



# Interactive virtual prototyping of a mechanical system considering the environment effect. Part 1: Modeling dynamics

Zheng Wang

*Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*

## ARTICLE INFO

### Article history:

Received 7 October 2010

Accepted after revision 10 June 2011

Available online 18 July 2011

### Keywords:

Dynamical systems

Virtual prototyping

Multi-body

Dynamic simulation

Environment effects

## ABSTRACT

Virtual prototyping often reduces the cost of overall development of a product, as well as the time required to develop it, and can increase innovation as well. The physically accurate simulation of a mechanical system enables the designer to investigate, explore, and experience the performance and behavior of an evolving product, and thus reduce the number of physical prototypes needed. For the sake of better support for designers to manipulate the simulation, we have developed a dynamic simulation package of a mechanical system considering its environment effect. The package incorporates physical behavior modeled by multi-body dynamics, fluid and contact dynamics with environment and user input. The main contribution of the simulator is in providing support that allows the users' interactive manipulation in terms of a specific environment parameter configuration in the simulation loop online as opposed to the traditional offline.

© 2011 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.

## 1. Introduction

The main advantage of virtual prototyping is that it is no longer necessary to build a prototype product intended for initial tests, and any adjustments will be made directly in virtual reality by simulation models. Usually, virtual prototyping of products can be freely explored, examined and experienced interactively. Operation tasks on virtual prototyping can be studied and simulated before implementation in the real world. The simulation system enhanced in this way will facilitate iterative “trial and error” or even damaging “tests” of the control strategies. For example, by studying the effects of the application of a force/torque in an operation, permanent damage to certain type of products can be avoided. Up to now, few attempts have been made in incorporating dynamic interaction modeling for virtual prototyping manipulation.

In the first part of this Note, we report our work in developing a simulator incorporating physical behaviors, environment effect by dynamics-based modeling. In Section 2, a brief overview of related works is given. The architecture of the simulator is presented in Section 3. Based on transferring data interface between CAD and the simulator, dynamics modeling and solution method of virtual prototyping will be seen in Section 4. Then for the environment effect, we derive fluid force/torque on the virtual prototype from the immersed boundaries method in Section 5. The implementation examples and simulation result are described in Section 6. Finally, the conclusion is given in Section 7.

## 2. Related works

The concept of a “physical model” is commonly used for the purpose of producing realistic animation of a scene involving rigid or deformable objects in computer graphics [1]. A similar concept was also applied to simulate robotic grasping tasks [2,3]. Although a great deal of work on a force feedback device has been conducted [4,5] and the need for force display is widely recognized in developing the user interface [6], practical development and implementation of physically behaved

E-mail address: [flair@126.com](mailto:flair@126.com).

mechanism manipulation is still limited. As for the articulation system, Bae and Haug [7] apply *D'Alembert's* principle of virtual work in its dynamics formulation. Keat [8] uses a velocity transformation. Rosenthal [9], as well as Anderson [10], use Kane's equation. Pradhan and Modi [11] employ *Lagrangian* method with velocity and position transformations. Among these formulations, Rosenthal's order- $n$  formulation is one of the best known because of its computational efficiency. His formulation is applied by Banerjee [12] to simulate the dynamic behavior of an extrusion beam undergoing a large deflection, whose degree of freedom changes with time. It also has been shown that inter-body constraint forces can be easily computed as a bonus feature of the algorithm without extra computational cost [13]. However, his formalism is limited to systems with open-chain topology. By using specific formalism borrowed from quaternion algebra, Tasora [14] developed and successfully tested a fast solution scheme for the *Lagrangian* multiplier method of  $n$ -body mechanisms, which handles problems like constraint stabilization, intermitting contacts, redundant joints and impacts, consequently allowing its application to complex problems of interactive dynamics. Meanwhile, for contact between virtual objects, Baraff overviewed the researches on unilateral contact in detail and consistently argued that the methods have to resort to complex model whose solution existence and multiplicity must be dealt with [15]. To overcome the discontinuity of the *Coulomb* friction law, Song et al. [16] introduced a smooth nonlinear friction law, which approximates *Coulomb* friction. Such a friction model can increase the efficiency of both rigid body and compliant contact simulation. Hippmann [17] presented a new contact algorithm for analyzing contacts of complex shaped bodies in multi-body dynamics, which is based on representation of the body surfaces by polygon meshes and contact force determination by the elastic foundation model. As for dynamic simulation applications of multi-body mechanical system, many companies have been resorting to simulation tools to improve their design process. A successful case [18] is the design of the Boeing 777 airplane. Haug et al. [19] developed a new class of implicit SDIRK (singly diagonal Runge–Kutta) methods for the solution of stiff ODE (Ordinary Differential Equation), which has been implemented in the commercial DADS software system for vehicle system engineering. Stribersky et al. [20] outline their techniques for the numerical simulation of the motion of a virtual system of a metro train. Various components of the virtual train were modularized into a database that allowed the authors to examine various configurations. Rigid and elastic car body modules were created. Some of their calculated results are compared with measurements made on an actual prototype rail vehicle. Andersson et al. [21] present a method which utilized virtual prototyping in the decision making process and concentrated on the development of inverse modeling methods and animation. The method is illustrated by a simulation and visualization of internal dynamic forces in a wheel loader when excavating granulated material such as gravel or seed. Damer et al. [22] in DigitalSpace created a virtual model of Bucket Wheel Excavator, which is a common mining vehicle adapted for lunar size and power. The application of this virtual vehicle in a lunar base/ISRU processor setting proves that a physics-based, force feedback joystick driven virtual vehicle simulation could be delivered to consumer personal computers via the Internet.

In order to improve model and simulation accuracy, some researchers are paying much more attention to environment effects on virtual prototyping. The environment effects are often modeled as a result of various hydrodynamic forces. While these forces result from incompressible fluid flow determined by the Navier–Stokes (distributed fluid-flow) equations [23], “lumped” approximations to these forces are always used. Yuh [24] and Ioi and Itoh [25] have identified four separate effects that need to be included in a dynamic simulation of submerged rigid bodies. Under limiting assumptions that the net hydrodynamic force on an object can be represented as a sum of separately identified components modeling the effects of added mass, drag, fluid acceleration, and buoyancy forces. However, the added-mass effect of an accelerating rigid body in a viscous flow is still in controversy [26].

### 3. Virtual prototyping architecture

The main objective in the design of our dynamic simulator is to develop an interactive dynamic simulation package that supporting users' interactive manipulation in the loop and could be run in real time in a virtual fluid environment. During the simulation process, users can change the constraint relationship between components, apply force/torque to interested components and change the parameters of the existed forces/torques. The simulator automatically updates the dynamic model of the multi-body system in real time, and then continuously simulates the behavior of the model on the current condition. Fig. 1 illustrates the architecture of our system, which contains several modules. The CAD-VE data interface handles the mechanism model data which was imported from CAD systems. The user interaction module enables users to control the simulation by 2D or 3D interaction. Employing the dynamic equations for each object, the simulation modeling and solver module calculates and returns the simulated result to the user. After the solution, updated dynamic information, such as joint angles, velocity, acceleration, generalized force, etc., is sent to the postprocessor module to update the display of the simulated result visually. It also exports the model and result data to files stored in disk, which can be exchanged between other simulation software.

### 4. Modeling multi-body dynamics for mechanical system from CAD model

#### 4.1. Data interface between CAD and dynamic simulator

There is a large amount of commercial and non-commercial applications (e.g. 3D\_Evolution or TransMagic) available on the market, offering native conversion between common standard (STEP, IGES) and other well-known (AutoCAD, CATIA,

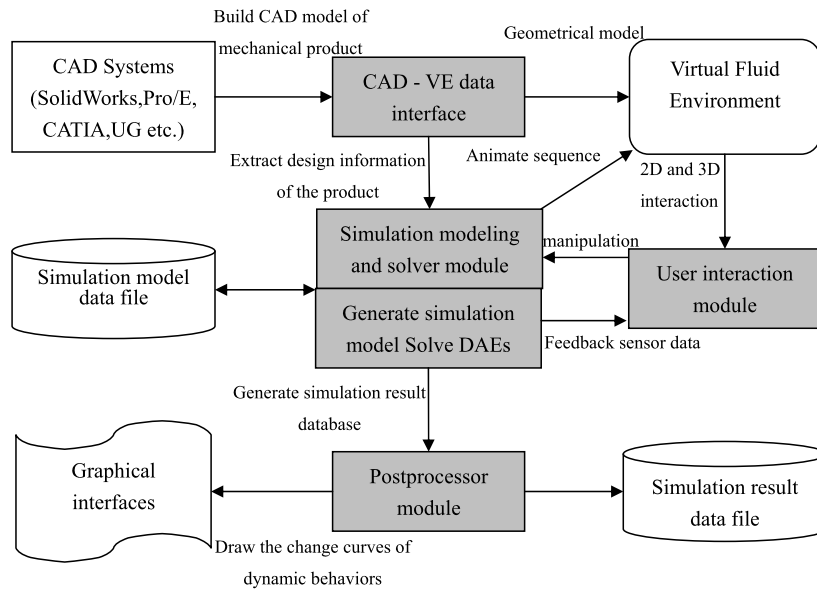


Fig. 1. Dynamic simulator architecture.

Inventor, Pro/Engineer, SolidWorks, Unigraphics, etc.) CAD data formats. Thus without subsequent restrictions it is assumed that the source data is available in the format that the CAD system used (Pro/Engineer) is able to import.

The translation from a CAD source file to dynamics description needs several steps. Basically this sequence will be followed during the conversion process.

**Step 1.** Assuming the CAD model has already been imported into Pro/Engineer, the hierarchy and geometry information of the actual model can be exported to VRML files simply through a click over the File menu “Save as...” command. Note that in a general case you get a main hierarchy file and the geometries of the subsequent parts in separate WRL files. Geometry information is only essential if one wants to model collision between the parts during the simulation.

**Step 2.** SimMechanics is a single-domain extension of the Simulink environment developed by MathWorks, and can be used for modeling and simulation of mechanical systems. Under Pro/Engineer environment the freely available Pro/E-to-SimMechanics plug-in, which lets one export the given CAD assembly to a single (so-called “Physical Modeling XML”) descriptor file, which is invented to ease the generation of SimMechanics models out of Pro/E data in an automated way [27]. The result XML file contains global hierarchy, constraint and physical parameter information (inertia, tensors, masses, etc.), but no geometries at all. Note that this tool can be used without having Matlab/Simulink or SimMechanics installed or started: only the intermediate PM XML file needs to be saved under Pro/Engineer in this step.

**Step 3.** An application called MotionInfo has been developed, the core logic of which processes the aforementioned XML descriptor file matching with VRML hierarchy and geometry files, thus generating an internal multi-body model out of the CAD information. In MotionInfo the internal (original) model can be extended interactively with various electromechanical elements (e.g. with parametrical motors from a model library) to form a more complex mechatronic (multi-domain) model. Finally, our tool is able to export its final mechatronic model to virtual prototyping models using the built-in conversion module, and by the same time it exports the geometry to DXF format files, in order to use those as visualizing shapes of virtual prototyping.

At the very beginning of our conversion process the physical information of the actual product’s CAD assembly will be exported to a single XML descriptor file. Such a “Physical Modeling XML” file enumerates all bodies in the original CAD assembly. Each XML rigid body entry contains information about the physical parameters (mass, inertia, surface area, volume, etc.) of the given part and has at least two coordinate frames in World space: the one that defines the location of the centre of gravity of the rigid body, the other that shows the origin transformation of the body’s geometry (CS1). XML bodies can carry any number of additional frames (CS2, CS3, ...), which all have a unique integer ID: these unique numbers are used by us to find the corresponding parts between joints. The XML file also contains information about joints, which represent the constraints of the original CAD assembly. Each XML joint ( $J_i$ ) has two integer IDs that are uniquely referencing two different XML bodies (these are named “Base” and “Follower” in a SimMechanics model). There can also be a series of joints between two bodies, representing various degrees of rotational or translational freedom between those two rigid

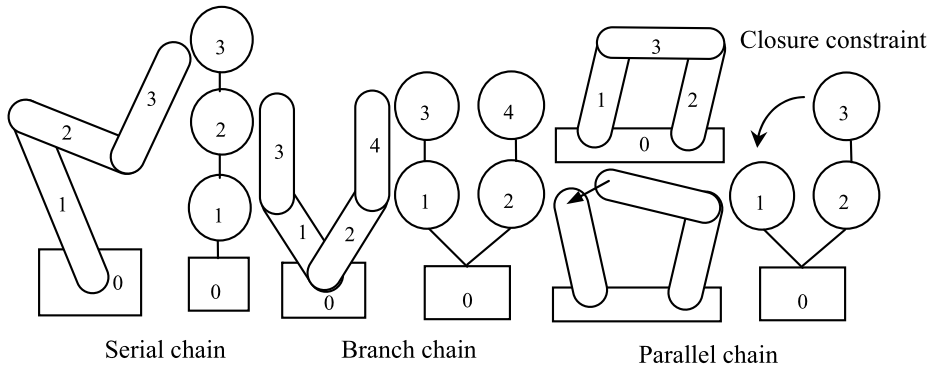


Fig. 2. Articulations of body.

parts. In the hierarchy this always implies the following sequence: “Follower” “J1”... “JN” “Base”, where “a b” shows that “a” is the child of “b” in the hierarchy (i.e. it inherits all transformation from that). Using the XML Joints’ frame references a skeleton (a draft hierarchy) of XML bodies can be built. Unfortunately this does not imply automatically that the final hierarchical model is immediately ready, the geometry of the (point-mass) bodies is still missing at this point.

In CAD systems it is quite often the case that more assembly parts share the same name (one can imagine a “template” part that has been used many times as building element). On the contrary, in case of our target simulator, the variable names must be unique inside each virtual prototyping model. First a pure mechanic model is created, via translating the source CAD information. This initial model contains only the parametrical bodies and the joints, which are connected by “connect” equations. In our simulator the uniqueness of variable names in the same visibility domain is essential. For this reason all exported bodies and joints must have an individual, unique instance name.

As the auto-generation of VRML and XML files are independently done, the partially auto-generated names inside the result files (e.g. “Obj01”, “Obj02” versus “Obj”, “Obj-1”) will be neither globally unique nor match each other. In order to find the corresponding entities both in VRML and in XML domains, a sophisticated procedure has to be followed. This is in the most cases inevitable, because there are usually less XML bodies than actual VRML geometries. You have to know which geometries form together a single rigid body, if you want to do a consistent collision handling during the simulation. All VRML geometry nodes have a homogeneous transformation matrix (which can arise derive from their respective parents recursively), from which one can derive their global pose (position and orientation) in the 3D world. This derived  $4 \times 4$  matrix is also used to transform the local vertices of a given VRML shape into the global coordinate system during rendering, for example. Fortunately the same pose information is also included in XML file with CS1 frame of the XML bodies.

First the position of all CS1 frames (extracted from XML) will be searched for matching with an origin frame location from VRML geometries being just imported. If there are more bodies having the same CS1 frame position, the filtering has to be continued on the “candidates” by differences in CS1 frame orientation. Assuming there are still more than one parts with the same global pose in CS1, the names of the XML bodies and VRML shapes (namely just their prefixes: e.g. “Obj01” will match with “Obj” or “Obj-1”) have to be compared to find the highly demanded single positive match. It is hardly imaginable that there are more parts in the CAD assembly with exactly the same pose and name. This should indicate that there is an error in the source CAD plan.

After finding all corresponding pairs of XML and VRML bodies, the final mechanic multi-body model can be generated. For this sake, the necessary physical parameters (masses, locations of CG frames, inertia tensors) have to be substituted into the final simulator’s parameters.

#### 4.2. Modeling articulation system

A typical environment contains one or more articulations of body to represent system components. Each articulation consists of rigid bodies with defined geometric shape connected together through joints in a tree structure, as in Fig. 2. Each body has only one parent body to which it is connected through its joint. The only exception is the root body of the system, which has no parent. Joints can describe topology and constraints, such as revolute, spherical, prismatic and universal relative movement between the connected bodies, a simplified bilateral contact shown in Fig. 3(a). For added realism, a damping element can be added to the joints to simulate mechanical energy dissipation due to joint friction or stiffness.

The above multi-body system dynamics description with Kane formulation from D’Alembert Lagrange method can be written as following:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = F_e + \Phi_q^T F_c \quad (1)$$

where  $q \in \mathbb{R}^n$  is a generalized coordinate vector;  $M(q)$  is an  $n \times n$  positive symmetric inertia matrix;  $C(q, \dot{q})\dot{q}$  is an  $n \times 1$  nonlinear inertia vector to represent centripetal and Coriolis force;  $G(q)$  represents the gravitational component; the external

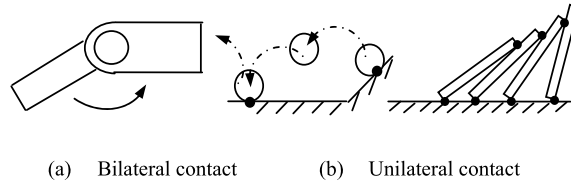


Fig. 3. Non-penetration constraint.

generalized force is denoted by  $F_e$  including control input  $u$ ;  $F_c$  denotes constraint force vector and a  $p \times n$  Jacobian matrix  $\partial\Phi/\partial q$  is denoted by  $\Phi_q$ , in which there exist  $p$  constraints

$$\Phi(q) = [\varphi_1(q), \dots, \varphi_p(q)]^T \geq 0 \tag{2}$$

#### 4.3. Modeling contact

Another important phenomenon that often happens is unilateral contact described by non-penetration constraint equations, as in Fig. 3(b). Nevertheless, the existence of non-smoothness requires special processes such as piecewise DAE, acceleration-force LCP (linear complementarity problem), penalty method, and velocity-impact time stepped method based on LCP as well [28].

Approximately continuous non-penetration constraint dynamic formulation is established based on compliant model to overcome the weakness of rigid model. It also ameliorates simulation realism in a virtual physics environment, and the stability of solution can be verified by singular perturbation theory.

We assume that there would be  $m$  contact points including  $r$  rolling and  $s$  sliding contacts at one moment,  $n$  and  $t$  denoting normal and tangent directions,  $s$  and  $r$  denoting sliding and rolling cases, respectively, then Jacobian matrix in Eq. (1) and constraint force can be represented as follows

$$\begin{aligned} \Phi_q^T &= [\Phi_{sq}^T \Phi_{nrq}^T \Phi_{trq}^T] \\ \Phi_{sq} &= [\Phi_{nsq} - \Phi_{tsq} \text{diag}(\mu_s \text{sign}(\dot{\Phi}_{tsq}))] \\ F_c &= [F_{cns}^T F_{ctr}^T F_{ctr}^T]^T \end{aligned} \tag{3}$$

where  $\mu$  is an  $s \times s$  diagonal matrix that contains all the coefficients of friction at the sliding contacts,  $\mu_s = -\text{diag}(\mu \text{sign}(\dot{\Phi}_{ts}))$ ,  $p = m + r = 2r + s$ . In normal direction, the non-penetration constraints should be

$$\phi_{ni} \geq 0, \quad F_{cni} \geq 0, \quad \phi_{ni} F_{cni} \geq 0 \tag{4}$$

and the complementarity equations of acceleration and force are

$$\ddot{\phi}_{ni} \geq 0, \quad F_{cni} \geq 0, \quad \ddot{\phi}_{ni} F_{cni} = 0 \tag{5}$$

In tangent direction, the displacement and friction force constraints should be

$$\mu F_{cni} - |F_{cti}| \geq 0, \quad \dot{\phi}_{ti} (\mu F_{cni} - |F_{cti}|) = 0, \quad \dot{\phi}_{ti} F_{cti} \leq 0 \tag{6}$$

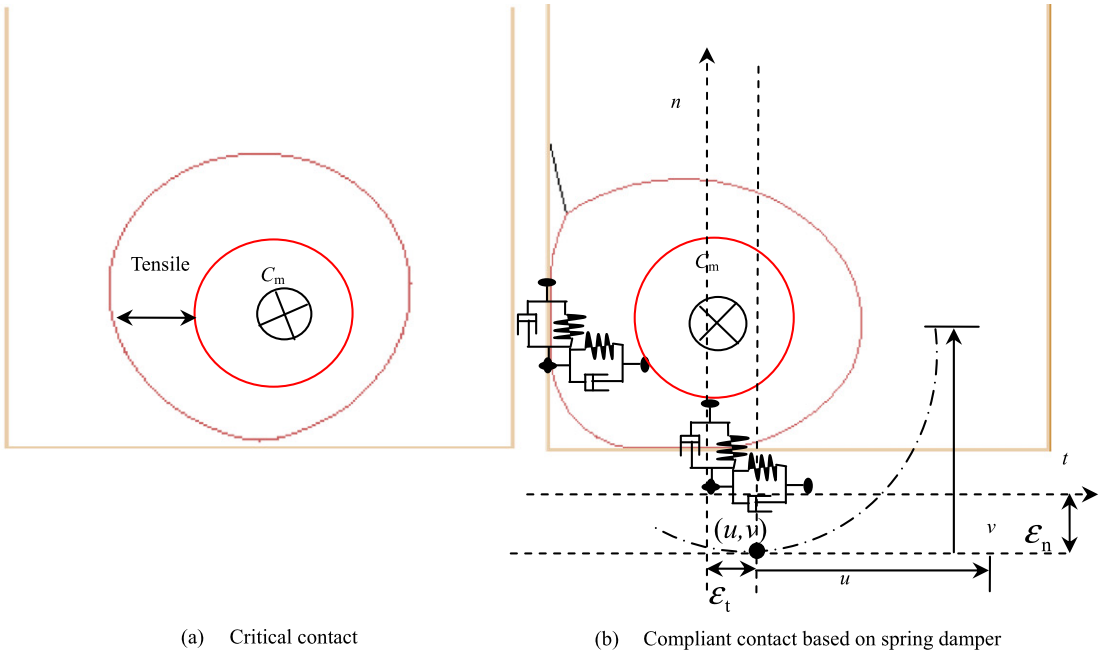
where  $\dot{\phi}_{ti}$  is a relative velocity at the  $i$ th contact. In sliding contact ( $\dot{\phi}_{ti} \neq 0$ ) case, we have

$$F_{cti} = -\text{sign}(\dot{\phi}_{ti}) \mu_i F_{cni} \tag{7}$$

where  $\mu_i$  is the  $i$ th entry of  $\mu$  and its complementarity constraint is

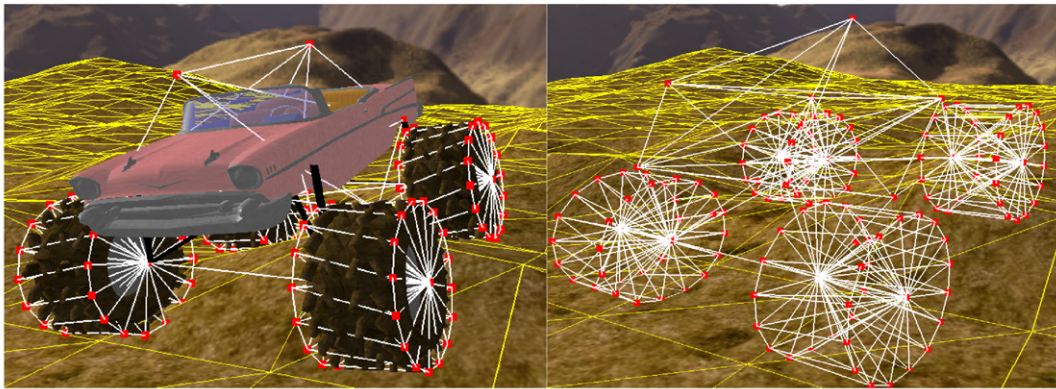
$$\ddot{\phi}_{ti} (\mu_i F_{cni} - |F_{cti}|) = 0 \tag{8}$$

In our system, we model compliant contact based on spring damper locally, in which the spring length (compliant layer) can be neglected relative to surface curvature of the body, and the same to its inertia. A test program of compliant contact model, as in Fig. 4, increases the deformation magnitude by setting parameters of the spring damper system for simplicity and clarity. The displacement ( $\varepsilon_n, \varepsilon_t$ ) introduced by spring deformation at contact ( $u, v$ ) will be determined by state variable ( $q, \dot{q}$ ) representing the multi-body system, so the actual contact position should be  $(u + \varepsilon_t, v + \varepsilon_n)$  after deformation. So  $(\varepsilon_n, \varepsilon_t)$  correlates with contact force by its derivative ( $\dot{\varepsilon}_n, \dot{\varepsilon}_t$ ), that is, the contact force is related with  $(q, \dot{q})$ . The tangential deformation will be a piecewise continuous function about tangent velocity for rolling (viscoelastic) compliant contact before transforming into sliding, while the traditional Coulomb law handles tangential force as a discontinuous function of velocity.

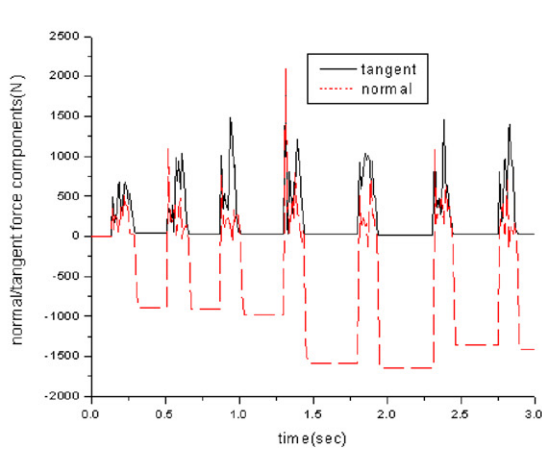


(a) Critical contact

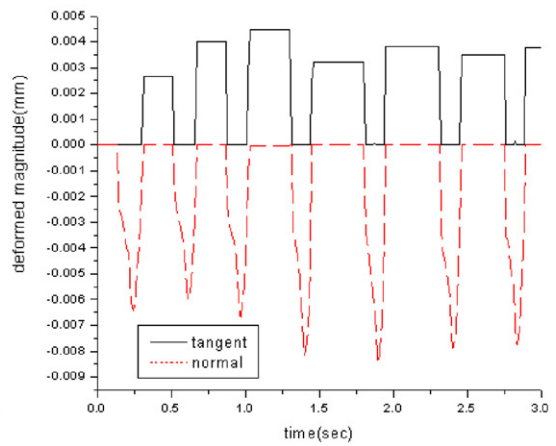
(b) Compliant contact based on spring damper



(c) Compliant contact model for an entity



d) A case of  $F_{cn} F_{ct}$



(e) A case of  $\epsilon_t, \epsilon_n$

Fig. 4. Non-penetration constraint based on compliance.

In Fig. 4, the normal and tangential contact force should be represented as follows

$$F_{cni} = K_{ni}(\varepsilon_{ni}) + \delta_{ni}(\varepsilon_{ni}, \dot{\varepsilon}_{ni}) \tag{9}$$

$$F_{cti} = K_{ti}(\varepsilon_{ti}) + \delta_{ti}(\varepsilon_{ti}, \dot{\varepsilon}_{ti}) \tag{10}$$

where  $K_{ni}$ ,  $K_{ti}$  is normal and tangential stiffness in spring damper system, and  $\delta_{ni}$ ,  $\delta_{ti}$  for corresponding damp coefficient,  $\varepsilon_{ni}$ ,  $\varepsilon_{ti}$  for corresponding deformation magnitude respectively.

#### 4.4. Solution

In our implementation, the solution of the above articulation system is based on the work in Ref. [29], where the different approaches are derived by forming an augmented system of equations and performing block matrix elimination. This yields a uniform derivation of the  $O(n)$  ABM and the  $O(n^3)$  CRBM.

As for solution of compliant contact, we firstly decouple the normal and tangential force, and make the tangential force follow Coulomb law,

$$|F_{cti}| \leq \mu_i F_{cni} \tag{11}$$

We use Kelvin–Voigt method to represent linear viscoelastic stiffness and damping in normal and tangent direction,

$$K = k_r \varepsilon, \quad \delta = \delta_r \dot{\varepsilon} \tag{12}$$

where  $k_r$  and  $\delta_r$  are the relevant stiffness and damping described by linear elasticity and viciousness in semi-space. For example,  $k_{rn}$  can be approximated with a Hertzian model. Normal/tangential force will be proportional to deformation in a linear spring system [30], but a special case may happen if  $k_{rn}$  ( $k_{rt}$ ),  $\delta_{rn}$  ( $\delta_{rt}$ ) = 0. A more accurate rigid impact model has been proposed by Hunt and Crossley as follows:

$$K = k_r \varepsilon^n, \quad \delta = \frac{3}{2} \alpha k_r \varepsilon^n \dot{\varepsilon} \tag{13}$$

where  $n$  can be given by Hertzian model and  $\alpha$  is a coefficient for impact velocity. This model possesses an apparent advantage when deformation vanishes while  $\dot{\varepsilon}$  is nonzero.

We define non-dimensional variables  $\varepsilon'_t = \varepsilon_t / (\mu F_{cn} / k_{rn})$ ,  $k'_{rt} = k_{rt} / (\mu k_{rn})$  and dimensionless time variable  $t' = t / T$ , where  $t$  a time variable and  $T$  is a characteristic time scale. Then we have  $F'_{ct} = F_{ct} / (\mu F_{cn})$ ,  $\delta'_t = \delta_t / (k_{rt} F_e)$  and tangential displacement,  $u'_t = \phi_t / (\mu F_{cn} / k_{rn})$ .  $F'_{ct}$  will depend on  $u'_t$  as the following cases:

- (1) Linear elastic case ( $k'_{rt} = 1.0$ ,  $\delta'_{rt} = 0$ );
- (2) Viscoelastic Kelvin–Voigt case represented in Eq. (12), where  $k_{rt} = 1.0$ ,  $\delta_t = 0.04$ ;
- (3) Nonlinear viscoelastic model represented in Eq. (13), where  $k'_{rt} = 0.6$ ,  $n = 1$ ,  $\alpha = 0.67$ ,  $\delta'_{rt} = 0.02$ .

So  $F'_{ct}$  will always be continuously correlative to  $(q, \dot{q})$ . The corresponding frictional force works in case (1) for tangential force dissipating energy, and viscous damping works in cases (2) and (3).

The above constraint equations can be alternative to Eqs. (5)–(8) and (4) can be transformed into

$$\varepsilon_{ni} = \max\{0, -\phi_{ni}(q)\} \tag{14}$$

$$\phi_{ni}(q) \geq -\varepsilon_{ni} \tag{15}$$

$$F_{cni} = K_{ni}(\varepsilon_{ni}) + \delta_{ni}(\varepsilon_{ni}, \dot{\varepsilon}_{ni}) \tag{16}$$

For rolling contact corresponding to Eq. (11) in tangent direction, we have

$$\dot{\varepsilon}_{ti} = -\dot{\phi}_{ti}(q) \tag{17}$$

$$v_{rsi} = 0 \tag{18}$$

$$F_{cti} = K_{ti}(\varepsilon_{ti}) + \delta_{ti}(\varepsilon_{ti}, \dot{\varepsilon}_{ti}) \tag{19}$$

For sliding contact, we have

$$F_{cti} = -\mu_i F_{cni} \text{sign}(v_{rsi}) \tag{20}$$

$$\dot{\varepsilon}_{cti} = h_{ti}(F_{cti} - K_{ti}(\varepsilon_{ti})) \tag{21}$$

$$v_{rsi} = \dot{\phi}_{ti} + \dot{\varepsilon}_{ti} \tag{22}$$

where  $h_{ti}$  is an inverse function about  $\delta_{ti}$ , if given  $\varepsilon_{ti}$ , the tangential deformation should be

$$\varepsilon_{ri} = \int_{t_0}^t \dot{\varepsilon}_{ti} dt + \varepsilon_{ti}(t_0) \tag{23}$$

Clearly, Eq. (22) can predict the type of contact, whether sliding or rolling and sliding direction. According to Eqs. (17)–(19) and considering viscoelasticity, if the force in Eq. (19) is inconsistent with friction constraint in Eq. (11), sliding will happen and contact force can be calculated from Eqs. (20)–(22), and  $\text{sign}(v_{rsi})$  will be opposite to direction of tangent force in Eq. (19). So, unique normal and tangential force solutions can be obtained from Eqs. (14)–(23) and only one  $\ddot{q}$  from positive matrix  $M$  in Eq. (1).

The compliant contact model will cost much more than the rigid, since the original rigid body system possesses  $n - (m + r)$  DOFs and  $2n - 2(m + r)$  dimensions of state space while the compliant model takes up  $2n$  dimensions of  $(q, \dot{q})$ . The uniqueness of normal and tangent force exclude multiplicity of solution, so the static non-determination will disappear. There is a solution case seen as in Fig. 4(d) and (e).

### 5. Modeling fluid dynamics for environment effect

#### 5.1. The mathematical formulation of the immersed boundary method

It is well established that the acceleration of a rigid body in an inviscid flow has the added-mass effect [31]. Because of the controversy about added-mass effect, the acceleration terms in our formulas will be derived from the contribution of the forces and torques of solid boundaries in the Navier–Stokes.

According to the immersed boundary method proposed by Peskin [32,33], we define the flow around a rigid object as the flow around a “rigid fluid” which replaces the object and undergoes the same rigid motion as the object. After nondimensionalized by a characteristic length scale, a characteristic velocity scale, and the fluid density, the “rigid fluid” is achieved by a boundary force and a body force in the momentum equation reading

$$\ddot{y} + \nabla \cdot (\dot{y}\dot{y}) = -\nabla p + \frac{1}{\text{Re}} \Delta \dot{y} + b + \int_S \rho \delta(y - y_s) d\alpha_1 d\alpha_2 \tag{24}$$

Above we use  $p$  to denote the pressure,  $t$  the time,  $b$  the body force contribution, and  $\text{Re}$  the *Reynolds* number. The boundary (multiple boundaries can be included in the same manner) is denoted as  $S$  and parameterized by *Lagrangian* parameters  $\alpha_1$  and  $\alpha_2$ . The boundary force concentrates at the boundary with the density  $\rho(\alpha_1, \alpha_2, t)$  through the *Dirac* delta function  $\delta(y - y_s)$ , where  $y_s(\alpha_1, \alpha_2, t)$  is coordinates at the boundary. The body force  $b$  is finite inside the boundary and zero outside. Inside it is given by

$$b = \ddot{\theta} \times (y - y_c) \tag{25}$$

The body force is piecewise continuous with the jump

$$[b] = -\ddot{\theta} \times (y_s - y_c) \tag{26}$$

where  $[\cdot] = (\cdot)_{s^+} - (\cdot)_{s^-}$  denotes a jump condition across the boundary ( $s^+$  denotes the boundary at the fluid side and  $s^-$  denotes the boundary at the object side), and  $y_c$  is coordinate and  $\theta$  is orientation of geometric center in the rigid fluid.

To present the boundary force, we introduce the following definitions:

$$T = \frac{\partial y_s}{\partial \alpha_1}, \quad B = \frac{\partial y_s}{\partial \alpha_2}, \quad N = T \times B \tag{27}$$

$$J_1 = \|T\|_2, \quad J_2 = \|B\|_2, \quad J = \|N\|_2 \tag{28}$$

$$n = \frac{N}{J}, \quad f = \frac{F}{J}, \quad f_n = f \cdot n, \quad f_t = f - f_n n \tag{29}$$

$$\omega = \nabla \times \dot{y}, \quad \omega_T = \omega \cdot T, \quad \omega_B = \omega \cdot B, \quad \omega_N = \omega \cdot N \tag{30}$$

$$r = y_s - y_c = R^T r_0 \tag{31}$$

where the normal direction  $n$  of the boundary is made point to the fluid side by the boundary parameterization, and  $r_0$  corresponds to  $r$  at reference time and  $R^T$  represents the orientation change of the body. We then have

$$f_t = -\frac{1}{\text{Re}} \left( \frac{\partial \dot{y}}{\partial n} \Big|_{s^+} - \dot{\theta} \times n \right) \tag{32}$$



$$\begin{aligned} \frac{\partial f_n}{\partial \alpha_1} &= T \cdot [b] - \left( \frac{2\dot{\theta} - \omega|_{s^+}}{\text{Re } J^2} \right) \cdot \left( J_1^2 \frac{\partial N}{\partial \alpha_2} - T \cdot B \frac{\partial N}{\partial \alpha_1} \right) - \frac{1}{\text{Re } J} \left( T \cdot B \frac{\partial \omega_T}{\partial n} \Big|_{s^+} - J_1^2 \frac{\partial \omega_B}{\partial n} \Big|_{s^+} \right) \\ &= T \cdot [b] + r_1 = \ddot{\theta} \cdot \left( \frac{\partial r}{\partial \alpha_1} \times r \right) + r_1 \end{aligned} \tag{33}$$

$$\begin{aligned} \frac{\partial f_n}{\partial \alpha_2} &= B \cdot [b] - \left( \frac{2\dot{\theta} - \omega|_{s^+}}{\text{Re } J^2} \right) \cdot \left( T \cdot B \frac{\partial N}{\partial \alpha_2} - J_2^2 \frac{\partial N}{\partial \alpha_1} \right) - \frac{1}{\text{Re } J} \left( J_2^2 \frac{\partial \omega_T}{\partial n} \Big|_{s^+} - T \cdot B \frac{\partial \omega_B}{\partial n} \Big|_{s^+} \right) \\ &= B \cdot [b] + r_2 = \ddot{\theta} \cdot \left( \frac{\partial r}{\partial \alpha_2} \times r \right) + r_2 \end{aligned} \tag{34}$$

so the  $f_n$  can be split into two parts

$$f_n = f_{nr} + f_{nq} \tag{35}$$

and  $f_{nr}$  and  $f_{nq}$  can be solved from the following Poisson equations with periodic boundary conditions after defining the Laplacian operator  $\Delta_\alpha$ .

$$\Delta_\alpha = \frac{\partial^2}{\partial \alpha_1^2} + \frac{\partial^2}{\partial \alpha_2^2} \tag{36}$$

$$\Delta_\alpha f_{nr} = \frac{\partial r_1}{\partial \alpha_1} + \frac{\partial r_2}{\partial \alpha_2} \tag{37}$$

$$\Delta_\alpha f_{nq} = \frac{\partial T \cdot [b]}{\partial \alpha_1} + \frac{\partial B \cdot [b]}{\partial \alpha_2} = \ddot{\theta} \cdot (\Delta_\alpha r \times r) = \ddot{\theta} \cdot (R^T (\Delta_\alpha r_0 \times r_0)) = \ddot{\theta} \cdot (R^T f_{n0}) \tag{38}$$

Though the right-hand side of the Poisson equation for  $f_{nq}$  is time-varying,  $f_{nq}$  can be obtained by solving a Poisson equation for  $f_{n0}$  at the reference time. Finally, we have

$$f_n = f_{nr} + \ddot{\theta} \cdot (R^T f_{n0}) \tag{39}$$

$$\rho = J(f_t + f_n n) = J f_t + f_{nr} N + (\ddot{\theta} \cdot (R^T f_{n0})) N \tag{40}$$

$$\begin{aligned} \int_S \rho \, d\alpha_1 \, d\alpha_2 &= \int_S (\ddot{\theta} \cdot (R^T f_{n0})) N \, d\alpha_1 \, d\alpha_2 - \int_S (J f_t + f_{nr} N) \, d\alpha_1 \, d\alpha_2 \\ &= \int_S (\ddot{\theta} \cdot (R^T f_{n0})) N \, d\alpha_1 \, d\alpha_2 - f_0 \\ &= R^T \left( \int_S N_0 f_{n0}^T \, d\alpha_1 \, d\alpha_2 \right) R \ddot{\theta} - f_0 \\ &= R^T Q_f R \ddot{\theta} - f_0 = A_f \ddot{\theta} - f_0 \end{aligned} \tag{41}$$

$$\begin{aligned} \int_S (y - y_c) \times \rho \, d\alpha_1 \, d\alpha_2 &= \int_S (\ddot{\theta} \cdot (R^T f_{n0})) ((y - y_c) \times N) \, d\alpha_1 \, d\alpha_2 \\ &\quad - \int_S (J(y - y_c) \times f_t + f_{nr} (y - y_c) \times N) \, d\alpha_1 \, d\alpha_2 \\ &= \int_S (\ddot{\theta} \cdot (R^T f_{n0})) ((y - y_c) \times N) \, d\alpha_1 \, d\alpha_2 - \tau_0 \\ &= R^T \left( \int_S ((y_0 - y_c) \times N_0) f_{n0}^T \, d\alpha_1 \, d\alpha_2 \right) R \ddot{\theta} - \tau_0 \\ &= R^T Q_\tau R \ddot{\theta} - \tau_0 = A_\tau \ddot{\theta} - \tau_0 \end{aligned} \tag{42}$$

Note that the matrices  $Q_f$  and  $Q_\tau$  need to be computed only once.

5.2. Derive incompressible viscous fluid force and torque on a rigid body

Then we derive the formulas for calculating the fluid force and torque on the “rigid fluid” object based on Newton dynamic equations, which read

$$m_i \ddot{y}_c = f_f + f_s + f_q \tag{43}$$

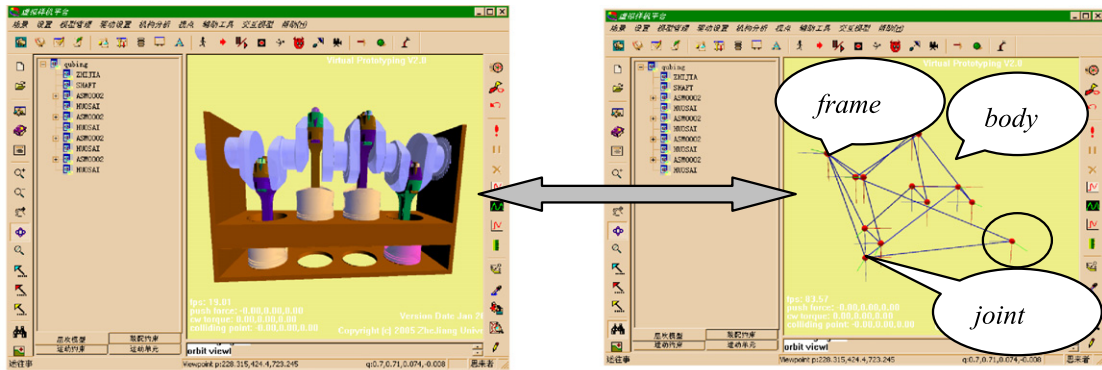
$$I_i \ddot{\theta}_c = \tau_f + \tau_s + \tau_q \tag{44}$$

where  $f_f$ ,  $f_s$ , and  $f_q$  ( $\tau_f$ ,  $\tau_s$ , and  $\tau_q$ ) are the forces (torques) on the “rigid fluid” due to the surrounding fluid, the boundary force, and the body force, respectively. The angular momentum and the torques are defined with respect to the center of mass of the “rigid fluid”. We use  $V$  denote the region occupied by the “rigid fluid” and we have

$$f_s = \int_S \rho \, d\alpha_1 \, d\alpha_2, \quad f_q = \int_V \ddot{\theta} \times (y - y_c) \, dV = 0 \tag{45}$$

$$\tau_s = \int_S (y - y_c) \times \rho \, d\alpha_1 \, d\alpha_2, \quad \tau_q = \int_V (y - y_c) \times (\ddot{\theta} \times (y - y_c)) \, dV \tag{46}$$

$$\begin{aligned} I_i \ddot{\theta}_c &= \int_V (y - y_c) \times \ddot{y} \, dV \\ &= \int_V (y - y_c) \times (\ddot{y}_c + \ddot{\theta} \times (y - y_c) + \dot{\theta} \times (\dot{\theta} \times (y - y_c))) \, dV \\ &= \tau_q + \int_V ((y - y_c) \cdot \dot{\theta}) ((y - y_c) \times \dot{\theta}) \, dV \end{aligned} \tag{47}$$



(a) Assembly and geometry

(b) Skeleton dynamics parameter information



(c) Simulation animation sequence

Fig. 5. Dynamics modeling and simulation results of an automotive engine.

After substituting expressions (45)–(47) into Eqs. (43) and (44), we obtain the formulas for the fluid force and torque on the body as

$$f_f = m_i \ddot{y}_c - \int_S \rho d\alpha_1 d\alpha_2 = m_i \ddot{y}_c - A_f \ddot{\theta} + f_0 \tag{48}$$

$$\begin{aligned} \tau_f &= \int_V ((y - y_c) \cdot \dot{\theta})(y - y_c \times \dot{\theta}) dV - \int_S (y - y_c) \times \rho d\alpha_1 d\alpha_2 \\ &= \left( R^T \left( \int_V (y_0 - y_c)(y_0 - y_c)^T dV \right) R \dot{\theta} \right) \times \dot{\theta} - A_\tau \ddot{\theta} + \tau_0 \end{aligned} \tag{49}$$

By introducing  $f_f$  and  $\tau_f$  as external generalized force into the right-hand side of Eq. (4) and combining with Eq. (5), we will have simulation results with fluid environment dynamics effect.

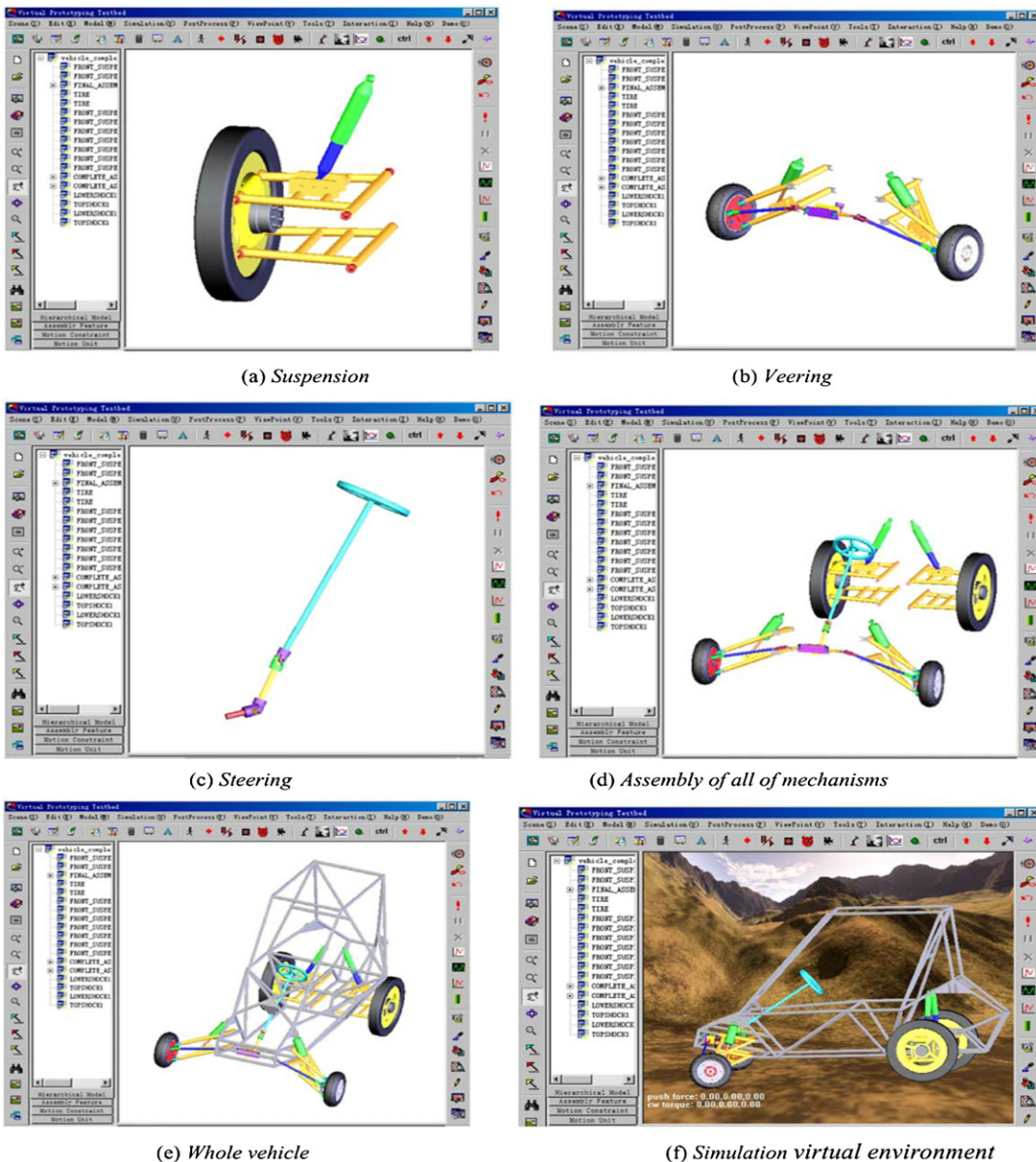


Fig. 6. Virtual prototyping of MiniBaja.

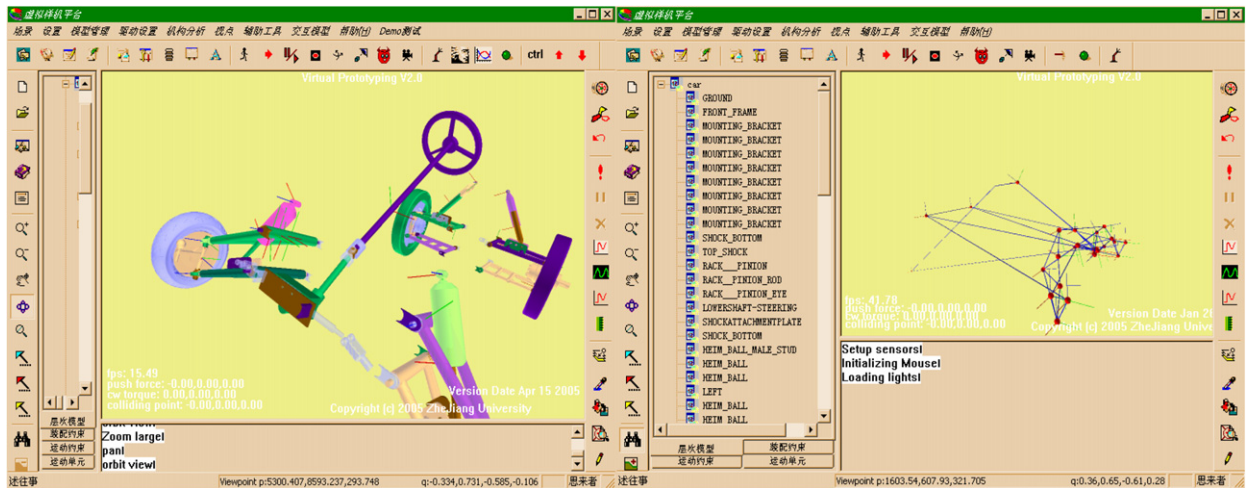


Fig. 7. Skeleton dynamics parameters information.

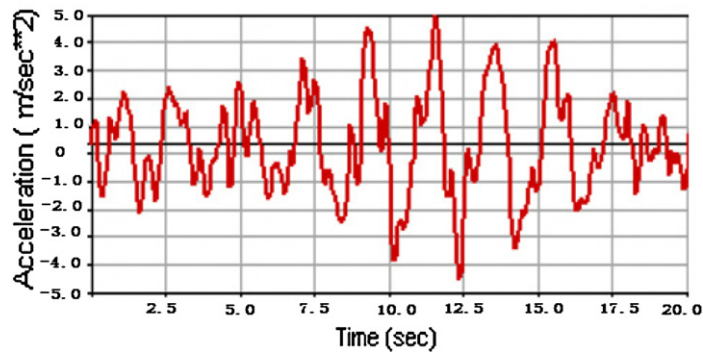


Fig. 8. Vertical acceleration of front wheel.

## 6. Application

### 6.1. An automotive engine

A classical analysis procedure for a slider crank mechanism of an automotive engine is presented in Fig. 5. Fig. 5(a) presents a geometrical model of this mechanism, which has been created and assembled in Pro/Engineer, and MotionInfo is used to extract its motion information including bodies, joints and frames etc, the topology of the mechanism skeleton as seen in Fig. 5(b). In Fig. 5(c), by inputting a valid driving force to the engine we get a sequence of animation.

### 6.2. MiniBaja

The *MiniBaja* project [34] is to design and build a prototype, four-wheeled, single-seater and off-road recreational vehicle, to meet given design specifications. The vehicle must be safe, easy to transport and service and fun to drive. It should be capable of negotiating rough terrain without damage. Apart from certain minimum safety requirements, the design is open to the initiative and ingenuity of the students.

We have built dynamics model for *MiniBaja* based on above mentioned method. The dynamics model of *MiniBaja* consists of two front suspensions, two rear suspensions, steering mechanism, veering mechanism and the tires. That makes a total of 32 rigid bodies. We can see our virtual mechanical system structure in Fig. 6, and the extracted skeleton dynamics parameters information represented in Fig. 7.

The dynamic behaviors of *MiniBaja* are easily obtained from the postprocessor module of our simulator during simulation. As shown in Fig. 8, given  $Re = 200$  for the environment, the curve demonstrates the change of vertical acceleration of front wheel when the vehicle running on an uneven terrain. Fig. 9 shows the vertical acceleration of vehicle body. The curve values in Fig. 9 ranges from  $-0.4$  to  $0.39$ , while the curve values in Fig. 8 ranges from  $-4.6$  to  $5.0$ . It clearly shows that the mean value of the body's acceleration is less than that of the wheel's, which reflects the reduction of the impact force between terrain and tire by the front suspension system in the vehicle. According to the curves, the design parameters of shock absorber in suspension system can be modified and simulated again so as to optimize the design of the vehicle.

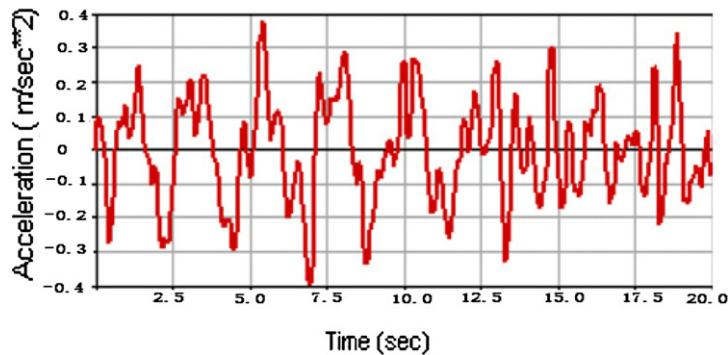


Fig. 9. Vertical acceleration of vehicle body.

## 7. Conclusion

We presented a simulation package for complex mechanical multi-body systems considering environment effect into dynamics model. The package incorporates physical behavior and a dynamic-based modeling approach in a mechanical system and fluid dynamics from immersed boundaries method; at the same time a compliant contact model has been built. As for extracting dynamics parameters from CAD model, we developed a utility named MotionInfo. Then two examples are implemented in the dynamic simulation of a car engine and *MiniBaja* to practice the above mentioned method, but without verifying the whole simulator as will be discussed in the second part of the article.

## Acknowledgement

The author will be grateful to the State Key Lab of CAD&CG in Zhejiang University for the original basis into this paper. We should also acknowledge the reviewers to the paper for a series of improvement suggestions.

## References

- [1] G. DeBunne, M. Desbrun, M.P. Cani, A. Barr, Dynamic real-time deformations using space time adaptive sampling, in: SIGGRAPH (2001), Computer Graphics Proceedings, 2001.
- [2] D. Pai, Strands: interactive simulation of thin solids using Cosserat models, in: Proceedings of Eurographics 2002, Computer Graphics Forum 21 (3) (2002) 347–352.
- [3] Z. Sun, J. Reif, On energy-minimizing paths on terrains for a mobile robot, in: Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA '03), Taipei, Taiwan, September 2003.
- [4] M.A. Srinivasan, C. Basdogan, Haptics in virtual environments: taxonomy, research status, and challenges, Computers and Graphics 21 (4) (1997) 393–404.
- [5] K. Hinckley, M. Sinclair, Touch-sensing input devices, in: Proceedings of the Sigchi Conference on Human Factors in Computing Systems, ACM Press, 1999, pp. 223–230.
- [6] M.C. Lin, M.A. Otaduy, Recent advances in haptic rendering & applications, in: SIGGRAPH Course 11, Los Angeles, CA, 2005.
- [7] D.S. Bae, E.J. Haug, A recursive formulation for constrained mechanical system dynamics: Part II, Closed loop systems, Mechanics of Structures and Machines 15 (4) (1987) 481–506.
- [8] J.E. Keat, Multi-body system order- $n$  dynamics formulation based on velocity transform method, Journal of Guidance, Control, and Dynamics 13 (2) (1990) 207–212.
- [9] D.E. Rosenthal, An order- $n$  formulation for robotic systems, Journal of the Astronautical Sciences 38 (4) (1990) 511–529.
- [10] K.S. Anderson, An order- $n$  formulation for the motion simulation of general constrained multi-rigid-body systems, Computers and Structures 43 (3) (1992) 565–579.
- [11] S. Pradhan, V.J. Modi, Order- $n$  formulation for flexible multi-body systems in tree topology: Lagrangian approach, Journal of Guidance, Control, and Dynamics 20 (4) (1997) 665–672.
- [12] A.K. Banerjee, Order- $n$  formulation of extrusion of a beam with large bending and rotation, Journal of Guidance, Control, and Dynamics 15 (1) (1992) 121–127.
- [13] A.K. Banerjee, S. Nagarajan, Efficient simulation of large overall motion of beams undergoing large deflection, Multi-Body System Dynamics 1 (1) (1997) 113–126.
- [14] A. Tasora, An optimized Lagrangian-multiplier approach for interactive multi-body simulation in kinematic and dynamical digital prototyping, in: International Symposium on Computer Simulation in Biomechanics, Politecnico di Milano, Italy, 2001, pp. 125–130.
- [15] D. Baraff, Analytical methods for dynamic simulation of non-penetrating rigid bodies, in: Proceedings of the ACM Computer Graphics Conference, 1989.
- [16] P. Song, P. Kraus, V. Kumar, P. Dupont, Analysis of rigid-body dynamic models for simulation of systems with frictional contacts, Journal of Applied Mechanics (2001) 118–124.
- [17] G. Hippmann, An algorithm for compliant contact between complexly shaped bodies, Multi-Body System Dynamics 12 (2004) 345–362.
- [18] <http://www.mscsoftware.com>, 2006.
- [19] E.J. Haug, D. Negrut, R. Serban, D. Solis, Numerical methods for high speed vehicle dynamic simulation, Mechanics of Structures and Machines 27 (4) (1999).
- [20] A. Stribersky, F. Moser, W. Rulka, Structural dynamics and ride comfort of a rail vehicle system, Advances in Engineering Software 33 (7–10) (2002) 541–552.
- [21] K. Andersson, U. Sellgren, Reality-driven virtual wheel loader operation, in: Proceedings of Virtual Concept 2005, Biarritz, France, 2005.

- [22] B. Damer, D. Rasmussen, P. Newman, et al., Design simulation of lunar exploration and ISRU prototype vehicles and mission scenarios, in: LEAG-SSR 2005, Conference, October 27, 2005.
- [23] I.H. Shames, *Mechanics of Fluids*, McGraw-Hill, New York, NY, 1962.
- [24] J. Yuh, Modeling and control of underwater robotic vehicles, *Systems, Man and Cybernetics, IEEE Transactions on* 20 (11–12) (1990) 1475–1483.
- [25] K. Ioi, K. Itoh, Modeling and simulation of an underwater manipulator, *Advanced Robotics* 4 (4) (1990) 303–317.
- [26] C.H.K. Williamson, R. Govardhan, Vortex-induced vibrations, *Fluid Mechanics* 36 (2004) 413–455.
- [27] MathWorks, CAD to SimMechanics Translators, <http://www.mathworks.com/products/simmechanics/description5.html>.
- [28] M. Cline, Rigid body simulation with contact and constraints, Master dissertation, University of British Columbia, BC Canada, 2002.
- [29] A. Shepherd, Analysis and training in information tasks, in: Dan Diaper (Ed.), *Task Analysis for Human–Computer Interaction*, Ellis Horwood, Chichester, 1989.
- [30] W.S. Howard, V. Kumar, A minimum principle for the dynamic analysis of systems with frictional contacts, in: *Proceedings of the 1993 IEEE Int'l Conference On Robotics and Automation*, New York, 1993, Publication of IEEE, New York, 1993, pp. 437–442.
- [31] G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.
- [32] Charles S. Peskin, Numerical analysis of blood flow in the heart, *Journal of Computational Physics* 25 (1997) 220–252.
- [33] Charles S. Peskin, Beth Feller Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, *Journal of Computational Physics* 105 (1993) 33–46.
- [34] <http://minibaja.olin.edu>, 2006.