\mathrm{n}})\right\} \qquad (5)$$

where $\psi_{i\mathrm{t}}$ is analogous to $\psi_{i\mathrm{n}}$ and represents the tangential displacement at contact i.

Equations (1-5) represent the basic dynamic model of the systems that dVC can currently simulate. However, there are also three useful variations: the quasi-static model formed by setting $M(q) = 0$, the first-order model formed by replacing $\dot{\nu}$ in equation (1) with $\dot{q}$, effectively creating a generalized damper world, and a compliant model formed by replacing the unknown contact forces λ_{n} and λ_{t} with functions of local deformations and the rates of deformations of the bodies. In the latter case, λ_{n} and λ_{t} are replaced by

unknown deformations, thus the total number of variables does not decrease. Furthermore, the region near the contact point can be discretized into multiple compliant contact regions, significantly increasing the number of variables and equations in the model, but also allowing more accurate model of systems exhibiting material deformation near the contact points (see [3] for details).

B. 2.5D Dynamic Model

There are many planar manipulation and assembly tasks in which we need to model surface friction between planar parts and a planar support surface. While this surface friction results in planar forces, they are dependent on the normal forces or the pressure distribution which contribute an out-of-plane component. The 2.5D model models the out-of-plane normal forces which do not directly play a role in determining the motion in the plane but do influence the frictional forces in the plane. We use the well-known model due to Mason [6] to model dry friction and linear damping models to model viscous friction.

III. DaVinci Code Overview

The next two sections outline the main aspects of dVC.

A. Bodies

In addition to force controlled bodies, dVC also supports position controlled bodies. A position controlled body is similar to an obstacle in that its motion is not affected by forces, however its position over time varies and is specified by a user supplied function.

Similarly, for force controlled bodies, a user can specify a force controller and apply external body forces as a function of time. This flexibility continues, allowing certain degrees of freedom of the body to be position controlled, while others are force controlled.

B. Collision Detection

dVC uses collision detection and distance computation in two different ways. Under the rigid body assumption, the formulation of a time-step problem requires knowledge of points in contact, contact normals, and the same information for points not quite in contact and for points that have penetrated (due to numerical or linearization errors). In the rigid body case, specific times and locations of collisions are not required; nonetheless, the solution of the time-step problem is consistent with the model at the end of the time step. On the other hand, if the bodies are assumed to have compliant surfaces, then distance queries must be used to find precise times of impact between bodies. This is required to properly calculate local deformations and for adapting the time step based on the effective local stiffness. In both cases, dVC maintains an *active constraint set*, which contains the distance and normal information for all geometric feature pairs that could come into contact during the ensuing time-step. This required a modification to the PQP collision detection [7] to determine all potential contact points, their relative connectivity, and the depth of penetration (for compliant models).

C. Complementarity Problem Solver

For simulation, a solution of the continuous-time model given by equations (1-5), is approximated at specific times by a time-stepping method. Replacing time derivatives by variable differences divided by the time step and after some algebraic manipulation, one arrives at a family of a time-stepping subproblems in the form of complementarity problems [8]. One possible formulation as a mixed linear complementarity problem (mixed LCP) is presented in the next section. dVC has implemented a wrapper plugin that uses the well-known PATH solver [5], [9] to solve the subproblems. In addition to solving the standard LCPs, the PATH solver can also solve mixed linear and nonlinear complementarity problems. PATH is closed source, but has a freely downloadable binary available from [10].

IV. Time-Stepping Formulations

We briefly discuss the development of time-stepping algorithms. The superscript ℓ denotes the value of the appropriate variable at time t^ℓ. The dynamic equations of motion (1) are discretized using an implicit, one-step scheme[1]:

$$M\nu^{\ell+1} = M\nu^{\ell} + h\left(W_{\mathrm{n}}^{\ell}\lambda_{\mathrm{n}}^{\ell+1} + W_{\mathrm{f}}^{\ell}\lambda_{\mathrm{f}}^{\ell+1} + \lambda_{\mathrm{app}}^{\ell}\right) \quad (6)$$

$$q^{\ell+1} = q^{\ell} + h\nu^{\ell+1} \quad (7)$$

The normal contact condition equation (3) can be linearized as follows:

$$0 \le p_{\mathrm{n}}^{\ell+1} \perp \psi_{\mathrm{n}}^{\ell} + \frac{\partial \psi_{\mathrm{n}}^{\ell}}{\partial q}\Delta q + \frac{\partial \psi_{\mathrm{n}}^{\ell}}{\partial t}\Delta t \ge 0 \quad (8)$$

where $p_{\mathrm{n}} = h\lambda_{\mathrm{n}}$, $\Delta q = q^{\ell+1} - q^{\ell}$ and $\Delta t = h$. Note that $W_{\alpha}^{\mathrm{T}} = \frac{\partial \psi_{\alpha}}{\partial q}$, for $\alpha \in \mathrm{n}, \mathrm{f}$.

In the plane, Coulomb's Law at a contact can be written as two linear complementarity conditions. Collecting the two conditions for all the contacts yields the following pair of conditions:

$$0 \le p_{\mathrm{f}}^{\ell+1} \perp E\sigma^{\ell+1} + \frac{\partial \psi_{\mathrm{f}}^{\ell}}{\partial q}\Delta q + \frac{\partial \psi_{\mathrm{f}}^{\ell}}{\partial t}\Delta t \ge 0 \quad (9)$$

$$0 \le \sigma^{\ell+1} \perp Up_{\mathrm{n}}^{\ell+1} - E^{\mathrm{T}}p_{\mathrm{f}}^{\ell+1} \ge 0 \quad (10)$$

where $p_{\mathrm{f}} = h\lambda_{\mathrm{f}}$, U is a diagonal matrix containing the coefficients of friction $U = diag(\mu_1, \ldots, \mu_{n_c})$, E is a block diagonal matrix, $E = diag(e_1, \ldots, e_{n_c})$ with e_i given as a column vector of length 2 with both elements equal to 1, and σ is the vector of sliding speeds at the contacts.

Equations (6,8,9,10) represent a mixed LCP that can be written in compact form as:

$$\begin{bmatrix} 0 \\ \rho_{\mathrm{n}}^{\ell+1} \\ \rho_{\mathrm{f}}^{\ell+1} \\ s^{\ell+1} \end{bmatrix} = \begin{bmatrix} -M & W_{\mathrm{n}} & W_{\mathrm{f}} & 0 \\ W_{\mathrm{n}}^{\mathrm{T}} & 0 & 0 & 0 \\ W_{\mathrm{f}}^{\mathrm{T}} & 0 & 0 & E \\ 0 & U & -E^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \nu^{\ell+1} \\ p_{\mathrm{n}}^{\ell+1} \\ p_{\mathrm{f}}^{\ell+1} \\ \sigma^{\ell+1} \end{bmatrix} + \begin{bmatrix} M\nu^{\ell} + hf \\ \frac{\psi_{\mathrm{n}}^{\ell}}{h} + \frac{\partial \psi_{\mathrm{n}}^{\ell}}{\partial t} \\ \frac{\partial \psi_{\mathrm{f}}^{\ell}}{\partial t} \\ 0 \end{bmatrix} \quad (11)$$

[1]Since the tangential direction t is unconstrained, we use the subscript f where we have introduced slack variables representing both the positive and negative components of t. This means a vector with a f subscript has a dimension twice as large as its corresponding t vector, and the wrench matrices have twice as many columns.

$$0 \leq \begin{bmatrix} \rho_{\mathrm{n}}^{\ell+1} \\ \rho_{\mathrm{f}}^{\ell+1} \\ s^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} p_{\mathrm{n}}^{\ell+1} \\ p_{\mathrm{f}}^{\ell+1} \\ \sigma^{\ell+1} \end{bmatrix} \geq 0 \tag{12}$$

where $\rho^{\ell+1} = \psi^{\ell+1}/h$. Inserting the value of $\nu^{\ell+1}$ obtained in the solution of this mixed LCP into equation (7) yields the updated value of the system configuration $q^{\ell+1}$.

a) Visco-Elastic Contacts: While the above time-stepping method provides accurate simulation for a large class of problems, it does not handle well impact dynamics with restitution or phenomena such as micro-slip. In [3] a model appropriate for such situations was derived and a time-stepping method analogous to that above was developed. In this method, the surfaces near each contact are discretized and sets of coupled lumped springs and dampers are inserted between pairs of interior points (in the rigid core of the body) and boundary points (on the compliant surface of the body) on the discretized contact patch. The introduction of the compliant layer on the body surfaces causes the dynamic equations to become numerically stiff. Consequently, when contacts exist in the system, the size of the time step must be reduced significantly, by many orders of magnitude when the springs are very stiff. In addition, this method requires the time when unilateral contact first occurs to be determined accurately. To accommodate these needs, we use an adaptive time step adjustment developed in [11], which sets the step size as a function of the derivative of the gap function. These requirements motivate our philosophy to include a range of models and time-stepping methods in dVC.

b) 2.5D formulation: The formulation for the 2.5D model incorporates additional elements corresponding to the tangential surface frictions and normal friction force vector.

$$\begin{bmatrix} 0 \\ \rho_{\mathrm{n}}^{\ell+1} \\ \rho_{\mathrm{f}}^{\ell+1} \\ s^{\ell+1} \end{bmatrix} = \begin{bmatrix} -M & W_{\mathrm{n}} & W_{\mathrm{f}} & 0 \\ W_{\mathrm{n}}^{\mathrm{T}} & 0 & 0 & 0 \\ W_{\mathrm{f}}^{\mathrm{T}} & 0 & 0 & E \\ 0 & \hat{U} & -E^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \nu^{\ell+1} \\ p_{\mathrm{n}}^{\ell+1} \\ p_{\mathrm{f}}^{\ell+1} \\ \sigma^{\ell+1} \end{bmatrix} + \begin{bmatrix} M\nu^{\ell} + hf \\ \frac{\psi_{\mathrm{n}}^{\ell}}{h} + \frac{\partial \psi_{\mathrm{n}}^{\ell}}{\partial t} \\ \frac{\partial \psi_{\mathrm{t}}^{\ell}}{\partial t} \\ \hat{p} \end{bmatrix} \tag{13}$$

$$0 \leq \begin{bmatrix} \rho_{\mathrm{n}}^{\ell+1} \\ \rho_{\mathrm{t}}^{\ell+1} \\ s^{\ell+1} \end{bmatrix} \perp \begin{bmatrix} p_{\mathrm{n}}^{\ell+1} \\ p_{\mathrm{f}}^{\ell+1} \\ \sigma^{\ell+1} \end{bmatrix} \geq 0 \tag{14}$$

where $W_{\mathrm{f}} \in \mathbb{R}^{3\times 2n_c+3n_d}$ has two friction directions for each planar contact and n_d directions for each surface contact, $\hat{U} = \begin{bmatrix} U & 0 \end{bmatrix}$, $\hat{p} = \begin{bmatrix} 0 \\ U_k p_{\mathrm{n}k} \end{bmatrix}$. The vector $p_{\mathrm{n}k} \in \mathbb{R}^3$ contains the support points' normal impulses, and $U_k \in \mathbb{R}^3$ is the matrix of support point friction coefficients. Lastly, $p_{\mathrm{f}} \in \mathbb{R}^{2n_c+n_d3}$, $\sigma \in \mathbb{R}^{n_c+3}$ and $E \in \mathbb{R}^{2n_c+3n_d\times n_c+3}$ are extended to incorporate the surface friction.

V. Simulated and Experimental Results

To illustrate the capabilities of dVC, several example systems are presented with experimental results. The first example describes how dVC can be used to help design dynamical systems with intermittent frictional contacts and uncertainty in the model. In the second example, dVC was used to design open loop plans for a planar micro-manipulation task. In the final example, dVC was used to analyze grasping strategies for a planar part; choosing strategies where success is likely even when "a guaranteed strategy does not exist" [12].

A. Design of a Part Reorienting Device

This example is taken from a parts feeding application [13]. It is a device with a cavity of complex geometry (as shown in Figure 2) designed to orient a cup-shaped part. Regardless of the part's initial orientation in the top chute, after it falls through the cavity it must enter the lower vertical chute with its center of gravity down.

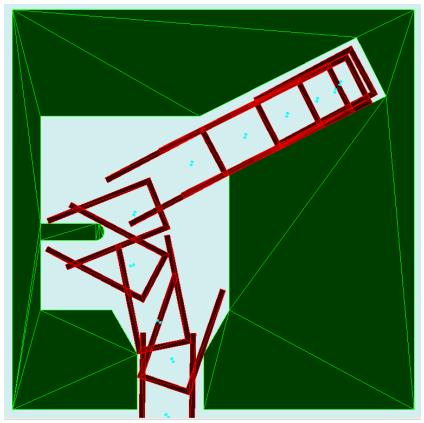


Fig. 2. Snapshots of the gravity-fed part in the feeder.

The problem consists of a design parameter space P and an uncertainty space M. Design parameters are parameters we have control over (in this case, 12 parameters) that define the geometry of the device. The uncertainty space exists because models are not perfect and certain physical properties (like friction coefficients) are difficult to measure and predict. The state of the system $x(t)$ was a function of: t, x_0, p, and m, where $p \in P$, $m \in M$ and x_0 is the initial condition. The goal was to find some $p \in P$ that works for all $m \in M$.

The solution approach was to randomly sample P for a design that was geometrically feasible (motion planning), then thoroughly sample M to verify the design. The dVC simulator lends itself nicely to this verification system through its modular design. Using dVC, a hierarchical algorithm was written and easily implemented:

1) Verify with geometric model
2) For all $m_i \in M_\mu$, verify with inelastic ST model
3) For all $m_i \in M$, verify with SPK model

where the M_μ uncertainty space consists of the unknown initial cup orientation and friction coefficients and M adds the additional unknown parameters arising from the locally deforming viscoelastic frictional contacts.

This hierarchical planning through the use of dVC allowed for a significant savings in design time, since the simpler model could be used to prune away failed designs without the need of testing with the more accurate, but also more computationally expensive model. Table II displays the running times of the verification algorithms under the naive approach, and our hierarchical approach for the geometric

and inelastic ST models. We saw of savings of 1462 seconds using our hierarchical planner because we did not run the more computationally expensive dynamic model on designs where no geometric solution existed.

TABLE II

HIERARCHICAL APPROACH TO DESIGN

	Naive Verification	Hierarchical Verification
Geometric	N/A	603 s
ST	5000 s	2935 s
Total	5000 s	3538 s

B. Needle pushing a planar slider

The second experiment is an assembly problem taken from a planar micro-manipulation experiment [14]. Figure 3 presents 2 pictures of the experimental setup as seen through a microscope. The goal in this problem is to find a pushing path for the needle (labeled probe in figure 3(a)) to push the block from the initial configuration to a goal configuration (shown in figure 3(b)). dVC was used to simulate candidate pushing plans from a space of possible pushing plans to find one capable of accomplishing the task.

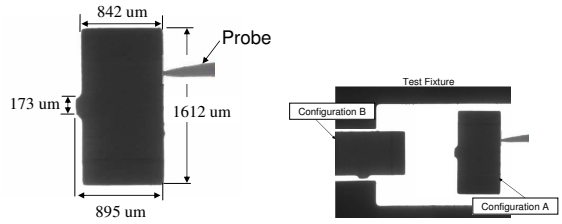


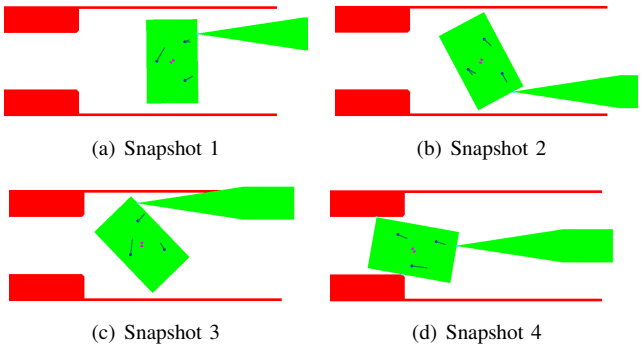
Fig. 3. Images taken from a microscope of the experimental micro part and assembly. The left image shows the dimensions of the part and the right image shows the initial (A) and goal (B) configurations.

Since friction in the plane of motion was important and inertial forces were an order of magnitude smaller than the frictional due to dimensions, for this problem we used a 2.5D quasi-static model, with a position-controlled trajectory of the needle. The locations of the support points and coefficients of friction were identified through experimentation. Figure 4 illustrates several frames of simulation. The support points are the 3 small dark circles inside the peg, and the lines extending from them are a visualization of the friction force components resisting the motion of block.

Figure 5 shows a comparison of the y component of the simulated and experimental trajectories. The path of the peg determined from the simulated pushing path matches closely when experimentally verified. The other components of the trajectory also match closely. The flexibility of dVC allows us to choose the desired physical model and also to use it as an open loop path planning tool. For videos of this example see [15].

C. Probabilistic Grasp Planning

In this example, dVC was used to simulate an earlier grasping experiment [12]. The goal of the experiment was to study the probability of success of a given grasping strategy under uncertainty in the friction forces between the gripper



(a) Snapshot 1 (b) Snapshot 2

(c) Snapshot 3 (d) Snapshot 4

Fig. 4. Four screen-shots from the simulated planar micro-manipulation task. The small dark circles inside the peg are the support points shown with corresponding tangential friction force vectors.

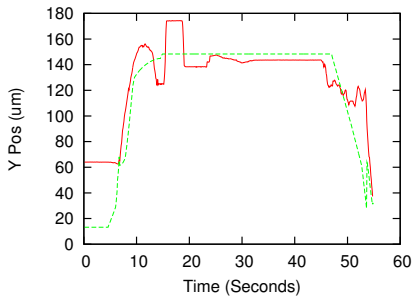


Fig. 5. Comparison of the y component of the 2.5D simulated and experimental trajectories.

and part and between the part and the support surface. The task is to grasp a planar part with a parallel-jaw gripper. The part and gripper are shown in figure 6, as well as the coordinate system used in the simulation. The part is slightly more than 40 mm in diameter. The midway point between the tips of the fingers is measured with respect to the center of the part (Figure 6). The goal is to estimate the probability of a successful grasp as a function of x and y.

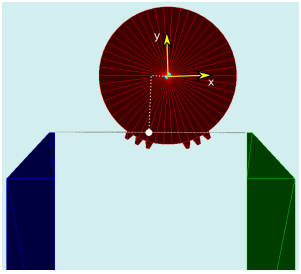


Fig. 6. The parameters x and y describe the position of the center point between the fingertips, relative to the center of the gear.

Similar to the parts reorienting device example, this example also contains a large uncertainty space: the location of the support points of the part, coefficient of surface friction, and coefficient of gripper friction. The location of the part in the world was known exactly. Similar to the micro-manipulation experiment, friction in the plane of motion is again important, but here we cannot ignore the inertial forces. For this problem we used a 2.5D dynamic model, with force-controlled parallel jaw grippers. Then for each (x_i, y_i) in a

20mm by 20mm box of initial positions (sampled at 1mm resolution), assign random values to the unknown parameters and simulate the grasp. Estimate the success probability for each (x_i, y_i) pair as the total number of successful grasps divided by the number of trials. The goal configuration (as defined in [12]) is when the tips of the fingers lie on the outer edges of the outer-most teeth.

Figures 7 and 8 illustrate two typical simulation results. The figures are 2D projections of the 3D histogram of successful grasps. These figures were created assuming a fixed support point tripod location on the lock, and selecting random values for the coefficients of friction in the specified ranges. Figure 7 is the result for medium surface contact friction coefficients and medium gripper contact friction coefficients. Figure 8 is the result for medium surface contact friction coefficients and high gripper contact friction coefficients. For the presented results, coefficients of friction from the medium group range in values between: (0.33, 0.66], and high: (0.66, 0.99].

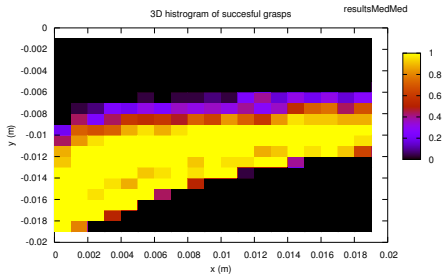


Fig. 7. 2D map of the 3D histogram of successful grasps for medium surface friction and medium gripper friction coefficients.

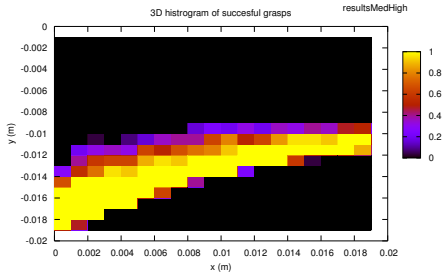


Fig. 8. 2D map of the 3D histogram of successful grasps for medium surface friction and high gripper friction coefficients.

Our initial results agree well with the experimental results presented in [12]. Current work focuses on developing a better understanding of the friction coefficient influences, developing better sampling strategies for the support point tripod, incorporating other physical models and developing hierarchical planning techniques similar to example 1.

VI. Conclusion and Future Work

We presented the first release of our configurable multi-body simulator, dVC, which is capable of simulating planar systems with multiple bodies in steady and/or intermittent unilateral frictional contact. It utilizes a plugin style architecture, allowing users to vary the collision detection, time-stepping, and solver methods on a per problem basis. Our immediate goals are to continue addressing two-dimensional manipulation problems. We are working on an order h^2 integration scheme [16] and a method that accurately formulates the time-stepping subproblems when the C-space is not locally convex [17].

VII. Acknowledgments

We thank Todd Munson (Argonne National Lab) for his assistance with the PATH solver. We thank David Cappelleri (Penn) for the experimental data for the micro-manipulation task, Thierry Ameil (Penn) for collecting the real-world feeder experimental data, and Patrick Marion (RPI) for the improvements to dVC. We thank Randy Brost for his encouragement and insightful discussions on the probabilistic grasp planning problem. This project was partially funded by NSF grants 013970, 0413227, and 0420703.

References

[1] M. Anitescu and F. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.

[2] D. Stewart and J. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *Int'l Jrnl. of Numerical Methods in Engineering*, vol. 39, pp. 2673–2691, 1996.

[3] P. Song, J.-S. Pang, and V. Kumar, "A semi-implicit time-stepping model for frictional compliant contact problems," *Int'l Jrnl. for Numerical Methods in Engineering*, vol. 60, pp. 2231–2261, 2004.

[4] N. Chakraborty, J. Peng, J. Mitchell, and S. Akella, "Proximity queries between convex objects: An interior point approach for implicit surfaces," in *Int'l Conference on Robotics and Automation*, 2006.

[5] M. C. Ferris and T. S. Munson, "Complementarity problems in GAMS and the PATH solver," Department of Computer Science, Univeristy of Wisconsin, Madison, Tech. Rep. MP-TR-1998-12, 1998.

[6] M. T. Mason, "Manipulator grasping and pushing operations," Ph.D. dissertation, Massachusetts Institute of Technology, June 1982, reprinted in *Robot Hands and the Mechanics of Manipulation*, MIT Press, Cambridge, Massachusetts, 1985.

[7] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," Department of Computer Science, University of N. Carolina, Chapel Hill, Tech. Rep. TR99-018, 1999.

[8] R. W. Cottle, J. Pang, and R. E. Stone, *The Linear Complementarity Problem*. Academic Press, 1992.

[9] S. Billups, S. Dirkse, and M. Ferris, "A comparison of large scale mixed complementarity problem solvers." *Computational Optimization and Applications*, vol. 7, pp. 3–25, 1997.

[10] M. C. Ferris, "Cpnet software," 2005, [accessed 20-Sept-2005]. [Online]. Available: http://www.cs.wisc.edu/cpnet/cpnetsoftware

[11] J. M. Esposito, V. Kumar, and G. J. Pappas, "Accurate event detection for simulating hybrid systems," in *Proc. of the 4th Int'l Workshop on Hybrid Systems*. London, UK: Springer-Verlag, 2001, pp. 204–217.

[12] R. C. Brost and A. D. Christiansen, "Probabilistic analysis of manipulation tasks: A conceptual framework," *Int'l Jrnl. of Robotics Research*, vol. 15, no. 1, pp. 1–23, Feb. 1996.

[13] P. Song, J. Trinkle, V. Kumar, and J. Pang, "Design of part feeding and assembly processes with dynamics," in *Int'l Conf. on Robotics and Automation*, 2004.

[14] D. J. Cappelleri, J. Fink, and V. Kumar, "Modeling uncertainty for planar meso-scale manipulation and assembly," in *ASME Int'l Design Engineering Technical Conference*, 2006.

[15] S. Berard, J. Trinkle, B. Nguyen, B. Roghani, J. Fink, and V. Kumar. [Online]. Available: http://www.seas.upenn.edu/~jonfink/davinci/peginhole

[16] F. Potra, M. Anitescu, B. Gavrea, and J. Trinkle, "A linearly implicit trapezoidal method for stiff multibody dynamics with contact, joints and friction," *Int'l Jrnl. for Numerical Methods in Engineering*, 2006.

[17] K. Egan, S. Berard, and J. Trinkle, "Modeling concave constraints using linear complementarity," Rensselaer Polytechnic Institute Department of Computer Science, Tech. Rep. 03-13, Dec. 2003.