Hybrid Modelling and Simulation of Rigid Bodies in Contact

Reza Pedrami

A Thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

September 15, 2005

Your file Votre référence
ISBN: 0-494-10270-5
Our file Notre référence
ISBN: 0-494-10270-5

Abstract

Reza Pedrami

Hybrid Modelling and Simulation of Rigid Bodies in Contact

Rigid body systems may undergo numerous types of dynamic interactions including collisions and continuous contact which considerably complicate systematic modelling and simulation of such problems. This thesis introduces a new modelling approach based on a hybrid system formulation to describe the dynamics of interacting rigid body systems. Interaction among physical objects occurs in two different forms: impulsive contact and continuous contact. Characteristics of impulsive and continuous contacts are different. Hence the modelling of each contact type requires the use of different approaches. While the impulse-modelling approach better simulates the impulsive contacts, our findings indicate that continues contact is much more accurately modeled using the sliding manifold method. The proposed hybrid system approach combines the impulse modelling method for collision interactions and the sliding manifold method to model the differential-algebraic equations associated with continuous contact interactions. Appropriate discrete states, events, reset maps, and threshold parameters that yield a hybrid automaton framework to describe interacting rigid body systems are developed. To illustrate the effectiveness of the proposed method a rolling ball simulation for a virtual reality system is presented.

Acknowledgments

This thesis was carried out in the Control and Information System Laboratory (CIS) at Concordia University, Montreal, Canada. I am grateful for having the opportunity to work and learn with the group of people from all corners of the world.

I would like to thank my advisors, Dr. Brandon Gordon and Dr. Ali Akgunduz, as they support me financially and intellectually.

I would like to express my gratitude to my parents, as they are never ending source of motivation and support. Without their love, support and encouragements this work was impossible.

To Mom and Dad

Table of Contents

List of Figures

Nomenclature

$\mathbf{q}$	Configuration space or position
$\mathbf{v}$	Tangent space or velocities
$\mathbf{z}$	Lagrange multiplier
w_i	Error variables
s_i	Sliding surfaces
μ_i	Positive constants
ε_i	Boundary layer thickness
$\mathbf{v}_s$	Control input
$\mathbf{J}_s$	Jacobian Matrix
$\mathbf{K}$	Gain Matrix
$d(t)$	Distance function
$\hat{\mathbf{n}}(t)$	Unit normal surface
$\mathbf{V}_c^i$	Contact point velocity
V_{ctx}, V_{cty}	Tangent contact velocities
V_{cn}	Normal contact velocity
V_{ci}	Linear velocity of mass centre
$\boldsymbol{\omega}_i$	Angular velocity
$\mathbf{P}$	Collision impulse

P_n	Normal component of collision impulse
γ	Collision Parameter
$\mathbf{p}_i$	Relative position vector of contact point
θ	The relative direction of sliding
C_r	Restitution Coefficient
C_f	Friction coefficient
$\mathbf{E}_l^i$	Local coordinate to global coordinate
$\mathbf{E}_n^{ij}$	Transformation matrix form the normal contact Coordinate between body i and body j to the global coordinate
$(\hat{\mathbf{t}}_x)_{ij}$ $\hat{\mathbf{n}}_{ij}$ $(\hat{\mathbf{t}}_z)_{ij}$	Unit normal vector of the normal coordinate
$\mathbf{z}_{ij}$	The contact force between body i and body j
$\mathbf{r}_{ij}$	The contact point between body i and body j with respect to local frame of body i
$\mathbf{a}_i$	Acceleration of body i
$w_{ij}^{np}, w_{ij}^{rx}, w_{ij}^{rz}$	Non-penetration and rolling constraints
n	Number of objects
N	Number of constraints
Π	Number of states
$\mathbf{Z}$	Extend the vector of variables
$\mathbf{v}'_k$	Control vector in mode k

$\mathbf{Z}'_{k}$	A vector of Lagrange multiplier in mode k
s'_{k}	Sliding manifolds in mode k
$(\mathbf{J}_{s})_{k}$	Jacobian Control in mode k
e_{1}^{ij}	Collision between the i^{th} and j^{th} body
e_{2}^{ij}	End of bouncing contact between the i^{th} and j^{th} body
e_{3}^{ij}	Vanishing of tangential velocity between the i^{th} and j^{th} body
e_{4}^{ij}	Separation between the i^{th} and j^{th} body
e_{5}^{ij}	Existence of tangential velocity between the i^{th} and j^{th} body
I_{ij}	Intermittent contact between the i^{th} and j^{th} body
Sl_{ij}	Slipping contact between the i^{th} and j^{th} body
Ro_{ij}	Rolling contact between the i^{th} and j^{th} body
Q_{1}^{ij}, Q_{2}^{ij} and Q_{3}^{ij}	Discrete states for model of contact between the i^{th} and j^{th} body

1 . Introduction

1.1 Motivation

Simulation and modelling is the process of making the computer into a device which imitates the physical world. The physical modelling of rigid body dynamic systems plays an important role for many applications such as computer animations, manufacturing, engineering processes, and the expanding area of virtual reality.

In the physical world, solid objects can interact with each other but not penetrate; unless there are physical deformations. The physics-based simulations, including real-time virtual reality applications, must handle the interaction between rigid bodies in contact. The goal is not only to produce an animation sequence that looks correct to a human, but also to calculate the forces that would prevent the bodies from interpenetrating.

Modelling of rigid bodies in contact is a complex problem due to the fact that different types of contacts between rigid bodies have their own physical properties. Although a large group of researchers has investigated the problem of modelling rigid bodies for each contact situation, there still exists an open area for the design and implementation of hybrid simulation systems which integrates all contact situations.

In the next sections, we briefly review the different contact configurations. The following discussion reviews the previous research for simulation of rigid bodies in contact. Also a corresponding mathematical characteristic of each contact situation is discussed in complete detail.

1.2 A Discussion on Different Contact Configurations

Two rigid bodies could be in different contact configurations. We can categorize them into two groups from a collision point of view:

- Non-colliding rigid bodies
- Colliding rigid bodies

In the first group, the focus is to calculate the forces between bodies that prevent them from interpenetration. The position and velocity of rigid bodies evolves in a smooth way with respect to time. In the case of collision, on the other hand, the change of velocity occurs on a much faster time scale than the time scale of the overall behaviour. In a more abstract way, this fast dynamic system can be modeled as an instantaneous jump in velocities upon collision. Rigid body systems show a very complicated behaviour due to discontinuities in velocities which are caused by colliding contacts. The problem of finding a proper reset map, which shows the relation between velocities before and after collision, is addressed as a collision response.

The first group implicitly implies that the rigid bodies are in permanent contact situation and also these continuous or permanent contacts can be categorized into three different types:

- Sticking contacts
- Rolling contacts
- Slipping contacts

It is known that the dynamics of a rigid body is characterized by the Euler-Lagrange equations. The Lagrange formulation constructs a set of ordinary differential equations. When two rigid bodies are in a continuous contact, the differential equations governing rigid bodies' motions are subjected to some algebraic equations. These algebraic equations are of different types such as geometric and kinematic. For instance, consider a ball rolling along a table. The motion of the ball centre should be located in a plane parallel with the table's surface. This is a geometric constraint. This geometric constraint is exactly equivalent to a non-penetration constraint. Ball motions may also be subject to some kinematic constraints such as rolling. This results in a system which is best described by a mixed set of differential and algebraic equations.

In the next section, we introduce two basic paradigms; impulse-based and constraint-based. Both methods are used by a large body of researchers to solve rigid bodies in contact.

1.2.1 Constraint-based Simulation via Impulse-based Simulation

The fact that no two bodies interpenetrate, impose unilateral constraints in dynamic simulation of rigid bodies. In consequence, a large body of researchers has proposed constraint-based approaches to simulate rigid-bodies in contact [1,2]. One of the early examples for the constraint-based approaches is the penalty method. In this method, a simple solution to separate penetrating bodies is presented by introducing arbitrary penalty forces. The constraint-based methods are very efficient in the case of continuous contact.

In contrast, impulse-based methods enforce no constraint on the configurations of the moving objects [3]. In this method, non-colliding contacts are modeled by a series of tiny micro collisions. The noteworthy advantage of the impulse-based method is its ability to unify all types of contacts under a single model [3]. This method is very efficient in the case of temporary collisions at a single contact point [4]. An analytical solution for calculating the impulsive forces between a single pair of bodies colliding at a single point is the basic problem of the impulse-based methods.

Designing and implementing a hybrid simulation system, which combines impulse-based methods and constraint-based methods, is a recent strategy to develop efficient simulators for the rigid body system in different contact situations [4,5]. In the next section, we briefly review the hybrid systems.

1.2.2 Hybrid Systems

The term hybrid system is used to describe a class of dynamical system that contains both continuous and discrete aspects. The focus on hybrid systems has increased significantly in recent years due to its capability of modelling wide ranges of engineering applications such as

- Manufacturing systems
- Traffic control systems
- Hierarchical control in process industry
- Electrical networks (circuits with diodes and switches)
- Mechanical systems with collision
- Embedded computation systems

The hybrid framework is ideal for modelling nonlinear physical systems, which include phenomena that occurs at multiple time scales. In these types of systems, the fast dynamics can be abstracted away and be treated as discrete changes affecting the slower dynamic [6].

A hybrid system is a mixture of continuous-time and discrete-event dynamics [6,7,8]. Therefore, the formulation of equations of the hybrid system motion consists of two main parts:

- Condition or event rules (for discrete event component)
- Evolution of continuous states in every possible modes

By combining differential equations and finite state automata, it is possible to describe a hybrid system. It is important to note that different set of differential equations describe the continuous dynamics in each hybrid systems mode. It means that dynamic structure of a system changes over the time in accordance with its event rules. In more details, the number of state variables and algebraic variables changes when switching among different modes of operation occurs. These systems are denoted as an evolvable structure. Frame works such as dynamic structure discrete event system specification and object oriented physical modelling (OOPM) are introduced for the representation of the systems with evolvable structure [9]. More recently, a hybrid system is developed to describe hybrid dynamics systems of variable structure with varying number of state and algebraic variables [10].

It is worthy to mention that a continuous dynamic in each mode can be described in different form of differential equations such as implicit/explicit Ordinary Differential Equations (ODE) and implicit/explicit Differential Algebraic Equations (DAE). DAEs are well suited when algebraic constraints on the variables exist [11]. In the next section, we discuss ODEs versus DAEs.

1.2.3 ODE versus DAE

In real life, when physical systems are modeled, several forms of relationships between variables of interest and some of their derivatives are generated. Therefore, the model takes the form of DAE. A variable usually has a physical meaning. Although changing a model to ODE format simplifies the solution of the differential equations, it may produce a meaningless variable as well. Solving ODEs are more convenient since several well-known techniques are available in the literature.

Obtaining direct solution of the DAEs has been the focus of research activities for over twenty years. The DAE systems, also known as descriptor, implicit, or singular systems are known to provide a more general description of dynamical systems than ODEs [12]. The DAE models can be utilized to model dynamic interactions among physical systems [13]. A wide variety of control methods applicable to the state space model, can be used for solving the DAE systems. This procedure, often referred to as DAE realization, has so far only been successful for limited classes of problems such as index one DAEs [14]. The index property of a DAE is a measure of the singularity of the system and can significantly affect the solution. Simulation of high index system is generally more difficult due to the numerical ill-conditioning [11,15].

1.3 Research Objectives

The dynamic of rigid body is characterized by Euler-Lagrange equations. The motion equations are subjected to some algebraic equations in a case of continuous contact interactions. These algebraic equations are of different types such as geometric and kinematics. Therefore, in continuous contact modes, the dynamics of rigid bodies are coupled together by corresponding algebraic constraint. Then, the system equations are described by a set of DAEs. A recent method known as the singularly perturbed sliding manifold (SPSM) approach is used to solve the realization problem associated with DAEs[16].

In the case of collision, change of velocity occurs on a much faster time scale than the time scale of overall behaviour. In a more abstract way, this fast dynamic system can be modeled as an instantaneous jump in the velocities upon the collision. Moreover, the rigid body system changes its dynamics over the time in accordance with its contact situation. For instance, a dynamic structure transform from ODEs (no contact) to DAEs (continuous contact). The number of state variables and algebraic variables change in accordance with the contact situation. A multiple time scale and evolvable structures conduct us to design a system with combination of continuous and discrete components.

The proposed hybrid dynamic system model combines an impulse modelling approach for collision interactions with a recently developed sliding manifold method to model the differential-algebraic equations associated with continuous contacts. Appropriate discrete states, events, reset maps, and threshold parameters of the generalized hybrid automaton are developed.

1.4 Literature Review

The problem of modelling rigid bodies in contact has been investigated separately for each mode of operation by a large body of researchers. The fact that no two bodies interpenetrate, impose unilateral constraint in dynamic simulation of rigid bodies. In consequence, a constraint-based approach is used to simulate rigid-bodies in resting contact (non-colliding) [1,2]. An analytic calculation of the forces between systems of rigid bodies in resting contact was developed. The collision is traced by observing relative motions of two bodies at each contact point in the direction of unit surface normal $\hat{\mathbf{n}}$. When the relative normal velocity is less than zero, the type of the contact is colliding contact. It is known that the relative normal velocity is zero for non-colliding contacts. To prevent inter-penetration, the relative acceleration in normal direction should be kept non-negative. It is apparent that the relative normal acceleration is a linear function of the contact force. The normal force, which exists in direction of unit surface normal, must be non-negative. The existence of either the normal force is zero or the relative normal velocity is zero introduces a quadratic objective function. Finally, the problem of computing forces is transformed to an optimization problem [1]. In a later work, the problem of computing forces as a linear complimentary problem (LCP) is considered. Consequently the need for complex optimization tool is eliminated. These methods are well suited for multiple contact points [2]. The drawback of these methods is that the collision impacts are handled separately instead of incorporating into the constraint model.

The earliest works in the collision response area present analytical solutions for calculating the impulsive forces between a single pair of bodies, which are colliding at a single point [17,18]. To eliminate the constraints on the configurations of the moving objects the impulse-based methods have been developed [3]. In this method, the non-colliding contacts are modeled by series of tiny micro collisions. The noteworthy advantage of the impulse-based method is its ability to unify all types of contacts under a single model [3]. This method is very efficient in the case of temporary collisions at a single contact point [4]. Modelling continuous contacts as series of tiny micro-collisions creates low amplitude high frequency vibrations. These vibrations generate a chattering behaviour in the normal relative velocity. The bound of chattering for normal velocity depends on the increment of integration during the impulse calculation. The impulse-based method is not well suited for continuous contact situation since tiny micro-collisions generate large cumulative errors during the simulation of them. An implicit time-stepping scheme is a recent impulse- based method. It is based on impulse-momentum equations. This method is distinct form the other impulse-based methods in: i) it does not require any explicit collision checking; ii) it can handle simultaneous collisions [19,20].

Designing and implementing a hybrid simulation system that combines impulse-based methods and constraint-based methods have not been studied much. Developing an efficient hybrid simulation is still an open problem [4,5]. Limited number of works in the area categorizes the physical systems to different classes of systems. Some of the physical systems are collision intensive. It means that collision play a major role in determining the dynamics. This class of system sometimes is denoted as collision intensive physical systems like part feeder system. On the other hand, some of physical systems are subject to permanent such as prismatic joint and revolute. It is clear that the constraint-based approach is the proper tool for such systems. Robotic systems and articulated rigid links are good example of such systems. It is important that so many physical applications are located somewhere between these two classes of the physical systems. In the third class of the physical systems, collision and constraints are important to determine the dynamic. These physical systems are denoted as transient contacts. Examples of latter class are billiard, bowling, interactive environments and mechanisms [5].

Moreover, the contact evolution problem has been studied recently. This strategy, which is suitable for the continuous contact configurations, falls into the constraint-based algorithms. The goal is to determine how a contact point evolves continuously over the time. The work focuses on the dynamics of piecewise smooth surface in a single surface-to-surface contact. The algorithm is based on using the reduced coordinate to evolve a single continuous contact between smooth parametric surfaces [21].

A recent hybrid impulse-based strategy combines two different strategies to calculate the impulse. A different scheme is used to calculate the impulses for different contact situations. Previously known impulse-based method [22] is used to compute the force in bouncing or temporary contact situations and implicit time-stepping scheme is used for steady or permanent contacts.

1.5 Thesis Contribution and the Outline of the Thesis

We first review some background material in the second chapter. A quick overview of DAE systems and high index DAE systems is then presented. Moreover, we show modelling of rigid body systems in the form of DAEs using Lagrange multipliers. Most importantly, a DAE realization process is developed by formulating a nonlinear control problem. In addition, a singularly perturbed sliding manifold approach is presented. Finally, we review impulse-momentum equations and a way to calculate collision impulse in the presence of friction.

In the third chapter, the definition of different contact situations is given. Then, a hybrid simulation system that combines impulse-based methods and constrained-based methods to simulate contact between two rigid bodies in different contact situations is presented. Representation of our simulator by using hybrid automaton framework is considered as the main contribution in this chapter since it opens a new mathematical way to analyze the rigid bodies in contact. Using the hybrid design, our simulator not only keeps the advantage of both constraint-based and impulse base methods but also eliminates some of their weakness such as elimination of the chattering of the relative normal velocity (in a case of continuous contact).

Our contribution in the fourth chapter is to generalize our hybrid scheme in a case that more than two rigid bodies can be present in the simulation. This characteristic enables the simulator to handle multi-contact problems. Firstly, a systematic approach to model a system of rigid bodies is introduced by utilizing synchronous product of automaton framework. The generalized model has different number of algebraic constraints in each mode and this multi-resolution nature of the problem is investigated. Therefore, a solution to multi- resolution problem under hybrid automaton framework is presented. This discussion is followed by a proper example. Moreover, as our design highly increases the complexity of the system, we propose a model reduction algorithm. Finally, we illustrate a systematic approach to eliminate the Zeno problem from our model.

The contributions in this thesis are summarized as follows:

- A new hybrid simulation approach for modelling multiple rigid bodies in contact under a hybrid automaton framework
- The approach combines the efficiency of the impulse-based method for intermittent contact and the SPSM approach for continuous contact
- A physics based hybrid model reduction algorithm is proposed to reduce model complexity
- A method is proposed to eliminate Zeno Behaviour

This thesis represents the first hybrid modelling approach that combines DAE constraint stabilization (SPSM) with the impulse method. The proposed approach is the first hybrid modelling approach suitable for hard real-time virtual reality applications.

2 . Background Material

2.1 DAE Modelling of Rigid Body Systems

In this section a brief review of DAE and rigid body systems will be given. The interested reader should refer to [11] for more details on the materials presented in the next subsections.

2.1.1 High Index DAE Systems

The most general DAE could be shown as the mixed set of implicit differential and algebraic equations given by:

$$\mathbf{F}(t, \mathbf{x}, \dot{\mathbf{x}}) = \mathbf{0} \tag{1}$$

where $\mathbf{x} \in \Re^n$, $\mathbf{F} : \Re \times \Re^n \times \Re^n \to \Re^n$. For a DAE system $[\partial \mathbf{F} / \partial \dot{\mathbf{x}}]$ is singular so that some of the state variables are algebraically constrained. The DAE system could be rewritten in the semi-explicit form as:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{z}) \tag{2}$$

$$\mathbf{0} = \mathbf{g}(t, \mathbf{x}, \mathbf{z}) \tag{3}$$

where $\mathbf{x} \in \Re^{n}$, $\mathbf{z} \in \Re^{m}$, $\mathbf{f}: \Re \times \Re^{n} \times \Re^{m} \rightarrow \Re^{n}$, and $\mathbf{g}: \Re \times \Re^{n} \times \Re^{m} \rightarrow \Re^{m}$. It is assumed that equations (2) and (3) are sufficiently differentiable and that a well-defined solution for **x** and **z** exists with consistent initial conditions. Under these assumptions an important structural property of a DAE known as the index can be defined [11].

Definition 1. The minimum number of times that all or part of the constraint equations (3) must be differentiated with respect to time in order to solve for $\dot{\mathbf{z}}$ as a continuous function of t, **x**, and **z** are the *index* of the DAE (2-3).

Calculating the differentiation gives:

$$\mathbf{0} = \frac{\partial \mathbf{g}}{\partial t} + \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\mathbf{f} + \frac{\partial \mathbf{g}}{\partial \mathbf{z}}\dot{\mathbf{z}} \tag{4}$$

It is clear that if the Jacobian [$\partial \mathbf{g} / \partial \mathbf{z}$] is nonsingular then it is possible to solve for $\dot{\mathbf{z}}$ and the system has an index of one. For high index systems (index > 1), the Jacobian is not invertible and the constraint equations are identically singular with respect to **z**. Such high index problems are common in many control applications [11]. A constrained rigid body system is a common example of a high index DAE system. In the following sub-section, we give a quick review on the Lagrange formulation for the equations of motion.

2.1.2 Lagrange Formulation

Consider a mechanical system subject to some algebraic constraints $\mathbf{h}(\mathbf{q})$:

$$h_i(\mathbf{q}) = 0, \quad i = 1,2,3,\dots,k < n \tag{5}$$

where $\mathbf{q}$ is the position. Constraints that are given as in equation (5) are called holonomic constraints. Since they depend only on the position variables, they are also called geometric constraints. The rigid body motion of such mechanical systems, which are subject to some geometric constraints, can be formulated in the form of Euler-Lagrange equations as shown below:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{6}$$

$$\frac{d}{dt}\frac{\partial}{\partial \mathbf{v}} L(\mathbf{q},\mathbf{v}) = \frac{\partial L}{\partial \mathbf{q}} + \mathbf{f}(\mathbf{q},\mathbf{v},t) + \mathbf{G}^T\mathbf{z} \tag{7}$$

$$0 = \mathbf{h}(\mathbf{q}) \tag{8}$$

where $\mathbf{G} = \partial\mathbf{h}/\partial\mathbf{q}$, L is Lagrangian, $\mathbf{z}$ is Lagrange multiplier, $\mathbf{v}$ is velocity , $\mathbf{f}(\mathbf{q},\mathbf{v},t)$ is external forces and t is the time. Equations (6-8) can be rewritten in DAE forms as:

$$\dot{\mathbf{q}} = \mathbf{v} \tag{9}$$

$$\frac{\partial^2 L}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q},\mathbf{v},t) + \mathbf{G}^T\mathbf{z} \tag{10}$$

$$0 = \mathbf{h}(\mathbf{q}) \tag{11}$$

where $\mathbf{f}'$ is given by:

$$\mathbf{f}'(\mathbf{q},\mathbf{v},t) = \frac{\partial L}{\partial \mathbf{q}} + \mathbf{f}(\mathbf{q},\mathbf{v},t) - \frac{\partial^2 L}{\partial \mathbf{q}\partial \mathbf{v}}\mathbf{v} \tag{12}$$

This DAE system in the unknown variables $\mathbf{v}, \mathbf{q}$ and $\mathbf{z}$ is linear in derivatives [23]. Equation (5) can be differentiated with respect to time t until the acceleration $\dot{\mathbf{v}}$ is obtained. It is shown that the resulting equation is linear in acceleration.

$$0 = \mathbf{h}(\mathbf{q}) \tag{13}$$

$$0 = \frac{\partial \mathbf{h}}{\partial \mathbf{q}}\dot{\mathbf{q}} = \mathbf{G}\mathbf{v} = \overline{\mathbf{h}}(\mathbf{q}, \mathbf{v}) \tag{14}$$

$$0 = \frac{\partial \overline{\mathbf{h}}}{\partial \mathbf{v}}\dot{\mathbf{v}} + \frac{\partial \overline{\mathbf{h}}}{\partial \mathbf{q}}\dot{\mathbf{q}} = \frac{\partial \overline{\mathbf{h}}}{\partial \mathbf{v}}\dot{\mathbf{v}} + \frac{\partial \overline{\mathbf{h}}}{\partial \mathbf{q}}\mathbf{v} \tag{15}$$

Therefore, the DAE realization for the holonomic case is of third index type if and only if the full rank condition for $\mathbf{G} = \frac{\partial \overline{\mathbf{h}}}{\partial \mathbf{v}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}}$ is satisfied [23].

System constraints whose expression involves positions and velocities in the following form:

$$\mathrm{a}_{\mathrm{i}}(\mathbf{q}, \mathbf{v}) = 0 \quad , \mathrm{i} = 1,2,\ldots,\mathrm{k} < \mathrm{n} \tag{16}$$

are referred as kinematic constraints. In rigid body dynamics, such constraints are usually encountered in the Pfaffian form:

$$\mathrm{a}_{\mathrm{i}}^{\mathrm{T}}(\mathbf{q})\mathbf{v} = 0, \quad \mathrm{i} = 1,2,\ldots,\mathrm{k} < \mathrm{n} \tag{17}$$

that is linear in the velocities. Although, the holonomic or geometric constraints imply the existence of kinematic constraints expressed as $(\partial \mathbf{h}/\partial \mathbf{q})\mathbf{v} = 0$, the converse is not necessarily true: it may happen that the kinematic constraints in equation (17) are not integrable and cannot be written in the form of equation (5). Therefore the mechanical system itself is called nonholonomic [24]. For instance, when two rigid bodies are rolling relative to one another, the constraint is a nonholonomic type. The fact that nonholonomic constraints may be realized as friction forces can help us to model rolling friction. In consequence, the DAE realization for the nonholonomic case is second index type if and only if the full rank condition $\partial\mathbf{a}/\partial\mathbf{v}$ is satisfied. The DAE system with kinematics constraints can then be expressed as:

$$\frac{\partial^2 \mathrm{L}}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q},\mathbf{v},\mathrm{t}) + \mathbf{a}(\mathbf{q})\mathbf{z} \tag{18}$$

$$\dot{\mathbf{q}} = \mathbf{v} \tag{19}$$

$$0 = \mathbf{a}^{\mathrm{T}}(\mathbf{q})\mathbf{v} \tag{20}$$

Finally a mechanical system with both geometric and kinematics constraints can be formulated as:

$$\frac{\partial^2 \mathrm{L}}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q},\mathbf{v},\mathrm{t}) + \begin{bmatrix}\mathbf{G}\\ \mathbf{a}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}\mathbf{z} \tag{21}$$

$$\dot{\mathbf{q}} = \mathbf{v} \tag{22}$$

$$0 = \mathbf{h}(\mathbf{q}) \tag{23}$$

$$0 = \mathbf{a}^{\mathrm{T}}(\mathbf{q})\mathbf{v} \tag{24}$$

In order to solve the rigid bodies in continuous contact, forming DAE systems shown above is essential.

2.2 Singularly Perturbed Sliding Manifolds

2.2.1 Modelling of High Index DAEs using Nonlinear Control

In general a semi explicit DAE is represented by equations (2,3). Nonlinear functions $\mathbf{f}$ and $\mathbf{g}$ respectively represent the ODE part and algebraic constraints of the DAE [11]. The mechanical system formulated in equations (21-24) is presented in a semi explicit DAE form. The semi explicit DAEs can be solved by sliding-manifold method [14,16,25].

A new approach for high index systems was developed in this section by formulating a nonlinear control problem that is equivalent to the realization process of the DAE system. This new connection is expressed in the following proposition.

Proposition 1. Consider the SISO nonlinear control problem defined by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(t,\mathbf{x},\mathbf{z}) \\ \dot{\mathbf{z}} &= \mathbf{v}_s \\ \mathbf{w} &= \mathbf{g}(t,\mathbf{x},\mathbf{z}) \end{aligned} \tag{25}$$

where $\mathbf{v}_s$ is an artificial input that drives $\mathbf{z}$ which is now assumed to be independent, and $\mathbf{w}$ is an output equal to the violation of the constraint equations. Then the relative degree of this problem is equal to the index of the DAE, and the zero dynamics represent the dynamics of the high index DAE.

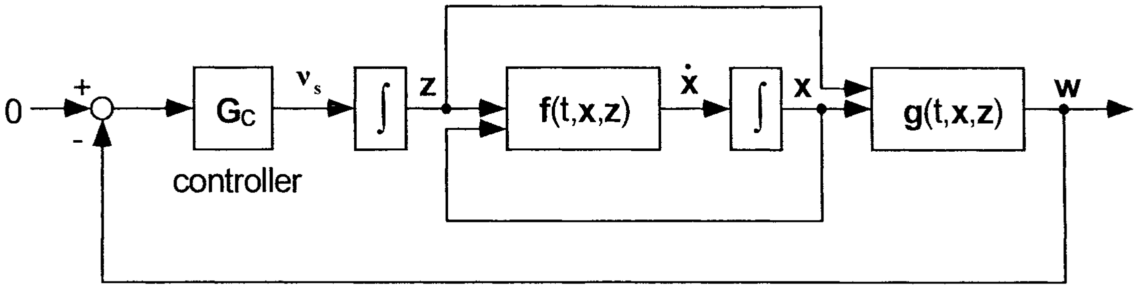


Figure 1. Block diagram of DAE realization approach

Figure 1 illustrates the block diagram of the nonlinear control system (25) associated with the DAE. The output of the system, which represents the violation of the algebraic equation $\mathbf{g}(t, \mathbf{x}, \mathbf{z})$, is fed back to create an input $\mathbf{v}_s$ that drives the output to zero. Thus, the realization of a high index DAE system can be interpreted as finding a controller that forces the defined output to zero.

2.2.2 DAE Realization of Rigid Body Systems

We start the application of sliding-manifold method by introducing the following error variables w_i as:

$$w_i = h_i(x) - h_i^d(x), \qquad i = 1,2,...k < m \tag{26}$$

$$w_j = a_j^T(x)u - a_j^{T^d}(x)u, \qquad j = k, k+1,...m \tag{27}$$

In general, each $\mathbf{w}$ may need to be differentiated a different number of times $(r_i - 1)$ for some components of $\mathbf{z}$ to appear. For the first k variables that represent geometric constraints, assume that $\mathbf{z}$ terms appear in $\ddot{\mathbf{w}}$ and for the remaining variables that represent nonholonomic constraints, assume that $\mathbf{z}$ terms appear in $\dot{\mathbf{w}}$. The appropriate choice of the state variables leads to the accuracy of our assumption [23]. Therefore, the sliding manifold is constructed as:

$$s_i = \mu_i^2 \ddot{w}_i + 2\mu_i \dot{w}_i + w_i, \quad i = 1,2,\ldots,k < m \tag{28}$$

$$s_j = \mu_j \dot{w}_j + w_j, \qquad j = k+1,\ldots,m \tag{29}$$

where μ_i is a positive parameter that determines the dynamics of the fast motion and s_i, s_j denote the sliding manifolds corresponds to the geometric and kinematic constraints. Hence the SPSM method designs a controller that forces the motion to satisfy all the kinematic and geometric constraints by a certain amount of error ε_i. Note that there exists other approaches for constraint stabilization, similar to proportional derivative control, such as Baumgart's method [11]. However, in this thesis we select the SPSM method due to its robustness properties and capability to allow more computationally efficient approximations than the other approaches.

In order to achieve the above goals we design a sliding controller that determines the value of the control input, which is $\mathbf{v}_s = \dot{\mathbf{z}}$. Next, the value of the Lagrange multipliers $\mathbf{z}$ is obtained by integrating $\mathbf{v}_s$. By differentiating s_i with respect to time we obtain:

$$\dot{s}_i = \mu_i^2 \dddot{w}_i + 2\mu_i \ddot{w}_i + \dot{w}_i, \quad i = 1,2,\ldots,k < m \tag{30}$$

$$\dot{s}_j = \mu_j \ddot{w}_j + \dot{w}_j, \qquad j = k+1,\ldots,m \tag{31}$$

$$\dddot{w}_i = \frac{\partial \ddot{w}_i}{\partial \mathbf{z}} \dot{\mathbf{z}} + \sigma_i, \quad i = 1,2,\ldots,k < m \tag{32}$$

$$\ddot{w}_j = \frac{\partial \dot{w}_j}{\partial \mathbf{z}} \dot{\mathbf{z}} + \sigma_j, \quad j = k+1, \ldots, m \qquad (33)$$

Now, the Jacobian matrix can be defined as:

$$\mathbf{J}_s^i = \mu_i^2 \frac{\partial \ddot{w}_i}{\partial \mathbf{z}}, \qquad i = 1,2,\ldots,k < m \qquad (34)$$

$$\mathbf{J}_s^j = \mu_j \frac{\partial \dot{w}_j}{\partial \mathbf{z}}, \quad j = k+1, \ldots, m \qquad (35)$$

The $\dot{\mathbf{s}}$ vector is packed in the following form:

$$\dot{\mathbf{s}} = \mathbf{J}_s \mathbf{v}_s + \boldsymbol{\alpha} \qquad (36)$$

where

$$\alpha_i = \mu_i^2 \sigma_i + 2\mu_i \ddot{w}_i + \dot{w}_i, \qquad i = 1,2,\ldots,k < m \qquad (37)$$

$$\alpha_j = \mu_j \sigma_j + \dot{w}_i, \qquad j = k+1, \ldots, m \qquad (38)$$

If the above equation is solved for $\mathbf{v}_s$ then we can steer the sliding motion into the desired boundary layer. Since determining the exact values of $\boldsymbol{\alpha}$ and $\mathbf{J}_s$ is computationally expensive, they are approximated by $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{J}}_s^{-1}$ respectively. It has been shown that if the $\mathbf{v}_s$ is computed by the following sliding controller:

$$\mathbf{v}_s = -\hat{\mathbf{J}}_s^{-1} \left(\hat{\boldsymbol{\alpha}} + \mathbf{K} \cdot \mathrm{diag}\left[\mathrm{sat}\left(\frac{s_i}{\varepsilon_i} \right) \right] \right) \qquad (39)$$

$$\mathrm{sat}(u) = \begin{cases} u & |u| < 1 \\ \mathrm{sign}(u) & |u| \geq 1 \end{cases} \qquad (40)$$

and the sliding conditions are satisfied then the motion will converge to its desired error boundary within a short reaching phase. These results are expressed by the following theorem.

Theorem 1. Assume that $\mathbf{\Gamma} = \left|\mathrm{diag}[\boldsymbol{\alpha} - \mathbf{J}_s\hat{\mathbf{J}}_s^{-1}\hat{\boldsymbol{\alpha}}\right|$. If the matrix of $\mathbf{J}_s\hat{\mathbf{J}}_s^{-1}\mathbf{K} - \mathbf{\Gamma}$ is diagonally dominant then a sliding condition is satisfied and it guarantees the following explicit error bounds

$$|s_i| < \varepsilon_i, \qquad i = 1,2,..m \tag{41}$$

$$\left|\frac{d^j w_i}{dt^j}\right| \le \frac{2^j \varepsilon_i}{\mu_i}, \quad i = 1,2,..,m \tag{42}$$

Proof. See reference [25].

2.3 Impulse Momentum Equations

There is an essential need to formulate the equations in an impulse momentum format at the moment of collision. This formulation helps us to calculate impulses caused by the collision. To improve the performance of the simulator, the friction impulse during the collision is taken into consideration. In addition to the coefficient of restitution, the friction is also attempted to be included in the formulation to improve the flexibility of the simulator.

2.3.1 Assumption for Calculating Impulse

The following assumptions must hold in order to achieve the proposed simulation:

- Infinitesimal collision time
- Poisson's hypothesis
- Coulomb Friction

Infinitesimal collision time assumption implies that the duration of a collision is very short. In consequent, the configuration of the colliding bodies can be assumed constant during the collision. Hence the impulsive force can instantaneously affect the velocities according to the fact that the fundamental dynamical axiom is the axiom of bounded momentum. In a more abstract way, this fast dynamic system can be modeled as an instantaneous jump in velocities upon collision.

The Poisson's hypothesis mentions that the collision has two phases: compression and restitution. Let P_{total} be the magnitude of the normal components of the impulse throughout the collision. Let P_{mc} be the part of the normal component of the impulse during the compression phase. It is important to notify that the point of the maximum compression impulse is simply the point at which the normal component relative contact velocity vanishes. According to Poisson's hypothesis

$$P_{total} = (1 + C_r)P_{mc} \tag{43}$$

where C_r is a constant between zero and one, depends on the object's material, and it is denoted as a coefficient of restitution as shown in figure 2.

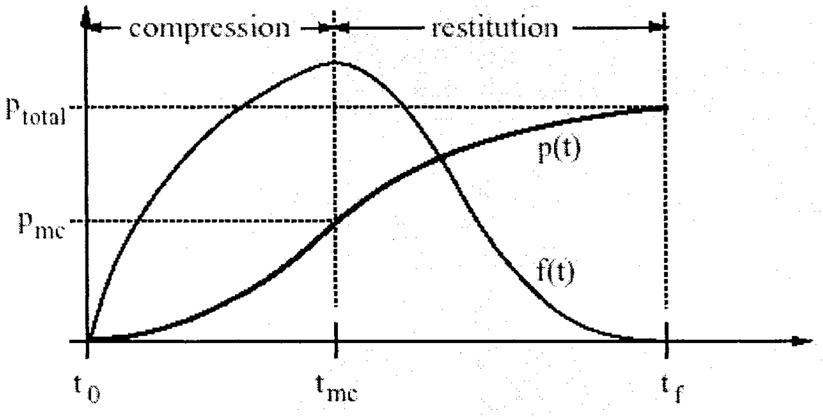


Figure 2. Compression and restitution phase of Poisson's Theorem [3]

Let ΔV_c be the relative contact velocity, ΔV_c^t be the tangential component of ΔV_c, and $\hat{u}_t$ be the unit vector in the direction of ΔV_c^t. Let F_f^n and F_f^t be the normal and the tangential (frictional) components of the force exerted. Coulomb Friction law implies that:

$$\begin{cases} F_f^t = C_f|F_n|, & \Delta V_c^t \neq 0 \\ F_f^t \leq C_f|F_n|, & \Delta V_c^t = 0 \end{cases} \tag{44}$$

where C_f is the coefficient of friction.

2.3.2 Impulse Calculation

Consider a collision between body 1 and body 2, as shown in figure 3. The contact velocities are:

$$\mathbf{V}_{c}^{i} = \mathbf{V}_{ci} + \boldsymbol{\omega}_{i} \times \mathbf{p}_{i} \tag{45}$$

The relative contact velocity $\Delta\mathbf{V}_{c}$ is simply $\mathbf{V}_{c}^{2} - \mathbf{V}_{c}^{1}$. If the $\hat{\mathbf{n}}$-component of $\mathbf{V}_{c}$ is negative, then the objects are colliding and the impulse computation is required. Collision impulse must be applied to find the new velocities after the collision. A new parameter, γ (collision parameter), was introduced to model the dynamics of the bodies during the collision. The collision parameter is a variable, which starts at zero, and continuously increases during the collision until it reaches to some final value γ_{f}. The collision parameter can be chosen as the normal component of the impulse, $\gamma = P_{n}$. All the velocities are functions of γ, where $\mathbf{P}(\gamma)$ is the impulse. The goal is to determine the total impulse($\mathbf{P}(\gamma_{f})$) and to reset the velocities.

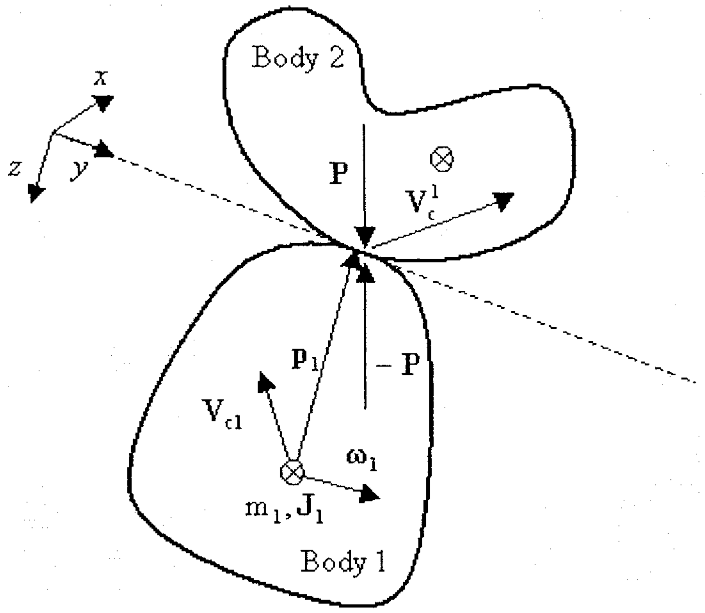


Figure 3. possible collision between rigid bodies

The linear and angular velocities for the first body can be described as a function of collision parameter γ:

$$\Delta \mathbf{V}_{c1} = \frac{1}{m_1} \mathbf{P}(\gamma) \tag{46}$$

$$\Delta \boldsymbol{\omega}_1 = \mathbf{J}_1^{-1}[\mathbf{r}_1 \times \mathbf{P}(\gamma)] \tag{47}$$

In consequent, the change in the contact velocity is defined as:

$$\Delta \mathbf{V}_c^1 = \left[\frac{1}{m_1} \mathbf{I} - \mathbf{r}_1^{\times} \mathbf{J}_1^{-1} \mathbf{r}_1^{\times} \right] \mathbf{P}(\gamma) \tag{48}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{r}_1^{\times}$ is the canonical 3×3 skew-symmetric matrix corresponding to $\mathbf{r}_1$. After carrying a similar analysis on the second body, the relative contact velocity is defined as:

$$\Delta \mathbf{V}_c(\gamma) = \mathbf{MP}(\gamma) \tag{49}$$

where $\mathbf{M}$ is a 3×3 matrix dependent only on the masses, the moment of inertia of the colliding bodies, and the location of the contact point relative to their centre of masses [13]. By the first assumption, $\mathbf{M}$ is constant over the entire collision. We can differentiate the above equation w.r.t. the collision parameter to obtain:

$$\mathbf{V}_c'(\gamma) = \mathbf{MP}'(\gamma) \tag{50}$$

Lemma 1. If the collision parameter γ is chosen to be P_n then while the bodies are sliding relative to one another P' is calculated as:

$$\mathbf{P}' = \begin{bmatrix} -\mathrm{C}_\mathrm{f} \cos\theta & -\mathrm{C}_\mathrm{f} \sin\theta & 1 \end{bmatrix}^\mathrm{T} \tag{51}$$

where $\theta(\gamma)$ is the relative direction of sliding during the collision.

Proof. See reference [3].

Finally, we obtain a nonlinear differential equation for V_c which is valid as long as the bodies are sliding relative to each other:

$$\begin{bmatrix} V'_{ctx} \\ V'_{cty} \\ V'_{cn} \end{bmatrix} = \mathbf{M} \begin{bmatrix} -C_f \frac{V_{ctx}}{\sqrt{V_{ctx}^2 + V_{cty}^2}} & -C_f \frac{V_{cty}}{\sqrt{V_{ctx}^2 + V_{cty}^2}} & 1 \end{bmatrix}^T \tag{52}$$

The basic impulse calculation algorithm proceeds as follows:

- Compute the initial $\mathbf{V}_c$
- Verify that V_{cn} is negative
- Numerically integrate $\mathbf{V}_c$ by using Eq. (48)

During this integration, V_{cn} will increase until it becomes zero where the point of maximum compression is reached. Then, the terminating value for the collision parameter γ_f is computed by multiplying the total normal impulse P_n during compression phase by $(1 + C_r)$. The integration continues until the terminating value is reached. At this point $\mathbf{P}(\gamma_f)$ is the desired total collision impulse. Finally, new angular and linear velocities are computed according to impulses.

3 . Single Contact Hybrid Simulation

We are introducing a hybrid automaton, which includes two modes of operation: i) continuous contact and ii) intermittent contact. In continuous contact mode, rigid bodies' dynamics are coupled together by constructing an algebraic constraint and the simulator model becomes a set of DAEs. On the other hand, the simulator model is described by a set of ODEs in the intermittent contact mode. Briefly, the rigid body system changes its dynamic over the time in accordance with its contact situation. A dynamic structure transforms from ODEs (no contact) to DAEs (continuous contact).

A recent method known as the singularly perturbed sliding manifold (SPSM) approach is used to solve the realization problem associated with the modelling of rigid bodies systems in continuous contact. A proper reset map is essential to reinitialize the state variables of the set of ODEs whenever the colliding contact makes jumps in the rigid bodies' velocities in intermittent contacts. In the first subsection, we give the definitions for different contact situation.

3.1 Contact Definitions

The Collision detection or the distance computation is one of the essential parts of the problem. A collision detection algorithm must be global; that is, they must report the Euclidean distance between two rigid bodies at each frame.

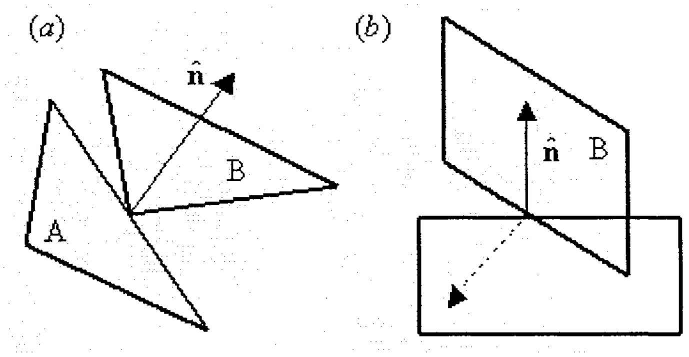


Figure 4. (a) Vertex-Plane contact (b) Edge-Edge contact

Definition 2. The shortest distance between two rigid bodies A and B is the Euclidean distance:

$$d(t) = \inf_{p\in A, q\in B} |\mathbf{p} - \mathbf{q}| \tag{53}$$

where $\mathbf{p}$ and $\mathbf{q}$ are the positions of the two arbitrary points at A and B respectively. The above equation could be rewritten as:

$$d(t) = \hat{\mathbf{n}}(t) \cdot (\mathbf{p}(t) - \mathbf{q}(t)) \tag{54}$$

where $\hat{\mathbf{n}}(t)$ is the unit normal surface. A normal surface direction can be defined as the outward unit surface normal of a plane for vertex-plane contact (see figure 4.a) and $\hat{\mathbf{n}}$ is defined as the unit vector mutually perpendicular to the two contacting edges for the edge-to-edge contact case (see figure 4.b). The distance function is utilized to examine the relative motion of two solid objects. The positive distance between objects ($d(t) > 0$) specifies that there is no contact between two bodies. The negative distance ($d(t) < 0$) is not acceptable and indicating inter-penetration. Also the zero distance ($d(t) = 0$) defines that two bodies are in contact. The relative velocity defines the types of contact when two bodies are in contact.

The interpenetration is not acceptable in rigid body simulations but some sort of tolerance is always acceptable to overcome the computational errors during the simulation. This condition can be expressed as:

$$\mathrm{d(t)} > -\mathrm{T}_{\mathrm{i}} \qquad (55)$$

This condition must be valid throughout the simulation. Two rigid bodies are in contact when the distance between two rigid bodies is below an acceptable threshold; that is mathematically expressed as:

$$d(t) < \mathrm{T}_d \qquad (56)$$

To handle the contact problem, we use the hybrid scheme. Motion equations for each mode are depending on the contact situation. To complete our hybrid simulation strategy, a report for change of contact situation should be generated. Later, we define different events that are responsible to report those changes. We can divide the contact points into to two types: intermittent and continuous.

Definition 3. Two rigid bodies are in *continuous contact* at time t_0 when they remain in contact at the next simulation step $t_0 + \Delta t$. Mathematically it means that $d(t_0)$ and $\dot{d}(t_0)$ are approximately zero.

It is assumed that the simulation time step Δt has been selected previously so that the simulation error is sufficiently small. Since the other simulation parameters and definitions depend on this parameter it should be selected first and the other parameters after that. There are different types of continuous contacts between rigid bodies that can be listed as follows:

1- Rolling Contact

2- Slipping Contact

3- Sticking Contact

In order to recognize each contact type, we must observe the relative tangential velocity between two rigid bodies at the contact points. The relative contact velocity is expressed as:

$$V_c = \sqrt{V_{ctx}^2 + V_{cty}^2} \tag{57}$$

where V_{ctx} and V_{ctz} defined as:

$$\begin{aligned} V_{ctx} &= \hat{t}_x(t).(\dot{p}(t) - \dot{q}(t)) \\ V_{ctz} &= \hat{t}_z(t) \cdot (\dot{p}(t) - \dot{q}(t)) \end{aligned} \tag{58}$$

Two bodies are slipping on each other when the velocity is not zero $V_c \neq 0$. Then, the friction force can be modeled according to Coulomb's law as:

$$\mathbf{F}_f = -C_f |F_n| \frac{V_{ctx}\hat{t}_x + V_{ctz}\hat{t}_z}{V_c} \tag{59}$$

The contact type is rolling or sticking when the relative tangential velocity is zero but these two cases are conceptually different. In the rolling case, the contact points are always changing; that is, the collision detection algorithm report different closet points $\mathbf{p}$ and $\mathbf{q}$ at each iteration. In contrast, the contact points are remaining constant during the simulation for sticking case.

Definition 4. Two rigid bodies are in *intermittent contact* at time t_0 when it's not of continuous type.

Note that the state in which two bodies are not in contact is categorized as intermittent contact. The separating contact can be considered as the intermittent type. In addition, the colliding contacts which introduces some sort of jump in the velocity falls into this family of contact situation.

Definition 5. Two rigid bodies are in *colliding contact at time* t_0 *if* the contact exists between the objects and the relative normal velocity is negative by accepting a proper threshold. This can be mathematically expressed as:

$$d(t_0) < T_d \wedge \dot{d}(t_0) < -T_C \tag{60}$$

In the colliding case, the velocities are reset. If not so, the Euclidean distance between colliding objects will violate the interpenetration condition at the next simulation step which means:

$$d(t_0 + \Delta t) < -T_i \tag{61}$$

Definition 6. Two rigid bodies are in *separating contact* at time t_0 if the contact exists between the objects and the relative normal velocity is positive by accepting a proper threshold. This can be mathematically expressed as:

$$\mathrm{d}(\mathrm{t}_0) < \mathrm{T}_\mathrm{d} \wedge \dot{\mathrm{d}}(\mathrm{t}_0) > \mathrm{T}_\mathrm{s} \tag{62}$$

Lemma 2. If the distance function is smooth then the following inequality holds:

$$T_c > \frac{T_d + T_i}{\Delta t} \tag{63}$$

Proof. We have the following approximation from smoothness of the distance function.

$$\mathrm{d}(\mathrm{t}_0 + \Delta \mathrm{t}) \approx \mathrm{d}(\mathrm{t}_0) + \dot{\mathrm{d}}(\mathrm{t})\Delta \mathrm{t} \tag{64}$$

Assume that the two bodies are in colliding contact at time t_0 so the equations (56,60) are valid. From equations (56,60,64), it is concluded that

$$\max(\mathrm{d}(\mathrm{t}_0 + \Delta \mathrm{t})) = \max(\mathrm{d}(\mathrm{t}_0) + \dot{\mathrm{d}}(\mathrm{t}_0)\Delta \mathrm{t}) = \max(\mathrm{d}(\mathrm{t}_0)) + \max(\dot{\mathrm{d}}(\mathrm{t}_0))\Delta \mathrm{t} = \mathrm{T}_\mathrm{d} - \mathrm{T}_\mathrm{c}\Delta \mathrm{t} \tag{65}$$

As long as the obtained maximum value still represents the colliding contact the equation (61) is valid. By obtaining

$$T_d - T_c \Delta t < -T_i \tag{66}$$

equation (63) is derived.

Lemma 3. If the distance function is smooth then the following inequality holds

$$T_s > \frac{T_d + T_i}{\Delta t} \tag{67}$$

Proof. The proof is similar to lemma *2*. In the proof, separating contact is used instead of colliding contact.

3.2 Switching Between Different Contact Situation

When two rigid bodies are in contact ($d(t)= 0$) and the relative normal velocity is zero ($\dot{d}(t) = 0$), then it is assumed that one geometric constraint exists between two rigid bodies. At this time, the continuous contact situation occurs. It physically means that two solid objects start to slide or roll on each other. If our assumption is wrong the calculated normal force is negative which is physically not possible. Hence it is concluded that the continuous contact as well as geometric constraint do not exist. This situation will be kept as far as the normal force is greater than zero. When the normal force goes to zero separation occurs.

Proposition 2 The continuous contact ends in two cases; either the magnitude of the normal force smoothly goes to zero or a sudden change in the direction of the normal force occurs.

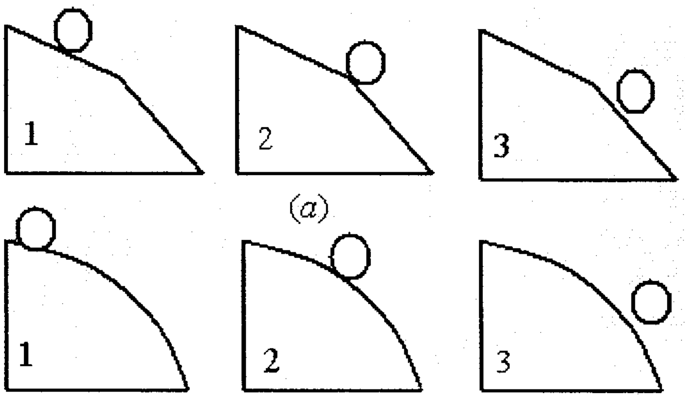


Figure 5. (a) Sudden jump (b) smooth release

Consider the magnitude of the normal force smoothly goes to zero, that is physically equivalent to the case where the constraint is released and became inactive. Secondly, A sudden change in the direction of the normal force implies that $\hat{\mathbf{n}}(t)$ is not smooth. The non-smoothness in $\hat{\mathbf{n}}(t)$ causes a jump in the distance between two bodies. These two cases are shown in the figure 5 for the rolling ball. Figure 5.a shows that ball detaches from the constraint when the normal force goes to zero and figure 5.b describes the effect of non-smoothness in $\hat{\mathbf{n}}(t)$.

When two solid objects are sliding or rolling on each other, there is a continuous contact between them. Not only the distance $d(t)$ continuously remains zero but also $\dot{d}(t)$ and the higher order derivatives remain zero. In these cases, a non-penetration constraint can be considered as geometric constraints as shown in equation (5). More important the normal force can be found by utilizing sliding manifold realization. In this approach, we can model the static friction during rolling by adding a rolling constraint to the model.

3.3 Hybrid Automaton for Single Contact

In this section, the goal is to the model rigid body systems with the help of the hybrid framework. A formal mathematical framework, hybrid automaton, is utilized to model and analyze the system.

Definition 7. A hybrid automaton H is a collection $\mathrm{H} = (\mathbf{Q}, \mathbf{X}, \mathbf{f}, \mathrm{Init}, \mathrm{D}, \mathrm{E}, \mathrm{G}, \mathrm{R})$, where

- $\mathbf{Q}$ is a discrete state space;
- $\mathbf{X} = \Re^{\mathrm{n}}$ is a continuous state space;
- $\mathbf{f} : \mathbf{Q} \times \mathbf{X} \to \Re^{\mathrm{n}}$ is a vector field ;
- $\mathrm{Init} \subset \mathbf{Q} \times \mathbf{X}$ is a set of initial states;
- $\mathrm{D} : \mathbf{Q} \to \mathrm{P}(\mathbf{X})$ is a domain;
- $\mathrm{E} \subset \mathbf{Q} \times \mathbf{Q}$ is a set of edges;
- $\mathrm{G} : \mathrm{E} \to \mathrm{P}(\mathbf{X})$ is a guard condition;
- $\mathrm{R} : \mathrm{E} \times \mathbf{X} \to \mathrm{P}(\mathbf{X})$ is a reset map.

For simplicity, consider a rigid body system subject only to a non-penetration constraint. The hybrid automaton for this system can be illustrated as:

- $\mathbf{Q} = \{Q_1, Q_2\}$ describes two different modes of operation. The First mode is the case where two solid objects are in intermittent contact and the second one is the case where two solid objects are in a continuous contact;
- $\mathbf{X} = [\mathbf{q}^T, \mathbf{v}^T, z]^T$ where $\mathbf{q}$ is the rigid body's position, $\mathbf{v}$ is the rigid body's velocity and z is the Lagrange multiplier. This vector of variable is the extend vector of variables which is the same in both modes.

The non-penetration constraint is a kind of inequality geometric constraint. It is known that error variable w which corresponds to a non-penetration constraint shows distance between two rigid bodies. The dynamic system, when there is no constraint (mode Q_1), can be shown as:

$$\begin{aligned} &\frac{\partial^2 L}{\partial \mathbf{v}^2} \dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q}, \mathbf{v}, t) \\ &\dot{\mathbf{q}} = \mathbf{v} \end{aligned} \qquad (68)$$

The variable z is not defined in mode Q_1.

Proposition 3 To extend the vector of variables in every mode, in such a way that the extended vector of variables is the same in all of them, their undefined variables are included and their time derivative are set to zero.

As mentioned in the proposition, it is desired to extend the vector of variables, in such a way that the extended vector of variables is the same in both modes. To do so,

$\mathbf{X} = [\mathbf{q}^T, \mathbf{v}^T, z]^T$ is utilized as continuous state variables. Then, equations of mode Q_1 can be rewritten as:

$$\begin{aligned} &\frac{\partial^2 L}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q}, \mathbf{v}, t) + \partial h/\partial \mathbf{q}^T z \\ &\dot{\mathbf{q}} = \mathbf{v} \\ &\dot{z} = 0 \end{aligned} \tag{69}$$

On the other hand, the dynamic system can be shown as below when the equality constraint is active (mode Q_2):

$$\begin{aligned} &\frac{\partial^2 L}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q}, \mathbf{v}, t) + \partial h/\partial \mathbf{q}^T z \\ &\dot{\mathbf{q}} = \mathbf{v} \\ &\dot{z} = \nu_s \\ &\nu_s = -\hat{J}_s^{-1}\left(\hat{\alpha} + K \cdot \mathrm{sat}\left(\frac{s}{\varepsilon}\right)\right) \end{aligned} \tag{70}$$

The dynamics of systems subject to a constraint is described by set of DAEs, which can be realized as a set of ODEs when the sliding manifold approach is used. Note that there exists other approaches for constraint stabilization, similar to proportional derivative control, such as Baumgart's method [11]. However, in this thesis we select the SPSM method due to its robustness properties and capability to allow more computationally efficient approximations than the other approaches.

The next step is to define the initial conditions. The acceptable set of initial conditions is described as:

$$\text{Init} = \{Q_1\} \times \{\mathbf{X} \mid w > 0 \wedge z = 0\} \cup \{Q_2\} \times \{\mathbf{X} \mid w < -\varepsilon \wedge z \geq 0\} \quad (71)$$

where ε is acceptable bound of an error for the geometric constraint. The domain set for each mode is characterized by following sets:

$$D(Q_1) = \{\mathbf{X} \mid w > \delta \vee \dot{w} > \beta\} \quad (72)$$

$$D(Q_2) = \{\mathbf{X} \mid w < \varepsilon \wedge \lambda \geq -\xi\} \quad (73)$$

where δ is a positive threshold for the distance, β is a threshold for the normal relative velocity and ξ is a positive constant which is of ε order ($\xi = O(\varepsilon)$). The definition of the rules that govern the transitions between modes of operation is the final step to complete a hybrid automaton system. The set of edges and guard conditions are describing mode transitions. The set of edges are as follows:

$$G(Q_1, Q_1) = \{\mathbf{X} \mid w < \delta \wedge \dot{w} < -\beta\} \quad (74)$$

$$G(Q_1, Q_2) = \{\mathbf{X} \mid w < \delta \wedge |\dot{w}| < \beta\} \quad (75)$$

$$G(Q_1, Q_2) = \{\mathbf{X} \mid w < \delta \wedge |\dot{w}| < \beta\} \quad (76)$$

The physical meaning of $G(Q_1, Q_1)$ is that the collision is detected and the relative normal velocity is less than zero with accepting β threshold. After hitting the guard condition, the continuous state resets to some new value in $R(Q_1, Q_1, \mathbf{X})$. Reset map shows a jump in the velocities at the time of collision. Therefore the velocities are reinitialized according to the method described in the section 2.3.2. The reset map can be expressed as:

$$R(Q_1, Q_1, \mathbf{X}) = \{\mathbf{q} = \mathbf{q}, z = z, \mathbf{v} = \mathbf{v}^+\} \qquad (77)$$

On edge $\{Q_1, Q_2\}$, a geometric constraint between two rigid bodies becomes active, the hybrid system then switch to the second mode of operation. Consequently, the interaction forces are calculated by sliding manifold approach in the continuous contact mode. At this discrete transition, the continuous state dose not jump. Hence there is no need to reinitialize the continuous states after switching to a new mode of operation. Briefly, the rest map correspond to this transition can be shown as:

$$R(Q_1, Q_2, \mathbf{X}) = \{\mathbf{q} = \mathbf{q}, z = z, \mathbf{v} = \mathbf{v}\} \qquad (78)$$

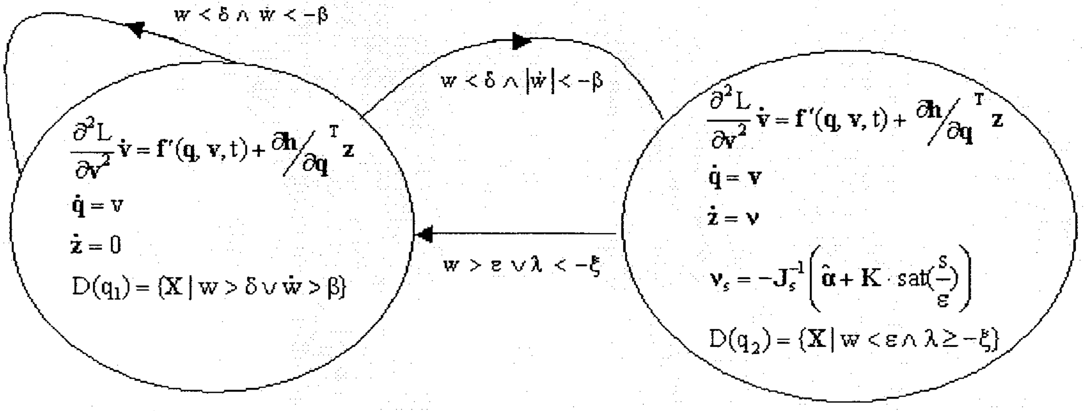

$w < \delta \wedge \dot{w} < -\beta$
$w < \delta \wedge |\dot{w}| < -\beta$
$w > \varepsilon \vee \lambda < -\xi$
$\frac{\partial^2 L}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q}, \mathbf{v}, t) + {\partial \mathbf{h}}/{\partial \mathbf{q}}^T \mathbf{z}$
$\dot{\mathbf{q}} = \mathbf{v}$
$\dot{\mathbf{z}} = 0$
$D(q_1) = \{\mathbf{X} \mid w > \delta \vee \dot{w} > \beta\}$
$\frac{\partial^2 L}{\partial \mathbf{v}^2}\dot{\mathbf{v}} = \mathbf{f}'(\mathbf{q}, \mathbf{v}, t) + {\partial \mathbf{h}}/{\partial \mathbf{q}}^T \mathbf{z}$
$\dot{\mathbf{q}} = \mathbf{v}$
$\dot{\mathbf{z}} = \mathbf{v}$
$\mathbf{v}_s = -\mathbf{J}_s^{-1}\left(\hat{\alpha} + \mathbf{K} \cdot \mathrm{sat}(\frac{s}{\varepsilon})\right)$
$D(q_2) = \{\mathbf{X} \mid w < \varepsilon \wedge \lambda \geq -\xi\}$

Figure 6. A hybrid system modelling of interacting rigid bodies

Edge $\{Q_2, Q_1\}$ physically implies the end of the continuous contact due to vanishing of normal force. The normal force may go to zero smoothly or non-smoothly. The non-smooth case occurs due to sudden change in the direction of normal force as mentioned in previous section. Therefore continuous state z may show a jump at this discrete transition and the corresponding rest map is as follows:

$$R(Q_2, Q_1, \mathbf{X}) = \{\mathbf{q} = \mathbf{q}, z = 0, \mathbf{v} = \mathbf{v}\} \tag{79}$$

Finally, the following modelling can be represented graphically in the figure 6.

3.3.1 2D Example

Consider a ball rolling on an arbitrary path. For simplicity, the equations that define the simulation are written in 2D case. The following geometric constraint exists when the ball rolls on the surface:

$$h(x, y, \theta) = (x - x_0)\cos\psi + (y - y_o)\sin\psi - R = 0 \tag{80}$$

$$\frac{\partial h}{\partial q} = \mathbf{G} = [\cos\psi \quad \sin\psi \quad 0] \tag{81}$$

Where (x, y, θ) is configuration space, (V_x, V_y, ω_z) is tangent space and ψ is an angel between the unit normal surface and the horizontal plane (see figure 7), R is a ball radius, and x_0 and y_0 are the locations of a single point on the path.

The nonholonomic kinematic constraint is a rolling condition and is expressed as:

$$a(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}) = \dot{x}\sin\psi - \dot{y}\cos\psi + R\dot{\theta} = [\sin\psi \quad -\cos\psi \quad R]\begin{bmatrix}\dot{x} \\ \dot{y} \\ \dot{\theta}\end{bmatrix} = 0 \tag{82}$$

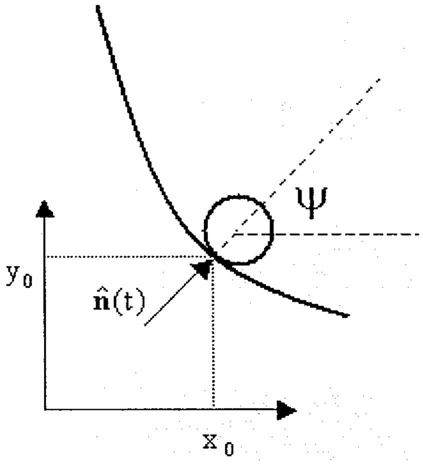


Figure 7. Ball rolls on a surface

Note that the SPSM method can stabilize this nonholonomic constraint in a manner similar to holonomic constraints. In that case the constraint just has an index of two instead of an index of three.

The equation of the motion for this system in the Cartesian coordinates forms a DAE problem as follows:

$$\dot{V}_x = \frac{1}{m} z_2 \sin\psi + \frac{1}{m} z_1 \cos\psi$$

$$\dot{V}_y = -\frac{1}{m} z_2 \cos\psi + \frac{1}{m} z_1 \sin\psi - g$$

$$\dot{\omega}_z = \frac{R}{J} z_2$$

$$\dot{x} = V_x \tag{83}$$

$$\dot{y} = V_y$$

$\dot{\theta} = \omega_z$

$\dot{z}_1 = \nu_1$

$\dot{z}_2 = \nu_2$

At this state, the error variables are described as:

$$w_1 = (x - x_0)\cos\psi + (y - y_o)\sin\psi - R \tag{84}$$

$$w_2 = \dot{x}\sin\psi - \dot{y}\cos\psi + R\dot{\theta} \tag{85}$$

and the sliding manifold is constructed as follows:

$$s_1 = \mu^2\ddot{w}_1 + 2\mu\dot{w}_1 + w_1 \tag{86}$$

$$s_2 = \mu\dot{w}_2 + w_2 \tag{87}$$

where μ is a positive constant. When a ball rolls on a trajectory, the control input is calculated with the sliding manifold approach that was described before. If the ball does not have a contact with the path then motion of rigid body in an intermittent mode can be described as follow:

$\dot{V}_x = 0$

$\dot{V}_y = -g$

$\dot{\omega}_z = 0$

$\dot{x} = V_x$

$$\dot{y} = V_y \tag{88}$$

$$\dot{\theta} = \omega_z$$

$$\dot{z}_1 = 0$$

$$\dot{z}_2 = 0$$

After calculating the collision impulses the velocities are reinitialized. The contact velocity then is expressed as:

$$V_{ctx} = V_x + R\omega_z \tag{89}$$

$$V_{cn} = V_y \tag{90}$$

The changes of velocities are:

$$\Delta V_{tx} = \frac{1}{m} P_x \tag{91}$$

$$\Delta V_n = \frac{1}{m} P_y \tag{92}$$

$$\Delta \omega_z = \frac{R}{J} P_X \tag{93}$$

Obtaining change of the contact velocity:

$$\Delta V_{ctx} = (\frac{1}{m} + \frac{R^2}{J}) P_x(\gamma) \tag{94}$$

$$\Delta V_{ctx} = (\frac{1}{m} + \frac{R^2}{J}) P_x(\gamma) \tag{95}$$

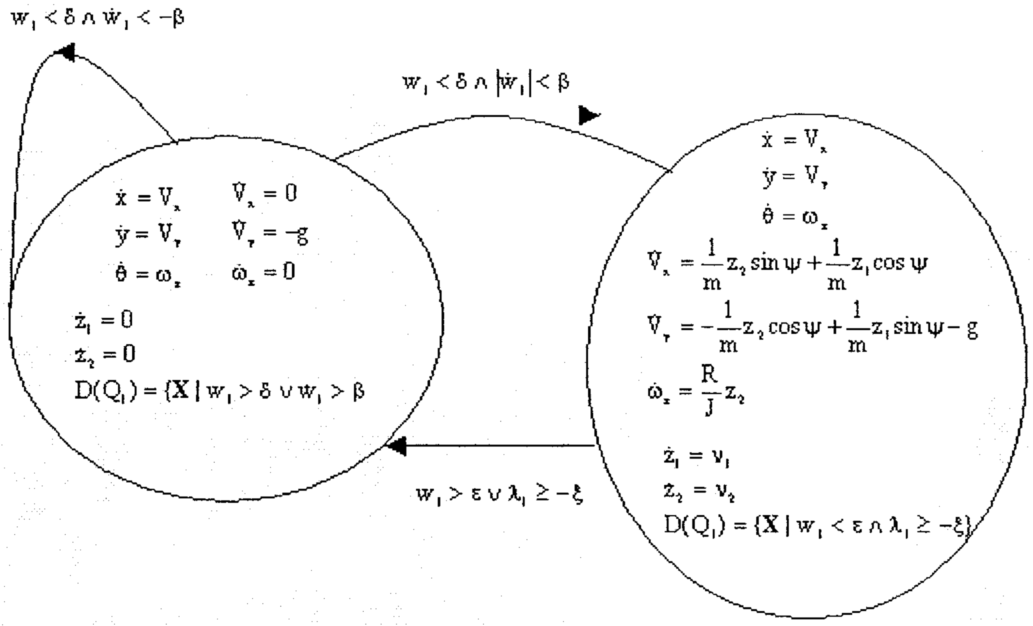
we can form the following nonlinear differential equation during the collision:

$$V'_{ctx} = -C_f\left(\frac{1}{m} + \frac{R^2}{J}\right)\frac{V_{ctx}}{|V_{ctx}|} \tag{96}$$

$$V'_{cn} = 1 \tag{97}$$

The term V_c can numerically be integrated by utilizing the equation given above. During this integration, V_{cn} will increase. The point of maximum compression is attained when V_n reaches to zero. Then the terminating value for the collision parameters γ_f is calculated. Finally, the integration continues to this point.

In order to simplify the problem, it is assumed that the geometric constraint and rolling constraint become active simultaneously. This physically meant that the short-time transition state from sliding to rolling is negligible. In a more detail modelling, one more mode should be added to the hybrid system for this transition state in which the geometric (non-penetration) constraint is active and the rolling constraint is inactive. The simplified hybrid automaton is shown in figure 8.



$w_1 < \delta \wedge \dot{w}_1 < -\beta$
$w_1 < \delta \wedge |\dot{w}_1| < \beta$
$\dot{x} = V_x$
$\dot{y} = V_y$
$\dot{\theta} = \omega_z$
$\dot{V}_x = 0$
$\dot{V}_y = -g$
$\dot{\omega}_z = 0$
$\dot{z}_1 = 0$
$\dot{z}_2 = 0$
$D(Q_1) = \{\mathbf{X} \mid w_1 > \delta \vee w_1 > \beta$
$\dot{x} = V_x$
$\dot{y} = V_y$
$\dot{\theta} = \omega_z$
$\dot{V}_x = \frac{1}{m} z_2 \sin\psi + \frac{1}{m} z_1 \cos\psi$
$\dot{V}_y = -\frac{1}{m} z_2 \cos\psi + \frac{1}{m} z_1 \sin\psi - g$
$\dot{\omega}_z = \frac{R}{J} z_2$
$\dot{z}_1 = v_1$
$\dot{z}_2 = v_2$
$D(Q_1) = \{\mathbf{X} \mid w_1 < \varepsilon \wedge \lambda_1 \geq -\xi\}$
$w_1 > \varepsilon \vee \lambda_1 \geq -\xi$

Figure 8. A simplified hybrid automaton for rolling ball example

Parameters $\varepsilon, \mu, \beta, \xi, \delta$ should be tuned in an appropriate way to avoid chattering. We test our method with different initial conditions and different paths, which are approximated as piecewise-constant convex paths. It can be seen from figure 9 that a ball follows a correct path if switching between the modes is done correctly. The behaviour of the system near discontinuity can be shown in figure 10 and it demonstrates that the constraint is released and hybrid automaton transits to intermittent mode. It can be seen form figure 11 that there is no chattering between the modes. The transition from continuous contact to intermittent one happens because of discontinuity. One of the advantages of the described method is to eliminate the chattering behaviour of relative normal velocity when two bodies have continuous contact with each other. The bound of chattering depends on the integration increment of collision parameter. Figure 12 illustrates these phenomena. The first and the second plots are using impulse-based method with different collision increment but the third one is the control-based method.

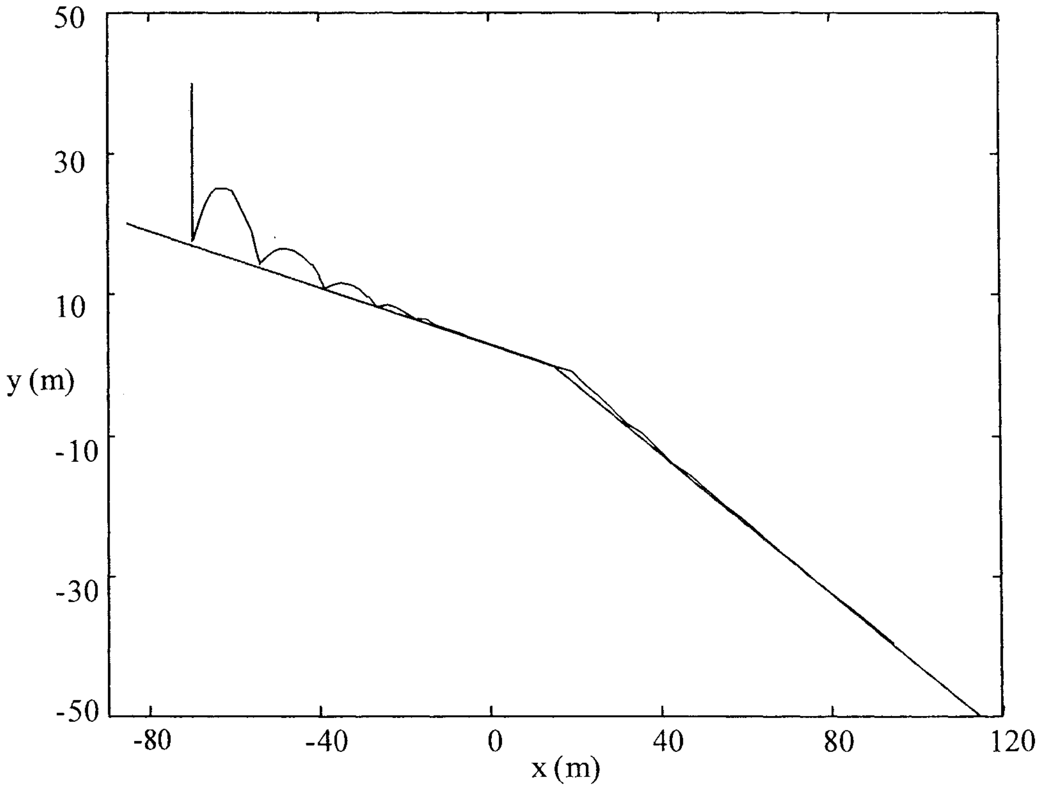


Figure 9. Ball trajectory

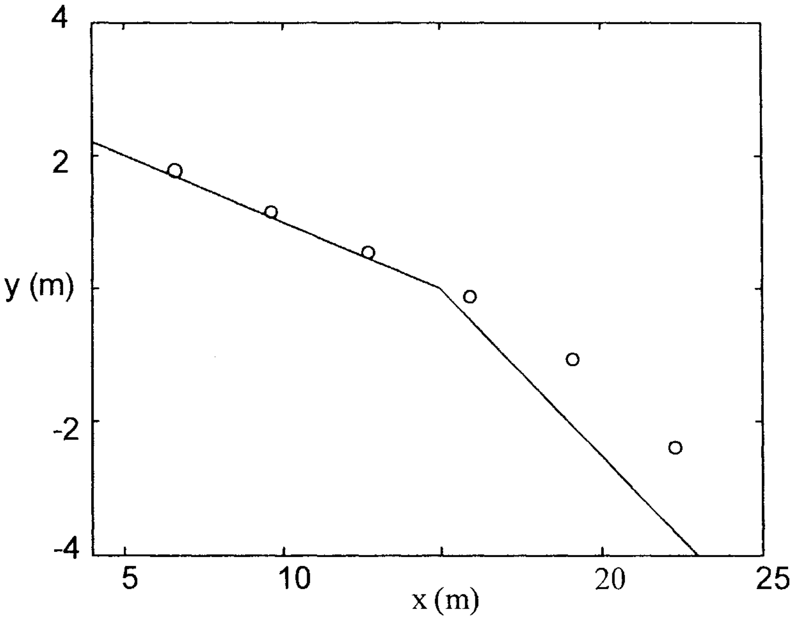


Figure 10. Ball detaches form the constraint near discontinuity

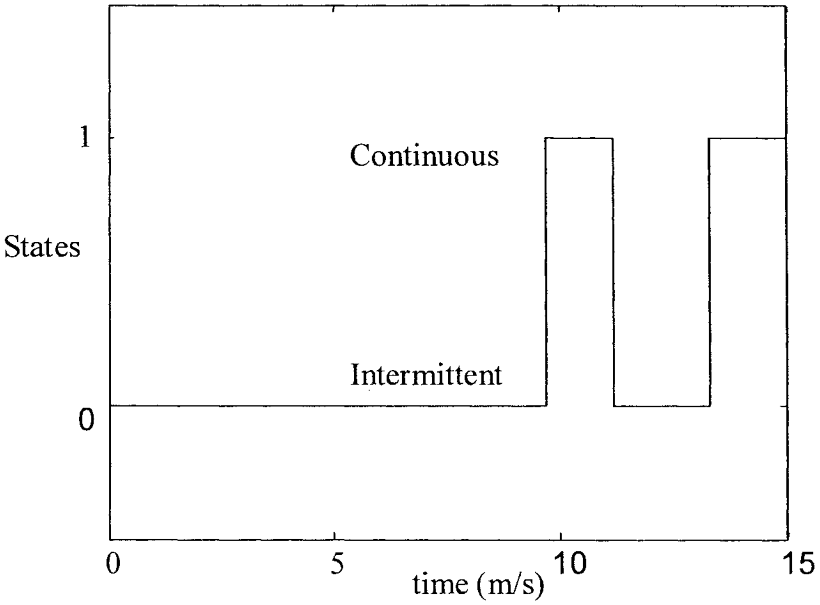


Figure 11. Mode Transition

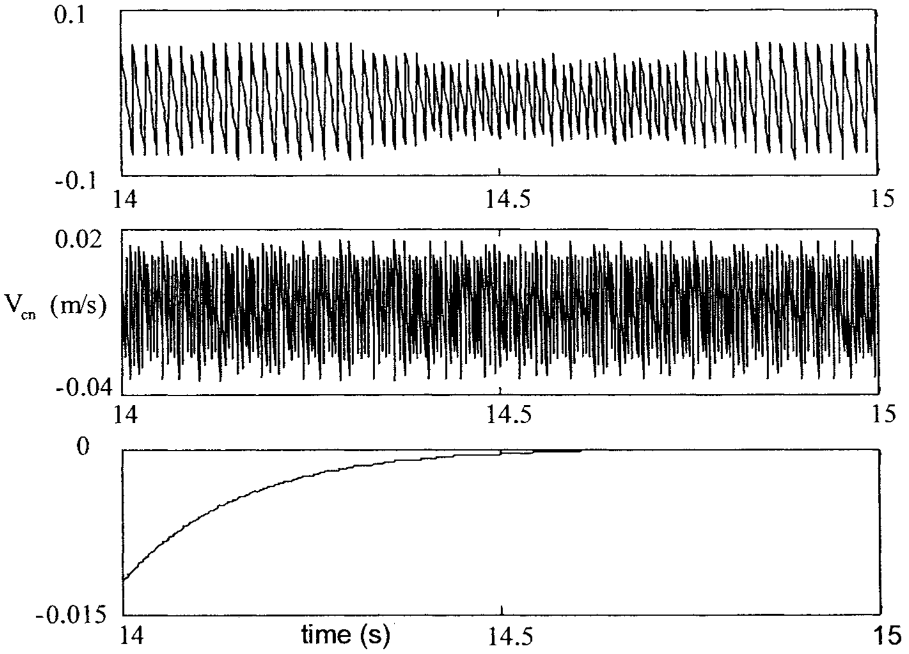


Figure 12. Chattering behaviours

3.3.2 Applications

In order to show the performance our simulator for real applications, some simulations are done in three dimensions. The first simulation shows the rolling ball example in pervious section in real 3D environment. A ball rolls down a ramp and then roll up another ramp and after that it follows a free body motion and then hits the ground and finally the bouncing is damped according to coefficient of restitution and again it starts to roll after short period of sliding (see figure 13). In the figure 14, the simulation is tested via different initial conditions and different trajectories of the ball are shown.

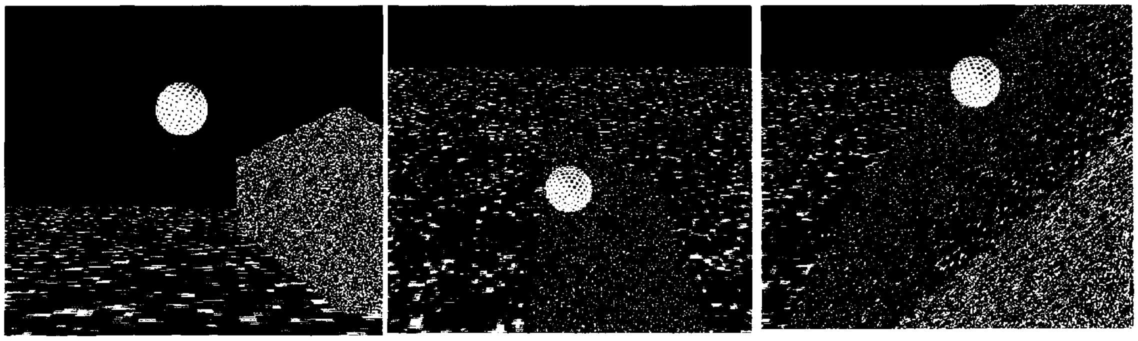


Figure 13. Ball is rolling down and up the ramp and hitting the ground

A ball on a spinning table is simulated to show that the control-based approach is well suited for continuous contact situations. It is known that ball is rolling when there is high coefficient of friction between ball and table. When the equations of motions are obeying from nonholonomic modelling and there is no external force on a ball, our result is that a ball rolls in circle, which is eccentric with respect to the table's axis of rotation [26]. (see figure 15).

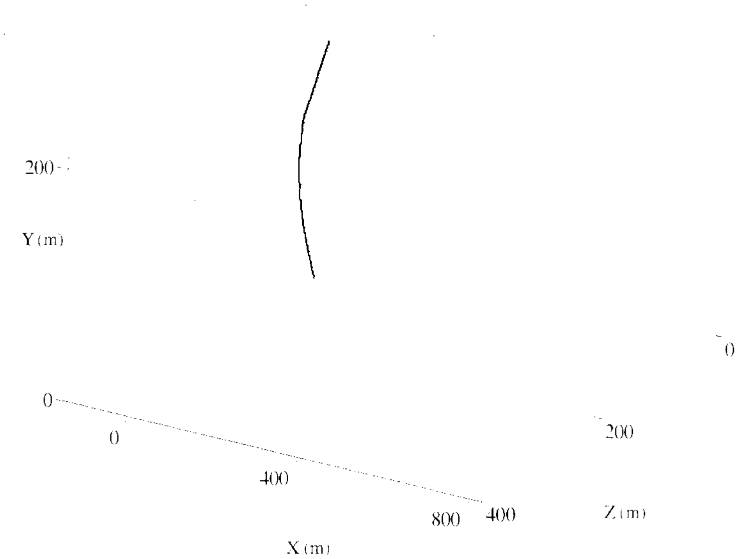


Figure 14. Ball trajectories according to different initial conditions

It seems that the method is acceptable for a wide range of applications, which are not collision intensive systems, and many mechanical systems fall into this category. For instance consider the cam-follower system. The cam mechanism is a familiar way to translate the rotational motion into a linear displacement. The cam and follower may be disconnected due to high velocities and this introduces collision to system and the simulator may handle this collision. The other mechanical example is to model contact between two gears.

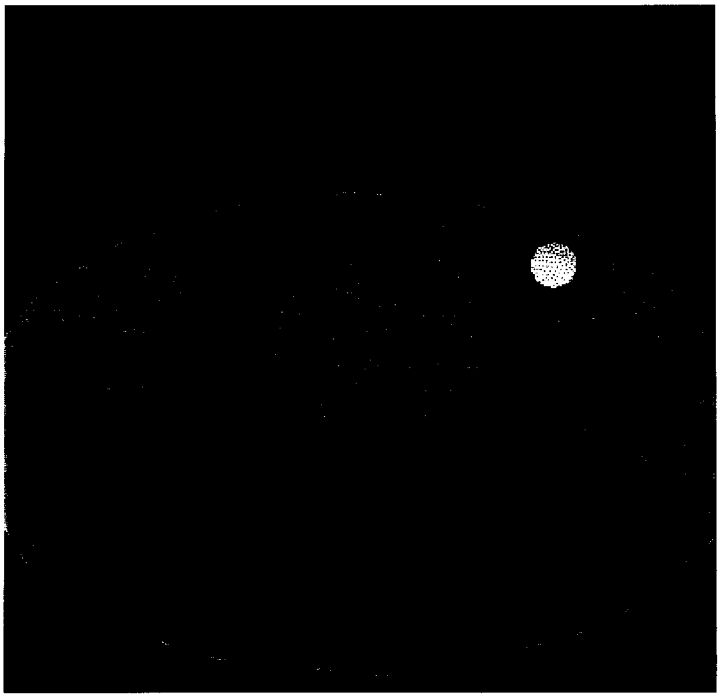


Figure 15. Ball trajectories according to different initial conditions

4 . Multiple Contact Hybrid Simulation

In this chapter, the main goal is to generalize our hybrid scheme for a case in which more than two rigid bodies are presented in the simulation. Throughout this chapter, we assume that all rigid bodies are convex. Only one contact point between two convex objects is possible if the number of the contact points is finite. Infinite number of contacts between two convex objects could be possible; that is degenerate case. Degenerate cases are not in the scope of this thesis.

Proposition 4 To handle the non-convex objects, we can split them to convex objects.

In the next section, we will derive the equations for the system of rigid bodies in continuous contacts in the general case.

4.1 Rigid bodies Systems in Continuous Contact

The local frame attached to a body is used to describe the motion of a rigid body. This local frame is located at the centre of mass with its axis aligned to the principal axis of the rotation. We denote a transformation matrix from local coordinate to global coordinate as $\mathbf{E}_1^i$, where i represents the local frame attached to the i^{th} body. Consider bodies i and j are in contact. It is required to define a normal contact coordinate between these two bodies. A 3×3 transformation matrix from the normal contact coordinate (between bodies i and j) to the global coordinate is denoted as $\mathbf{E}_{\mathrm{n}}^{\mathrm{ij}}$. This transformation matrix can be expressed as:

$$\mathbf{E}_{\mathrm{n}}^{\mathrm{ij}} = [(\hat{\mathbf{t}}_{\mathrm{x}})_{\mathrm{ij}} \quad \hat{\mathbf{n}}_{\mathrm{ij}} \quad (\hat{\mathbf{t}}_{\mathrm{z}})_{\mathrm{ij}}] \tag{98}$$

The contact forces between body i and body j are given by 3×1 vector $\mathbf{z}_{\mathrm{ij}}$ in the normal contact coordinate $\mathbf{E}_{\mathrm{n}}^{\mathrm{ij}}$. The position of the contact point in the i^{th} body frame is denoted by $\mathbf{r}_{\mathrm{ij}}$. A function, Λ_{ij}, is introduced to report the continuous contact situation as:

$$\Lambda_{\mathrm{ij}} = \begin{cases} 1 & \text{if body i and body j are in continuous contact} \\ 0 & \text{otherwise} \end{cases} \tag{99}$$

It is apparent from definition that $\Lambda_{\mathrm{ij}} = \Lambda_{\mathrm{ji}}$. It is also known from the action and reaction law that we have $\mathbf{z}_{\mathrm{ij}} = -\mathbf{z}_{\mathrm{ji}}$. The motion equations of each object can be shown in general case as follows:

$$\begin{gathered} \mathrm{m}_{\mathrm{i}}\mathbf{a}_{\mathrm{i}} = \mathbf{f}_{\mathrm{i}} + \mathrm{m}_{\mathrm{i}}\mathbf{g} + \sum_{\substack{\mathrm{j}=1 \\ \mathrm{j}\neq\mathrm{i}}}^{\mathrm{n}} \Lambda_{\mathrm{ij}}(\mathbf{E}_{\mathrm{n}}^{\mathrm{ij}}\mathbf{z}_{\mathrm{ij}}) \\ \mathbf{J}_{\mathrm{i}}\dot{\boldsymbol{\omega}}_{\mathrm{i}} + \boldsymbol{\omega}_{\mathrm{i}} \times \mathbf{J}_{\mathrm{i}}\boldsymbol{\omega}_{\mathrm{i}} = \boldsymbol{\tau}_{\mathrm{i}} + \sum_{\substack{\mathrm{j}=1 \\ \mathrm{j}\neq\mathrm{i}}}^{\mathrm{n}} \Lambda_{\mathrm{ij}}(\mathbf{r}_{\mathrm{ij}} \times {\mathbf{E}_{\mathrm{l}}^{\mathrm{i}}}^{-1}\mathbf{E}_{\mathrm{n}}^{\mathrm{ij}}\mathbf{z}_{\mathrm{ij}}) \end{gathered} \tag{100}$$

For a case in which some bodies are in continuous contacts, we must incorporate proper constraints into the dynamic equations. Normally, these constraints are expressed by set of algebraic differential equations. As we explained earlier in the thesis, there are different types of continuous contacts. Hence, each type introduces its own set of constraint equations. It is important to note that the non-penetration constraint is common constraint between all those types. We can show the non-penetration constraint between bodies i and j as:

$$\mathrm{w}_{\mathrm{ij}}^{\mathrm{np}} = \mathrm{d}_{\mathrm{ij}} = \mathbf{E}_{\mathrm{n}}^{\mathrm{ij}} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \cdot (\mathbf{p} - \mathbf{q}) = \hat{\mathbf{n}}_{\mathrm{ij}} \cdot (\mathbf{p} - \mathbf{q}) \tag{101}$$

In a case of slipping contact, we have an inequality constraint:

$$(\hat{\mathbf{t}}_{\mathrm{x}})_{\mathrm{ij}} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) + (\hat{\mathbf{t}}_{\mathrm{z}})_{\mathrm{ij}} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) \neq 0 \tag{102}$$

When two objects are rolling on each other in 3D space, two more constraints must be added to the dynamic simulation. Those constraints can be shown as:

$$\mathrm{w}_{\mathrm{ij}}^{\mathrm{rx}} = (\hat{\mathbf{t}}_{\mathrm{x}})_{\mathrm{ij}} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) = 0 \tag{103}$$

$$\mathrm{w}_{\mathrm{ij}}^{\mathrm{rz}} = (\hat{\mathbf{t}}_{\mathrm{z}})_{\mathrm{ij}} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) = 0 \tag{104}$$

We can apply the same constraint equations for the sticking case. The dynamic equations include a set of ODEs and algebraic differential equations at the same time. Basically, the whole system is subject to set of the DAEs. This DAE can be realized as a set of ODEs by sliding manifold approach. It is clear that the number of constraints is changing due to the change of the contact configuration. As mentioned earlier, this is one of the main reasons that leads us to use the hybrid scheme for the simulation of rigid bodies in contact. In this hybrid scheme, we have different number of constraint equations in each mode. For instance, two more algebraic constraints must be added to the dynamic equations when the system switches from the slipping contact to the rolling contact. To apply sliding manifold method, we should introduce control inputs correspond to each constraint. We define the control input as:

$$\begin{aligned}
\nu_{ij}^{np} &= (0 \quad 1 \quad 0)^{T} \cdot \dot{\mathbf{z}}_{ij} \\
\nu_{ij}^{rx} &= (1 \quad 0 \quad 0)^{T} \cdot \dot{\mathbf{z}}_{ij} \\
\nu_{ij}^{rz} &= (0 \quad 0 \quad 1)^{T} \cdot \dot{\mathbf{z}}_{ij}
\end{aligned} \tag{105}$$

When body i and body j are slipping on each other, the three elements of $\mathbf{z}_{ij}$ are not independent. From Coulomb's law of friction (59), we conclude that only the second element of $\mathbf{z}_{ij}$ is an independent variable. This implies that just one control input must be introduced for the independent element for the slipping case. Also equations (105) must be modified to the following equations.

$$\begin{aligned}
(1 \quad 0 \quad 0)^{T} \cdot \mathbf{z}_{ij} &= -C_f[(0 \quad 1 \quad 0)^{T} \cdot \mathbf{z}_{ij}]\frac{V_{ctx}}{|\mathbf{V}_c|} \\
\nu_{ij}^{np} &= (0 \quad 1 \quad 0)^{T} \cdot \dot{\mathbf{z}}_{ij} \\
(0 \quad 0 \quad 1)^{T} \cdot \mathbf{z}_{ij} &= -C_f[(0 \quad 1 \quad 0)^{T} \cdot \mathbf{z}_{ij}]\frac{V_{ctz}}{|\mathbf{V}_c|}
\end{aligned} \tag{106}$$

We introduced another function Γ_{ij} to report the type of the continuous contact. The function also enables us to find the number of constraints in each mode of our hybrid simulator. The function mathematically is expressed as:

$$\Gamma_{ij} = \begin{cases} 1 & \text{if body i and body j are in slipping contact} \\ 0 & \text{otherwise} \end{cases} \tag{107}$$

In general, each $\mathbf{w}_{ij}$ must be differentiated a different number of times $(r_i - 1)$ for some components of $\mathbf{z}_{ij}$ to appear. For the non-penetration constraints, it can be seen that $\mathbf{z}_{ij}$ terms appear in $\ddot{\mathbf{w}}_{ij}$. It is also assumed that $\mathbf{z}_{ij}$ terms appear in $\dot{\mathbf{w}}_{ij}$ for rolling constraints. Differentiating the non-penetration constraint with respect to time, we obtain:

$$\begin{aligned} w_{ij}^{np} &= \hat{\mathbf{n}}_{ij} \cdot (\mathbf{p} - \mathbf{q}) \\ \dot{w}_{ij}^{np} &= \dot{\hat{\mathbf{n}}}_{ij} \cdot (\mathbf{p} - \mathbf{q}) + \hat{\mathbf{n}}_{ij} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) \\ \ddot{w}_{ij}^{np} &= \ddot{\hat{\mathbf{n}}}^{ij} \cdot (\mathbf{p} - \mathbf{q}) + 2\dot{\hat{\mathbf{n}}}_{ij} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) + \hat{\mathbf{n}}_{ij} \cdot (\ddot{\mathbf{p}} - \ddot{\mathbf{q}}) \end{aligned} \tag{108}$$

Also the differentiation of the rolling constraint can be expressed as:

$$\begin{aligned} w_{ij}^{rx} &= (\hat{\mathbf{t}}_x)_{ij} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) \\ w_{ij}^{rz} &= (\hat{\mathbf{t}}_z)_{ij} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) \\ \dot{w}_{ij}^{rx} &= (\dot{\hat{\mathbf{t}}}_x)_{ij} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) + (\hat{\mathbf{t}}_x)_{ij} \cdot (\ddot{\mathbf{p}} - \ddot{\mathbf{q}}) \\ \dot{w}_{ij}^{rz} &= (\dot{\hat{\mathbf{t}}}_z)_{ij} \cdot (\dot{\mathbf{p}} - \dot{\mathbf{q}}) + (\hat{\mathbf{t}}_z)_{ij} \cdot (\ddot{\mathbf{p}} - \ddot{\mathbf{q}}) \end{aligned} \tag{109}$$

The velocity of contact points, $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$, are expressed as:

$$\begin{aligned} \dot{\mathbf{p}} &= \dot{\mathbf{r}}_i + \mathbf{E}_1^i(\boldsymbol{\omega}_i \times \mathbf{r}_{ij}) \\ \dot{\mathbf{q}} &= \dot{\mathbf{r}}_j + \mathbf{E}_1^j(\boldsymbol{\omega}_j \times \mathbf{r}_{ji}) \end{aligned} \tag{110}$$

where $\mathbf{r}_i$ is the centre of mass position of the i^{th} object. Differentiating with respect to time one more time gives:

$$\begin{aligned} \ddot{p} &= \ddot{r}_i + E_1^i[\dot{\omega}_i \times r_{ij} + \omega_i \times (\omega_i \times r_{ij})] \\ \ddot{q} &= \ddot{r}_j + E_1^j[\dot{\omega}_j \times r_{ji} + \omega_j \times (\omega_j \times r_{ji})] \end{aligned} \tag{111}$$

Combining equations (100) and (111) gives:

$$
\begin{aligned}
\ddot{p} &= \frac{1}{m_i}[f_i + m_i\mathbf{g} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \Lambda_{ij}(\mathbf{E}_n^{ij}\mathbf{z}_{ij})] + \\
&\quad E_l^i[J_i^{-1}(\tau_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} \Lambda_{ij}(\mathbf{r}_{ij} \times \mathbf{E}_l^{i^{-1}}\mathbf{E}_n^{ij}\mathbf{z}_{ij}) - \boldsymbol{\omega}_i \times J_i\boldsymbol{\omega}_i) \times r_{ij} + \omega_i \times (\omega_i \times r_{ij})] \\
\ddot{q} &= \frac{1}{m_j}[f_j + m_j\mathbf{g} + \sum_{\substack{i=1 \\ i\neq j}}^{n} \Lambda_{ji}(\mathbf{E}_n^{ji}\mathbf{z}_{ji})] + \\
&\quad E_l^j[J_j^{-1}(\tau_j + \sum_{\substack{i=1 \\ i\neq j}}^{n} \Lambda_{ji}(\mathbf{r}_{ji} \times \mathbf{E}_l^{j^{-1}}\mathbf{E}_n^{ji}\mathbf{z}_{ji}) - \boldsymbol{\omega}_j \times J_j\boldsymbol{\omega}_j) \times r_{ji} + \omega_j \times (\omega_j \times r_{ji})
\end{aligned}
\tag{112}
$$

The next step is to construct sliding manifold corresponds to each constraint. Therefore, the sliding manifold is constructed as:

$$s_{ij}^{np} = (\mu_{ij}^{np})^2 \ddot{w}_{ij}^{np} + 2\mu_{ij}^{np}\dot{w}_{ij}^{np} + w_{ij}^{np} \tag{113}$$

$$s_{ij}^{rx} = \mu_{ij}^{rx}\dot{w}_{ij}^{rx} + w_{ij}^{rx} \tag{114}$$

$$s_{ij}^{rz} = \mu_{ij}^{rz}\dot{w}_{ij}^{rz} + w_{ij}^{rz} \tag{115}$$

4.2 Hybrid Design

As we mentioned before, the number of constraints in each mode is different. It depends on the contact configuration of each mode. It is assumed that all objects are convex so each pair of objects must be in contact at only one point. The number of constraints is calculated as:

$$N = 3\times(\sum_{i=1}^{n}\sum_{j=i+1}^{n}\Lambda_{ij}) - 2\times(\sum_{i=1}^{n}\sum_{j=i+1}^{n}\Gamma_{ij}) \tag{116}$$

The number of modes is equal to the number of the possible contact configurations. Therefore, it can be calculated as:

$$\Pi = 3^{\binom{n}{2}} \tag{117}$$

The case of a single contact between two rigid bodies is completely investigated in the previous chapter. In this chapter, our goal is to propose a systematic way to handle general cases. A formal mathematical framework, hybrid automaton, is utilized to model and analyze the system (see the definition 7 for details).

When two or more objects exist in an environment, their state vectors can be combined into a new configuration vector $\mathbf{q} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \cdots \quad \mathbf{q}_n^T]^T$, where n is the number of independent objects in the environment and $\mathbf{q}_i$ is the state vector for the i^{th} object. All the elements of $\mathbf{q}$ are independents for a case in which none of the objects are in contact. On the other hand, one or more of the terms in $\mathbf{q}$ is no longer independent when some of the objects are in contact. Therefore, some constraints equations must be added to the system. In consequence, the dynamic equations in no longer described by a set of ODEs.

To feed the dynamic equations into the hybrid automaton framework, we have some restriction such as:

- The equations must be in the form of ODEs (see the definition *7* for $\mathbf{f}$) instead of DAEs.
- The vector of variables should be of the same dimension in all modes

To achieve those goals, each DAE system is realized as an ODE system with the help of the sliding manifold approach by introducing each $\mathbf{z}_{ij}$. The next step is to extend the vector of variables, in such a way that the extended vector of variables is the same in all modes (see proposition *3*). Finally, a continuous state variable is introduced as:

$$\begin{aligned}
&\mathbf{X} = [\mathbf{q}_1^T \quad \mathbf{q}_2^T \quad \cdots \quad \mathbf{q}_n^T \quad \mathbf{Z}^T]^T \\
&\mathbf{Z} = [\mathbf{z}_{12}^T \quad \mathbf{z}_{13}^T \quad \cdots \quad \mathbf{z}_{1n}^T \quad \mathbf{z}_{23}^T \quad \mathbf{z}_{24}^T \quad \cdots \quad \mathbf{z}_{2n}^T \quad \cdots \quad \mathbf{z}_{(n-2)(n-1)}^T \quad \mathbf{z}_{(n-2)n}^T \quad \mathbf{z}_{(n-1)n}^T]^T \\
&\text{where} \\
&\mathbf{z}_{ij} = [z_{ij}^{rx} \quad z_{ij}^{np} \quad z_{ij}^{ry}]^T
\end{aligned} \tag{118}$$

As we mentioned before, the tangential frictional force and the normal force are not independent any more for the case of slipping contact. Hence the equations (100) should be modified as follows:

$$\begin{aligned}
&m_i\mathbf{a}_i = \mathbf{f}_i + m_i\mathbf{g} + \sum_{\substack{j=1 \\ j\neq i}}^{n} \Lambda_{ij}(1-\Gamma_{ij})(\mathbf{E}_n^{ij}\mathbf{z}_{ij}) + \sum_{\substack{j=1 \\ j\neq i}}^{n} \Gamma_{ij}\mathbf{E}_n^{ij}z_{ij}^{np}\begin{pmatrix} -\frac{V_{ctx}}{|\mathbf{V}_c|} & 1 & -\frac{V_{ctz}}{|\mathbf{V}_c|} \end{pmatrix}^T \\
&\mathbf{J}_i\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \mathbf{J}_i\boldsymbol{\omega}_i = \boldsymbol{\tau}_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} \Lambda_{ij}(1-\Gamma_{ij})(\mathbf{r}_{ij} \times \mathbf{E}_1^{i-1}\mathbf{E}_n^{ij}\mathbf{z}_{ij}) \\
&\qquad\qquad + \sum_{\substack{j=1 \\ j\neq i}}^{n} \Gamma_{ij}(\mathbf{r}_{ij} \times \mathbf{E}_1^{i-1}\mathbf{E}_n^{ij}z_{ij}^{np}\begin{pmatrix} -\frac{V_{ctx}}{|\mathbf{V}_c|} & 1 & -\frac{V_{ctz}}{|\mathbf{V}_c|} \end{pmatrix}^T)
\end{aligned} \tag{119}$$

where

$$z_{ij}^{np} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T \cdot \mathbf{z}_{ij}$$

In order to complete the ODE equations, we must define $\dot{\mathbf{Z}}$ with respect to the other state variables. This vector can be generally shown as:

$$\dot{\mathbf{z}}_{ij} = \begin{bmatrix} \dot{z}_{ij}^{rx} \\ \dot{z}_{ij}^{np} \\ \dot{z}_{ij}^{rz} \end{bmatrix} = \Lambda_{ij}(1-\Gamma_{ij}) \begin{bmatrix} v_{ij}^{rx} \\ v_{ij}^{np} \\ v_{ij}^{rz} \end{bmatrix} + \Gamma_{ij} \begin{bmatrix} 0 \\ v_{ij}^{np} \\ 0 \end{bmatrix} \tag{120}$$

4.2.1 Discussion on the dimension of Control Jacobian

Before using the sliding manifold, it is necessary to decrease the dimension of $\mathbf{Z}$ in each mode. To decrease the dimension, we just keep the defined elements (ones which have nonzero derivative with respect to time). We denote this new vector variable as $\mathbf{Z}'_k$ (k shows the corresponding contact configuration). This is exactly equivalent of eliminating the undefined variables of each mode (see proposition *3*). It is clear from equation (120) that the dimension of this new vector variable $\mathbf{Z}'_k$ is highly depends on the contact configuration. Consequently, it is equal to the number of constraints, N, which can be found from equation (116). In the last step, we must give a solution to find the control inputs corresponding to each constraint $\mathbf{v}'_k = \dot{\mathbf{Z}}'_k$ for a new reduced system, where $\mathbf{v}'_k$ is the control vector in the corresponding mode. The SPSM method then designs a controller that forces the motion to satisfy constraints with certain amount of error ε_{ij}. In order to achieve the above goals, we design a sliding controller that determines the value of the control input, which is $\mathbf{v}_{ij} = \dot{\mathbf{z}}_{ij}$. The value of $\mathbf{z}_{ij}$ is then obtained by integrating $\mathbf{v}_{ij}$. Differentiating $\mathbf{s}_{ij}$ with respect to time we obtain:

$$\dot{s}^{np}_{ij} = (\mu^{np}_{ij})^2 \dddot{w}^{np}_{ij} + 2\mu^{np}_{ij} \ddot{w}^{np}_{ij} + \dot{w}^{np}_{ij} \tag{121}$$

$$\dot{s}^{rx}_{ij} = \mu^{rx}_{ij} \ddot{w}^{rx}_{ij} + \dot{w}^{rx}_{ij} \tag{122}$$

$$\dot{s}^{rz}_{ij} = \mu^{rz}_{ij} \ddot{w}^{rz}_{ij} + \dot{w}^{rz}_{ij} \tag{123}$$

After rearranging the equations, we have:

$$\dot{s}^{np}_{ij} = (\mu^{np}_{ij})^2 (\frac{\partial \ddot{w}^{np}_{ij}}{\partial \mathbf{z}'_k} \dot{\mathbf{z}}'_k + \sigma^{np}_{ij}) + 2\mu^{np}_{ij} \ddot{w}^{np}_{ij} + \dot{w}^{np}_{ij} \tag{124}$$

$$\dot{s}^{rx}_{ij} = \mu^{rx}_{ij} (\frac{\partial \dot{w}^{rx}_{ij}}{\partial \mathbf{z}'_k} \dot{\mathbf{z}}'_k + \sigma^{rx}_{ij}) + \dot{w}^{rx}_{ij} \tag{125}$$

$$\dot{s}^{rz}_{ij} = \mu^{rz}_{ij} (\frac{\partial \dot{w}^{rz}_{ij}}{\partial \mathbf{z}'_k} \dot{\mathbf{z}}'_k + \sigma^{rz}_{ij}) + \dot{w}^{rz}_{ij} \tag{126}$$

Therefore the control Jacobian matrix is defined as:

$$\mathbf{J}^{np}_{ij} = (\mu^{np}_{ij})^2 (\frac{\partial \ddot{w}^{np}_{ij}}{\partial \mathbf{z}'_k} \dot{\mathbf{z}}'_k) \tag{127}$$

$$\mathbf{J}^{rx}_{ij} = \mu^{rx}_{ij} \frac{\partial \dot{w}^{rx}_{ij}}{\partial \mathbf{z}'_k} \dot{\mathbf{z}}'_k \tag{128}$$

$$\mathbf{J}^{rz}_{ij} = \mu^{rz}_{ij} \frac{\partial \dot{w}^{rz}_{ij}}{\partial \mathbf{z}'_k} \dot{\mathbf{z}}'_k \tag{129}$$

At this moment, the dimension of the Jacobian matrix is $\Pi \times N$. For the constraints, which are not active, the corresponding row of Jacobian is zero. After eliminating those rows, we reach to the square Jacobian matrix with N dimensions. In a more compact form, we can write:

$$\dot{\mathbf{s}}'_k = (\mathbf{J}_s)_k \mathbf{v}'_k + \boldsymbol{\alpha}_k \tag{130}$$

where s' is a reduced vector of sliding manifolds corresponds to each mode, and $\boldsymbol{\alpha}_k$ is calculated as:

$$(\alpha_{ij}^{np})_k = (\mu_{ij}^{np})^2(\sigma_{ij}^{np}) + 2\mu_{ij}^{np}\dot{w}_{ij}^{np} + \ddot{w}_{ij}^{np} \tag{131}$$

$$(\alpha_{ij}^{rx})_k = \mu_{ij}^{rx}(\sigma_{ij}^{rx}) + \dot{w}_{ij}^{rx} \tag{132}$$

$$(\alpha_{ij}^{rz})_k = \mu_{ij}^{rz}(\sigma_{ij}^{rz}) + \dot{w}_{ij}^{rz} \tag{133}$$

If the above equation is solved for $\mathbf{v}'_k$ then we can steer the sliding motion into the desired boundary layer. Since determining the exact values of $\boldsymbol{\alpha}_k$ and $(\mathbf{J}_s)_k$ is computationally expensive, they are approximated by $\hat{\boldsymbol{\alpha}}_k$ and $(\hat{\mathbf{J}}_s)_k^{-1}$ respectively. It has been shown that if the $\mathbf{v}'_k$ is computed by the following sliding controller:

$$\mathbf{v}'_k = -(\hat{\mathbf{J}}_s)_k^{-1}\left(\hat{\boldsymbol{\alpha}}_k + \mathbf{K}_k \cdot \mathrm{diag}\left[\mathrm{sat}\left(\frac{s'_i}{\varepsilon_i}\right)\right]\right) \tag{134}$$

$$\mathrm{sat}(u) = \begin{cases} u & |u| < 1 \\ \mathrm{sign}(u) & |u| \geq 1 \end{cases} \tag{135}$$

where K_k is the gain for mode k, then the motion will converge to its desired error boundary after a short reaching phase according to theorem *1*.

Theorem 2. Assume that matrix $\boldsymbol{\Gamma}_k = \left|\mathrm{diag}[\boldsymbol{\alpha}_k - (\mathbf{J}_s)_k(\hat{\mathbf{J}}_s)_k^{-1}\hat{\boldsymbol{\alpha}}_k\right|$. If the matrices $(\mathbf{J}_s)_1(\hat{\mathbf{J}}_s)_1^{-1}\mathbf{K}_1 - \boldsymbol{\Gamma}_1, (\mathbf{J}_s)_2(\hat{\mathbf{J}}_s)_2^{-1}\mathbf{K}_2 - \boldsymbol{\Gamma}_2, \ldots, (\mathbf{J}_s)_\Pi(\hat{\mathbf{J}}_s)_\Pi^{-1}\mathbf{K}_\Pi - \boldsymbol{\Gamma}_\Pi$ (where Π denotes the number of modes) are diagonally dominant then a sliding condition is satisfied and it guarantees the following explicit error bounds throughout the hybrid simulation:

$$\begin{aligned}&\left|s_{ij}^{np}\right| < \varepsilon_{ij}^{np}\\ &\left|s_{ij}^{rx}\right| < \varepsilon_{ij}^{rx}, \quad i = 1,2,..,n \text{ and } j = i, i+1...,n\\ &\left|s_{ij}^{rz}\right| < \varepsilon_{ij}^{rz}\end{aligned} \tag{136}$$

$$\begin{aligned}&\left|\frac{d^p w_{ij}^{np}}{dt^p}\right| \le \frac{2^p \varepsilon_{ij}^{np}}{\mu_{ij}^{np}}\\ &\left|\frac{d^p w_{ij}^{rx}}{dt^p}\right| \le \frac{2^p \varepsilon_{ij}^{rx}}{\mu_{ij}^{rx}}, \quad i = 1,2,...n, j = i, i+1,...,n\\ &\left|\frac{d^p w_{ij}^{rz}}{dt^p}\right| \le \frac{2^p \varepsilon_{ij}^{rz}}{\mu_{ij}^{rx}}\end{aligned} \tag{137}$$

where n denotes the number of objects.

Proof. From the diagonal dominancy of matrix of $(\mathbf{J}_s)_k(\hat{\mathbf{J}}_s)_k^{-1}\mathbf{K}_k - \boldsymbol{\Gamma}_k$ and theorem *1.*, it is concluded that the following bounds are satisfied in the k^th mode. So each error variables is bounded in each mode and this implies the equation (137)

It is very important to note that the dimension of square Jacobian matrix is changing according to the contact configuration. These Jacobian matrices are illustrated by the proper example in the next sub section.

4.2.2 V-Groove Example:

The V-Groove problem is a very good example to illustrate different contact situations. As it is shown in the figure 16, this problem includes three convex rigid bodies. It is important to note that V-Groove is a non-convex object so it should be divided into two convex objects.

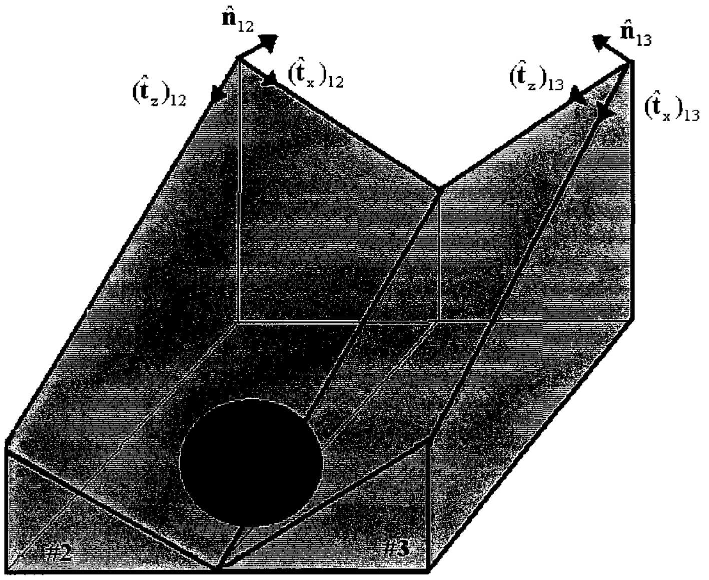

$\hat{\mathbf{n}}_{12}$
$(\hat{\mathbf{t}}_z)_{12}$
$(\hat{\mathbf{t}}_x)_{12}$
$(\hat{\mathbf{t}}_z)_{13}$
$\hat{\mathbf{n}}_{13}$
$(\hat{\mathbf{t}}_x)_{13}$
#2
#3

Figure 16. V-Groove problem

Consider a case that ball rolls on the second object, there are three constraints, which restrict the movement of the ball; two rolling constraint along $(\hat{\mathbf{t}}_x)_{12}, (\hat{\mathbf{t}}_z)_{12}$ directions and one non-penetration constraint along unit normal direction $\hat{\mathbf{n}}_{12}$. The constraint equations can be found as:

$$
\begin{aligned}
w_{12}^{rx} &= \dot{\mathbf{r}}_1 \cdot (\hat{\mathbf{t}}_x)_{12} + r\omega_1 \cdot (\hat{\mathbf{t}}_z)_{12} \\
w_{12}^{np} &= (\dot{\mathbf{r}}_1 - \mathbf{c}_2) \cdot \hat{\mathbf{n}}_{12} - r \\
w_{12}^{rz} &= \dot{\mathbf{r}}_1 \cdot (\hat{\mathbf{t}}_z)_{12} - r\omega_1 \cdot (\hat{\mathbf{t}}_x)_{12}
\end{aligned}
\tag{138}
$$

where $\mathbf{c}_2$ shows the contact point on the second object. The Jacobian matrix for this contact configuration is calculated as:

$$
(\mathbf{J}_s)_1 = \begin{bmatrix} \mu_{12}^{rx}(\frac{1}{m}+\frac{r^2}{J}) & 0 & 0 \\ 0 & (\mu_{12}^{np})^2(\frac{1}{m}) & 0 \\ 0 & 0 & \mu_{12}^{rz}\left(\frac{1}{m}+\frac{r^2}{J}\right) \end{bmatrix}
\tag{139}
$$

It is assumed that this contact configuration corresponds to the first mode. Therefore the control input for this mode can be shown as:

$$
v_1' = \begin{bmatrix} \dot{z}_{12}^{rx} & \dot{z}_{12}^{np} & \dot{z}_{12}^{rz} \end{bmatrix}^T
\tag{140}
$$

Consider systems begin to roll inside the V-groove (second mode), this means that six constraints are active. The constraints are as follows:

$$
\begin{aligned}
w_{12}^{rx} &= \dot{\mathbf{r}}_1 \cdot (\hat{\mathbf{t}}_x)_{12} + r\omega_1 \cdot (\hat{\mathbf{t}}_z)_{12} \\
w_{12}^{np} &= (\dot{\mathbf{r}}_1 - \mathbf{c}_1) \cdot \hat{\mathbf{n}}_{12} - r \\
w_{12}^{rz} &= \dot{\mathbf{r}}_1 \cdot (\hat{\mathbf{t}}_z)_{12} - r\omega_1 \cdot (\hat{\mathbf{t}}_x)_{12}
\end{aligned}
\tag{141}
$$

$$
\begin{aligned}
w_{13}^{rx} &= \dot{\mathbf{r}}_1 \cdot (\hat{\mathbf{t}}_x)_{13} + r\omega_1 \cdot (\hat{\mathbf{t}}_z)_{13} \\
w_{13}^{np} &= (\dot{\mathbf{r}}_1 - \mathbf{c}_3) \cdot \hat{\mathbf{n}}_{13} - r \\
w_{13}^{rz} &= \dot{\mathbf{r}}_1 \cdot (\hat{\mathbf{t}}_z)_{13} - r\omega_1 \cdot (\hat{\mathbf{t}}_x)_{13}
\end{aligned}
\tag{142}
$$

where c_2, c_3 shows the contact point on the second and the third object. The Jacobian matrix for this contact configuration is calculated as:

$$(\mathbf{J}_s)_2 = \begin{bmatrix} \mathbf{J}_s^{11} & \mathbf{J}_s^{21} \\ \mathbf{J}_s^{21} & \mathbf{J}_s^{22} \end{bmatrix} \tag{143}$$

where

$$\mathbf{J}_s^{11} = \begin{bmatrix} \mu_{12}^{rx}(\frac{1}{m}+\frac{r^2}{J}) & 0 & 0 \\ 0 & (\mu_{12}^{np})^2(\frac{1}{m}) & 0 \\ 0 & 0 & \mu_{12}^{rz}(\frac{1}{m}+\frac{r^2}{J}) \end{bmatrix} \tag{144}$$

$$\mathbf{J}_s^{12} = \begin{bmatrix} \mu_{12}^{rx}(\Psi_{11}\frac{1}{m}+\Psi_{33}\frac{r^2}{J}) & \mu_{12}^{rx}(\Psi_{12}\frac{1}{m}) & \mu_{12}^{rx}(\Psi_{13}\frac{1}{m}-\Psi_{31}\frac{r^2}{J}) \\ (\mu_{12}^{np})^2\Psi_{21}(\frac{1}{m}) & (\mu_{12}^{np})^2(\Psi_{22}\frac{1}{m}) & (\mu_{12}^{np})^2\Psi_{23}(\frac{1}{m}) \\ \mu_{12}^{rz}(\Psi_{31}\frac{1}{m}-\Psi_{13}\frac{r^2}{J}) & \mu_{12}^{rz}(\Psi_{32}\frac{1}{m}) & \mu_{12}^{rz}(\Psi_{33}\frac{1}{m}+\Psi_{11}\frac{r^2}{J}) \end{bmatrix} \tag{145}$$

$$\mathbf{J}_s^{21} = \begin{bmatrix} \mu_{13}^{rx}(\Psi_{11}\frac{1}{m}+\Psi_{33}\frac{r^2}{J}) & \mu_{13}^{rx}(\Psi_{21}\frac{1}{m}) & \mu_{13}^{rx}(\Psi_{31}\frac{1}{m}-\Psi_{13}\frac{r^2}{J}) \\ (\mu_{13}^{np})^2\Psi_{12}(\frac{1}{m}) & (\mu_{13}^{np})^2(\Psi_{22}\frac{1}{m}) & (\mu_{13}^{np})^2\Psi_{32}(\frac{1}{m}) \\ \mu_{13}^{rz}(\Psi_{13}\frac{1}{m}-\Psi_{31}\frac{r^2}{J}) & \mu_{13}^{rz}(\Psi_{23}\frac{1}{m}) & \mu_{13}^{rz}(\Psi_{33}\frac{1}{m}+\Psi_{11}\frac{r^2}{J}) \end{bmatrix} \tag{146}$$

$$\mathbf{J}_s^{11} = \begin{bmatrix} \mu_{13}^{rx}(\frac{1}{m}+\frac{r^2}{J}) & 0 & 0 \\ 0 & (\mu_{13}^{np})^2(\frac{1}{m}) & 0 \\ 0 & 0 & \mu_{13}^{rz}(\frac{1}{m}+\frac{r^2}{J}) \end{bmatrix} \tag{147}$$

and also each element of matrix $\mathbf{\Psi}$ shows the angle between the two different contact coordinates. This matrix is defined as:

$$\mathbf{\Psi} = \begin{bmatrix} (\hat{\mathbf{t}}_x)_{12} \cdot (\hat{\mathbf{t}}_x)_{13} & (\hat{\mathbf{t}}_x)_{12} \cdot \hat{\mathbf{n}}_{13} & (\hat{\mathbf{t}}_x)_{12} \cdot (\hat{\mathbf{t}}_z)_{13} \\ \hat{\mathbf{n}}_{12} \cdot (\hat{\mathbf{t}}_x)_{13} & \hat{\mathbf{n}}_{12} \cdot \hat{\mathbf{n}}_{13} & \hat{\mathbf{n}}_{12} \cdot (\hat{\mathbf{t}}_z)_{13} \\ (\hat{\mathbf{t}}_z)_{12} \cdot (\hat{\mathbf{t}}_x)_{13} & (\hat{\mathbf{t}}_z)_{12} \cdot \hat{\mathbf{n}}_{13} & (\hat{\mathbf{t}}_z)_{12} \cdot (\hat{\mathbf{t}}_z)_{13} \end{bmatrix} \tag{148}$$

Finally, control input can be shown as:

$$v'_2 = \begin{bmatrix} \dot{z}_{12}^{rx} & \dot{z}_{12}^{np} & \dot{z}_{12}^{rz} & \dot{z}_{13}^{rx} & \dot{z}_{13}^{np} & \dot{z}_{13}^{rz} \end{bmatrix}^T \tag{149}$$

As it is shown, the resolution of the Jacobian matrix and the control input change according to the contact situation. In this example, the resolution changes from three to six when the system switches from the first mode to the second one. It is important to note that $\mathbf{Z}$ vector in all modes have the same dimension. For instance, $\dot{\mathbf{Z}}$ for the first and the second mode can be shown as:

$$\begin{bmatrix} v_{12}^{rx} & v_{12}^{np} & v_{12}^{rz} & 0 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} \dot{z}_{12}^{rx} & \dot{z}_{12}^{np} & \dot{z}_{12}^{rz} & \dot{z}_{13}^{rx} & \dot{z}_{13}^{np} & \dot{z}_{13}^{rz} \end{bmatrix}^T = \dot{\mathbf{Z}} \tag{150}$$

$$\begin{bmatrix} v_{12}^{rx} & v_{12}^{np} & v_{12}^{rz} & v_{13}^{rx} & v_{13}^{np} & v_{13}^{rz} \end{bmatrix}^T = \begin{bmatrix} \dot{z}_{12}^{rx} & \dot{z}_{12}^{np} & \dot{z}_{12}^{rz} & \dot{z}_{13}^{rx} & \dot{z}_{13}^{np} & \dot{z}_{13}^{rz} \end{bmatrix} = \dot{\mathbf{Z}} \tag{151}$$

The next step to design hybrid automaton is to define proper events and guard conditions. They will be developed in the following section.

4.3 Discussion on States and Events

In order to define the set of edges, guard conditions and domains, we should borrow some concepts from automaton literature. The definition of events is highly dependent on the definition of contacts (section 3.1). Consider the bodies i and j are in contact. A hybrid model, which describes the contact between these two arbitrary pair of objects, is depicted in figure 17 (events and modes are shown in the nomenclature).

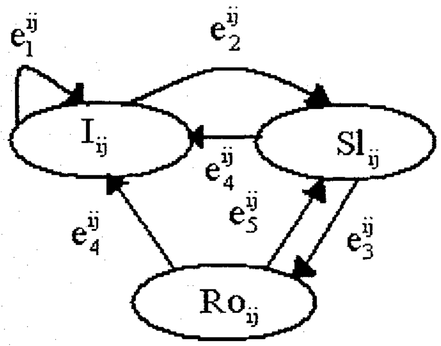


Figure 17. Hybrid Modelling for pair of objects in contact

Given events are describing physical phenomena. Also it is describing the guard conditions and domains in hybrid automaton framework. It is apparent that there are three possible contact configurations between a pair of objects. I_{ij}, Ro_{ij}, Sl_{ij} are showing these different cases; intermittent, rolling and slipping. The discrete states variables Q_1^{ij}, Q_2^{ij} and Q_3^{ij} can be shown mathematically with the help of functions Λ_{ij} and Γ_{ij} :

$$Q_1^{ij} = (1-\Lambda_{ij}) \qquad (152)$$

$$Q_2^{ij} = \Lambda_{ij}\Gamma_{ij} \tag{153}$$

$$Q_3^{ij} = \Lambda_{ij}(1-\Gamma_{ij}) \tag{154}$$

Five events completely are responsible for switching between the modes. Therefore the set of edges for this hybrid model can be shown as:

$$E = \{(Q_1^{ij},Q_1^{ij}),(Q_1^{ij},Q_2^{ij}),(Q_2^{ij},Q_3^{ij}),(Q_2^{ij},Q_1^{ij}),(Q_3^{ij},Q_1^{ij}),(Q_3^{ij},Q_2^{ij}\} \tag{155}$$

The guard conditions $G(Q_1^{ij},Q_1^{ij})$ physically mean that the collision is detected. This implies that the definition (*5*) must be taken into the consideration in the representation of the guard. Therefore, we have:

$$e_1^{ij} : G(Q_1^{ij},Q_1^{ij}) = \{X \mid w_{ij}^{np} < T_d \wedge \dot{w}_{ij}^{np} < -Tc\} \tag{156}$$

On edge $\{Q_i^{ij},Q_2^{ij}\}$, a continuous contact between two rigid body starts. Hence the hybrid system switches to the other mode of operation. According to the definition (*3*) and (*6*), the guard is described by:

$$e_2^{ij} : G(Q_1^{ij},Q_2^{ij}) = \{X \mid w_{ij}^{np} < T_d \wedge -T_c < \dot{w}_{ij}^{np} < \beta < T_s\} \tag{157}$$

where β is a positive constant. Two guards $\{Q_2^{ij},Q_1^{ij})$ and $\{Q_3^{ij},Q_1^{ij}\}$ imply the end of the continuous contact but from different modes. These guards can be defined with the help of definition (*6*) as:

$$e_4^{ij} : G(Q_3^{ij},Q_1^{ij}) = G(Q_2^{ij},Q_1^{ij}) = \{X \mid w_{ij}^{np} > T_s \vee z_{ij}^{np} < -\lambda\} \tag{158}$$

where λ is a positive threshold for the normal force. These conditions physically imply the end of the continuous contact, which is caused by the vanishing of the normal force in a smooth or non-smooth way. The last guard $G(Q_2^{ij}, Q_3^{ij})$ physically is about the vanishing of the tangential velocity, equation (57). Therefore, we have

$$e_3^{ij}G(Q_2^{ij}, Q_3^{ij}) = \{X \mid |w_{ij}^{rx}| < T_{cv} \wedge |w_{ij}^{rz}| < T_{cv}\} \tag{159}$$

where T_{cv} is a positive constant considered as tolerance for tangential velocity. The last event is responsible for switching from the rolling mode to the slipping mode; that is the existence of tangential velocity.

$$e_5^{ij}G(Q_2^{ij}, Q_3^{ij}) = \{X \mid |w_{ij}^{rx}| > T_{cv} \vee |w_{ij}^{rz}| > T_{cv}\} \tag{160}$$

After constructing the hybrid automaton model for a pair of objects, we can construct the general hybrid automaton model for the system of rigid bodies with the help of synchronous product or parallel composition.

Definition 8. Synchronous product is product between two automata $\vartheta_1 = (X_1, \Sigma_1, \eta_1), \vartheta_2 = (X_2, \Sigma_2, \eta_2)$; where X_1, X_2 is a set of discrete states corresponding to ϑ_1 and $\vartheta_2 . \Sigma_1, \Sigma_2$ are set of events and Finally η_1, η_2 are the states transition partial functions $X \times \Sigma \rightarrow X$. The product $\vartheta_1 \otimes \vartheta_2 = (X, \Sigma, \eta)$ is defined as follows:

$$X = X_1 \times X_2 \tag{161}$$

$$\Sigma = \Sigma_1 \cup \Sigma_2 \tag{162}$$

$$\eta((x_1,x_2),\sigma)=\begin{cases}(\eta_1(x_1,\sigma),\eta_2(x_2,\sigma)) & \sigma\in\Sigma_1\cap\Sigma_2,\text{if } \eta_1(x_1,\sigma)!,\eta_2(x_2,\sigma)! \\ (\eta_1(x_1,\sigma),x_2) & \sigma\in\Sigma_1-\Sigma_2,\text{if } \eta_1(x_1,\sigma)! \\ (x_1,\eta_2(x_2,\sigma)) & \sigma\in\Sigma_2-\Sigma_1,\text{if}\eta_2(x_2,\sigma)! \\ \text{otherwise it's not defined} & \end{cases} \tag{163}$$

As it is shown in figure 17, the set of modes for hybrid model between ith and jth body is $X_{ij}=\{I_{ij},Sl_{ij},Ro_{ij}\}$ and set of events can be expressed as $\Sigma_{ij}=\{e^1_{ij},e^2_{ij},e^3_{ij},e^4_{ij}\}$. Transition partial function is described by:

$$\begin{aligned}&\eta_{ij}(I_{ij},e^{ij}_1)=I_{ij}\\&\eta_{ij}(I_{ij},e^{ij}_2)=Sl_{ij}\\&\eta_{ij}(Sl,e^{ij}_4)=I_{ij}\\&\eta_{ij}(Sl_{ij},e^{ij}_3)=Ro_{ij}\\&\eta_{ij}(Ro_{ij},e^{ij}_4)=I_{ij}\\&\eta_{ij}(Ro_{ij},e^{ij}_5)=Sl_{ij}\end{aligned} \tag{164}$$

We can conclude that the hybrid model for each individual pair of rigid bodies can be shown by $\vartheta_{ij}(X_{ij},\Sigma_{ij},\eta_{ij})$. According to the definition of synchronous product (*8*), the complete model could be found as:

$$\vartheta=\vartheta_{12}\otimes\vartheta_{13}\otimes\cdots\vartheta_{1n}\otimes\vartheta_{23}\otimes\vartheta_{24}\otimes\cdots\varphi_{2n}\cdots\vartheta_{(n-2)(n-1)}\otimes\vartheta_{(n-2)n}\otimes\vartheta_{(n-1)n} \tag{165}$$

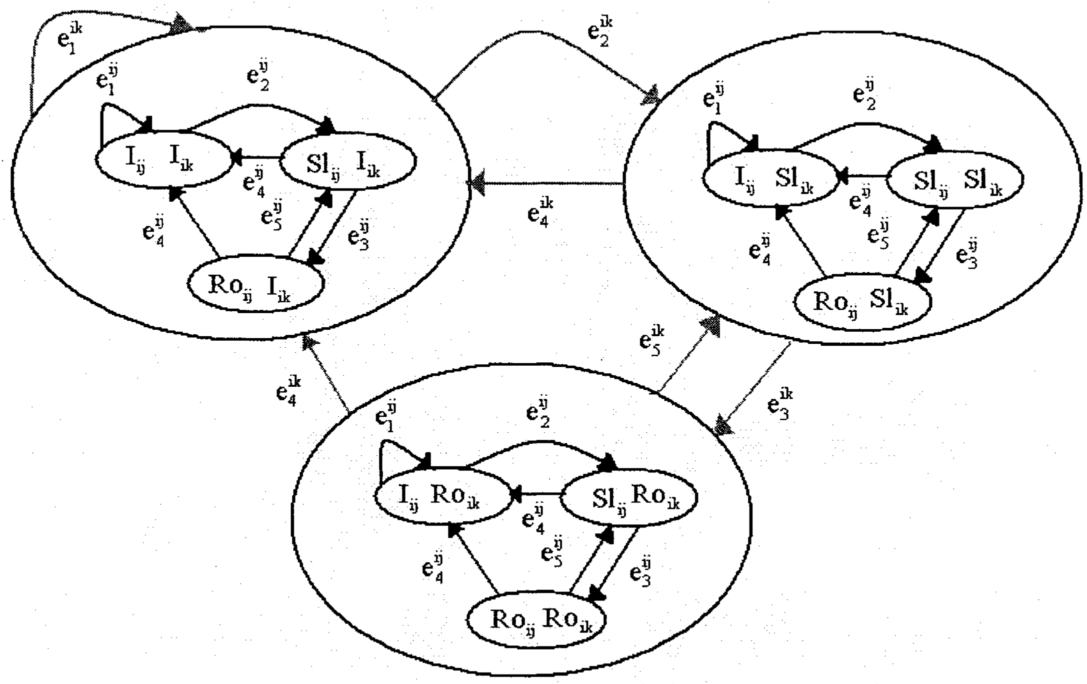


Figure 18. Synchronous products $\vartheta_{ij} \otimes \vartheta_{ik}$

For example consider the case in which three objects are present in the simulation. Therefore, the model can be constructed as $\vartheta = (\vartheta_{12} \otimes \vartheta_{13}) \otimes \vartheta_{23}$. The figure 18 shows the synchronous product graphically. It looks very similar to the figure 17 but each upper layer mode consists of some sub-modes. Each event for upper layer mode (shown by grey lines) could be occurred for its individual sub-modes. At the moment, the events is triggered at any sub-mode, systems goes to the corresponding sub-mode of the other upper layer mode. For example, consider a case event e_3^{ik} happens when the system is located at (Sl_{ij}, Sl_{ik}). Then system switches to (Sl_{ij}, Ro_{ik}) since a corresponding sub-mode is Sl_{ij}.

4.4 Reducing the Complexity of the Hybrid Automaton

The number of modes is highly dependent on the number of objects in the simulation. It can be calculated according to the equation (117). It is clear that increase in the number of modes makes the problem more complicated. In this section, we are looking for introducing ways to reduce this number.

It is known that collision detection problem could be broken up into two sub-problems: broad phase and narrow phase. In a broad phase, we identify the pair of objects we need to consider for possible collision. On the other hand, we consider the distance computation problem as well as collision response in the narrow phase. For the first phase, a kind of simple bounding volume is used to solve the problem [27]. Therefore, it is quick to eliminate objects that cannot possibly interact with each other. We denote the number of objects that cannot possibly interact with each other as n'. Therefore we can calculate the number of states as:

$$\Pi = 3^{\binom{n-n'}{2}} \qquad (166)$$

For example, consider a simulation that we have 10 objects. Assume that more often minimum 6 of the objects cannot interact with each other. For this case, we decrease the number of states from $2.95e+21$ to maximum 729 states with the help of narrow and broad phase. Figure 19 shows that maximum four objects can interact with each other in the shaded bounding volume.

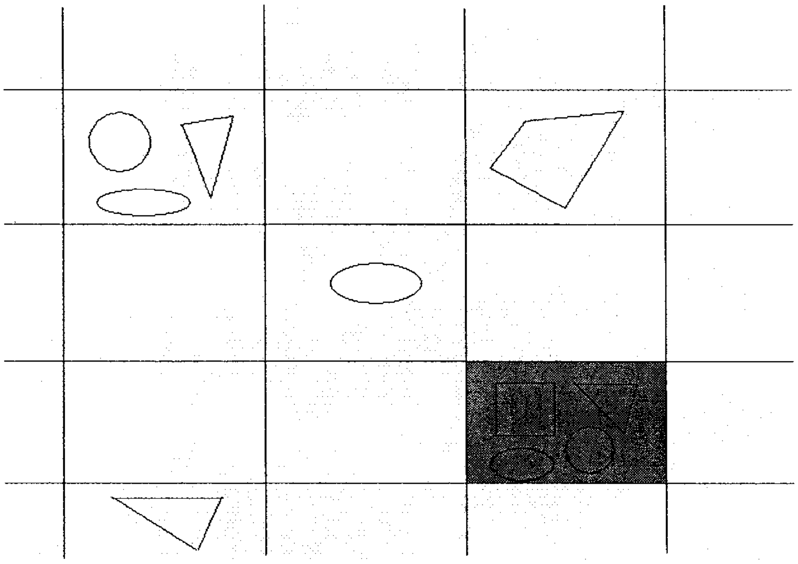


Figure 19. Bounding volume and broad phase

Although we decrease the number of states, it seems that it should be reduced more. We consider three possible contact configurations between each pair of objects. In most of the cases, the slipping mode is too short and it is like a transient period to rolling or sticking mode. The duration of this transient period is highly dependent on the coefficient of dynamic friction. More friction implies the less transient time. The alternative is to model slipping contact as series of micro-collisions. Hence the model used for intermittent mode can also be used for the transient slipping mode. The same strategy is done in the impulse-based simulations. In impulse-based method, all contact configurations (rolling, slipping, and sticking) are modeled as the series of tiny micro-collision. Unlike the impulse-based method, we do not model a rolling or sticking mode as series of micro-collision. Our decision is based on two main reasons. First, the tiny micro-collisions behave like very low-amplitude, high frequency impulses on each body. In consequence, the time integral of force times velocity (work done) show increase with respect to time. It gradually adds energy into the system. The amount of energy in a short transient slipping period is small. Therefore it doesn't affect the simulation too much. Secondly, considering another mode for sticking and rolling case helps the system to avoid from Zeno behaviour. Finally, we reduced the number of contact situations between each pair from three to two. Then, the number of states is given by:

$$\Pi = 2^{\binom{n-n'}{2}} \tag{167}$$

In the second step the number of states are reduced from 729 to 64. Briefly, we can summarize the reduction algorithm in the following proposition. The proposed algorithm is graphically summarized in figure 20.

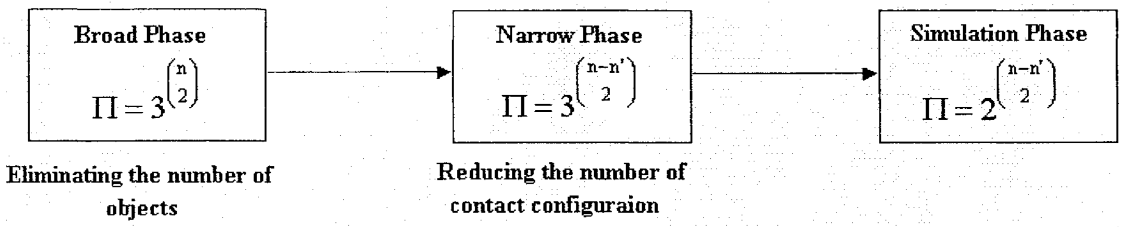

Broad Phase
$\Pi = 3^{\binom{n}{2}}$
Narrow Phase
$\Pi = 3^{\binom{n-n'}{2}}$
Simulation Phase
$\Pi = 2^{\binom{n-n'}{2}}$
Eliminating the number of objects
Reducing the number of contact configuraion

Figure 20. Reduction algorithm

Proposition 5 To reduce the hybrid automaton complexity, first the problem is divided into two sub-problems: broad phase and narrow phase. Objects that cannot possibly interact with each other are eliminated in the broad phase. Secondly, the number of contact configurations between a pair of object is reduced from three to two configurations by modelling a slipping contact as series of tiny micro-collision.

4.4.1 Zeno Behaviour

In this section, firstly we introduce Zeno paradox. It is desired to move from point A to point B in a finite time. But to reach B, we must first reach M; the midpoint of AB, and to reach that, we must first reach the midpoint of AM. Since space is infinitely divisible, we have to reach an infinite number of midpoints in a finite time. This phenomenon is called as Zeno paradox.

An execution of a hybrid system may exhibit infinitely many discrete jumps in finite time. This phenomenon is denoted as Zeno behaviour. As mentioned in the previous section, considering another mode for the sticking and rolling cases helps the system to avoid from Zeno behaviour. To define the Zeno mathematically, it is a need to define a hybrid time set.

Definition 9. A *hybrid time set* is a finite or infinite sequences of interval $T = \{I_i\}_{i=0}^{N}$ such that

- $I_i = [t_i, t_i']$ for all $i < N$;
- if $N < \infty$ then either $I_N = [t_n, t_n']$ or $I_N = [t_n, t_n']$;
- $t_i \leq t_i' = t_{i+1}$ for all i.

Note that the right end point of one interval coincides with the left end point of the following interval. The interpretation is that these are the times at which events are triggered. Consider a set, which includes all event times, $S_{\text{event}} : \{t_1, t_2, \ldots, t_n\}$. If this set is closed and countable and also $0 \in S_{event}$, we called that set as an admissible event times set. With the help of hybrid time set, we are able to give a mathematical representation of Zeno behaviour.

Definition 10. An admissible event time sets S_{event} is *left Zeno free* if

- for all $t' > t, (t, t') \cap S_{event}$ is not empty

and is *right Zeno free* if

- for all $t' < t, (t', t) \cap S_{event}$ is not empty.

In the following subsection, we show how we can solve the Zeno problem by introducing a new mode of sticking.

4.4.1.1 Example

A model for a bouncing ball can be represented as a simple hybrid system with one mode. A dimension of the continuous state is two. x_1 is the vertical position of the ball , x_2 is the velocity and g represent the gravitational force. The ball bounces when $x_1 = 0$. At each bounce, the ball loses a fraction of its energy. This happens when we reset the velocities according to the Poisson's law of restitution $x_2 := -cx_2$.

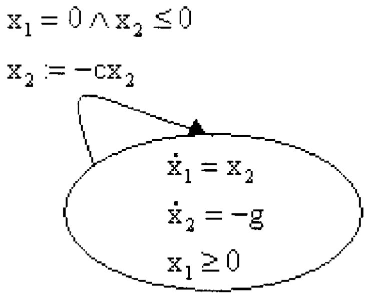


Figure 21. The simplest hybrid model for bouncing ball

Assume $x_1(0)=0$ and $x_2(0)=x_2^0$, event times are related through $t_{i+1}=t_i+\frac{2c^i x_2^0}{g}$.

This sequence has finite limit $t^* = \frac{2x_2^0}{g-gc}$. Ball is at rest within finite time span, but after infinitely many bounces. Therefore, the set of event times has right Zeno.

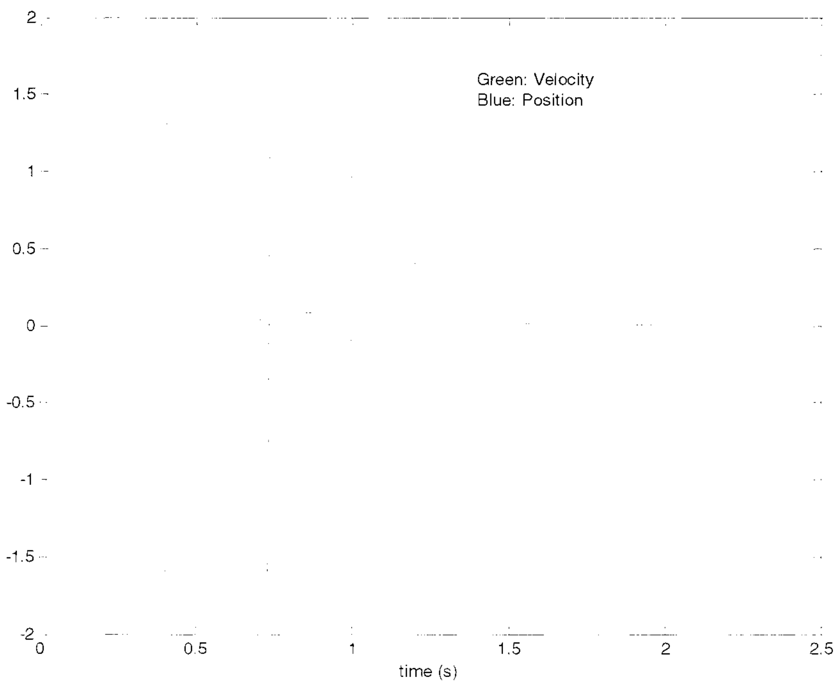


Figure 22. Zeno behaviour for bouncing ball

Zeno behaviour is clearly seen in the figure 22. The jumps in velocity happen as a result of collision. To solve the problem, it is required to introduce a new mode for our hybrid scheme. This new hybrid system is shown in the figure 23.

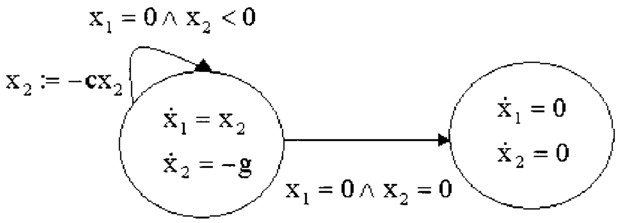


Figure 23. Modification of hybrid system to solve the Zeno problem

Finally, the solution for Zeno problem is depicted in figure 24. The second mode of operation illustrates the sticking mode for the ball.

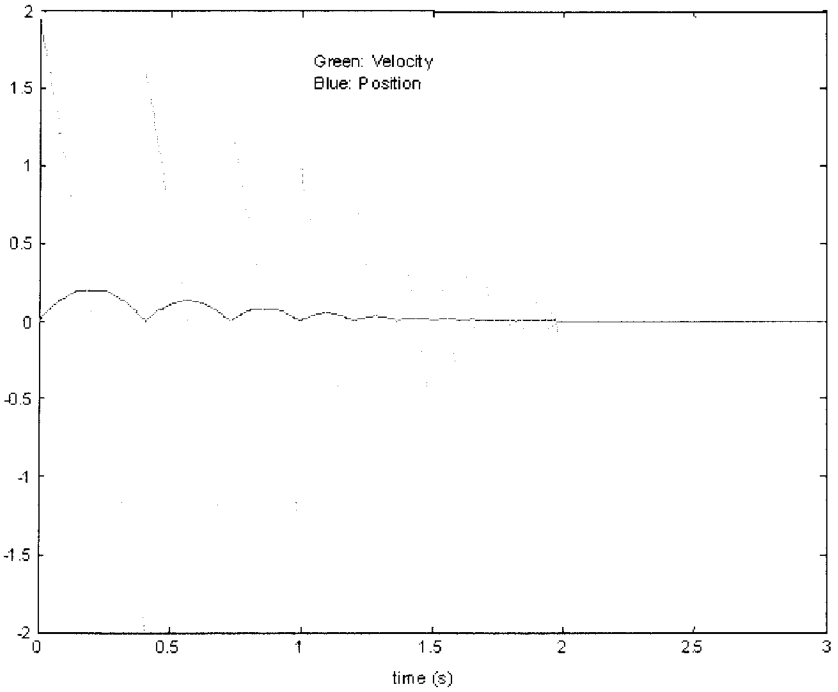


Figure 24. Solution of the Zeno problem

Finally, the solution for Zeno problem is depicted in figure 24. The second mode of operation illustrates the sticking mode for the ball.

4.5 Applications in 3D Simulations

A reduced hybrid model, which describes the model for V-Groove problem is depicted in figure 25. There are only four modes of operation in the reduced model. It is important to note that the slipping contact is modeled as a series of micro collision. Consequently, modes of operations correspond to slipping contact; (I_{12}, Sl_{13}), (Sl_{12}, I_{13}) and (Sl_{12}, Sl_{13}), are merged in mode (I_{12}, I_{13}).

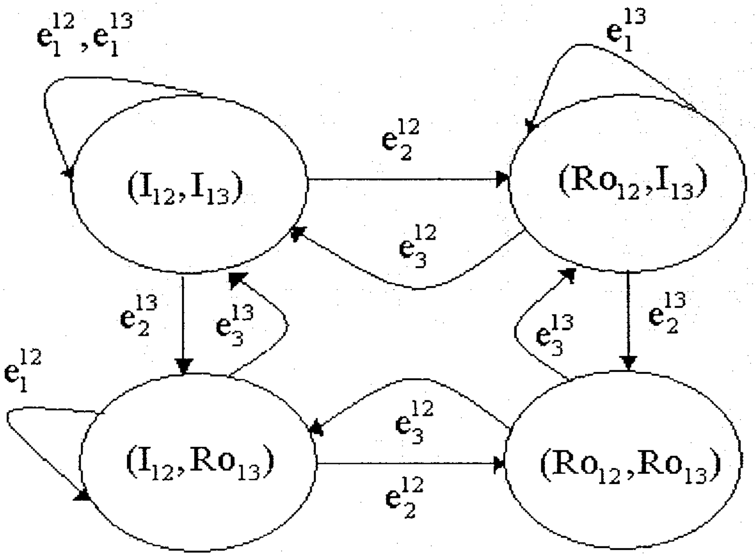


Figure 25. Hybrid Model for V-Groove problem

Five events completely are responsible for switching between the modes. The events e_1^{12} and e_1^{13} physically mean that collision is detected between the ball and surfaces of V-groove. They can be shown as:

$$e_1^{12} := \{X \mid w_{12}^{np} < T_d \wedge \dot{w}_{12}^{np} < -T_C\} \tag{168}$$

$$e_1^{13} := \{X \mid w_{13}^{np} < T_d \wedge \dot{w}_{13}^{np} < -T_C\} \tag{169}$$

When event e_2^{12} or e_2^{13} is triggered, a rolling or sticking contact between the ball and the corresponding surface of V-groove starts. They are mathematically expressed as:

$$e_2^{12} := \{X \mid w_{12}^{np} < T_d \wedge -T_c < \dot{w}_{12}^{np} < \beta \wedge \sqrt{(w_{12}^{rx})^2 + (w_{12}^{rz})^2} < T_{cv}\} \tag{170}$$

$$e_2^{13} := \{X \mid w_{13}^{np} < T_d \wedge -T_c < \dot{w}_{13}^{np} < \beta \wedge \sqrt{(w_{13}^{rx})^2 + (w_{13}^{rz})^2} < T_{cv}\} \tag{171}$$

The last events e_3^{12} and e_3^{13} are responsible for switching from the rolling modes to intermittent/Slipping modes. This may happen due to the existence of tangential velocity or vanishing of a normal force. They are given by:

$$e_3^{12} := \{X \mid w_{12}^{np} > T_s \vee z_{12}^{np} < -\lambda \vee \sqrt{(w_{12}^{rx})^2 + (w_{12}^{rz})^2} > T_{cv}\} \tag{172}$$

$$e_3^{13} := \{X \mid w_{13}^{np} > T_s \vee z_{13}^{np} < -\lambda \vee \sqrt{(w_{13}^{rx})^2 + (w_{13}^{rz})^2} > T_{cv}\} \tag{173}$$

Finally, we test our method with different initial conditions for V-groove problem. The simulation shows the rolling ball and V-groove example in real 3D environment. A ball rolls down the surface of V-groove and then hits another surface and after that it goes down the V-groove. The ball and V-groove 3D simulation is shown in the figure 26.

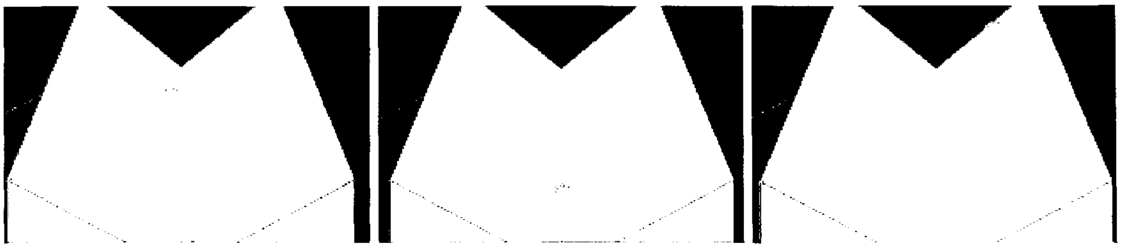


Figure 26 Ball and V-Groove 3D simulation

The simulation is tested and ball trajectories is shown in figure 27 . Firstly, ball is in the ballistic motions. Then hits the first surface and bounces. After a period of bouncing, ball starts to rolls on the surface until the moment at which hits the second surface. After hitting the second surface, ball bounces back and forth between two surfaces and finally rolls down the V-groove.

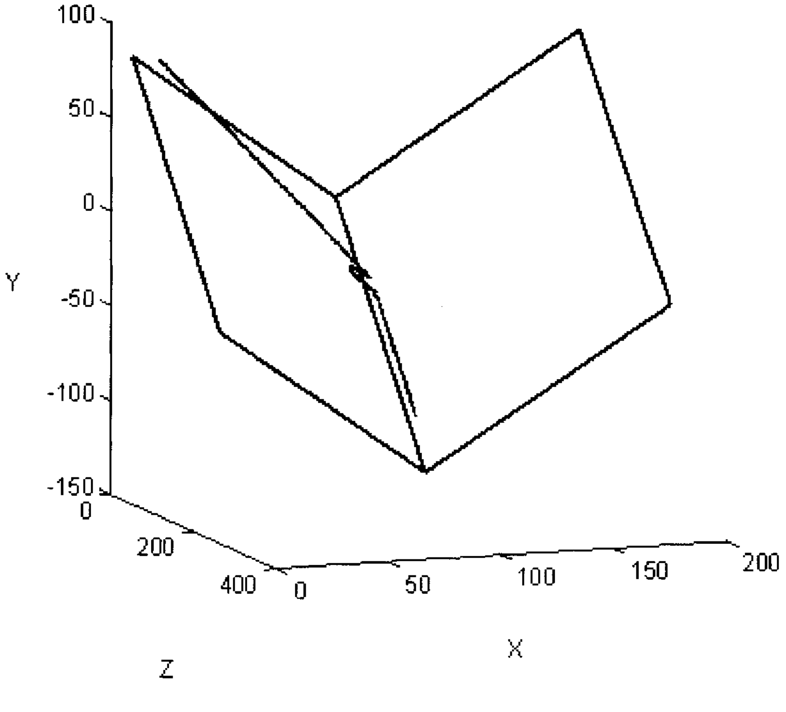


Figure 27 Ball and V-Groove 3D simulation

Most importantly, the mode transition is investigated via different initial conditions. It is clear from figures 28 and 29 that there is no chattering between the modes. An integer number is assigned to each mode, where $1,2,3$ and 4 are respectively representing the states (I_{12},I_{13}), (Ro_{12},I_{13}), (I_{12},Ro_{13}) and (Ro_{12},Ro_{13}). The system is initially located at the first mode. After a while bouncing contacts switches to continuous rolling contact. In consequence, system switches to the second mode. The transition from continuous rolling contact to intermittent one happens due to the collision to the second surface. Hence, system switches back to the first mode. Finally, ball rolls down the V-groove and system switches to the fourth mode. It is important to note that the switching from the first mode to the fourth mode is not possible according to our automaton design. The zoom in-figures are clearly showing this fact.

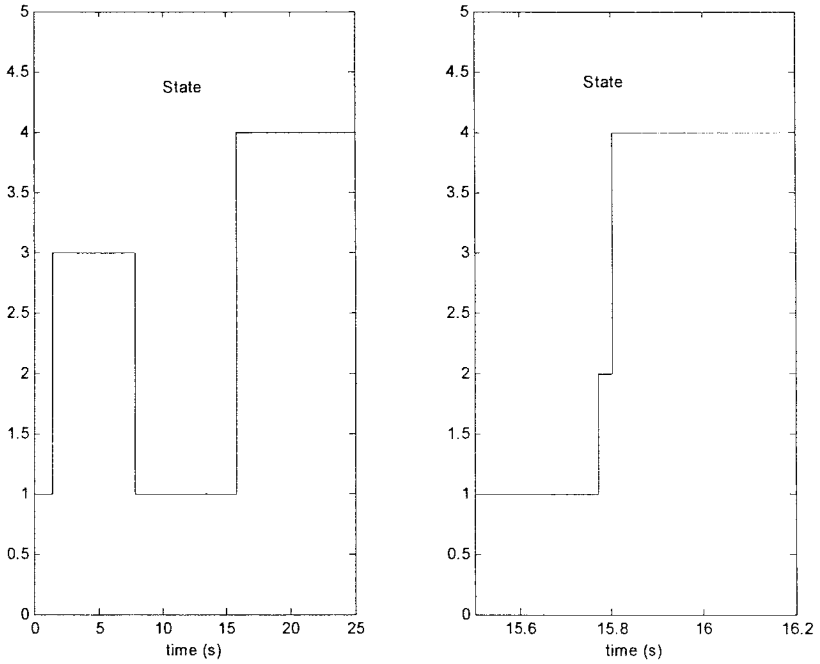


Figure 28 State transitions for a case in which ball is dropped near the first surface

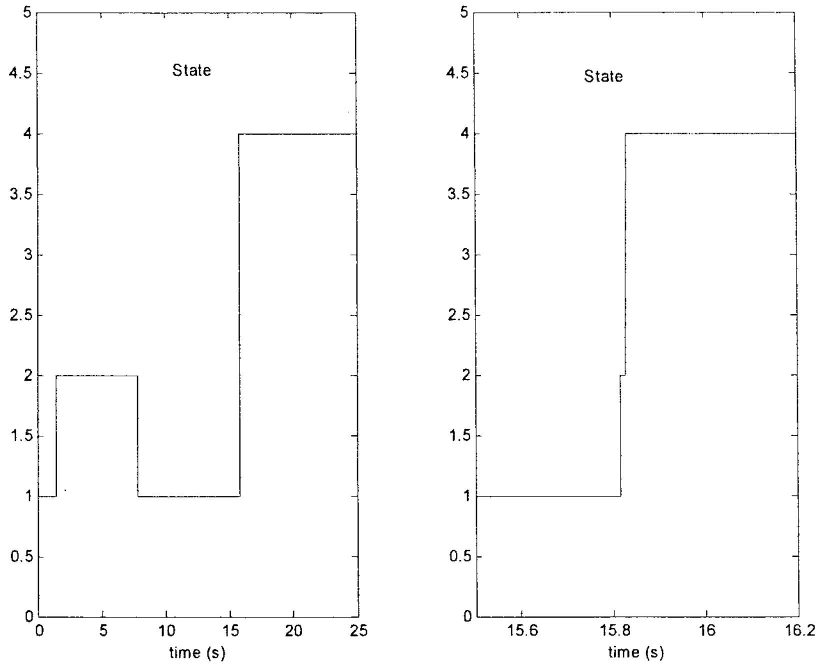


Figure 29 State transitions for a case in which ball is dropped near the second surface

5 . Conclusions and Future Work

5.1 Conclusions

Constrained-based simulation is a well-studied topic due to its applicability to a large class of physical problems. On the other hand, the impulse-based simulation is a recent approach that is particularly suitable for collision intensive systems. However, neither technique is perfect to address physical system simulations during both collision and non-collision modes. Hence, a hybrid model that utilizes both methods effectively under a hybrid automaton framework enables much more realistic physical simulation. This thesis introduces a new hybrid design for simulating rigid bodies in various contact situations.

In the third chapter, a hybrid simulation system that combines impulse-based methods and constrained-based methods is presented. It is shown that the collision response problem can be modeled effectively by combining the singularly perturbed sliding manifold DAE realization with the impulse-based method. A recent DAE realization, singularly perturbed sliding manifold approach, is utilized for simulation rigid bodies in continuous contact. An explicit state space approximation of these DAEs is constructed consequently. Furthermore, a hybrid system that transform from ODEs (no contact) to DAEs (continuous contact) is analyzed within the hybrid automaton framework. Some examples are provided to demonstrate the efficiency of the proposed method to model two rigid bodies in contact.

In the fourth chapter, generalized hybrid automation framework for more than two rigid bodies is introduced. A solution to the problem of multi-resolution under hybrid automaton framework is proposed to implement the hybrid simulation. In addition, a model reduction algorithm is presented. Model reduction algorithm reduces the number of possible contact situations in two consecutive steps. First, the number of modes decreases with the help of broad and narrow phase strategy. Next the number of states is decreased by modelling a slipping contact as a series of tiny micro collisions. Proper examples are given to demonstrate the effectiveness of the proposed algorithms. Finally, a unique solution to Zeno problem for the proposed hybrid systems is suggested.

The contributions in this thesis are summarized as follows:

- A new hybrid simulation approach for modelling multiple rigid bodies in contact under a hybrid automaton framework
- The approach combines the efficiency of the impulse-based method for intermittent contact and the SPSM approach for continuous contact
- A physics based hybrid model reduction algorithm is proposed to reduce model complexity
- A method is proposed to eliminate Zeno Behaviour

This thesis represents the first hybrid modelling approach that combines DAE constraint stabilization (SPSM) with the impulse method. The proposed approach is the first hybrid modelling approach suitable for hard real-time virtual reality applications.

5.2 Future Work

In this section, we state the possible future directions of this research work. Generally speaking, one of the main problems of hybrid systems is the chattering problem. The hybrid system literature is loose in this area. In consequent, a tuning procedure for the parameters is a need to avoid chattering. Finding a systematic or hierarchical way to tune the parameters could be a great step to make the hybrid design more effective. Briefly, finding a systematic method for selecting simulation parameters is the main direction of our research.

Implementing the hybrid simulation on the cluster of computers helps us to improve the real time performance of the simulator. For instance, a computation of each state can be assigned to a different computer. It is apparent from a broad phase algorithm that each bounding volume of space is independent from the other ones; this makes the problem to suit well for the cluster of computers.

Our future direction can be summarized as:

- Systematic methods for selecting simulation parameters
- Distributed simulation for computational clusters
- Theoretical analysis of chattering and Zeno behaviour

6 . References

[1] Baraff. D., 1989, "Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies," *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pp. 223-232.

[2] Baraff. D., 1994, "Fast Contact Force Computation for Non-Penetrating Rigid Bodies," *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp.23-34.

[3] Mirtich, B., and Canny, J., 1995, "Impulse-based Simulation of Rigid Bodies," *Proceedings of the 1995 symposium on Interactive 3D graphics*, Monterey, California, pp. 181-ff.

[4] Sauer, J., Schomer, E., and Lennerz C., 1998, "Real-Time Rigid Body Simulations Of Some Classical Mechanics Toys," *10^{th} European Simulation Symposium and Exhibition,* pp. 93-98

[5] Mirtich, B., 1995, "Hybrid Simulation: Combining Constraints and Impulses," *Proceedings of First Workshop on Simulation and Interaction in Virtual Environments*

[6] Mosterman, P. J., Zhao F., and Biswas, G., 1997, "Model Semantics and Simulation for Hybrid Systems Operating in Slidng Reimes," *Proceedings of the AAAI Fall Symposium 97 on Model-directed Autonomous Systems*, Boston, MA, pp. 48-55.

[7] João P. Hespanha., " Lecture notes on hybrid control and switched systems", http://www.ece.ucsb.edu/~hespanha/ece229/, University of California at Santa Barbara

[8] K. H. Johansson, J. Lygeros, and S. Sastry. "Modelling of Hybrid systems," In H. Unbehauen, Ed., Encyclopedia of Life Support Systems (EOLSS), Theme 6.43: Control Systems, Robotics and Automation. Developed under the auspices of UNESCO, 2004. Article-level contribution. Invited paper

[9] Barros, F. J., Zeigler B. P., and Fishwick, P. A., 1998,"Multimodel and Dynamic Strcuture Models: An Integration of DSDE/DEVS and OOPM," *Winter Simulation Conference*, Washington DC, pp.413-419.

[10] Urquia, A., and Dormido, S., 2003, "Object-Oriented Description of Hybrid Dynamic Systems of Variable Structure," SAGE J. Simulation, **79(9)**, 485-493.

[11] Brenan, K., Campbell, S., and Petzold, L., 1989, *Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, Notrh-Holland, Amsterdam.

[12] Luenberger, D. G., 1977, "Dynamic Equations in Descriptor Form," IEEE Trans. Autom. Control, **22(3),** pp. 312-321.

[13] Gu, B., and Asada, H. H., 2004, "Co-Simulation of Algebraically Coupled Dynamic Subsystems Without Disclosure of Proprietary Subsystem Models," ASME J. Dyn. Syst., Meas., Control, **126,** pp. 1-13.

[14] Gordon, B. W., and Liu, S., 1998, "A Singular Perturbation Approach for Modeling Differential-Algebraic Systems," ASME J. Dyn. Syst., Meas., Control, **120,** pp. 541-545.

[15] Ascher, U., Chin, H., Petzold, L., Reich, S., 1995, "Stabilization of Constrained

Mechanical Systems with DAEs and Invariant Manifolds," J. Mech. Struct. Mach., **123**, pp. 135-158.

[16] Gordon, B. W., and Asada, H., 2000, "Modeling, Realization, and Simulation of Thermo-Fluid Systems Usuing Singularly Perturbed Sliding Manifolds," ASME J. Dyn. Syst., Meas., Control, **122,** pp. 699-707.

[17] Hahn, K. J., 1988, "Realistic Animation of Rigid Bodies," *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp. 299-308.

[18] Moore, M., and Wilhelms, J., 1988, "Collision Detection and Response for Computer Animation," *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp.289-298.

[19] Stewart, D. E., and Trinkle J. C., "An implicit time stepping scheme for rigid-body dynamics with inelastic collisions and coulomb friction," International Journal **of** Numerical Methods for Engineering, volume 39, pages 2673-2691, 1996.

[20] Stewart, D. E., and Trinkle J. C., "An implicit time-stepping scheme for rigid-body dynamics with coulomb friction, " *Proc. IEEE Int. Conf. Robot.Autom. (ICRA),* pages 162-169, San Francisco, CA, 2000.

[21] Kry, P. G., and Pai, D. K., "Continuous contact simulation for smooth surfaces," ACM Trans. Graph. 22, 1 (Jan. 2003), 106-129

[22] Wookho, S., Trinkle, J. C., and Amato, N.M., 2001, "Hybrid dynamic simulation of rigid-body contact with Coulomb friction," *IEEE International Conference on Robotics and Automation*, 2001. Proceedings 2001 ICRA. Volume 2, 2001 Page(s):1376 – 1381

[23] Rabier, P. J., and Rheinboldt, W. C., 2000, *Nonholonomic Motion of Rigid*

Mechanical Systems form DAE Viewpoint, SIAM, Philadelphia.

[24] Luca, A. D., and Oriolo, G., 1995, "Modelling and Control of nonnholonomic mechanical systems," www.dis.uniroma1.it/~labrob/pub/papers/CISM95.pdf

[25] Gordon, B. W., 1999, "State Space Modeling of Differential-Algebraic Systems Using Singularly Perturbed Sliding Manifolds," Ph.D thesis, MIT, Mechanical Engineering Dept., Cambridge.

[26] Lewis, A. D., and Murray, R. M., 1995, "Variational principals for constrained systems: theory and experiment," International Journal of Non-linear Mechanis, **30(6)**, pp. 793–815

[27] Guibas, L. J., Xie F., and Zhang L., "Kinetic Collision Detection: Algorithms and Experiments", *Proc. IEEE Int. Conf. Robot.Autom. (ICRA)* , 2001