

KANE'S EQUATIONS FOR HAPTIC DISPLAY OF MULTIBODY SYSTEMS

R. Brent Gillespie

University of Michigan
Department of Mechanical Engineering
1015 Beal Ave.
Ann Arbor, MI 48109
email: brentg@umich.edu

ABSTRACT

The design of a general-purpose multibody simulator that runs in real-time and features haptic display is presented. The repertoire of this simulator includes systems subject to holonomic constraints, nonholonomic constraints, and notably, systems subject to changing constraints. In contrast to the use of generic equations of motion, Kane's method is used by a computerized symbolic manipulator to produce custom-built, compact, and computationally efficient ordinary differential equations for each virtual system to be simulated. A method is presented whereby new equations, reflecting the presence of a transient constraint, may be formulated on-line by a recombination of terms comprising the nominal equations. This paper also serves as a tutorial introduction to Kane's method and includes a detailed derivation of Kane's equations with a geometric interpretation. An illustrative example featuring changing constraint conditions is developed in the body of the paper while two more examples, chosen to highlight certain features within Kane's method, are presented at the conclusion.

1 INTRODUCTION

To render a system of virtual objects that responds dynamically when touched or manipulated through a haptic interface requires real-time integration of that virtual system's equations of motion. For some systems, generic equations of motion, say for a single body or an open chain, can be adapted to the target dynamics by setting parameters, but such an approach usually severely limits the family of virtual objects that can be simulated. Instead, customized equations can be formulated using a variety of techniques, and a particular technique might be carried out either by hand with pencil and paper, or by a computer equipped with symbolic manipulation or specialized numerical routines. Naturally,

computerized production of the equations of motion is desirable when the target system is large or complex. This paper presents the design of a general-purpose multibody simulator with haptic display, one that automatically formulates the equations given a system description (geometry, interconnections, inertia properties, and constitutive laws for springs, dampers, or control elements). Once formulated, the governing equations are passed to a real-time solver that couples the virtual system to a human user through the action of motors and sensors in a haptic interface.

Of the available techniques for formulating equations of motion for multibody systems, Kane's method was chosen for its relative ease of computerization and its computational efficiency. Efficiency may be interpreted here both as producing equations efficiently (with the fewest symbolic operations) and producing efficient equations (which require the fewest numerical operations for their solution). Also, Kane's method produces equations in ordinary differential form (ODEs) even for nonholonomically constrained systems, which can be accommodated using standard solvers.

The label 'general-purpose', when applied to a simulator, refers to the relative ease with which a new virtual system may be described, set-up, and run by the simulator to address the needs or desires of the user/analyst. Within this 'general' scope, however, attention shall be restricted in this paper to multibody systems, or systems of rigid bodies interconnected by various joints but also bodies variously making or breaking contact with one another. The inclusion of target systems whose bodies collide, slide, roll, or sometimes merely rest on one another is a significant feature, since a human user's haptic exploration is often aimed at inciting such changing contact conditions.

Under the hood of any multibody simulator is one of several methods for holding interconnected bodies together or maintaining the pertinent constraint conditions. Satisfaction of the

constraints can be guaranteed either by embedding the constraint equations into the equations of motion (*i.e.*, eliminating dependent variables) or by applying the appropriate constraint forces, computed as part of the simulation. Alternatively, violation of the constraints can be permitted to some degree and so-called penalty forces applied to ensure that such violation is minimal. These penalty forces may be computed by multiplying the maximum interpenetration between bodies by a constant and their action may be interpreted as that of virtual springs. Naturally, these various methods for maintaining constraints can be mixed within a given simulator design. Perhaps the constraints associated with the joints of a mechanism that will not be broken during simulation would be embedded while the constraints associated with a rolling or sliding contact that are subject to change during simulation would be treated by application of constraint forces or the penalty method. For example, in the simulator in (Haug et al., 1986) and more recently (García de Jalón and Bayo, 1994), certain holonomic constraints are embedded using Lagrange's equations, while nonholonomic constraints and constraints subject to change during runtime are handled using the method of undetermined multipliers. Changing constraint conditions are reflected by swapping out the constraint equations, which are appended to the invariant core equations. Baraff (Baraff, 1989), (Baraff, 1994) has shown that constraint force computation can be posed as a linear complementarity problem (LCP), for which efficient numerical techniques have been developed. Even the multibody dynamics and impulse resolution problems can be posed as LCPs and thus combined in a common architecture with computation of the constraint forces. Thus in the simulator design by Baraff (Baraff, 1995), certain holonomic constraints are embedded using Lagrange's method, while those subject to change are handled by constraint force computation. Rather than imposing the forces of constraint, Mirtich (Mirtich and Canny, 1994) applies repeated impulses, sized to hold the constraints. Even a block resting on a table experiences many repeated tiny collisions rather than a constraint force.

Concentrating on simulators that feature haptic rendering, the design by (Constantinescu et al., 2000) and (Cadoz et al., 1984) use the penalty method to compute constraint forces. These simulators, however, do not feature any constraint embedding or multibody methods, rather they use generic Newton-Euler equations for each body separately. The simulator by Ruspigi (Ruspini and Khatib, 2001) and further elaborated in (Khatib et al., 2002) and the simulator by Berkelman (Berkelman et al., 1999) follow the Baraff design and also feature haptic display. The Mirtich method has been adopted by (Chang and Colgate, 1997). The simulator by Gillespie (Gillespie, 1993), (Gillespie, 1994) features constraints that are fully embedded. Changing constraints in this simulator are handled by swapping the entire equations of motion, passing final conditions as initial conditions at each exchange.

Rather than computing constraint forces as an auxiliary process to the forward dynamics problem, the work in (Nahvi et al.,

1998) uses an inverse dynamics formulation. The resulting force, computed in response to sensed motion, is rendered by a haptic interface to allow a user feel constraints imposed by a virtual mechanism whose motion is fully prescribed by the haptic interface motion. However, if objects within the virtual environment have internal degrees of freedom or are not fully constrained by their connection to the haptic interface, then a forward dynamics model must be simulated, at least to compute the certain motions. A topic not covered in detail here, but related to the choice of a forward or inverse dynamic model for simulation is the choice of haptic rendering approach: whether impedance display or admittance display. Forces are displayed and motion sensed by the haptic interface in impedance display while motion is displayed (often using an inner control loop) and forces sensed in admittance display. Most haptic interface devices are by design better force sources than motion sources, so impedance display is more common, the work of Yoshikawa (Yoshikawa et al., 1995) and MacLean (MacLean and Durfee, 1995) being exceptions. A convenient link between a haptic interface using impedance display and a forward-dynamics simulation is provided by the virtual coupler (Brown and Colgate, 1997). Forces for haptic display and equal and opposite forces for imposing on the forward dynamics model are computed by multiplying the vector difference of the image of the operational point of the haptic interface and the point of contact with the virtual environment by a spring constant (and perhaps multiplying velocity by a damping constant, together which form the virtual coupler). A second advantage of the virtual coupler approach is the clean separation it imposes between hardware and simulator, which allows a coupled stability analysis to be undertaken. In particular, the sampled data passivity analysis may be reduced to a discrete time passivity analysis (Colgate, 1994), (Brown and Colgate, 1997) (Colgate and Schenkel, 1997), (Miller et al., 2000) .

The simulator presented in this paper uses embedded constraint handling. The simulator is outfitted with a set of symbolic manipulation routines that produce a new model, custom-tailored to each new target system. In this way, the structural properties of each target system can be incorporated to the fullest extent to yield compact equations. Just before runtime, the equations are wrapped in a numerical integration routine to yield the simulation code. The present system employs either AUTOLEV (Schaechter, 1988), (Kane and Levinson, 1999), (Autolev, 2003) or our own toolbox in Matlab/MAPLE to perform the symbolic algebra. Constraints that are subject to change, however, are optionally not embedded until run-time as will be shown below. Briefly, constraint equations are formulated on-line for each change in contact condition within the system and embedded in the model (used to reduce the model to its minimal realization) in real-time. There are two major components of this technique: the automated selection of generalized speeds, developed by Reckdahl, and efficient constraint embedding, first presented by Wampler et al. (Wampler et al., 1985) and further refined by Mitiguy (Mitiguy, 1995). In this paper, these components are

assembled into a complete method for real-time constraint embedding.

The presentation of this simulator is intended to serve in part as an introduction to Kane's method for the haptics community. A step-by-step discussion of the model construction process is included and references to the Lagrange formalism are made throughout to increase the tutorial value of the paper. Designers of virtual environments who are familiar with the Newton-Euler and Lagrange methods, but have perhaps only heard of Kane's method, make up the intended audience for this paper.

The body of this paper is organized around the following three steps for the creation of a virtual environment simulator:

Step 1, presented in section 2: A mathematical model is constructed by naming variables and parameters. Initial conditions are assigned to the variables as appropriate and values are assigned to the parameters according to the material or geometric properties for which those parameters stand.

Step 2, discussed in section 3: Kinematic, kinetic and mass distribution quantities are formed and these quantities are manipulated according to a principle of mechanics to formulate the equations of motion.

Step 3, discussed in section 4: The equations of motion are incorporated into an interactive simulator with haptic display. Constraint conditions, if transient, may be embedded on-line in an automated fashion.

An illustrative example featuring a changing constraint is developed stepwise in each of the above sections, and two more examples are presented in Section 5. One of these examples demonstrates the influence of the choice of model variables on the compactness of the resulting equations of motion, and the other example demonstrates various methods for embedding constraints.

2 STEP 1: MODEL CONSTRUCTION

To begin the formulation of the equations of motion, one chooses a set of parameters and variables to describe certain mass distribution, kinematic, and kinetic quantities in the target system. Choosing parameters and variables sounds deceptively simple, but the decisions made here exert a strong influence on the eventual form and numerical efficiency of the equations of motion. This section introduces the variables used in Kane's equations, in particular generalized speeds, and compares them to those used in Lagrange's equations.

Before treating the parameters and variables used in formulating the equations of motion, mention must be made of a second major part of the system description, the geometric model. Whereas the equations of motion (which might be called the kinematic/dynamic model) are differential equations in terms of configuration and motion variables that locate reference points and reference frames, the geometric model is an algebraic description of the boundary of each body with respect to its reference point and reference frame. While the kinematic/dynamic

model is solved through numerical integration to evolve the system configuration forward in time, the geometric model is handled by a collision detector to determine when and at what points bodies within the system collide. Upon detecting a collision, integration must be halted, impulses resolved, new constraints imposed if appropriate, and integration re-started with the appropriate initial conditions.

This paper will concentrate on the construction of the kinematic/dynamic model and its use within a differential equation solver for simulation. Neither the geometric model nor the collision detector that makes use of the geometric model will be considered in detail. Instead, this paper assumes that a mechanism to detect changing contact conditions is already in place. The literature contains presentations of several specialized geometric models and associated collision detectors. Perhaps most successful are the Lin/Canny algorithm (Lin and Canny, 1991), (Lin and Manocha, 1997) and the GJK algorithm (Gilbert et al., 1988) (Gilbert and Foo, 1990) (Ong and Gilbert, 1997), (Ong and Gilbert, 2001). The Lin/Canny algorithm has been adapted for use with haptic rendering (Gregory et al., 2000) (Gregory et al., 1999). For good reviews on the issue of collision detection, see (Lin et al., 1996) and (Jimenez et al., 2001). See also (Hollerbach, 2000) and (Mark et al., 1996) for reviews with focus on haptic interface to virtual environments.

2.1 Kinematic/Dynamic Model

To construct the kinematic/dynamic model for a system S , one chooses variables to describe the configuration of S , or variables that specify the location of a reference point and orientation of a reference frame fixed within each body of S . These variables may be called *configuration variables*. In Lagrange's method, one uses *generalized coordinates*, commonly labelled q_i ($i = 1, \dots, n$), as configuration variables. Generalized coordinates specify the location of reference points and orientation of reference frames defined for S relative to one another rather than all relative to a single origin and single Newtonian reference frame. The generalized coordinates describe only the allowed configurations and thus encapsulate certain *configuration constraints*. For example, if a revolute joint connects two bodies, only the joint angle is needed to describe the configuration of the second body if the configuration of the first is known. Compared to a set of variables that specifies the location and orientation of each body relative to a common ground, a set of generalized coordinates is reduced in number i.e., $n < 6v$, where v is the number of bodies of S and spatial motion is being considered. If M is the number of configuration constraints encapsulated by the choice of generalized coordinates, then $n = 6v - M$.

A set of generalized coordinates is usually found by inspection. In some cases, as in the case of closed kinematic chains, however, an independent set is difficult to find. It is then convenient to carry along a configuration constraint equation as part of the system description (or to carry along a differentiated configuration constraint i.e., motion constraint, as elaborated below).

Lagrange's equations are second order differential equations in the generalized coordinates q_i ($i = 1, \dots, n$). These may be converted to first-order differential equations or into state-space form in the standard way, by defining an additional set of variables, called *motion variables*. To convert Lagrange's equations, one defines the motion variables simply as configuration variable derivatives, sometimes called *generalized velocities*. Then the state vector is made up of the configuration and motion variables: the generalized coordinates and generalized velocities. Table 1 shows the configuration and motion variables used in Lagrange's method in the first column.

In Kane's method, generalized coordinates are also used as configuration variables. However, the motion variables in Kane's equations are defined as *functions* that are linear in the configuration variable derivatives and in general nonlinear in the configuration variables. The use of such functions can lead to significantly more compact equations, as will be demonstrated in an example in section 5 below. The name given to these new motion variables is *generalized speeds* and the symbol commonly used is u , as shown in the second column of Table 1.

More explicitly, each generalized speed is defined by the analyst as a linear combination of the generalized coordinate derivatives:

$$\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = Y \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} + Z \quad (1)$$

where the elements of the $(n \times n)$ matrix Y and $(n \times 1)$ matrix Z are functions of q_i ($i = 1, \dots, n$) and possibly the time t . Reciprocal relations express the generalized coordinate derivatives in terms of the generalized speeds:

$$\begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} = W \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} + X \quad (2)$$

where the $(n \times n)$ matrix W and $(n \times 1)$ matrix X are functions of q_i ($i = 1, \dots, n$) and possibly the time t . The expressions of the generalized coordinate derivatives in terms of the generalized speeds (Eqs. 2) are called *kinematical differential equations* and form the first part of the state equations (equations of motion). The equations of motion will take on a particularly compact (and thus computationally efficient) form with the effective use of generalized speeds.

For certain physical reasons, a second type of constraint equation may arise, restricting not the allowable configurations that the system may occupy, but restricting the types of motion that it may undergo. These are called motion constraints¹. Just

as configuration constraints may be encapsulated within the definition of a set of generalized coordinates, *motion constraints* may be encapsulated in the definition of a set of generalized speeds. By judicious choice of the functions defining the generalized speeds in terms of the generalized coordinate derivatives and generalized coordinates, the motion constraint equations can be automatically satisfied. Thus if a system is subject to m motion constraints, the number of generalized speeds is $p = n - m$. In Kane's equations, the motion constraints may be fully embedded, yielding ordinary differential equations in independent variables. Whereas in Lagrange's method, if a motion constraint exists, it must be carried along as part of the system description. Guaranteeing the satisfaction of a motion constraint in Lagrange's method is then handled using the method of Undetermined Multipliers.

Motion constraints may be conveniently expressed as functions among a non-minimal set of generalized speeds. Note also that differentiated configuration constraints may be considered motion constraints; such treatment of configuration constraints is often convenient. Motion constraints that arise for physical reasons (e.g., rolling constraints) may be expressed in Pfaffian form, as a linear relationship among the generalized speeds and are called *simple* motion constraints by Kane. Thus, simple motion constraints may be expressed

$$A \cdot B \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} + C = 0 \quad (3)$$

where the elements of the matrices $A \cdot B$ and C are functions of the generalized coordinates and time.

Given m motion constraints, there will be m dependent generalized speeds in the set u_r ($r = 1, \dots, n$), which may be expressed in terms of the remaining $p = n - m$ independent generalized speeds. After selecting which generalized speeds are to be considered independent and re-ordering (a process which will be considered again in Section 4) Eq. (3) can be re-arranged as an expression for the m dependent generalized speeds in terms of the p independent generalized speeds:

$$A \begin{bmatrix} u_{p+1} \\ \vdots \\ u_n \end{bmatrix} = B \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} + C. \quad (4)$$

One must choose the A and B matrices so that Eq. (4) may be

in favor of the roughly equivalent terms 'holonomic constraint' and 'nonholonomic constraint' in this paper so that a differentiated holonomic constraint (which is still properly referred to as a holonomic constraint) may be grouped with nonholonomic constraints under one heading. Simply put, configuration constraints are expressed as equations involving configuration variables and motion constraints are expressed as equations involving motion and configuration variables.

¹The terms 'configuration constraint' and 'motion constraint' have been used

Table 1. Options For The State Variables

M : # configuration constraints; m : # motion constraints;

n : # independent generalized coordinates; $p = n - m$: # independent generalized speeds.

	Lagrange's method	Kane's method
Configuration Variables	generalized coordinates $q_i (i = 1, \dots, n)$	generalized coordinates $q_i (i = 1, \dots, n)$
Motion Variables	generalized velocities $\dot{q}_i (i = 1, \dots, n)$	generalized speeds $u_i (i = 1, \dots, p)$
number of constraints encapsulated	M	$M + m$

solved uniquely for the dependent generalized speeds and the results written in terms of $D = A^{-1}B$ and $E = A^{-1}C$ as:

$$\begin{bmatrix} u_{p+1} \\ \vdots \\ u_n \end{bmatrix} = D \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} + E, \quad (5)$$

where the elements of D and E are functions of $q_i (i = 1, \dots, n)$ and possibly the time t . Now that the motion constraints have been expressed in an explicit linear form, the construction of the reduced set may be completed by substitution. This process is called embedding the motion constraints.

One more note regarding generalized speeds is in order. The definition of the set of generalized speeds is up to the analyst. The definition of the generalized speeds can have a profound effect on the compactness of the equations of motion, although the effect does not become apparent until the equations are actually formulated in Step 2. Mitiguy (Mitiguy and Kane, 1996) has developed a set of directives for the optimal choice of generalized speeds for systems with open chains interconnected by revolute, Hooke's, or ball-and-socket joints.

To summarize Table 1, two choices have been presented for the set of variables for a kinematic/dynamic model. Although in both sets, M configuration constraints have been encapsulated by the choice of $n = 6v - M$ generalized coordinates, differences exist in the encapsulation of m motion constraints. Column 1, with n generalized velocities, carries along m motion constraint equations. Column 2 uses p generalized speeds $u_r (r = 1, \dots, p)$ where $p = n - m$. Thus column 2 embeds both the configuration and the motion constraints and forms what may be considered a set of *independent variables*.

Example 1 Figure 1 shows a planar system comprising a uniform disk B of radius r , mass m , and central moment of inertia J in contact with a ramp A fixed in a Newtonian reference frame N . Let the horizontal line X and vertical line Y , both fixed in N , intersect at the point O on the edge of A and use ϕ to designate the angle between X and the edge of A .

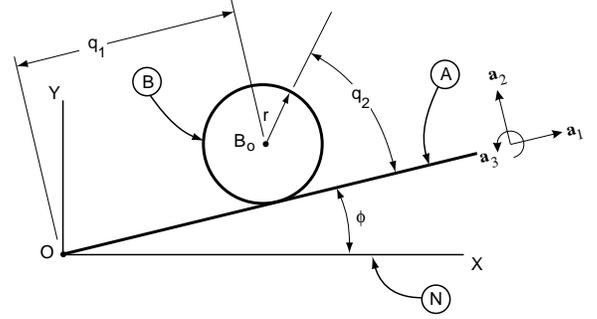


Figure 1. A rolling disk

To characterize the configuration of B in N , the X - Y coordinates x and y of the mass center B_o of B may be used, along with the angle q_2 between \mathbf{a}_2 and a line fixed in B that is initially parallel to the ramp edge. However, if B is to remain in contact with A , the coordinate set $\{x, y, q_2\}$ satisfies a configuration constraint: $x \cos(\phi) - y \sin(\phi) = r$. After introducing mutually perpendicular unit vectors $\mathbf{a}_1, \mathbf{a}_2$, and \mathbf{a}_3 fixed in A , with \mathbf{a}_1 parallel to the edge of A , \mathbf{a}_3 normal to the X - Y plane and $\mathbf{a}_2 = \mathbf{a}_3 \times \mathbf{a}_1$ as shown in Figure 2, this configuration constraint may be expressed $\mathbf{p}^{OB_o} \cdot \mathbf{a}_2 = r$, where \mathbf{p}^{OB_o} is the position vector from O to B_o . If we define $q_1 \triangleq \mathbf{p}^{OB_o} \cdot \mathbf{a}_1$, then the set of generalized coordinates $\{q_1, q_2\}$ encapsulates the configuration constraint.

For motion variables, the generalized velocities \dot{q}_1 and \dot{q}_2 are used in Lagrange's formulation, whereas in Kane's equations generalized speeds can be defined as linear functions of the generalized coordinate derivatives. Although for this example no particular advantage is available by defining the generalized speeds as anything different than the generalized velocities, we shall, for purposes of exposition, define $u_1 \triangleq \dot{q}_1$ and $u_2 \triangleq r\dot{q}_2$.

If B slides on A , then both u_1 and u_2 are independent variables. However, if B rolls on A , a motion constraint² may be

²When rolling takes place in a plane, as in this example, the rolling constraint is not nonholonomic; it can be integrated to form a holonomic constraint, and then used to reduce the set of generalized coordinates down to a single independent member. For rolling in 3D, a rolling constraint is nonholonomic and cannot be used to express dependencies among the generalized coordinates, only among the

written: $u_1 + u_2 = 0$. When the rolling constraint is enforced, the equations may be written in terms of a single generalized speed (but still two generalized coordinates), a process which will be taken up for this example in the next sections.

Although one rarely bothers to write down the matrices Y, Z, W, X, D , and E defined in Eqs. (1), (2), and (5) when working Kane’s method by hand, they are generally stored in memory by a computerized (symbolic manipulator) implementation of Kane’s method. For this example, they are:

$$Y = \begin{bmatrix} 1 & 0 \\ 0 & r \end{bmatrix}; Z = [0]; W = \begin{bmatrix} 1 & 0 \\ 0 & 1/r \end{bmatrix}; X = [0], \quad (6)$$

and when the constraint is imposed:

$$D = [-1]; E = [0], \quad (7)$$

which will become useful in Section 4 below.

2.2 Handling of Constraints

Let us further discuss the handling of constraints given their relevance to changing contact conditions. In Table 1, generalized coordinates and generalized speeds were introduced along with methods for defining them in such a way that all constraints (both configuration and motion constraints) may be embedded. The result is a set of independent variables, a *minimal* state description which can be used to produce a particularly compact set of equations of motion.

Now it is time to re-consider whether a minimal description is actually desired. In fact, some constraints may be subject to change during the period of interest and should therefore not be embedded, at least not prior to run-time. Instead, it would be useful to pass a dependent set of state variables accompanied by a set of possibly transient constraint expressions on to the simulator construction Steps 2 and 3. The transient constraints are then handled, by embedding or by other means, during runtime, where the various factors affecting their transience are available. For this purpose, the exposition of steps 2 and 3 below will include the development of an on-line method for embedding constraints in the equations of motion.

In addition to embedding constraints, another operation must be automated: the selection of the independent generalized speeds, for each time a constraint comes into play, the set of generalized speeds that may be considered independent is subject to change. Usually, the selection is made by the analyst, but for handling changing constraints, this task must be given to the runtime code. Additionally, singular configurations may occur at certain times during a simulation, producing ambiguities in the

generalized speeds. It is always possible, however, to treat a holonomic constraint as if it were nonholonomic, that is, to treat it as a *motion* constraint, and that is the approach taken in this example.

state description. Although generalized coordinates and generalized speeds can usually be chosen with foresight to avoid singular configurations, occasionally one would rather handle singularities by re-defining the set of independent generalized speeds or generalized coordinates during simulation. The automated selection of independent generalized speeds to be developed in Section 4 supports changing constraints and additionally enables the accommodation of singularities.

3 STEP 2: FORMULATE EQUATIONS

After completing the first step, the analyst has a model in hand that describes the configuration and motion of the target system in terms of certain configuration and motion variables. The next step is to form various vector and scalar quantities, including kinematic quantities, kinetic quantities, quantities that describe mass distribution, and, optionally, quantities that describe kinetic energy, potential energy, and work functions. Then, these quantities are related to one another by applying a principle of mechanics. The resulting equations will involve derivatives of the motion variables and are called *dynamical differential equations*. During simulation, the dynamical differential equations together with the kinematical differential equations will govern the behavior of the model as it interacts with a human user through the haptic interface.

The various approaches for producing equations of motion can be classified as either 1) vector approaches based on the direct use of D’Alembert’s or Jourdain’s principles such as Kane’s method, or 2) scalar approaches based on Hamilton’s and Gauss’ principles such as Lagrange’s equations or the Gibbs-Appell equations. The vector approaches employ a projection or dot-product onto certain subspaces while the scalar approaches are each based on a variational principle. Generally, the vector approaches require the expression of accelerations or angular momentum derivatives, which can, especially without the judicious use of generalized speeds, involve significant amounts of algebra. In Lagrange’s equations, the algebra enters not in the computation of accelerations, since only velocities are needed, but in the computation of partial derivatives and derivatives of scalar expressions.

Though the equations produced using the various approaches are equivalent in the sense that they produce the same numerical results in simulation, Kane’s method holds some unique advantages. Both Lagrange’s equations and Kane’s equations use generalized coordinates and thus embed configuration constraints. Therefore mechanisms, including robots and haptic interface devices, that are generally composed of many bodies yet possess few degrees of freedom, are better served by either one of these *multibody system* methods than by standard application of the equations of Newton and Euler (Gillespie and Colgate, 1997). Kane’s method, because it uses generalized coordinates yet is based on a vector approach, has been called “Lagrange’s form of D’Alembert’s Principle”. But Kane’s method also uses generalized speeds, which allows motion constraints to be em-

bedded. So Kane's method has been compared to the Principle of Virtual Power or Jourdain's Principle and Kane's equations have been likened to the Gibbs-Appell equations and Maggi's equations (Baruh, 2000), (Papstavridis, 1998). The use of generalized speeds in Kane's method enables the selection of an independent set of motion variables for nonholonomically constrained systems and the production of a model in ordinary differential equation form. In Lagrangian mechanics, where only generalized velocities are available, one must use the Method of Undetermined Multipliers to handle motion constraints. As mentioned above, the use of generalized speeds can lead to more compact equations of motion. Indeed, a special choice of generalized speeds can produce equations that are diagonal in the generalized speed derivatives, that is, uncoupled dynamical equations (Lesser, 1992). A further advantage of Kane's method over Lagrange's method derives from its membership in the vector approach classification: it is a projection method. The projection of certain non-contributing forces (sometimes called non-working constraint forces) which are known to lie in the null-space of the basis vectors formed as part of the method allows one to disregard such forces at the outset of the analysis, which leads to savings in the operations required during formulation.

Actually, one can use generalized speeds in the scalar approaches and realize some of the associated advantages. In such case the generalized speeds are known as *quasi-velocities*, and in an extension of Lagrange's equations there appear related quantities called *quasi-coordinates*. In fact, quasi-velocities were developed by Hamel and Jourdain and used to extend Lagrange's method in the early 20-th century (Papstavridis, 1998), thus the concept and term *quasi-velocity* or *nonholonomic variable* predates the term *generalized speed* by about 50 years. Numerous papers have appeared demonstrating the interrelationships between Kane's equations, Lagrange's equations, and Hamilton's equations (Mingori, 1995), (Desloge, 1987), (Townsend, 1992). Certainly it can be said that generalized speeds are more commonly known and used than quasi-velocities in the engineering community today. One indicator that supports this claim is the number and success of computer programs for dynamic system analysis that employ Kane's method internally (see (Trinkle, 2002)).

The success of Kane's method within the engineering community is perhaps due to Kane's own academic interests, which are primarily pedagogical, and his approach, which is algorithmic. By the same token however, a portion of the engineering community and the greater applied mechanics community, especially those whose background is in the more classical methods, find Kane's method at first unintuitive and unmotivated³. Further, Kane's method has been adopted to much greater extent by the spacecraft dynamics community than by the robotics community. With the explicit intention of addressing this situation,

³One member of this community commented that learning Kane's method from Kane's book (Kane, 1985) is like learning C from the book by Kernigan and Ritchie: the material is all there, just difficult to digest.

a derivation of Kane's equations is undertaken in the following. A geometric interpretation, following in part (Lesser, 1992) but also (Baruh, 2000) and (Papstavridis, 1998), using the language of differential geometry (which is often encountered within the field of robotics) forms the basis of the following derivation. First, Kane's equations for a system of particles are developed, highlighting the projection of Newton's equations onto a space tangent to the configuration space. Then the production of basis vectors for the tangent space used for the projection is discussed explicitly, including the case for motion constrained systems. Finally, the method is extended to rigid bodies.

3.1 Derivation of Kane's Dynamical Equations

Consider a system S of particles P_i ($i = 1, \dots, v$) whose positions with respect to an arbitrary point fixed in a Newtonian reference frame N are given by the position vectors \mathbf{p}^i ($i = 1, \dots, v$). Newton's law relates the force \mathbf{F}_i acting on P_i to m_i , the mass of P_i and ${}^N\mathbf{a}^i$, the acceleration of P_i in N as follows:

$$\mathbf{F}_i - m_i {}^N\mathbf{a}^i = \mathbf{0} \quad (i = 1, \dots, v), \quad (8)$$

where ${}^N\mathbf{a}^i = \frac{d}{dt} {}^N\mathbf{v}^i$ and ${}^N\mathbf{v}^i = \frac{d}{dt} \mathbf{p}^i$. To generate scalar equations from these v vector equations, one may dot-multiply with various vectors, including, for example, any three non-parallel vectors fixed in N , or, optionally, any v sets of 3 non-parallel vectors. Use of such vectors would produce $3v$ scalar equations, which could, conceivably, be solved for the N -measure numbers of ${}^N\mathbf{a}^i$ ($i = 1, \dots, v$), which are $3v$ in number. However, if the configuration of S may be parameterized by a set of n generalized coordinates q_i , ($i = 1, \dots, n$) and time t , *i.e.*, if vector functions \mathbf{p}^i may be written:

$$\mathbf{p}^i = \mathbf{p}^i(q_1, \dots, q_n, t) \quad (i = 1, \dots, v), \quad (9)$$

then the motion of S evolves on an n -dimensional manifold embedded within the $3v$ -dimensional coordinate space. That manifold may be called the configuration space or configuration manifold for S and the generalized coordinates q_i ($i = 1, \dots, n$) are curvilinear coordinates that parameterize that manifold. If the vector functions \mathbf{p}^i ($i = 1, \dots, v$) (or equivalently, the configuration constraints embodied by the choice of generalized coordinates) do not depend on time, then the manifold is fixed in the coordinate space. If the vector functions do depend on time, then the configuration manifold moves within the coordinate space, or it may be considered a one-parameter family of manifolds, where t is the parameter.

Given the parameterization of the configuration space by q_i ($i = 1, \dots, n$), a set of $n < 3v$ vectors can be found that spans the configuration manifold's tangent space and these n vectors shall be very useful for dot-multiplying Eq. (8) to produce scalar equations. Let us for the moment postpone the discussion of

how such a spanning set of vectors, or a *basis* for the tangent space, is found and simply designate the members of this set $\boldsymbol{\tau}_r^{P_i}$, ($r = 1, \dots, n$), where we note that there are n such vectors for each particle P_i . Dot-multiplication of Eq. (8) with each of $\boldsymbol{\tau}_r^{P_i}$, ($r = 1, \dots, n$) leads to the scalar equations

$$\boldsymbol{\tau}_r^{P_i} \cdot \mathbf{F}_i - \boldsymbol{\tau}_r^{P_i} \cdot m_i \mathbf{a}^{P_i} = 0 \quad (r = 1, \dots, n; i = 1, \dots, v) \quad (10)$$

After summing over the v particles of S , one arrives at:

$$\sum_{i=1}^v \boldsymbol{\tau}_r^{P_i} \cdot \mathbf{F}_i - \sum_{i=1}^v \boldsymbol{\tau}_r^{P_i} \cdot m_i \mathbf{a}^{P_i} = 0 \quad (r = 1, \dots, n) \quad (11)$$

Note that there are now only n such equations which may be solved for the n quantities \ddot{q}_r (the generalized velocity derivatives) that show up in expressions for \mathbf{a}^{P_i} .

But now we go a step further and introduce a linear transformation to the tangent space basis vectors $\boldsymbol{\tau}_r^{P_i}$ ($r = 1, \dots, n$) to produce a new set of tangent space basis vectors $\mathbf{v}_r^{P_i}$ ($r = 1, \dots, n$). This is where Kane's method departs from the traditional Lagrangian approach. We will again defer the full definition of this transformation for a moment, but note that any independent linear combination of all members of the basis set $\boldsymbol{\tau}_r^{P_i}$ ($r = 1, \dots, n$) may be used to produce a new basis set. The tangent space itself remains invariant. As we shall see below, the production of the new basis vectors $\mathbf{v}_r^{P_i}$ ($r = 1, \dots, n$) by differentiation with respect to the curvilinear coordinates q_r ($r = 1, \dots, n$) will no longer be possible as it is with $\boldsymbol{\tau}_r^{P_i}$ ($r = 1, \dots, n$), however, this does not matter. It just so happens that there is no coordinate system on the configuration manifold in terms of which the basis vectors $\mathbf{v}_r^{P_i}$ ($r = 1, \dots, n$) can be generated.

Using the new basis vectors for the tangent space, we now write:

$$\sum_{i=1}^v \mathbf{v}_r^{P_i} \cdot \mathbf{F}_i - \sum_{i=1}^v \mathbf{v}_r^{P_i} \cdot m_i \mathbf{a}^{P_i} = 0 \quad (r = 1, \dots, n) \quad (12)$$

The first term on the left is used to define the *generalized active forces* F_r for S in N :

$$F_r \triangleq \sum_{i=1}^v \mathbf{v}_r^{P_i} \cdot \mathbf{F}_i \quad (r = 1, \dots, n) \quad (13)$$

and the second term is used to define the *generalized inertia forces* F_r^* for S in N :

$$F_r^* \triangleq - \sum_{i=1}^v \mathbf{v}_r^{P_i} \cdot m_i \mathbf{a}^{P_i} \quad (r = 1, \dots, n). \quad (14)$$

Thus Eq. (12) yields

$$F_r + F_r^* = 0 \quad (r = 1, \dots, n) \quad (15)$$

which are called Kane's dynamical equations.

If the system S is subject to motion constraints (assume that the configuration constraints have already been accounted for by the choice of generalized coordinates,) then the motion of S can be characterized by $p < n$ independent generalized speeds u_r ($r = 1, \dots, p$) as discussed in the previous section. In such case the motion of S evolves in a p -dimensional sub-manifold embedded within the configuration manifold. We say that S still has n *global* (position or configuration) degrees of freedom (dof), but $n - m$ *local* (velocity or motion) dof (Papastavridis, 1998). Geometrically speaking, this means that S can still reach all points in the configuration manifold, but not by any path; *i.e.*, the motion constraints impose local restrictions (rather than global restrictions imposed by configuration constraints). Given the motion constraints, a set of basis vectors with only p members can be found that spans the space tangent to the p -dimensional submanifold. Let us call the members of this set $\tilde{\mathbf{v}}_r^{P_i}$, ($r = 1, \dots, p$). Now Eq. (8) may be dot-multiplied with each of $\tilde{\mathbf{v}}_r^{P_i}$, ($r = 1, \dots, p$) which leads to, after summing over the v particles:

$$\sum_{i=1}^v \tilde{\mathbf{v}}_r^{P_i} \cdot \mathbf{F}_i - \sum_{i=1}^v \tilde{\mathbf{v}}_r^{P_i} \cdot m_i \mathbf{a}^{P_i} = 0 \quad (r = 1, \dots, p) \quad (16)$$

After using the first term to define the constrained generalized active forces \tilde{F}_r ($r = 1, \dots, p$) for S in N and the second term (including the negative sign) to define the constrained generalized inertia forces \tilde{F}_r^* ($r = 1, \dots, p$) we may write Kane's dynamical equations for the constrained system S

$$\tilde{F}_r + \tilde{F}_r^* = 0 \quad (r = 1, \dots, p) \quad (17)$$

Let us summarize the operations performed so far and reiterate the geometric interpretation. We began with the force balance equations for each particle (Eq. (8)) and then took the dot-product of each term with the basis vectors $\mathbf{v}_r^{P_i}$ ($r = 1, \dots, n$) (or, for the motion-constrained system, $\tilde{\mathbf{v}}_r^{P_i}$ ($r = 1, \dots, p$)). Then we summed over all the particles to yield n scalar equations (15) (for the constrained system, p scalar equations (17).) which are first order differential equations in the generalized speeds u_r . Hence the r -th differential equation is the sum of all force balances in the direction defined by the r -th basis vector \mathbf{v}_r (or $\tilde{\mathbf{v}}_r$). The individual force balances have been projected onto the tangent space of the configuration manifold (or in the case of the constrained system, onto the space tangent to the motion-constraint submanifold).

Because Kane's method is fundamentally a projection method, a further advantage is realized, namely, noncontributing forces may be dropped from the analysis at the outset. To see this, we may decompose the total forces \mathbf{F}_i acting on the particles into two components: the applied forces \mathbf{F}_i^a and the constraint forces \mathbf{F}_i^c , thus $\mathbf{F}_i = \mathbf{F}_i^a + \mathbf{F}_i^c$. The constraint forces, whether they arise from configuration constraints or motion constraints, are sometimes called non-working forces since they are orthogonal to the

tangent space of the configuration manifold (or in the case of a motion-constrained system, orthogonal to the tangent space of the motion sub-manifold) and thus they do no work on the system. D'Alembert's principle of constraint is another statement of this fact:

$$\mathbf{F}_i^c \cdot \mathbf{v}_r^{P_i} = 0 \quad (i = 1, \dots, \nu) \quad (18)$$

We may thus re-define the generalized active force: $F_r \triangleq \sum_{i=1}^{\nu} \mathbf{v}_r^{P_i} \cdot \mathbf{F}_i^a$ ($r = 1, \dots, n$) (or $\tilde{F}_r \triangleq \sum_{i=1}^{\nu} \tilde{\mathbf{v}}_r^{P_i} \cdot \mathbf{F}_i^a$ ($r = 1, \dots, p$)). Hence in the construction of Kane's dynamical equations, one need not consider any forces of constraint in the analysis, which often greatly simplifies the task. Even if included in the analysis, constraint forces will be automatically eliminated, since they are orthogonal to the tangent space. Note that constraint forces include forces of interaction between bodies such as the forces which hold a joint together in a multibody system. They do not include, however, friction forces acting at a joint or spring or damper forces when such an element spans the axis of a joint.

Construction of the tangent vectors

Let us now take up the topic deferred above, the construction of a set of basis vectors $\boldsymbol{\tau}_r$ for the tangent space. To begin, let us apply the chain rule to the vector functions \mathbf{p}^{P_i} (Eq. (9)), to arrive at an expression for the velocity ${}^N \mathbf{v}^{P_i}$ in terms of \dot{q}_i ($i = 1, \dots, n$):

$${}^N \mathbf{v}^{P_i} = \begin{bmatrix} {}^N \frac{\partial \mathbf{p}}{\partial q_1} & \dots & {}^N \frac{\partial \mathbf{p}}{\partial q_n} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} + \frac{{}^N \partial \mathbf{p}}{\partial t} \quad (i = 1, \dots, \nu) \quad (19)$$

where the left superscript indicates that differentiation is performed in frame N (see (Kane, 1985) Chapter 1, for a discussion of the importance of keeping track of reference frames during differentiation of vectors). The coefficients $\frac{{}^N \partial \mathbf{p}^{P_i}}{\partial q_r}$ ($r = 1, \dots, n$) of the generalized velocities in fact form a set of vectors that spans the tangent space of the configuration manifold. We may thus use these vectors in place of $\boldsymbol{\tau}_r^{P_i}$ used above in the derivation of Kane's equations. Each $\frac{{}^N \partial \mathbf{p}^{P_i}}{\partial q_r}$ represents the direction in which \mathbf{p}^{P_i} is affected by q_r . Thus if one obtains, by any means, an expression for ${}^N \mathbf{v}^{P_i}$ in terms of the generalized velocities, one may simply read off the coefficients of \dot{q}_r ($r = 1, \dots, n$) to obtain a basis for the tangent space.

In the treatment so far, including Eq. (19), the particles of system S have been dealt with one at a time. If instead one writes down column matrices with vector functions as elements, then certain very satisfying items make appearances, including a Jacobian matrix. For a matrix formulation of Kane's equations, see (Blajer, 1990) and for some discussion of the Jacobian matrix, see (Scott, 1988) and (Storch and Gates, 1989).

The tangent vectors that appear in Eq. (19) play an important role in the generation of relationships between Kane's equations and more classical equations. From Eq. (19) and a similar expression for ${}^N \mathbf{a}^{P_i}$, one can easily conclude from the definition of partial derivative that

$$\frac{{}^N \partial \mathbf{p}^{P_i}}{\partial q_r} = \frac{{}^N \partial \mathbf{v}^{P_i}}{\partial \dot{q}_r} = \frac{{}^N \partial \mathbf{a}^{P_i}}{\partial \ddot{q}_r}. \quad (20)$$

The first equation plays an important role in establishing inter-relationships between Kane's equations in the form (10) and Lagrange's equations (Baruh, 2000). The second equality is used to relate Kane's equations to the Gibbs-Appel equations (Mingori, 1995) (Desloge, 1987).

Generalized speeds and a transformation of basis vectors

As mentioned above, Kane introduced a linear transformation to the set of basis vectors $\boldsymbol{\tau}_r$, ($r = 1, \dots, n$) (or equivalently to the set $\frac{{}^N \partial \mathbf{p}}{\partial q_r}$, ($r = 1, \dots, n$)) that leads to significant advantages in certain cases. Specifically, the set of vectors $\boldsymbol{\tau}_r$ shall be subjected to a linear transformation using the matrix W that we first encountered in the definition for the generalized speeds, Eq. (1). As above, the transformed basis vectors are denoted $\mathbf{v}_r^{P_i}$, so the transformation reads:

$${}^N \mathbf{v}_r^{P_i} = \sum_{j=1}^n \boldsymbol{\tau}_j^{P_i} W_{jr} = \sum_{j=1}^n \frac{{}^N \partial \mathbf{p}^{P_i}}{\partial q_r} W_{jr} \quad (21)$$

We see that the elements of each column of W spell out the particular linear combination of the basis vectors $\boldsymbol{\tau}_j^{P_i}$. The basis vectors $\mathbf{v}_r^{P_i}$ are given the name *partial velocities* by Kane, for reasons that will become apparent soon. To see the utility of this transformation, substitute the inverted generalized speed definitions (the kinematical differential equations) Eq. (2) into Eq. (19) to obtain

$${}^N \mathbf{v}^{P_i} = \begin{bmatrix} {}^N \frac{\partial \mathbf{p}}{\partial q_1} & \dots & {}^N \frac{\partial \mathbf{p}}{\partial q_n} \end{bmatrix} W \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} + \begin{bmatrix} {}^N \frac{\partial \mathbf{p}}{\partial q_1} & \dots & {}^N \frac{\partial \mathbf{p}}{\partial q_n} \end{bmatrix} X + \frac{{}^N \partial \mathbf{p}}{\partial t} \quad (22)$$

The r^{th} partial velocity $\mathbf{v}_r^{P_i}$ of P_i is then defined as the coefficient of u_r ($r = 1, \dots, n$), consistent with Eq. (21). The portion of \mathbf{v}^{P_i} not containing any u_r is denoted $\mathbf{v}_t^{P_i}$ and called the velocity remainder of P_i in N . Rewriting Equations (22) yields

$${}^N \mathbf{v}^{P_i} = \begin{bmatrix} \mathbf{v}_1^{P_i} & \dots & \mathbf{v}_n^{P_i} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} + \mathbf{v}_t^{P_i} \quad (23)$$

The motivation for the term *partial velocity* is now evident. After one has in hand an expression for the velocity \mathbf{v}^{P_i} in terms of the generalized speeds (generated by kinematic operations, not necessarily by performing the operations in Eqs. (22) and (23)) one simply reads off the coefficients of the r -th generalized speed to obtain the r -th partial velocity. The r -th partial velocity is in fact a partial derivative of \mathbf{v}^{P_i} with respect to u_r :

$$\mathbf{v}_r^{P_i} = \frac{\partial \mathbf{v}^{P_i}}{\partial u_r} \quad (24)$$

In practice (when applying Kane's method by hand) one does not carry out the transformation Eq. (21) but rather uses Eq. (2) to substitute for the \dot{q} 's in expressions for particle velocities. Then the matter of identifying partial velocities (and arranging them into a table, as is often convenient) is simply a matter of reading off coefficients of the u 's.

Motion Constraints

For systems subject to motion constraints, the operations are not much more difficult: one simply additionally uses Eq. (5) to substitute for the dependent generalized speeds before reading off the partial velocities as the coefficients of the u 's. Only the independent u 's will appear in the velocity expressions and there will be only p of them.

For further use later, let us develop a linear transformation relating the constrained basis vectors $\tilde{\mathbf{v}}_r^{P_i}$ to the unconstrained (but transformed in terms of generalized speeds) basis vectors $\mathbf{v}_r^{P_i}$, analogous to the transformation of Eq. (21). Equation (23) can be partitioned as:

$${}^N \mathbf{v}^{P_i} = [\mathbf{v}_1^{P_i} \dots \mathbf{v}_p^{P_i}] \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} + [\mathbf{v}_{p+1}^{P_i} \dots \mathbf{v}_n^{P_i}] \begin{bmatrix} u_{p+1} \\ \vdots \\ u_n \end{bmatrix} + \mathbf{v}_i^{P_i} \quad (25)$$

and then the right hand side of Eq. (5) can be substituted for u_i ($i = p+1, \dots, n$) to produce:

$${}^N \mathbf{v}^{P_i} = ([\mathbf{v}_1^{P_i} \dots \mathbf{v}_p^{P_i}] + [\mathbf{v}_{p+1}^{P_i} \dots \mathbf{v}_n^{P_i}] D) \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} + \mathbf{v}_i^{P_i} + [\mathbf{v}_{p+1}^{P_i} \dots \mathbf{v}_n^{P_i}] E \quad (26)$$

Now the coefficient of u_r ($r = 1, \dots, p$) in Eq. (26) is the definition of the *constrained partial velocity* $\tilde{\mathbf{v}}_r^{P_i}$ ($r = 1, \dots, p$) of P_i in N and the portion of ${}^N \mathbf{v}^{P_i}$ not containing any u_r is the definition of the *constrained partial velocity remainder* $\tilde{\mathbf{v}}_i^{P_i}$ of P_i in N . Rewriting Eq. (26) now gives:

$${}^N \mathbf{v}^{P_i} = [\tilde{\mathbf{v}}_1^{P_i} \dots \tilde{\mathbf{v}}_p^{P_i}] \begin{bmatrix} u_1 \\ \vdots \\ u_p \end{bmatrix} + \tilde{\mathbf{v}}_i^{P_i}, \quad (27)$$

which also demonstrates that the constrained partial velocity vectors can be read off by inspection of velocity expressions.

Extension to rigid bodies

To extend the projection method to rigid bodies requires the availability of a tangent vector for each point of each rigid body within the mechanism. Although this initially sounds like a daunting task, it does not turn out bad at all since the forces of interaction between points in a single rigid body are non-contributing and since a new quantity (the angular velocity) makes an appearance for each rigid body, enabling the formation of simplified expressions. Let us begin with a single rigid body B . From standard kinematics, the velocity ${}^N \mathbf{v}^P$ in N of an arbitrary point P of B may be expressed:

$${}^N \mathbf{v}^P = {}^N \mathbf{v}^{B_o} + {}^N \boldsymbol{\omega}^B \times \mathbf{p}^{B_o P} \quad (28)$$

where ${}^N \mathbf{v}^{B_o}$ is the velocity of the center of mass B_o of B in N , ${}^N \boldsymbol{\omega}^B$ is the angular velocity of B in N and $\mathbf{p}^{B_o P}$ is a position vector from B_o to point P . Now the angular velocity of body B that appears in Eq. (28) can be expressed in terms of the generalized velocities, just as we expressed the velocity of a particle P in terms of the generalized velocities in Eq. (19). However, since the angular velocity is itself a defined quantity, that is, it cannot be constructed by differentiating an angular displacement, the coefficients appearing in front of the generalized velocities are not partials of an angular displacement with respect to the generalized speeds, but rather partials of the angular velocity with respect to the generalized velocities (see Eq. (20)).

$${}^N \boldsymbol{\omega}^B = \left[\frac{{}^N \partial {}^N \boldsymbol{\omega}^B}{\partial \dot{q}_1} \dots \frac{{}^N \partial {}^N \boldsymbol{\omega}^B}{\partial \dot{q}_n} \right] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} + \frac{{}^N \partial {}^N \boldsymbol{\omega}^B}{\partial t} \quad (29)$$

We shall give the coefficients of the generalized velocities the names *partial angular velocities* and denote them $\boldsymbol{\omega}_r$ ($r = 1, \dots, n$) and the second term the name *partial angular velocity remainder* and denote it $\boldsymbol{\omega}$ ⁴. Now, using an equation analogous to Eq. (19) for ${}^N \mathbf{v}^{B_o}$ and Eqs. (28) and (29), the velocity of the arbitrary point P of B can be expressed:

$${}^N \mathbf{v}^P = \sum_{r=1}^n (\boldsymbol{\tau}_r^{B_o} + \boldsymbol{\omega}_r^B \times \mathbf{p}^{B_o P}) \dot{q}_r + \boldsymbol{\tau}_i^{B_o} + \boldsymbol{\omega}_i^B \times \mathbf{p}^{B_o P}, \quad (30)$$

which is analogous to the expression for the velocity of a particle in Eq. (19). The coefficient of \dot{q}_r can be identified as the r -th

⁴Here, for simplicity of notation, we have chosen not to use different symbols for the angular velocity tangent vectors defined as the partials with respect to the generalized velocities and those defined as the partials with respect to the generalized speeds, as we did for the velocity tangent vectors $\boldsymbol{\tau}_r$ and \mathbf{v}_r . Later, we shall generalize the definition (invoke a transformation on the basis) and reuse the same symbols.

tangent vector for the point P on body B . However, the sum and cross product contained within this coefficient need not be carried out inside the sum.

Now if \mathbf{R}^B is the resultant of all forces acting on B and \mathbf{T}^B is a bound vector applied at B_o , and \mathbf{H}^B is the sum of all moments acting on B so that \mathbf{R}^B and \mathbf{T}^B may be considered a *replacement* for all forces and moments acting on B , then the Newton-Euler equations for B may be expressed

$$\mathbf{R}^B - \dot{\mathbf{L}}^B = \mathbf{0}; \quad \mathbf{T}^B - \dot{\mathbf{H}}^B = \mathbf{0} \quad (31)$$

where $\mathbf{L}^B = m_B {}^N \mathbf{v}^{B_o}$ is the linear momentum of B and \mathbf{H}^B is the angular momentum of B . Note that $\dot{\mathbf{L}}^B = m_B {}^N \mathbf{a}^{B_o}$ where m_B is the mass of B and ${}^N \mathbf{a}^{B_o}$ is the acceleration in N of B_o . Also, $\dot{\mathbf{H}}^B$ may be expressed as $(\mathbf{I}^B \cdot {}^N \boldsymbol{\alpha}^B + {}^N \boldsymbol{\omega}^B \times \mathbf{I}^B \cdot {}^N \boldsymbol{\omega}^B)$ where \mathbf{I}^B is the central inertia dyadic of B and ${}^N \boldsymbol{\alpha}^B$ is the angular acceleration of B in N . Alternate expressions exist for $\dot{\mathbf{H}}^B$, which are more efficient (fewer operations needed) in certain situations (Kane, 1985).

To produce scalar equations from Eq. (31), we dot-multiply the first of Eqs. (31) with $\boldsymbol{\tau}_r$ and the second of Eqs. (31) with $\boldsymbol{\omega}_r$ and sum the two. This operation projects the equations onto the appropriate tangent spaces. If there are N bodies in a system S , indexed by the subscript i , we may sum over each of the bodies to obtain:

$$F_r = \sum_{i=1}^N (\mathbf{R}^{B_i} \cdot \boldsymbol{\tau}_r^{B_i} + \mathbf{T}^{B_i} \cdot \boldsymbol{\omega}_r^{B_i}) \quad (r = 1, \dots, n), \quad (32)$$

called the generalized active force for the system of bodies S in N , and

$$F_r^* = \sum_{i=1}^N (\dot{\mathbf{L}}^{B_i} \cdot \boldsymbol{\tau}_r^{B_i} + \dot{\mathbf{H}}^{B_i} \cdot \boldsymbol{\omega}_r^{B_i}) \quad (r = 1, \dots, n), \quad (33)$$

called the generalized inertia force F_r^* for S in N , so that the dynamical differential equations may be expressed simply as:

$$F_r + F_r^* = 0 \quad (r = 1, \dots, n), \quad (34)$$

which looks no different than (15). Indeed, for a system of particles and rigid bodies, one simply sums the contribution to F_r of the collection of particles, formed according to Eq. (13) with the contribution to F_r of the collection of bodies, formed according to Eq. (32). Likewise, F_r^* is formed according to Eqs. (14) and (33).

Finally, note that the transformations that were applied to the basis vectors $\boldsymbol{\tau}_r$ to produce \mathbf{v}_r for the system of particles above may be extended to the system of rigid bodies without modification. The same arguments hold, that a linear combination of basis

vectors spans the same tangent space. Thus the transformed tangent vectors are defined $\mathbf{v}_r^{B_o} \triangleq \frac{{}^N d \mathbf{v}^{B_o}}{d u_r}$ and $\boldsymbol{\omega}_r^B = \frac{\partial {}^N \boldsymbol{\omega}^B}{\partial u_r}$. Equation (29) now reads:

$${}^N \boldsymbol{\omega}^B = [\boldsymbol{\omega}_1^B \dots \boldsymbol{\omega}_n^B] \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} + \boldsymbol{\omega}_r^B \quad (35)$$

while a similar expression (much like Eq. (23)) may be written for \mathbf{v}^{B_o} . These new tangent vectors are used in place of $\boldsymbol{\tau}_r^{B_o}$ and $\boldsymbol{\omega}_r^B$ in Eq. (34) above. Likewise, for a system of motion-constrained rigid bodies, the use of the constraint equations (5), expressed as relationships among the generalized speeds may be used to eliminate the dependent generalized speeds and form a reduced set of p independent basis vectors $\tilde{\mathbf{v}}_r^{B_o}$ and $\tilde{\boldsymbol{\omega}}_r^B$ that span the tangent space to the motion-constrained sub-manifold. The same formalism and geometric interpretation of the equations still applies. Finally, note that non-working forces, including forces of constraint, need not be included in the analysis of rigid bodies.

Example 1 (continued) The rolling disk considered above will now be used to demonstrate the formulation of Kane's equations. First, one forms expressions for velocities of particles and mass centers and points to which forces are applied and angular velocities of bodies. For the rolling/sliding disk, the velocity ${}^N \mathbf{v}^{B_o}$ of point B_o in N and angular velocity ${}^N \boldsymbol{\omega}^B$ in terms of the generalized speeds are required:

$$\begin{aligned} {}^N \mathbf{v}^{B_o} &= u_1 \mathbf{a}_1 \\ {}^N \boldsymbol{\omega}^B &= (u_2/r) \mathbf{a}_3 \end{aligned} \quad (36)$$

From these expressions, a table of partial velocities may be constructed by inspection:

	$\mathbf{v}_r^{B_o}$	$\boldsymbol{\omega}_r^B$
$r = 1$	\mathbf{a}_1	$\mathbf{0}$
$r = 2$	$\mathbf{0}$	\mathbf{a}_3/r

To complete the kinematic analysis, expressions for the accelerations of particles and body mass centers and body angular accelerations are formed:

$$\begin{aligned} {}^N \mathbf{a}^{B_o} &= \frac{{}^N d {}^N \mathbf{v}^{B_o}}{d t} = \dot{u}_1 \mathbf{a}_1; \\ {}^B \boldsymbol{\alpha}^B &= \frac{{}^N d {}^N \boldsymbol{\omega}^B}{d t} = \frac{\dot{u}_2}{r} \mathbf{a}_3 \end{aligned} \quad (37)$$

The forces acting on B , including the gravity force acting on the mass center B_o and the contact forces acting at the point of contact with A may be resolved into a resultant $\mathbf{R}^B = -m \mathbf{g} \mathbf{n}_2 - F_r \mathbf{a}_1 + N \mathbf{a}_2$ applied at B_o and a torque $\mathbf{T}^B = -r F_r \mathbf{a}_3$,

where F_f is the friction force, N is the normal force of contact, g is the local gravitational constant, and \mathbf{n}_2 is a unit vector directed vertically upward. The equations of motion are then composed by performing the dot products called for in Eqs. (32) and (33). For $r = 1$,

$$\begin{aligned} F_1 &= \mathbf{v}_1^{B_o} \cdot \mathbf{R}^{B_o} + \omega_1^B \cdot \mathbf{T}^B = -mg \sin(\phi) - F_f \\ F_1^* &= \mathbf{v}_1^{B_o} \cdot (-m^N \mathbf{a}^{B_o}) + \omega_1^B \cdot (-J^N \boldsymbol{\alpha}^B) = -m\dot{u}_1 \end{aligned} \quad (38)$$

and for $r = 2$,

$$\begin{aligned} F_2 &= \mathbf{v}_2^{B_o} \cdot \mathbf{R}^{B_o} + \omega_2^B \cdot \mathbf{T}^B = F_f \\ F_2^* &= \mathbf{v}_2^{B_o} \cdot (-m^N \mathbf{a}^{B_o}) + \omega_2^B \cdot (-J^N \boldsymbol{\alpha}^B) = -\frac{J}{r^2} \dot{u}_2 \end{aligned} \quad (39)$$

so the unconstrained equations of motion are:

$$\begin{aligned} -mg \sin(\phi) - F_f - m\dot{u}_1 &= 0 \\ -F_f - \frac{J}{r^2} \dot{u}_2 &= 0 \end{aligned} \quad (40)$$

which may be easily solved for \dot{u}_1 and \dot{u}_2 .

To formulate the constrained equations, with the constraints embedded, one first substitutes for the dependent generalized speeds in expressions for velocities, angular velocities, accelerations, and angular accelerations and forms a new table of partial velocities and partial angular velocities. For this example, we substitute $u_2 = -u_1$ to obtain: ${}^N \boldsymbol{\omega}^B = -u_1 / r \mathbf{a}_3$ and ${}^N \boldsymbol{\alpha}^B = -\dot{u}_1 / r \mathbf{a}_3$ while ${}^N \mathbf{v}^{B_o}$ and ${}^N \mathbf{a}^{B_o}$ remain unchanged. The new table of partial velocities and partial angular velocities reads:

	$\tilde{\mathbf{v}}_r^{B_o}$	$\tilde{\boldsymbol{\omega}}_r^B$
$r = 1$	\mathbf{a}_1	$-\mathbf{a}_3 / r$

The constrained generalized active force \tilde{F}_1 and constrained generalized inertia force \tilde{F}_1^* are formulated according to the dot products specified in Eqs. (32) and (33) using the constrained partial velocity and partial angular velocity:

$$\begin{aligned} \tilde{F}_1 &= -mg \sin(\phi) - F_f + F_f \\ \tilde{F}_1^* &= -m\dot{u}_1 - \frac{J}{r^2} \dot{u}_1 \end{aligned} \quad (41)$$

so that the equation of motion now reads:

$$-mg \sin(\phi) - \left(m + \frac{J}{r^2} \right) \dot{u}_1 = 0 \quad (42)$$

An alternative method for formulating the constrained equations, by taking combinations of the unconstrained equations, will be introduced in the next section and applied to the present example. The alternative method is proposed for on-line (during simulation) formulation of the constrained equations for systems subject to changing constraint conditions.

4 STEP 3: SIMULATE

Now that the equations of motion have been formulated symbolically for the target system, they may be wrapped in a numerical integration routine and solved within an interactive simulator featuring haptic display. If the constraints are embedded, which can be accommodated by Kane's method even for non-holonomically constrained systems as presented above, then the equations are in ODE form and minimal in number. A system with m motion constraints has $p = n - m$ dynamical differential equations, one for each independent generalized speed. Determination of values, by an ODE solver, for the p independent generalized speeds uniquely determines the values of the m dependent generalized speeds through Eq. (5). In contrast, Lagrange's equations supplemented with Undetermined Multipliers require the solution, by a Differential Algebraic Equation (DAE) solver, of $n + m$ equations in $n + m$ unknowns (the p independent and the m dependent generalized velocities, and m measure numbers of the constraint forces).

Certainly constraint embedding appears to have its advantages, both in terms of efficiency and simplicity of form in the resulting equations. However, if certain constraint equations are subject to change during simulation, then the process of embedding those constraints must take place on-line, so that their variation may be driven by the human user through a haptic interface during interactive simulation. In addition, the selection of independent generalized speeds must be completed on-line, since new constraint conditions can lead to changes in whether a given generalized speed is independent, dependent, or may be optionally grouped with either the dependent or independent set. Because options usually exist, the choice of which generalized speeds are to be considered independent is usually made by an analyst. Auspiciously, Reckdahl has developed a method by which independent generalized speeds may be chosen automatically. In this section, a simulator architecture is presented that includes Reckdahl's automated selection of independent generalized speeds, automated constraint embedding, and numerical solution of the resulting equations of motion in independent variables.

Using equations with embedded constraints, the following must occur with each detection of a collision and possible change of constraint: integration of the equations must be stopped, the impulses resolved, a new set of independent generalized speeds chosen, the dependent generalized speeds expressed in terms of the independent ones, the new constraint-embedded model constructed, then integration re-started. This rather involved process can be simplified, however, if all possible constraint conditions can be enumerated prior to simulation time. In such case, all pertinent equations, each with constraints embedded, may be pre-formulated and assembled into a hybrid dynamical finite state machine. The finite state machine then handles interactive sequencing during runtime. Mechanisms featuring latches, locks, stops, escapements such as the piano action can be neatly simulated using this approach (Gillespie, 1996).

If, however, all constraint conditions cannot be anticipated or if constructing the finite state machine would be too arduous because of necessary size or complexity, then an automated constraint-handling method outlined in the following 4 steps can be applied. 1) The constraint equations themselves are formulated numerically or literally on-line by a symbolic manipulator that is integrated into the run-time code. 2) The essentially numerical algorithm of Reckdahl (Reckdahl, 1997), further discussed below, is used to determine a set of independent generalized speeds. 3) The re-ordering of the generalized speeds or equivalently, the partitioning of matrix A_B into matrices A and B of Eq. (4) is completed on-line by a symbolic manipulator. 4) Finally, the constrained equations are determined using $D = A^{-1}B$ of Eq. (5) through the operations of Eq. (44) and (45) developed below.

These four steps are organized in a flow chart in Figure 2 that also indicates the initial formulation of the unconstrained equations $F_r + F_r^*$, numerical integration of the constrained equations $\tilde{F}_r + \tilde{F}_r^*$, and a set of operations that conditionally invoke constraint deletion, constraint imposition, and singularity avoidance. Steps 1), 2), and 3) lie outside the main simulation loop while step 4) lies within and is invoked each time step. Various conditional statements tie together the auxiliary loops, most notably a check on the constraint forces, a collision detector, and a check on the condition number of the matrix A of Eq. (4). Each of these conditionals, if satisfied, initiates an excursion from the main simulation loop. The excursion initiated by the check on constraint forces is labelled *constraint deletion* (Haug et al., 1986); it enables the simulation of unilateral constraints, or noninterpenetration constraints. The excursion initiated by the collision detector can be called *constraint imposition*. Like constraint deletion, this conditional invokes steps 1), 2) and 3) to take care of changes to the set of active constraints. A shorter excursion, invoking only steps 2) and 3), is conditionally initiated by a check on the condition number of matrix A . This excursion is called *singularity avoidance*; it allows impending singular configurations, indicated by small condition numbers to be side-stepped by re-selection of the independent generalized speeds.

Blocks 1) through 4) will be further discussed below. The collision detector will not be addressed in this paper as mentioned earlier, nor will the impulse resolver or the check on the constraint forces, beyond a few notes made now. For a method for resolving impulses and further notes on constraint imposition in the notation of Kane's equations as in Step 2) above, see (Djerassi, 1994), (Djerassi, 1999). The check on constraint forces, which enables unilateral constraints, initially appears to undermine the whole motivation for embedded constraint handling, since these forces are the very ones eliminated from the analysis using Kane's method. However, the constraint forces can be evaluated using an analytical procedure that complements constraint embedding. An orthogonal complement to the tangent space of the configuration manifold may be formed using the set of vectors used as a basis for the tangent space (the partial veloc-

ities) (Lesser, 1992). The equations so produced are called the *kinetostatic equations* (Papstavridis, 1998).

Block 1): Formulate Constraint Equations

The constraint equations in the form of Eq. (3) are formulated, where the elements of the matrix A_B are expressed symbolically in terms of parameters of the geometric model. It is assumed that the geometric model has been segmented into a connected graph of entities like vertices, edges, and surfaces for use by the collision detector. The same geometric representation can be used to formulate constraint equations once a certain pair of entities is known to be in contact.

Block 2): Select Independent Generalized Speeds

The procedure for identifying the independent generalized speeds is based on the construction of an orthonormal basis for the row space of matrix A_B . An orthonormal basis may be constructed by performing a numerical singular value decomposition (SVD) of the matrix A_B :

$$A_B = U\Sigma V^T, \quad (43)$$

where the parameters, if expressed symbolically, have been assigned values. The number m of non-zero singular values in Σ gives the number of independent constraints. Linearly dependent constraints, if they appear in Eq. (3), will be identified using the SVD. A row-wise inspection, according to an algorithm that can be automated, of the elements of V^T can be used to identify the independent generalized speeds. See (Reckdahl, 1997).

Block 3): Re-ordering the Generalized Speeds

After the independent generalized speeds are identified, matrix A_B is partitioned into matrices A and B of Eq. (4), which involves a re-ordering of the generalized speeds.

Block 4): Embedding the Constraints

Inside the main simulation loop, the constraints may be embedded using Eqs. (4.4.3) and (4.11.4) in (Kane, 1985), as recommended by (Mitiguy, 1995) and also detailed by (Wampler et al., 1985). Rather than substituting the right hand side of Equation (5) for each dependent generalized speed before forming partial velocities as was demonstrated in the formulation of the constrained equations $\tilde{F}_r + \tilde{F}_r^* = 0$ in the previous section, the constrained equations can be formulated directly by taking a certain combination of terms appearing in the unconstrained equations. That combination is directed by the constraint equations.

More explicitly, the new constrained generalized active force \tilde{F}_r ($r = 1, \dots, p$) and the constrained generalized inertia force \tilde{F}_r^* ($r = 1, \dots, p$) may be formed from the unconstrained

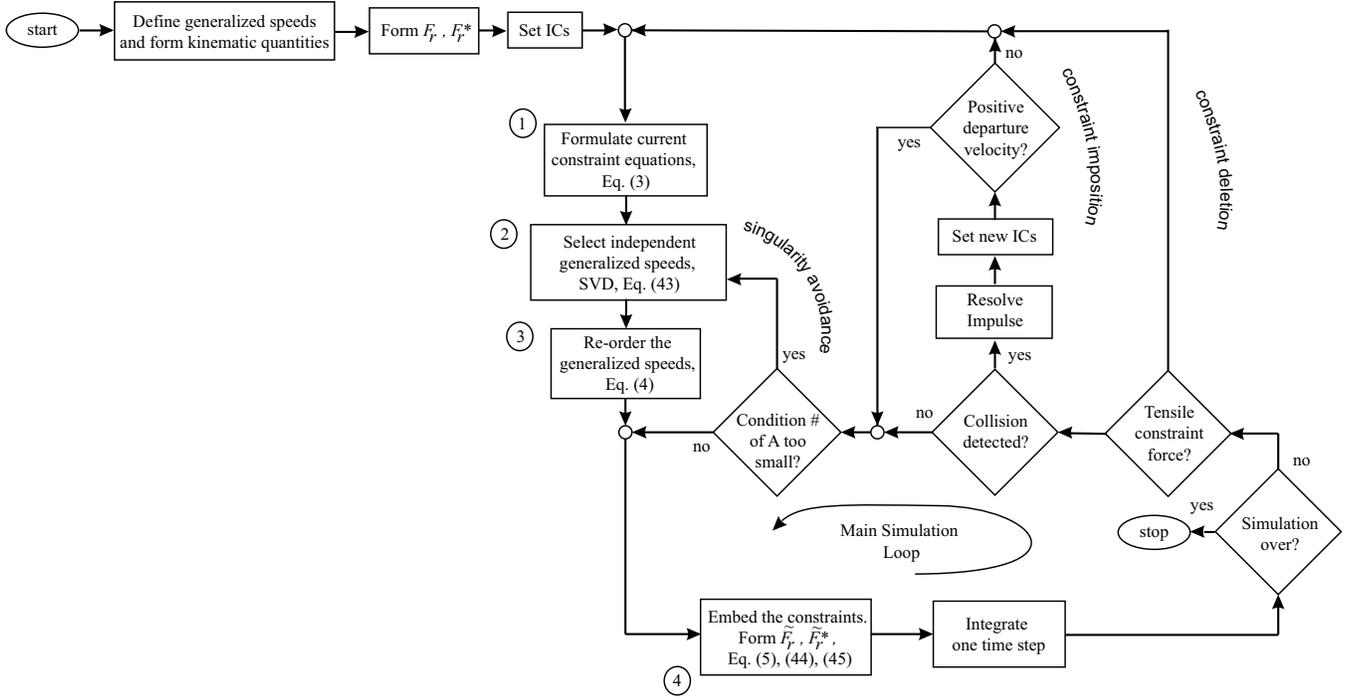


Figure 2. *Simulation Flow Chart*

generalized active force F_r ($r = 1, \dots, n$) and generalized inertia force F_r^* ($r = 1, \dots, n$) as follows:

$$\tilde{F}_r = F_r + \sum_{s=p+1}^n D_{sr} F_s \quad (r = 1, \dots, p) \quad (44)$$

$$\tilde{F}_r^* = F_r^* + \sum_{s=p+1}^n D_{sr} F_s^* \quad (r = 1, \dots, p) \quad (45)$$

where $D = A^{-1}B$ is defined in Eq. (5). To eliminate the dependent generalized speeds and dependent generalized speed derivatives from the resulting expressions, one may substitute from the motion constraints (Eq. (5)) and differentiated motion constraints. To establish the validity of Eqs. (44) and (45), simply substitute the definition of the constrained partial velocity (See Eq. (26)) into Eqs. (44) and (45) and rearrange terms to arrive at Eq. (13) and (14). The equations of motion in the independent generalized speeds are then simply formed as:

$$\tilde{F}_r + \tilde{F}_r^* = 0 \quad (r = 1, \dots, p) \quad (46)$$

which are only p equations in the p unknowns \dot{u}_i ($i = 1, \dots, p$).

Figure 2, then, presents a simulation paradigm in which the equations to be integrated are ODEs in a set of independent variables, no matter what constraints might hold at a particular time. This enables the use of a standard ODE solver, the stability properties of which have been extensively studied (Andrew and Humphries, 1998), which allows certain stability guarantees to be established (Brown and Colgate, 1997). Most operations in the flow chart can be implemented numerically, except Block 1), formulation of constraint equations. This block involves the assembly of terms as a function of the newly determined set of contact conditions and requires symbolic algebra. Alternatively, the constraint equations may be composed by indexing a pre-arranged matrix of variables according to the interconnection information provided by the collision detector. Note that the relatively expensive SVD operation in block 2) is performed outside the main integration loop while

The design of the simulator outlined in Figure 2 is based on full constraint embedding, which enables the use of an ODE solver and therefore allows the main simulation loop to run as fast as possible. (Note that checking the condition number of

matrix A can be performed efficiently *e.g.*, Gaussian elimination). Of course if simulation is to take place in real-time, all loops must be traversable in a single step, which may be difficult given complexity of impulse resolution, the symbolic nature of block 1), and the relative expense of the SVD operation in block 2). However, it is argued here that a fast main simulation loop could be used to develop a multi-rate scheme featuring a kind of predictor-corrector approach to constraint handling. Using a main simulation loop running faster than real-time, human behavior is predicted a short time ahead, collisions are anticipated, their resolution pre-computed, and the ensuing equations pre-formulated. Upon actual occurrence, the pre-computed collision event and associated resolution could be corrected before adoption and display through the haptic interface. A second motivation for the full constraint embedding scheme presented here follows from the relative ease of stability or passivity analysis that may be taken when a ODE solver is used rather than a DAE or other solver.

Example 1 (continued) The equations of motion for the rolling/sliding disk, with the rolling constraint embedded, may be formulated according to the method of Eq. (44) and (45) rather than the traditional approach employed in the previous section. Since there is one constraint and two generalized coordinates, $p = n - m = 2 - 1 = 1$, and from Eq. (7) $D = [-1]$, so Eqs. (44) and (45) read:

$$\begin{aligned}\tilde{F}_1 &= F_1 - F_2 \\ \tilde{F}_1^* &= F_1^* - F_2^*\end{aligned}\quad (47)$$

which, when Eqs. (38) and (39) are used to substitute for F_r and F_r^* ($r = 1, 2$), produce the same constrained equations of motion obtained for this example in the previous section.

One final note regarding this example is in order. Note that the normal force N is a noncontributing force and was eliminated from the analysis at the outset. Yet Newton's Law of Friction would express the measure number F_f of the force of friction as the product of N and a coefficient of friction μ . Several methods exist for bringing N into evidence, that is, to produce an equation for N . Kane outlines a method he calls the use of *auxiliary generalized speeds* in (Kane, 1985), which is related to a process called *relaxation of the constraints* (Papstavridis, 1998). The method of Lagrange multipliers can also be used with Kane's method. A third method is also available: to construct a basis for the orthogonal complement of the basis to the tangent space and use such basis to project the Newton-Euler equations (Lesser, 1992) (Papstavridis, 1998). Whatever method is used to bring N into evidence, great care must be taken to apply the force of friction in the appropriate direction *i.e.*, opposite the relative motion, which couples the dynamic and kinetostatic equations. This coupling can lead to some very interesting problems in wedging and sticking having to do with the existence and uniqueness of solutions (Trinkle et al., 1997) (Song et al., 2001).

5 FURTHER EXAMPLES

Example 2 To highlight the effect that the choice of generalized speeds has on the form of the equations of motion for a given system, consider the spherical robot wrist depicted in Figure 3. Rigid links A , B , and C form an open kinematic chain

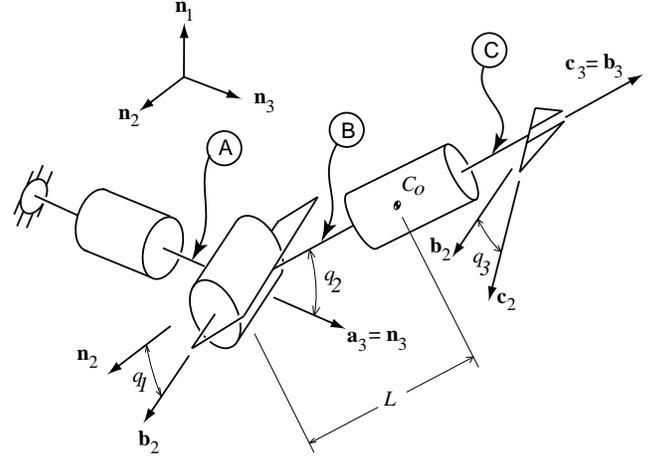


Figure 3. A spherical robot wrist

interconnected by revolute joints anchored in a Newtonian reference frame N . Let unit vectors \mathbf{n}_i ($i = 1, 2, 3$) fixed in N with \mathbf{n}_3 horizontal and parallel to the axis of A , \mathbf{n}_1 directed vertically upward and $\mathbf{n}_2 = \mathbf{n}_3 \times \mathbf{n}_1$. Similarly, dextral sets of orthogonal unit vectors $\mathbf{a}_i, \mathbf{b}_i$, and \mathbf{c}_i ($i = 1, 2, 3$), are fixed in A, B , and C , with $\mathbf{a}_3 = \mathbf{n}_3$ and $\mathbf{b}_2 = \mathbf{a}_2$ and $\mathbf{c}_3 = \mathbf{b}_3$. The quantities q_1, q_2 , and q_3 denote the angles between \mathbf{n}_2 and \mathbf{a}_2 , \mathbf{a}_3 and \mathbf{b}_3 , and \mathbf{b}_2 and \mathbf{c}_2 , respectively. The simplest way to introduce generalized speeds u_1, u_2 and u_3 is to define each as the first time derivative of the corresponding generalized coordinate: ($u_i = \dot{q}_i$ ($i = 1, 2, 3$)), in which event the associated kinematical differential equations are (See, for example (Kane et al., 1983)):

$$\begin{aligned}\dot{q}_1 &= u_1 \\ \dot{q}_2 &= u_2 \\ \dot{q}_3 &= u_3\end{aligned}\quad (48)$$

Alternatively, the generalized speeds may be defined as $u_i \triangleq {}^N \boldsymbol{\omega}^C \cdot \mathbf{c}_i$ ($i = 1, 2, 3$) in which case the associated kinematical differential equations are

$$\begin{aligned}\dot{q}_1 &= (\sin(q_3)u_2 - \cos(q_3)u_1) / \sin(q_2) \\ \dot{q}_2 &= \sin(q_3)u_1 + \cos(q_3)u_2 \\ \dot{q}_3 &= u_3 - \cos(q_2)(\sin(q_3)u_2 - \cos(q_3)u_1) / \sin(q_2)\end{aligned}\quad (50)$$

Certainly the kinematical differential equations for the second set of generalized speed definitions are more complex. To compare

$$\begin{aligned}
\dot{u}_1 &= (T_1 - L(F_2 + GM(s_1c_3 + s_3c_1c_2)) - (I_3 - I_2 - ML^2)u_2u_3)/(I_1 + ML^2) \\
\dot{u}_2 &= (T_2 + L(F_1 + GM(s_1s_3 - c_1c_2c_3)) + (I_3 - I_1 - ML^2)u_1u_3)/(I_2 + ML^2) \\
\dot{u}_3 &= (T_3 + (I_1 - I_2)u_1u_2)/I_3
\end{aligned} \tag{49}$$

Table 3. Operations count:

	Eq. (48)	Eqs. not shown	Eq. (50)	Eq. (49)
multiply, divide	0	790	10	29
add, subtract	0	285	4	16
sin, cos	0	372	9	10

the dynamical differential equations for the two sets of definitions, let A and B be massless and let M denote the mass of C with mass center C_o located on the axis of C a distance L from the joint connecting A to B . Let I_1, I_2 and I_3 denote the central moments of inertia of C about lines parallel to $\mathbf{c}_1, \mathbf{c}_2$ and \mathbf{c}_3 , respectively and replace the set of all contact forces acting on C with a couple of torque $T_1\mathbf{c}_1 + T_2\mathbf{c}_2 + T_3\mathbf{c}_3$ together with a force $F_1\mathbf{c}_1 + F_2\mathbf{c}_2 + F_3\mathbf{c}_3$ applied at C_o . Let the only distance force acting on C be the force of gravity. Then the dynamical differential equations corresponding to Eqns (48) are so long as to preclude their inclusion here, while those associated with Eqns (50) are shown in Eqn (49):

Table 3 compares the number of operations by category in the dynamical differential equations associated with the customary generalized speed definition, (leading to the kinematical differential equations (48)) and those shown in Eq. (49), which are associated with a carefully defined set of generalized speeds (leading to the kinematical differential equations (50)). Clearly, the dynamical equations associated with the specialized generalized speeds are much shorter. For this particular example, one might notice that the symmetry in Eq. (49) is reminiscent of Euler's equations for the torque-free motion of an axisymmetric body. Indeed, the kinematical differential equations in Eq. (50) may be found among those tabulated in (Kane et al., 1983) for each of the 24 sets of orientation angles. However, the particular definition of generalized speeds which leads to the most efficient equations for a given system depends on many factors. General directives are difficult to produce. For a few guidelines, see (Mittag and Kane, 1996).

Example 3 To illustrate the formulation of equations of motion using Kane's method and to further highlight the process of embedding constraints, an example with variable constraint structure is presented. The target system is a shopping cart with

a single castered front wheel whose vertical axis (the caster axis) is subject to sticking. When the front caster rotates freely, the shopping cart possesses two degrees of freedom; whereas when the front caster sticks, an additional motion constraint leaves only a single degree of freedom.

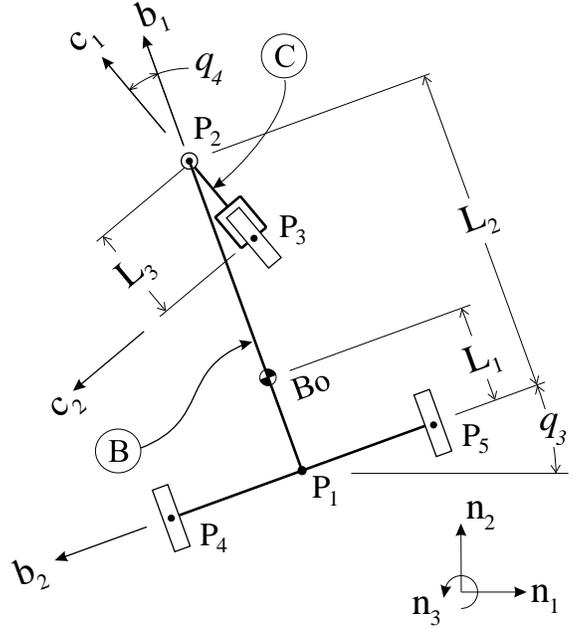


Figure 4. Schematic representation of a shopping cart, top view.

Figure 4 shows a top view of the shopping cart rolling on a flat horizontal plane N , in which orthogonal unit vectors \mathbf{n}_1 and \mathbf{n}_2 are fixed. Body B , of mass m_b and with central moment of inertia about a vertical axis I_b , represents the basket of the shopping cart. Unit vectors \mathbf{b}_1 and \mathbf{b}_2 are fixed in B , with \mathbf{b}_1 oriented along the long axis of B and \mathbf{b}_2 parallel to the common axis of the two back wheels, which are located at points P_4 and P_5 . Point P_1 lies midway between P_4 and P_5 and point B_o , representing the mass center of B , lies a distance L_1 along \mathbf{b}_1 from P_1 . A single castered wheel is mounted on B at point P_2 , located a distance L_2 along \mathbf{b}_1 from P_1 . Body C , in which orthogonal unit vectors \mathbf{c}_1 and \mathbf{c}_2 are fixed as shown, holds the castered wheel in contact with N at point P_3 , located a distance L_3 along $-\mathbf{c}_1$ from P_2 . Assume that the inertia properties of C may be described by a particle of mass m_c located at P_3 .

A human user pushes on B , exerting a moment $\tau \mathbf{b}_3$ and a force $F \mathbf{b}_1$ applied at point P_1 . The equations of motion shall be formulated considering N a Newtonian reference frame, both when the caster at P_2 is rotating freely and when it is sticking. These equations would be suitable for rendering the intermittent behavior (including the changing impedance or feel) of the cart through a three axis (two translations and one rotation in the plane) haptic interface.

Step 1 One may define a set of generalized coordinates q_i , ($i = 1, \dots, 4$) and generalized speeds u_i , ($i = 1, \dots, 3$) as follows. Let q_1 and q_2 be the N -measure numbers of a position vector locating P_1 with respect to a point fixed in N ⁵. Define q_3 as the angle between \mathbf{n}_1 and \mathbf{b}_2 and define q_4 as the angle between \mathbf{b}_1 and \mathbf{c}_1 . Define the generalized speeds using

$$u_1 \triangleq {}^N \mathbf{v}^{P_1} \cdot \mathbf{b}_1 \quad (51)$$

$$u_2 \triangleq {}^N \boldsymbol{\omega}^B \cdot \mathbf{n}_3 \quad (52)$$

$$u_3 \triangleq {}^N \boldsymbol{\omega}^C \cdot \mathbf{n}_3 \quad (53)$$

Note that, since the wheels at P_4 and P_5 allow no motion parallel to their common axis \mathbf{b}_2 , the velocity of P_1 in N may be expressed in terms of a single generalized speed:

$${}^N \mathbf{v}^{P_1} = u_1 \mathbf{b}_1. \quad (54)$$

Thus only three generalized speeds are needed. In effect, a reduced set of generalized speeds has been defined, one that already encapsulates the rolling constraint of the wheels at P_4 and P_5 .

Looking at expressions for ${}^N \mathbf{v}^{P_1}$, ${}^N \boldsymbol{\omega}^B$, and ${}^N \boldsymbol{\omega}^C$, written either in terms of \dot{q}_i , ($i = 1, \dots, 4$) or u_i , ($i = 1, \dots, 3$),

$$\begin{aligned} {}^N \mathbf{v}^{P_1} &= \dot{q}_1 \mathbf{n}_1 + \dot{q}_2 \mathbf{n}_2 = u_1 \mathbf{b}_1 \\ {}^N \boldsymbol{\omega}^B &= \dot{q}_3 \mathbf{n}_3 = u_2 \mathbf{n}_3 \\ {}^N \boldsymbol{\omega}^C &= (\dot{q}_3 + \dot{q}_4) \mathbf{n}_3 = u_3 \mathbf{n}_3 \end{aligned} \quad (55)$$

one may express the definitions of the generalized speeds in terms of the generalized coordinate derivatives using (see Eq. (1))

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} c_3 & s_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \quad (56)$$

and the reciprocal relations that become the kinematical differ-

⁵ q_1 and q_2 would be called *coordinates* of P_1 if coordinate frames (defined by bound unit vectors) were used in place of reference frames (defined by free unit vectors).

ential equations read (see Eq. (2)):

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} c_3 & 0 & 0 \\ s_3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (57)$$

Step 2 The set of generalized speeds $\{u_1, u_2, u_3\}$ contains dependent variables since a rolling constraint exists at the wheel at P_3 . If the caster is sticking, yet another constraint equation exists. Let us first develop the equations of motion for the system without caster sticking, with only the rolling constraint embedded. Then we shall develop the equations of motion for the system with caster sticking, with both the rolling constraint and the sticking constraint embedded. Thereafter, the recombination method by which the sticking equations can be developed directly from the non-sticking equations will be demonstrated.

Non-sticking system equations The wheel at P_3 permits no motion parallel to its axis,

$$\mathbf{v}^{P_3} \cdot \mathbf{c}_2 = 0, \quad (58)$$

which produces a constraint equation that may be assembled into the matrix form of Eq. (3):

$$\begin{bmatrix} -s_4 & L_2 c_4 & -L_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = 0 \quad (59)$$

Let us choose u_3 to be the dependent generalized speed, in which case Eq. (5) reads :

$$[u_3] = \begin{bmatrix} -s_4/L_3 & L_2 c_4/L_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (60)$$

To embed the constraint, we now substitute the right hand side of Eq. (60) wherever u_3 appears in kinematic quantities (for example, in ${}^N \mathbf{v}^{P_3}$). Likewise, at this point an expression for \dot{u}_3 in terms of \dot{u}_1 and \dot{u}_2 and q_i , ($i = 1, \dots, 4$) can be formed for use in eliminating \dot{u}_3 .

Partial velocities for mass centers and points to which force is applied and partial angular velocities for massive bodies and bodies to which moments are applied may be assembled into a table:

	$\mathbf{v}_r^{P_1}$	$\mathbf{v}_r^{B_0}$	$\mathbf{v}_r^{P_3}$	$\boldsymbol{\omega}_r^B$
$r = 1$	\mathbf{b}_1	\mathbf{b}_1	$\mathbf{b}_1 + s_4 \mathbf{c}_2$	0
$r = 2$	0	$L_1 \mathbf{b}_2$	$L_2 \mathbf{b}_2 - L_2 c_4 \mathbf{c}_2$	\mathbf{b}_3

The generalized active forces F_r ($r = 1, 2$) may be now be formulated according to Eq. (13)

$$F_r = \mathbf{v}_r^{P_1} \cdot (F \mathbf{b}_1) + \boldsymbol{\omega}_r^B \cdot (\mathbf{t} \mathbf{n}_3) \quad (61)$$

and the generalized inertia forces F_r^* ($r = 1, 2$) may be formulated according to Eq. (14):

$$F_r^* = \mathbf{v}_r^{B_o} \cdot (-m_B {}^N \mathbf{a}^{B_o}) + \mathbf{v}_r^{P_3} \cdot (-m_C {}^N \mathbf{a}^{P_3}) + \boldsymbol{\omega}_r^B \cdot (-I_B {}^N \boldsymbol{\alpha}^B) \quad (62)$$

Expressions for ${}^N \mathbf{a}^{B_o}$, ${}^N \mathbf{a}^{P_3}$ and ${}^N \boldsymbol{\alpha}^B$ may be obtained by differentiating the expressions for ${}^N \mathbf{v}^{B_o}$, ${}^N \mathbf{v}^{P_3}$, and ${}^N \boldsymbol{\omega}^B$. When u_3 and \dot{u}_3 appears, substitutions are made from the right-hand-sides of Eq (60) and its derivative.

Carrying out the dot products and referring to the table of partial velocities, the dynamical differential equations are produced by summing the F_r and F_r^* ($r = 1, 2$) shown in Eq. (63).

Sticking system equations During those periods when the caster sticks, an additional constraint arises since ${}^B \boldsymbol{\omega}^C = 0$ or $\dot{q}_4 = 0$. In terms of the generalized speeds this adds a row to the matrix form of Eq. (3):

$$\begin{bmatrix} -s_4 & L_2 c_4 & -L_3 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{0}, \quad (64)$$

which may be used to eliminate 2 variables from the set of 3 generalized speeds. Let us declare u_2 and u_3 the dependent generalized speeds. Then

$$u_3 = u_2 = s_4 / (L_2 c_4 - L_3) u_1 \quad (65)$$

An expression for \dot{u}_2 can be obtained by differentiation. A new table of partial velocities and partial angular velocities reads:

	$r = 1$
$\mathbf{v}_r^{P_1}$	\mathbf{b}_1
$\mathbf{v}_r^{B_o}$	$\mathbf{b}_1 - (L_1 s_4 / Z_2) \mathbf{b}_2$
$\mathbf{v}_r^{P_3}$	$\mathbf{b}_1 - (L_2 s_4 / Z_2) \mathbf{b}_2 + (L_3 s_4 / Z_2) \mathbf{c}_2$
$\boldsymbol{\omega}_r^B$	$-(s_4 / Z_2) \mathbf{b}_3$

Carrying out the dot products called for in Eq. (61) and (62), and after defining the following intermediate variables

$$\begin{aligned} Z_1 &= \dot{q}_4 = u_3 - u_2 \\ Z_2 &= L_3 - L_2 c_4 \\ Z_3 &= L_2 m_C (L_2 - L_3 c_4) \\ Z_4 &= L_1 m_B + L_2 m_C \\ Z_5 &= I_B + Z_3 + M_b L_1^2 \\ Z_6 &= s_4 Z_1 (Z_5 + m_C L_3^2 - L_2 L_3 m_C c_4) / Z_2^2, \end{aligned} \quad (66)$$

one arrives at dynamical differential equations shown in Eq. (67)

$$\begin{aligned} \ddot{F}_1 &= F - s_4 \tau / Z_2 \\ \ddot{F}_1^* &= (Z_4 - L_2 L_3 m_C s_4^2 / Z_2) u_2^2 - L_3 m_C (c_4 - L_2 s_4^2 / Z_2) u_3^2 - \\ &\quad Z_6 c_4 u_1 + L_2 Z_6 s_4 u_2 - s_4 (Z_4 - L_3 m_C c_4) / Z_2 u_1 u_2 - \\ &\quad (m_B + Z_5 s_4^2 / Z_2^2 + m_C (1 - L_3 s_4^2 / Z_2)) \dot{u}_1 \end{aligned} \quad (67)$$

Step 3

The components of the equations of the shopping cart may be organized into the interactive simulator design of Figure 2. Eqs. (67) would be formulated on-line according to Eqs. (44) and (45), but only when called for by the active constraint. (The matrix D and the number of equations p are declared and used as variables.) Then a virtual coupler is used to link the forward dynamics simulation of the cart to the haptic interface and in turn to the human user. The human user can be regarded as an admittance, controlling the motions of the haptic interface and feeling the reaction force and torque. The handle of the haptic interface is regarded as one end of the virtual coupler while the values of the generalized coordinates locating the shopping cart handle are regarded as the other end. The force and torque (resultant and couple), used as input to the simulation and displayed at the haptic interface, is determined by the relative displacement of the haptic interface and the image of the handle (given by the generalized coordinate values). A three-axis haptic interface is assumed available, capable of displaying two translations and one rotation in the plane. The linear and rotational stiffness and damping of the virtual coupler are tuned according to the properties of the haptic interface hardware and simulation stepsize in order to provide good coupling yet ensure system stability (Brown and Colgate, 1997), (Gillespie, 1996).

6 SUMMARY

The design of a simulator with haptic display based on Kane's method has been presented. An integral part of the simulator design is a symbolic manipulator that constructs custom-built equations of motion for each system to be simulated. The use of generalized coordinates and generalized speeds allows maximally efficient and maximally uncoupled equations of motion to be constructed. Changing constraint conditions within the system are handled on-line using an automated constraint embedding technique. Solution of the equations of motion by a numerical solver and incorporation of the solver into a set of routines for collision detection, resolution, and constraint handling was presented. Three examples were presented, the first highlighting a changing motion constraint, subject kineto-static laws of friction, the second highlighting the effect of the choice of generalized speeds and the third highlighting the permanent embedding of a motion (nonholonomic) constraint and variable embedding of a changing motion constraint.

In Step 2, Kane's method was presented in a tutorial fashion with a geometric interpretation. It was shown how the partial ve-

$$\begin{aligned}
F_1 &= F \\
F_2 &= \tau \\
F_1^* &= (L_1 m_B + L_2 m_C c_4^2) \dot{u}_2^2 - L_3 m_C c_4 u_3^2 - m_C s_4 c_4 u_1 u_2 - (m_B + m_C c_4^2) \dot{u}_1 - L_2 m_C s_4 c_4 \dot{u}_2 \\
F_2^* &= m_C L_2^2 s_4 c_4 u_2^2 - L_2 L_3 m_C s_4 u_3^2 - (L_1 m_B + L_2 m_C s_4^2) u_1 u_2 - L_2 m_C s_4 c_4 \dot{u}_1 - (I_B + m_B L_1^2 + m_C L_2^2 s_4^2) \dot{u}_2
\end{aligned} \tag{63}$$

locity vectors span the tangent space of the manifold on which the system dynamics are constrained to evolve. These partial velocity vectors play a central role in Kane's method, since through the dot product in the definitions of F_r and F_r^* they effect a projection of the Newton-Euler equations onto the constrained manifold. It was also shown how the basis formed by the set of partial velocity vectors (defined as partials with respect to generalized speeds) is a linear transformation of a basis formed by a set of vectors defined as partials with respect to generalized velocities.

In Step 1, the analyst chooses variables (generalized coordinates and generalized speeds) to describe the target system and thereby defines the directions of the partial velocity vectors. In particular, the definition of the generalized speeds in terms of the generalized coordinate derivatives produces the linear transformation between the two bases discussed in Step 2. The judicious choice of generalized speeds can have a profound effect on the form and compactness of the eventual equations of motion. For example, the choice of generalized speeds impacts the form of the eventual equations by determining the complexity of expressions for accelerations and by determining the alignment of partial velocities with applied forces. Also, a particular choice of generalized speeds can produce an orthogonal basis of partial velocity vectors. Such an orthogonal basis produces dynamical equations which are decoupled in the generalized speed derivatives, or said another way, the mass matrix is diagonalized. A diagonal mass matrix is easy to invert, even symbolically, which is advantageous when it comes to real-time simulation of complex systems.

In Step 3, a design was presented in which practically all steps of the process of producing and simulating customized equations was automated, except of course for the operations of Step 1, the definition of variables. This simulator design features a combined approach with both symbolic algebra formulation and numerical solution of the equations of motion. Because all constraints are embedded, the model for numerical integration is p ordinary differential equations, where p is the (velocity) degrees of freedom. Thus the numerical solution of the model is a particularly efficient operation.

ACKNOWLEDGEMENTS

The author would like to thank Professors Tom Kane and Ed Colgate for their very helpful reviews of earlier drafts of this paper. Discussions with Paul Mitiguy were also instrumental. Thanks also to the reviewers, whose comments contributed to

significant improvement and also the comment about Kernigan and Ritchie. This work was supported by a NSF postdoctoral research grant CISE-9625756 and PECASE award CISE-0093290.

REFERENCES

- Andrew, S. and Humphries, A. R. (1998). *Dynamical Systems and Numerical Analysis*. Cambridge Univ Press.
- Autolev (accessed July, 2003). Autolev.com web site. web. www.autolev.com.
- Baraff, D. (1989). Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Computer Graphics*, volume 23-3, pages 223–232. ACM.
- Baraff, D. (1994). Fast contact force computation for nonpenetrating rigid bodies. In *Computer Graphics*, pages 23–34. ACM.
- Baraff, D. (1995). Interactive simulation of solid rigid bodies. *IEEE Computer Graphics and Applications*, 15(3):63–75.
- Baruh, H. (2000). Another look at the describing equations of dynamics. *Journal of the Chinese Society of Mechanical Engineers*, 21(1):15–24.
- Berkelman, P., Hollis, R., and Baraff, D. (1999). Interaction with a real time dynamic environment simulation using a magnetic levitation haptic interface device. In *Proceedings 1999 IEEE International Conference on Robotics and Automation*, volume 4, pages 3261–3266, Detroit, MI, USA.
- Blajer, W. (1990). A useful matrix form of Kane's equations. *Mechanics Research Communications*, 17(5):311–318.
- Brown, M. and Colgate, J. E. (1997). Passive implementation of multibody simulations for haptic display. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume DSC.
- Cadoz, C., Luciani, A., and Florens, J.-L. (1984). Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the cordis system. *Computer Music Journal*, 8(3):60–73.
- Chang, B. and Colgate, J. E. (1997). Real-time impulse-based simulation of rigid body systems for haptic display. In *Proceedings of the ASME Dynamic Systems and Control Division*, pages 145–152, Dallas, TX, USA.
- Colgate, J. E. (1994). Coupled stability of multiport systems - theory and experiments. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 116(3):419–428.
- Colgate, J. E. and Schenkel, G. G. (1997). Passivity of a class of sampled-data systems: Application to haptic interfaces. *Journal of Robotic Systems*, 14(1):37–47.
- Constantinescu, D., Chau, I., DiMaio, S. P., Filipozzi, L., Salcudean, S. E., and Ghassemi, F. (2000). Haptic rendering of planar rigid-body motion using a redundant parallel mechanism. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2440–2445, San Francisco, CA, USA.
- Desloge, E. A. (1987). Relationship between Kane's equations and the Gibbs-Appell equations. *Journal of Guidance Control Dynamics*, 10:120–122.
- Djerassi, S. (1994). Imposition of constraints. *Transactions of the ASME*, 61:434–439.
- Djerassi, S. (1999). Dynamic analysis of systems with varying constraints. *Transactions of the ASME. Journal of Applied Mechanics*, 66(3):786–793.
- Garcia de Jalón, J. and Bayo, E. (1994). *Kinematic and Dynamic Simulation of Multibody Systems: the Real-Time Challenge*. Springer-Verlag.

- Gilbert, E. and Foo, C. (1990). Computing the distance between general convex objects in three-dimensional space. *IEEE Transactions on Robotics and Automation*, 6(1):53–61.
- Gilbert, E., Johnson, D., and Keerthi, S. (1988). A fast procedure for computing distance between convex objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, pages 193–203.
- Gillespie, B. (1993). Interactive dynamics with haptic display. In *Advances in Robotics, Mechatronics and Haptic Interfaces ASME Winter Annual Meeting*, volume DSC-Vol 49, pages 65–72. ASME.
- Gillespie, B. (1994). The virtual piano action: design and implementation. In *Proceedings of the International Computer Music Conference, Held: Aarhus, Denmark*, pages 167–170. ICMA.
- Gillespie, B. (1996). *Haptic Display of Systems with Changing Kinematic Constraints: The Virtual Piano Action*. PhD thesis, Stanford University.
- Gillespie, R. B. and Colgate, J. E. (1997). A survey of multibody dynamics for virtual environments. In *Proceedings of the ASME Dynamic Systems and Control Division*.
- Gregory, A., Lin, M., Gottschalk, S., and Taylor, R. (1999). A framework for fast and accurate collision detection for haptic interaction. In *Proceedings IEEE Virtual Reality*, pages 38–45, Houston, TX, USA.
- Gregory, A., Lin, M., Gottschalk, S., and Taylor, R. (2000). Fast and accurate collision detection for haptic interaction using a three degree-of-freedom force-feedback device. *Computational Geometry: Theory and Applications*, 15:69–89.
- Haug, E. J., Wu, S. C., and Yang, S. M. (1986). Dynamics of mechanical systems with coulomb friction, stiction, impact and constraint addition-deletion –I. *Mechanism and Machine Theory*, 21(5):401–406.
- Hollerbach, J. M. (2000). Some current issues in haptics research. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 1, pages 757–762, San Francisco, CA, USA.
- Jimenez, P., Thomas, F., and Torras, C. (2001). 3D collision detection: A survey. *Computers and Graphics*, 25(2):269–85.
- Kane, T. and Levinson, D. (1999). A multibody motion stability analysis. *Multibody System Dynamics*, 3(3):287–299.
- Kane, T. R. (1985). *Dynamics, Theory and Applications*. McGraw-Hill, New York.
- Kane, T. R., Likins, P. W., and Levinson, D. A. (1983). *Spacecraft Dynamics*. McGraw Hill.
- Khatib, O., Brock, O., Chang, K.-S., Conti, F., Ruspini, D., and Sentes, L. (2002). Robotics and interactive simulation. *Communications of the ACM*, 45(3):46–51.
- Lesser, M. (1992). A geometrical interpretation of Kane’s equations. *Proceedings of the Royal Society of London, Series A*, 436:69–87.
- Lin, M. and Manocha, D. (1997). Efficient contact determination in dynamic environments. *International Journal of Computational Geometry and Applications*, 7(1-2):123–51.
- Lin, M. C. and Canny, J. F. (1991). A fast algorithm for incremental distance calculation. In *IEEE International Conference on Robotics and Automation*, pages 1008–14. Sacramento, California.
- Lin, M. C., Manocha, D., Cohen, J., and Gottschalk, S. (1996). Collision detection: Algorithms and applications. In *Proc. of the 2nd Workshop on Algorithmic Foundations of Robotics*, pages 129–142.
- MacLean, K. E. and Durfee, W. K. (1995). An apparatus to study the emulation of haptic feedback. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume 57-2, pages 615–621.
- Mark, W. R., Randolph, S. C., Finch, M., van Verth, J. M., and Taylor III., R. M. (1996). Adding force feedback to graphics systems: Issues and solutions. In *Proc. SIGGRAPH*, pages 447–52.
- Miller, B. E., Colgate, J. E., and Freeman, R. A. (2000). Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation*, 16(6):712–719.
- Mingori, D. L. (1995). Lagrange’s equations, Hamilton’s equations, and Kane’s equations: Interrelations, energy integrals, and a variational principle. *Journal of Applied Mechanics*, 62:505–510.
- Mirtich, B. and Canny, J. (1994). Impulse-based dynamic simulation. In *Proceedings, Workshop on Algorithmic Foundations of Robotics*.
- Mitiguy, P. (1995). *Efficient formulation and solution of equations of motion*. PhD thesis, Stanford University.
- Mitiguy, P. C. and Kane, T. R. (1996). Motion variables leading to efficient equations of motion. *International Journal of Robotics Research*, 15(5):522–532.
- Nahvi, A., Nelson, D. D., Hollerbach, J. M., and Johnson, D. E. (1998). Haptic manipulation of virtual mechanisms for mechanical CAD designs. In *Proceedings of the International Conference on Robotics and Automation*, number 1, pages 375–380, Leuven, Belgium.
- Ong, C. and Gilbert, E. (2001). Fast versions of the Gilbert-Johnson-Keerthi distance algorithm: Additional results and comparisons. *IEEE Transactions on Robotics and Automation*, 17(4):531–39.
- Ong, C. J. and Gilbert, E. G. (1997). The Gilbert-Johnson-Keerthi distance algorithm: A fast version for incremental motions. In *Proc. 1997 IEEE International Conference on Robotics and Automation*, pages 1183–89.
- Papastavridis, J. G. (1998). A panoramic overview of the principles and equations of motion of advanced engineering dynamics. *Applied Mechanics Reviews*, 51(4):239–265.
- Reckdahl, K. J. (1997). *Dynamics and control of mechanical systems containing closed kinematic chains*. PhD thesis, Stanford University.
- Ruspini, D. and Khatib, O. (2001). Haptic display for human interaction with virtual dynamic environments. *Journal of Robotic Systems*, 18(12):769–783.
- Schaechter, D. B. (1988). Interactive computerized symbolic dynamics for the dynamist. *Journal of the Astronautical Sciences*, 36(4):365–388.
- Scott, D. (1988). Can a projection method of obtaining equations of motion compete with Lagrange’s equations? *American Journal of Physics*, 56:451–456.
- Song, P., Kraus, P., and Kumar, V. (2001). Analysis of rigid-body dynamic models for simulation of systems with frictional contacts. *Transactions of the ASME. Journal of Applied Mechanics*, 68(1):118–128.
- Storch, J. and Gates, S. (1989). Motivating Kane’s method of obtaining equations of motion for dynamic systems. *Journal of Guidance*, 12(4):593–595.
- Townsend, M. A. (1992). Kane’s equations, Lagrange’s equations, and virtual work. *Journal of Guidance*, 15(1):277–280.
- Trinkle, J. (updated 2002). Simulation packages. website. http://www.cs.rpi.edu/trink/sim_packages.html.
- Trinkle, J., Pang, J.-S., Sudarsky, J.-S., and Lo, G. (1997). On dynamic multi-rigid-body contact problems with Coulomb friction. *Zeitschrift für Angewandte Mathematik und Mechanik*, 77(4):267–279.
- Wampler, C., Buffington, K., and Shu-hui, J. (1985). Formulation of equations of motion for systems subject to constraints. *Journal of Applied Mechanics*, 52:465–470.
- Yoshikawa, T., Yokokohji, Y., Matsumoto, T., and Zheng, X.-Z. (1995). Display of feel for the manipulation of dynamic virtual objects. *Journal of Dynamic Systems, Measurement, and Control*, 117:554–558.