Pattern-Guided Simulations of Immersed Rigid Bodies

Haoran Xie*
The University of Tokyo

Kazunori Miyata†
Japan Advanced Institute of Science and Technology

Abstract

This paper proposes a pattern-guided framework for immersed rigid body simulations involving unsteady dynamics of a fully immersed or submerged rigid body in a still flow. Instead of the heavy computation of fluid-body coupling simulations, a novel framework considering different flow effects from the surrounding flow is constructed by parameter estimation of force coefficients. We distinguish the flow effects of the inertial, viscous and turbulent effects to the rigid body. It is difficult to clarify the force coefficients of viscous effect in real flow. In this paper we define the control parameters of viscous forces in rigid body simulator, and propose a energy optimization strategy for determining the time series of control parameters. This strategy is built upon a motion graph of motion patterns and the turbulent kinetic energy. The proposed approach achieves efficient and realistic immersed rigid body simulation results, and these results are relevant to the real-time animations of body-vorticity coupling.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: rigid body, motion patterns, parameter subspace

1 Introduction

Coupling between a rigid body and its surrounding flow is an important subject in computer animation. Immersed rigid body dynamics represents the motion of a rigid body fully immersed or submerged under real flow of water or air, that is strongly coupled with the surrounding flow. The strongly coupled fluid body interactions involve vortex shedding and flow instabilities. For clarity, T_f and T_r represent the characteristic timescales of fluid motion and coupled rigid body motion, respectively. If $T_f \gg T_r$, then the rate of change of the surrounding flow can be neglected in comparison with the rate of change of a body, e.g., a coffee cup falling to the ground, a conventional rigid body simulator is suitable in this case. If $T_f \ll T_r$, then, rather than solving coupling dynamics, a quasi-steady approximation of drag and lift forces using force coefficients, which is a simplified representation of aerodynamics, is sufficient. For a strongly coupled interaction, $T_f \approx T_r$, the quasi-steady approximation is invalid any more. Numerical difficulty arises from the nonlinearity and unsteadiness of the coupled motion due to the generation and detachment of vortices from the body's edge.

The simulations of immersed rigid bodies moving through still fluid flows require a combination of quantitative experiments and numerical models that characterize the state of the passive dynamics of the rigid bodies. This subject has attracted researchers' attention for over one hundred years, because of the intricate interaction between the motion of rigid bodies and the induced fluid reaction, and it is also challenging to achieve the natural simulation results in computer graphics. This paper aims to develop a data-driven approach for immersed thin rigid body dynamics without the computation of fluid motion while considering the flow effects. Examples of such bodies are abundant in everyday phenomena and computer animations, e.g., the motions of leaves, snowflakes, confetti, and petals. Although these phenomena about immersed rigid bodies commonly happen in a short timescale and people may have difficulty to distinct the motion details, this work becomes significant because (1) common user has subconscious understanding of everyday phenomena, (2) slow motions are useful visual effects in graphics productions, and (3) a realistic simulation approach is helpful for related fabrication designs involving unsteady aerodynamics.

In contrast to the conventional data-driven techniques based on the motion captures, the motion capture of an immersed rigid body (e.g., leaf) is difficult and inaccurate with the following reasons: (1) the object is so small and lightweight that the use of common marker is unacceptable; (2) the motion is so fast that high speed cameras are required; (3) the motion is so chaotic that numerous experimental and analysis works are required. We call this work pattern-guided because the proposed approach is based on the captured transitions of motion patterns, rather than the captured trajectory. The capture of the motion patterns is easier and more accurate than the direct motion capture of body's trajectory. In this framework, the captured data is recorded as transition probabilities in a motion graph to obtain control parameters.

A challenging problem is how to estimate the viscous forces without the computation of fluid motions. In order to solve this problem and efficiently simulate immersed rigid body dynamics, we present the dynamical model that considers inertial and viscous flow effects. For inertial effect, we propose a proxy geometry approximation method of thin rigid bodies, and compute added-mass tensors analytically to alleviate the limitations of numerical approaches. For viscous effect, we decompose the total viscous forces into three components, i.e., drag, translational lift, and rotational lift forces with unknown control parameters. Based on the proposed dynamical model, we herein present an data analysis of the three-dimensional motion patterns and their parameter subspaces for an individual body. Also, we present a parameter estimation of control parameters that adopts a specified motion graph to represent motion transitions among motion patterns, and reflects the influences of turbulent effects.

2 Related work

Rigid body simulation has a long history in computer graphics and is a significant starting point for character and deformable body simulations [Baraff 1993]. Traditional rigid body solvers do not consider influences from the surrounding flow in which a rigid body always falls vertically. The coupling of a rigid body with fluids has also been studied extensively in computer graphics. Two main strategies are commonly adopted: an Euler fluid and Lagrangian rigid body dynamics [Batty et al. 2007; Robinson-Mosher et al. 2008] and fully Lagrangian meshless methods [Becker et al. 2009; He et al. 2012]. These methods are too computationally heavy for

---

*e-mail: xiehr@acm.org
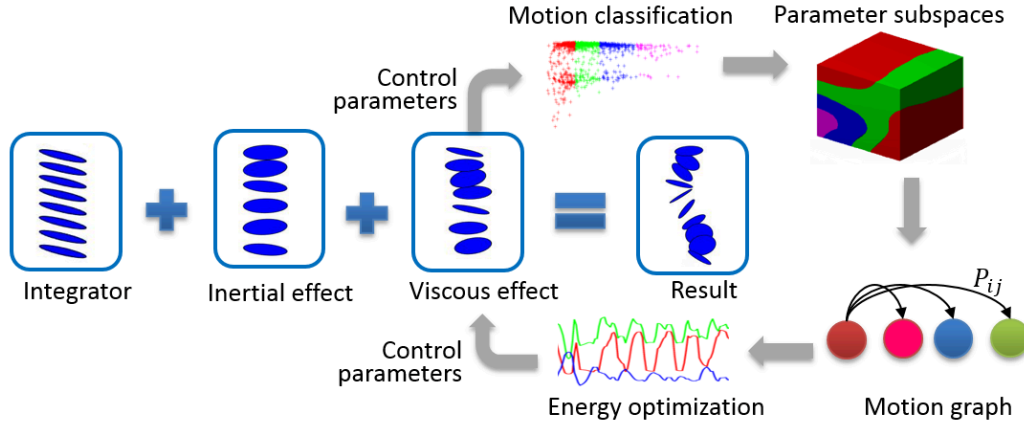
†e-mail: miyata@jaist.ac.jp

Figure 1: *The work flow of the proposed pattern-guided framework consists of a rigid body simulator, data training of motion patterns, and parameter estimation of control parameters. Note that the data training and parameter estimation need to be computed only once.*

immersed rigid body simulations.

Immersed rigid body simulations are closed to the aerodynamics animations, which have also been widely utilized in computer graphics. [Wei et al. 2004] proposed a lattice Boltzmann method for solving fluid simulation and the Kutta-Joukowski theorem for body dynamics. [Ozgen et al. 2010] presented a fractional derivatives method for representing the historic force of underwater cloth in a low Reynolds number flow. [Weissmann and Pinkall 2012] introduced the Kirchhoff tensor to explain inertial effects in inviscid flows. The others have explored the motions of immersed rigid body dynamics using Markov models [Reissell and Pai 2001], particle trajectory databases [Shi et al. 2005], stochastic model [Xie and Miyata 2013], and motion graph techniques [Xie and Miyata 2014]. Different with the previous approaches, this work simulates the unsteady dynamics of an immersed rigid body considering both inertial and turbulent effects from the surrounding flow.

In aerodynamics or hydrodynamics animations, the force coefficients of drag and lift forces play an important role. In previous work, the force coefficients have been considered heuristic constant [Selino and Jones 2013], function of Reynold number [Mihalef et al. 2009], function of angle of attack [Wu and Popović 2003; Okamoto et al. 2009; Ju et al. 2013], and function of both of them [Umetani et al. 2014]. In this work, the force coefficients are handled as explicit functions of angle of attack and implicit functions of Reynolds number to represent the unsteady dynamics of an immersed rigid body considering flow effects.

3 Overview

As illustrated in Figure 1, the whole work flow is divided into two parts: runtime simulator (in blue box) and precomputed steps (with gray arrow). The runtime simulator is constructed from the conventional rigid body integrator. In the dynamical model of rigid bodies, the flow effects are represented from two sides: the inertial effect of added mass tensors in an inviscid flow, and the viscous effect due to drag and lift force due to viscous flow are analysed separately.

Note that the flow effects from the surrounding flow are complicated. The simple starting point to explain these effects is potential flow theory [Umberto 2005], but the numerical model for flow motions in potential flow lacks two significant components in real flow situations: viscous forces and vorticity. However, because the added mass contributions do not change due to the rotational effects in a viscous flow [Mougin and Magnaudet 2002], i.e., the viscous

flow has the same inertial effect from potential theory. It is reasonable to combine the inertial and viscous effect together in the same dynamical model. Therefore, the flow effects have three sources: inertial effect force that can be calculated accurately, viscous effect force due to flow viscosity, and turbulent effect related to vortex shedding from the surrounding flow [Biesheuvel and Hagmeijer 2006].

To realize all the flow effects, the turbulent effect is implicitly embedded in viscous effect via estimations of control parameters in the dynamical model. Rather than motion capture of the motion trajectories of individual bodies, this work utilizes the capture of motion patterns which can be more efficient and accurate from falling objects experiments. After achieving the motion patterns and their transition probabilities, all the data is saved as motion graph. To connect the motion patterns found in experiments with the proposed dynamical model, the numerical motion patterns are extracted from numerous simulations in whole control parameter space. Therefore, the different parameter subspaces are constructed based on the motion classification in all sampled simulation data. Using the parameter subspace and motion graph, the turbulent effect is justified via a energy optimization process to access the possible parameter sequences, which could be utilized in runtime simulations directly.

We discuss the dynamical model with inertial and viscous effects in Section 4, and the training process of parameter subspaces from simulations data in Section 5. To estimate the control parameters of viscous forces, we describes the curvature-based motion planning method with turbulent energy optimization in Section 6.

4 Flow Effects

To obtain the dynamical model of the immersed rigid body in a real flow, this section describes the flow effects from two aspects: the inertial effect from potential theory, and the viscous effect generated from the surrounding flow.

4.1 Inertial Effect

Here, assume a rigid body B with mass m moving in an inviscid flow domain Ω with fluid density ρ_f at rest at infinity, the flow velocity field v has a scalar potential field ϕ in the flow domain due to the irrotational flow condition, and ϕ satisfies the following

Laplace equation:

$$\Delta\phi(\boldsymbol{z}) = 0 \quad \boldsymbol{z} \in \Omega \tag{1}$$

$$\nabla\phi(\boldsymbol{z})\cdot\boldsymbol{n} = \boldsymbol{u}_n(\boldsymbol{z}) \quad \boldsymbol{z} \in \partial B \tag{2}$$

$$\phi(\boldsymbol{z}) = 0 \quad \|\boldsymbol{z}\| \to \infty \tag{3}$$

where $\boldsymbol{n}$ is a surface normal vector and $\boldsymbol{u}_n(\boldsymbol{z})$ is the normal velocity of the rigid body at the surface point. From Equations (1) and (3), the fundamental solution of the Laplace equation in 3D is given by $G(x, y) = \frac{1}{4\pi r}$ with $r = \|x - y\|$, i.e., the distance from a source point x to a collocation point y at the body's surface. Then, the kinetic energy of the surrounding flow can be expressed as follows:

$$E = \frac{1}{2}\boldsymbol{u}\cdot(\mathbf{M}\boldsymbol{u}) + \frac{1}{2}\boldsymbol{\omega}\cdot(\mathbf{J}\boldsymbol{\omega}) \tag{4}$$

where u and ω are the translational and angular velocities of the immersed body, respectively; $\mathbf{M}$ and $\mathbf{J}$ are second-order tensors representing the added mass and added moment of inertia of flow due to the translational and rotational motions of the body, respectively. The values of the tensors are determined from the velocity potential value $\phi_i, 1 \le i \le 6$, which can be calculated using the boundary element method [Hess and Smith 1967].

$$K_{ij} = \rho_f \int_{\partial B} \phi_i \frac{\partial \phi_j}{\partial n} ds \tag{5}$$

Note that the added tensor $\mathbf{K}$ is symmetric in terms of Green's theorem, i.e., $\mathbf{K}_{ij} = \mathbf{K}_{ji}$. Here, $\mathbf{M}$ and $\mathbf{J}$ correspond to the value of $\mathbf{K}$ while $1 \le i, j \le 3$ and $4 \le i, j \le 6$, respectively. According to Equation (5), the added tensors are not dependent on the velocities of the body and are only a function of the body surface geometry.

Rigid bodies exhibiting sensitive viscous effects from the surrounding flow have common features: low body-to-fluid density ratio and thin structure with a large frontal area, such as leaves and pieces of paper. The results of numerical approaches [Hess and Smith 1967; Weissmann and Pinkall 2012] become inaccurate for thin structures, because the solution becomes nearly singular when the distance r is quite small. To solve this issue, we determined that an ellipsoid proxy [Zheng and James 2010] is efficient and sufficient for thin-structure rigid bodies. For more complex models, we suggest the adoption of the singularity weakening numerical approaches like sinh transformation [Johnston et al. 2013].

Proxy geometry approximation. The algorithm to obtain the axes of an input model is similar to bounding ellipsoid construction. For N vertices $\boldsymbol{v}_i$ of meshes, $\max\limits_{1 \le i \le N}\{\boldsymbol{v}_i \cdot \boldsymbol{r}_j\} - \min\limits_{1 \le i \le N}\{\boldsymbol{v}_i \cdot \boldsymbol{r}_j\}$ gives the values of three axes (a, b, c) of the proxy, where vectors $\boldsymbol{r}_j$ are the natural axes of the set of vertices $\boldsymbol{v}_i$ ($j \in [1, 3]$) and their corresponding eigenvalues $\lambda_1 \ge \lambda_2 \ge \lambda_3$). Considering that a model might not be ellipsoid, the model's proxy is a mixture of the body and its surrounding fluid with average density $\bar{\rho}$ is equal to $(\rho_r V_r + \rho_f V_f)/V$, where $V = (V_r + V_f)$ are the volumes of the body and the fluid in the proxy, and ρ_r is body density.

Dynamical model. The motion of an immersed rigid body is represented in Euclidean group SE(3) with $(\mathbf{R}(t), \boldsymbol{x}(t))$, where $\mathbf{R}(t)$ and $\boldsymbol{x}(t)$ are the orientation matrix and position of the body's center of mass (COM) in the world frame, respectively. The translational velocity $\boldsymbol{u}$ and angular velocity $\boldsymbol{\omega}$ are defined in a body-fixed frame. The kinematic equations of the immersed rigid body are given as:

$$\begin{pmatrix} \dot{\mathbf{R}} \\ \dot{\boldsymbol{x}} \end{pmatrix} = \begin{pmatrix} \mathbf{R}\hat{\omega} \\ \mathbf{R}\boldsymbol{u} \end{pmatrix} \tag{6}$$

where $\hat{\omega}$ is the skew matrix of $\boldsymbol{\omega}$. The fluid effect in an inviscid flow is inertial, and the dynamical equations of motion is given as

Kirchhoff equation:

$$\mathbf{M}_a\dot{\boldsymbol{u}} = (\mathbf{M}_a\boldsymbol{u}) \times \boldsymbol{\omega} + \boldsymbol{F}_g \tag{7}$$

$$\mathbf{I}_a\dot{\boldsymbol{\omega}} = (\mathbf{I}_a\boldsymbol{\omega}) \times \boldsymbol{\omega} + (\mathbf{M}\boldsymbol{u}) \times \boldsymbol{u} + \boldsymbol{\Gamma}_g \tag{8}$$

Here, $\mathbf{M}_a = m\mathbf{E}+\mathbf{M}$, $\mathbf{E}$ is a unit tensor, and $\mathbf{I}_a = \mathbf{I}+\mathbf{J}$, where $\mathbf{I}$ is the moment of inertia of the body[1] . $\mathbf{M}$ and $\mathbf{J}$ are obtained from the analytic solution (see Appendix A). $\boldsymbol{F}_g$ is the buoyancy-corrected gravity $\boldsymbol{F}_g = (\bar{\rho} - \rho_f)V\mathbf{R}^T\boldsymbol{g}$ and its torque $\boldsymbol{\Gamma}_g = \rho_f V\boldsymbol{r} \times \mathbf{R}^T\boldsymbol{g}$. where $\boldsymbol{g}$ is gravitational acceleration, $\boldsymbol{r}$ represents the vector from the position of the COM to the center of buoyancy.

4.2 Viscous Effect

The quasi-steady force assumptions based on the Kutta-Joukowski lift theorem have the limitation to account for the viscous effect in unsteady dynamics. From recent numerical analyses [Veldhuis et al. 2009] and experimental observations [Varshney et al. 2013], the viscous effect force has the following three components: drag $\boldsymbol{F}_D$, rotational lift $\boldsymbol{F}_{L1}$ and translational lift $\boldsymbol{F}_{L2}$ forces (Figure 2). These force coefficients are unknown and changed instantaneously with the body states. These instantaneous and complicated coefficients play a significant role in deciding the sensitive motions of the immersed bodies, which are related to the body geometry, the angle of attack of the body α, and the Reynolds number of flow Re [Umberto 2005; Zastawny et al. 2012]. First, we consider the explicit relationships between the drag and lift coefficients and angle of attack. According to experimental observations in previous work [Varshney et al. 2013], a generalized parametric force model is given as follows:

$$\begin{pmatrix} \boldsymbol{F}_D \\ \boldsymbol{F}_{L1} \\ \boldsymbol{F}_{L2} \end{pmatrix} = \frac{1}{2}\rho_f\|\boldsymbol{u}\|^2 A \begin{pmatrix} -C_D \sin^2\alpha\boldsymbol{e}_1 \\ C_{L1} \sin(2\alpha)\boldsymbol{e}_2 \\ C_{L2} \cos(2\alpha)\boldsymbol{e}_3 \end{pmatrix} \tag{9}$$

Here, (C_D, C_{L1}, C_{L2}) are given as the control parameters of the force coefficients. α is the angle of attack calculated by $\alpha = \tan^{-1}(\|\boldsymbol{u}_n\|/\|\boldsymbol{u}_t\|)$, where $\boldsymbol{u}_n$ and $\boldsymbol{u}_t$ are the normal and tangent velocity components. A is the frontal area of the body with an approximation of $ab\pi$. $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ is defined by the directions of linear and angular velocities $\boldsymbol{u}$ and $\boldsymbol{\omega}$.

$$(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3) = (\boldsymbol{u}, \boldsymbol{\omega} \times \boldsymbol{u}, (\boldsymbol{\omega} \times \boldsymbol{u}) \times \boldsymbol{u}) \tag{10}$$
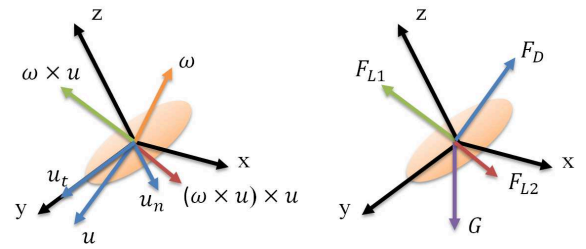

Figure 2: *Illustration of the drag, rotational lift and translational lift forces in a body frame corresponding to translational velocity (blue) and angular velocity (yellow) and their cross products (left).*

Then, the torque induced by the vortex-flow force is given as follows:

$$\boldsymbol{\Gamma}_M = \boldsymbol{p} \times (\boldsymbol{F}_D + \boldsymbol{F}_{L1} + \boldsymbol{F}_{L2}) \tag{11}$$

---

[1]$\mathbf{I} = diag(\frac{1}{5}m(b^2+c^2), \frac{1}{5}m(a^2+c^2), \frac{1}{5}m(a^2+b^2))$ for an ellipsoid structure, .

Algorithm 1 Rigid body simulator using geometric integrator.

```
 1: // pre-computations
 2: calculate axes (a, b, c) of proxy geometry
 3: calculate analytic added tensors M and J (see Appendix A)
 4:
 5: function SIMULATOR(S₀, C)
 6:     for tⱼ ∈ [t₀, tₙ] do
 7:         f_ext = (F_g, Γ_g)
 8:         if nargin > 1 then
 9:             calculate force directions ← Eq.(10)
10:             calculate F_D, F_{L1}, F_{L2} and Γ_M(C_j) ← Eq.(9)
11:             f_ext = f_ext + (F_D + F_{L1} + F_{L2}, Γ_M)
12:         end if
13:         S_j = integrator(S_{j-1}, f_ext) ← Eqs.(7)(8)
14:     end for
15: end function
```

where the vector p is from the COM to the center of pressure of the proxy. Under the Kutta condition, p can be one-quarter of the major axis for an ideal flow. In this work, we take $\|p\| = (1 - \sin^3 \alpha) a/4$ to account for the viscous effect.

The dynamical model in this work is implemented as outlined in Algorithm 1. Because the proxy geometry approximation and analytic added tensors are only depend on body's geometry, these calculations are implemented in pre-computation steps. The input argument $\mathbf{S}$ denotes the body state $(\mathbf{R}, x, u, \omega)$ and C for the control parameters (C_D, C_{L1}, C_{L2}). *nargin* represents the number of input arguments as MATLAB function. Our rigid body simulator utilizes a geometric Lie group integrator [Kobilarov et al. 2009], which enables stable numerical results for various time steps.

5 Motion Patterns

In this work, the simulations of the immersed rigid-bodies are guided by motion patterns as $S(t, Q, C)$, where t denotes the time series, Q is motion patterns which we observed in numerical simulations and physics experiments, and $C \in \mathbb{C}$ denotes the simulation control parameters. We consider the motion patterns of an individual body with various control parameters and equal initial conditions, including release angle, velocities (set 0 from rest), aspect ratio, and body density.

5.1 Motion Classification

From the numerical simulations in the parameter space $\mathbb{C}$, we observed four typical motion patterns in 3D space (Figure 3): (1) steady descent (**SD**), a body falls vertically; (2) zigzag (**ZZ**), a fluttering motion by which the body turns left and right; (3) autorotation (**AR**), a tumbling motion such that the body falls broadside to one side; (4) autorotation and zigzag (**AZ**), a chaotic motion that transitions between AR and ZZ motions. These patterns agree with previous 2D and quasi-2D physics experiments ([Tanabe and Kaneko 1994; Field et al. 1997; Andersen et al. 2005]), and numerical fluid-structure numerical experiments [Auguste et al. 2013].

With regard to the effects from the surrounding flow, note that the inertial effect of the added mass tensors causes vertical oscillations in the horizontal plane, as shown in Figure 3(c). The amplitude of the oscillation increases with inertial effect, e.g., the motion of a coin in water has larger ZZ motion amplitude [Zhong et al. 2011]. The viscous effect of drag and lift forces causes the immersed rigid body to move far away from the vertical direction to an autorotation motion (Figure 3(d)). There is a trade-off between these two effects; the body exhibits a chaotic mixture of AR and ZZ. As shown at the
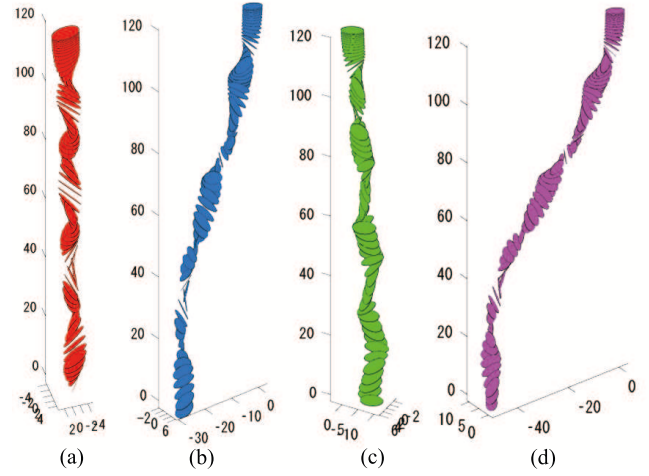
Figure 3: *Four observed motion patterns for a card falling from rest (release angle = $10°$, $a = 3.5cm$, $b = 2cm$). From left to right: (a) SD ($c = (1.81, 0.31, 0.31)$); (b) AZ ($c = (0.61, 1.81, 1.21)$); (c) ZZ ($c = (0.31, 0.91, 0.61)$); (d) AR ($c = (0.61, 2.71, 0.91)$). Spatial coordinates are given in cm.*

end of the pattern in Figure 3(c), AR becomes ZZ motion.

To classify the motion patterns of the proposed dynamical model from the numerical experiments, the trajectories are projected on the horizontal XY plane. Although the body's orientation is absent in this process, motion patterns can be discriminated correctly [Zhong et al. 2011]. Based on the scattered 2D points, principal components analysis is utilized to obtain the characteristic features of the motions. Here, the eigenvalues and eigenvectors of the data are represented as the axes of an eclipse. The eclipse's center is calculated by the mean value of the data. We then adopt a pair value (d, e) to label each motion, where $d \geq 0$ is the distance from center of the eclipse to the original release point, and $e \in (0, 1)$ is the eccentricity of the eclipse, as shown in Figure 4.
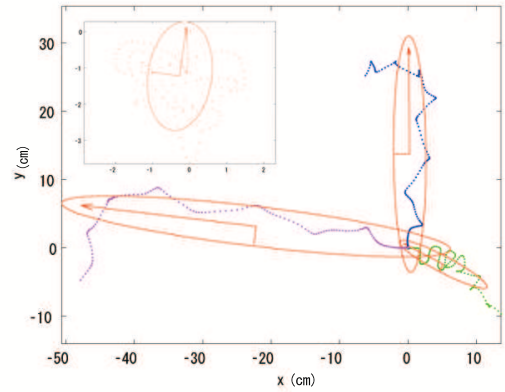
Figure 4: *Distinction among four motion patterns. SD (red points, top-left subfigure; $d = 1.25, e = 0.72$); ZZ (green points; $d = 5.69, e = 0.79$); AZ (blue points; $d = 13.72, e = 0.93$); and AR (magenta points; $d = 22.43, e = 0.92$). These projected points correspond to the trajectories in Figure 3 with the same colors. For better appearance, the AZ and ZZ points are rotated by $\pi/2$ and π, respectively.*

In the current numerical experiments, four motion patterns have been observed; therefore, the values (d, e) of all sampling data

136

are clustered by the k-means method with four clusters. As shown in Figure 5, the trajectories of an elliptical paper card and a leaf are clustered into motion patterns SD, ZZ, AZ, and AR in different parameter space ($\mathbb{C} = (0, \pi) \times (0, \pi) \times (0, \pi)$, Figure 5(a); $(0, 2\pi) \times (0, 2\pi) \times (0, 2\pi)$, Figure 5(b)). The classification results are similar in both cases based on (d, e); therefore, the parameter space $(0, \pi) \times (0, \pi) \times (0, \pi)$ is sufficient for this work. When d is small, the trajectory is close to the original release location where the motion is SD. When d increases and e decreases, ZZ and AZ motions are evident. When d further increases and $e \rightarrow 1$, the motion becomes planar as AR motion. There are more samples in the ZZ zone than other zones, which is consistent with previously reported results [Field et al. 1997].
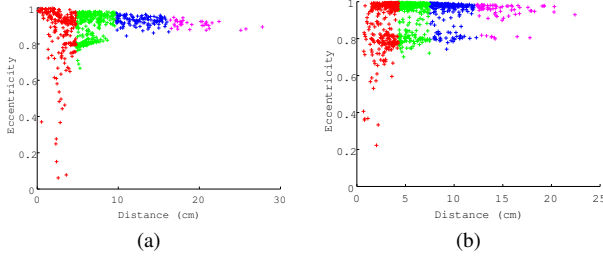

Figure 5: *Clustering results of falling card and falling leaf motions(SD, red; ZZ, green; AZ blue; AR, magenta). Card simulations from 729 samples (a), and leaf simulation from 1331 samples (b) in parameter space.*

5.2 Subspaces Construction

According to the clustering results of the numerical experiments with unified sampling in parameter space, the whole space can be divided into different parameter subspaces corresponding to each pattern. The process is completed by indexing data (d, e) with $\mathbb{C}$.
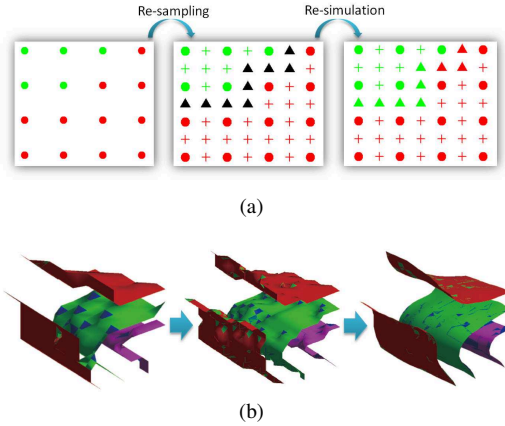

Figure 6: *Low-high-resolution sampling process. (a) Two dimensions; red and green denote different patterns. (+) denotes new sampled data, black triangles denote data on the boundary to be resimulated; (b) 3D subspaces boundaries after resampling.*

To reduce computational cost of numerical experiments in parameter space, a resampling process is required to obtain smooth boundaries among the subspaces. In this work, a simple sampling method (Figure 6) is proposed. First, low-resolution sampling grids are subdivided into high-resolution grids, and the pattern of the sampling

points is determined by nearest neighbors (six neighbors in three dimensions). If neighbors are in the same pattern, the point takes this pattern value. For simplicity, this process does not consider points without pattern information. If the patterns differ, the point should be resimulated to obtain pattern information, i.e., boundary data (black triangles, Figure 6). Finally, the parameter subspaces are constructed on the basis of the whole parameter space $\mathbb{C}$. The proposed method reduces computational complexity of the sampling process from $O(N^3)$ to $O(N^2)$. As shown in Figure 7, the parameter space is divided into five regions. The SD pattern occupies two regions; the other three patterns each occupy only one region. Note that a relatively smooth boundary can be achieved by resampling iteratively (Figure 6(b)); however, we found that for subspace sampling, only a few steps are sufficient for parameter estimation.
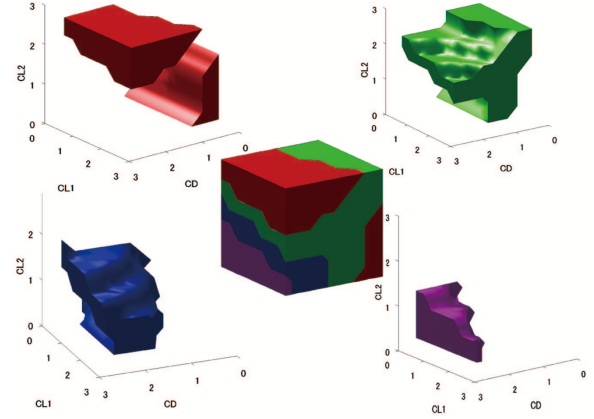

Figure 7: *Parameter subspaces of different falling card motion patterns (SD, red; ZZ, green; AZ, blue; AR, magenta).*

6 Parameter Estimation

For an immersed rigid body moving in real flow environments, there is a significant effect that is unaccounted for by the inertial or viscous effects, i.e., the turbulent effect due to coupling of the body with generated vortices [Andersen et al. 2005]. This effect is particularly apparent at the turning points of the motion trajectory. In this work, the parameter estimation of instantaneous control parameters of force coefficients is proposed based on a curvature-based motion planning method using a motion graph with turbulent kinetic energy (TKE) optimization.

6.1 Turbulent Kinetic Energy

We model TKE using a stochastic approach based on synthetic turbulence to approximate the energy transfer between the body and the surrounding flow. This approach takes the turbulent effects due to vorticity into consideration. The proposed method assumes that objects are suspended particles in a fluid domain [Xie and Miyata 2013; Selino and Jones 2013].

For statistically isotropic turbulence, velocity increments of the body can be represented by the following first-order stochastic differential equation.

$$d\boldsymbol{u}^*(t) = \alpha\boldsymbol{u}^*(t)dt + \sqrt{\beta}d\boldsymbol{W}(t) \qquad (12)$$

This equation is also known as the Langevin model [Pope 2011], where $\boldsymbol{u}^*(t)$ is the intermediate velocity of the immersed rigid body due to flow viscosity and vorticity. Here, α and β are the drift and

137

diffusion functions, which are the functions of the turbulent kinetic energy k, and its dissipation rate ε of turbulent flow from turbulent model computation (see appendix C).

$$\alpha = -(0.5 + 0.75 C_k)\lambda, \qquad \beta = C_k \varepsilon \qquad (13)$$

where $\lambda = \varepsilon/k$ is turbulent frequency. $C_k = 2.1$ is a Kolmogorov coefficient. $\boldsymbol{W}(t)$ denotes Wiener process implemented as white noise in this work. In contrast to adding noise to body's velocity, the TKE method is verified more realistic [Selino and Jones 2013].

6.2 Motion Graph

Motion transition. Hundreds of experiments of falling immersed rigid bodies were executed to identify motion patterns. A high-speed camera was used to record the motion. The experiments involved piece of different shaped pieces of papers and leaves moving in air, and coins and plastic materials moving in water. The experiment results indicate that the motions transits in the sequence of SD→AR→AZ→ZZ. These observations are consistent with the thousands of experimental results reported in previous work [Razavi 2010; Xie and Miyata 2014]. The analysis of our experimental results is based on a previously reported phase diagram [Field et al. 1997]. When the object falls in a real flow due to gravity, the instantaneous Reynolds number increases because of the increasing body velocity. Then, the surrounding flow reveals unsteady dynamics to generate turbulence, and the motion becomes sensitive to the flow. In that sense, the motion cannot transfer from ZZ to SD.
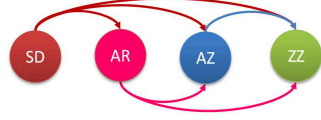
Graph structure. We model the unsteady dynamics of the immersed rigid bodies using a motion graph [Kovar et al. 2002], which is a finite directed graph of motion patterns: $G = (V, E)$. A node $i \in V$ in the graph corresponds to a motion pattern $Q_i = S(t, Q_{i-1}, \boldsymbol{C}_i)$ with a prior distribution function of the predefined four motion patterns. An arc $(i, j) \in E$ represents a transition from node i to j with a transition probability $\mathbf{P}_{ij} = P(Q_j | Q_i)$ with the given pattern Q_i (see Appendix B).

6.3 Motion Planning

The motion of an immersed rigid body becomes most sensitive when it reaches the turning point of the motion trajectory [Andersen et al. 2005]. In this sense, the motion transitions occur at the maximum curvature points (Figure 8). Let an immersed rigid body move in a time interval $[t_0, t_n]$ with initial motion patterns Q_0. The motion patterns $Q(t)$ is defined by n functions:

$$Q(t) = \begin{cases} S(t, Q_0, \boldsymbol{C}_1) & t \in [t_0, t_1] \\ \cdots \\ S(t, Q_{n-1}, C_n) & t \in [t_{n-1}, t_n] \end{cases} \qquad (14)$$

To guarantee the continuity of motion transitions, $\boldsymbol{Q}_i$ is set to be $Q_i = S(t, Q_{i-1}, \boldsymbol{C}_i), i \in (0, n)$. In this work, the key issue for motion planning is the parameter estimation of control parameters $\{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_n\}$ at time sequences $\{t_0, t_1, \ldots, t_{n-1}\}$. The estimation should exhibit the features as an implicit function of the force coefficients related to the Reynolds number. We propose an efficient curvature-based motion-planning solver to realize the motion graph and turbulent energy transition of immersed rigid body dynamics.

Algorithm 2 Parameter estimation by curvature-based motion planning algorithm.

Require: transition matrix $\mathbf{P}$, motion pattern data $\mathbb{C}$
1: // pre-computations
2:     velocity field of mean flow by semi-Lagrangian method
3:     calculate turbulent parameter series (k, ε) (see Appendix C)
4: // initialization of motion pattern
5: $Q_0 = randi(4)$
6: **for** current time step $t_j \in [t_0, t_n]$ **do**
7:     **if** previous curvature $\kappa_{j-1} = \max \kappa_1, \cdots, \kappa_{j-2}$ **then**
8:         Q_j s.t. $P(Q_j | Q_{j-1}) \in \mathbf{P}$
9:     **else**
10:         $Q_j = Q_{j-1}$
11:     **end if**
12:     $S_j^* = $ SIMULATOR(S_{j-1}) (See Algorithm 1)
13:     calculate approximated TKE $\boldsymbol{E}_2 \leftarrow$ Eqs.(12)(17)
14:     **for all** $\boldsymbol{c} \in \mathbb{C}(Q_j)$ **do**
15:         minimize energy consumption $\leftarrow$ Eq.(9)(15)(16)
16:         obtain optimal control parameters $\boldsymbol{C}_j$
17:     **end for**
18:     $S_j = $ SIMULATOR$(S_{j-1}, \boldsymbol{C}_j)$ (See Algorithm 1)
19:     calculate trajectory curvature κ_j
20: **end for**
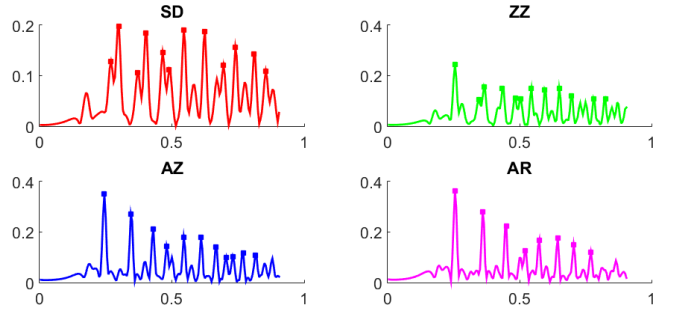21: save control parameters $\{\boldsymbol{C}_1, \ldots, \boldsymbol{C}_n\}$

Figure 8: *Curvature curves corresponding to motion patterns in Figure 3. The square points on the motion curves are the maximum values detected on their curvature curves.*

Curvature-based motion planning. The motion pattern Q_i at each time step $S(t, Q_{i-1}, \boldsymbol{C}_i), i = 1, \ldots, n$ is determined by parameters $\boldsymbol{C}_i$. From the training data step (Section 5), the motion pattern database of parameter subspaces is constructed from motion patterns $\{\mathbb{C}_i, i \in [1, 4] | (SD, AR, AZ, ZZ)\}$ in the whole parameter space $\mathbb{C}$. The subspaces are disjointed and $\bigcup \mathbb{C}_i = \mathbb{C}$. We create a motion graph $G = (V, E), V \in \mathbb{C}$ and determine the patterns of each time step q_i on the basis of a prior distribution function and transition probabilities. As outlined in Algorithm 2, the realization algorithm requires motion transition from Section 6.2 and training motion pattern data from Section 5. At each timestep as referred line 6 to 20 in Algorithm 2, the current motion pattern is determined by the calculated previous curvature. If this curvature arrives the maximum value, i.e., candidate of turning points, the motion pattern should be decided from transition probabilities, otherwise, the previous pattern info is inherited. While the current motion pattern is obtained, the estimation of control parameters is processed by energy optimization step as follows.

Energy optimization. The force coefficients of the dynamical system are instantaneously related to angle of attack and Reynolds number, thus, we assume the control parameters demonstrates time variance at each time step. Therefore, the energy transition among the body and the surrounding flow is calculated by the minimum

138

time derivatives of the drag and lift forces with the energy consumption of body $\mathbf{E}_1$ and the approximated TKE of flow $\mathbf{E}_2$. The following object function of energy optimization is derived from Equations (9) and (12).

$$\min_{\boldsymbol{c}} \quad \sum_{i=1}^n \|d\mathbf{E}_1(t_i) - d\mathbf{E}_2(t_i)\|^2 \tag{15}$$

$$\mathbf{E}_1(t) = \quad (\boldsymbol{F}_D + \boldsymbol{F}_{L1} + \boldsymbol{F}_{L2})(\boldsymbol{u}^*, \boldsymbol{\omega}^*, \boldsymbol{c}) \tag{16}$$

$$\mathbf{E}_2(t) = \quad \sqrt{C_k \varepsilon(t)} \boldsymbol{\xi} - (0.5 + 0.75 C_k)\lambda(t)\boldsymbol{u}^* \tag{17}$$

Here, $\|\cdot\|$ denotes the vector norm, $\boldsymbol{u}^*$ and $\boldsymbol{\omega}^*$ are the intermediate general velocity and translational velocity of the body, which are calculated without viscous forces from Equations (7) and (8) as illustrated in line 12 of Algorithm 2. $\boldsymbol{\xi}$ is a random vector defined by $\mathrm{Norm}(0, 1)$ as the Wiener process using the Box-Muller method. In contrast to path planning techniques for rigid body simulations [Popović et al. 2003], the proposed method is a high-level planning approach to estimate the control parameters of force coefficients in the proposed dynamical model that reflects a statistical description of turbulent flow. After the computation of TKE from turbulent model from Section 6.1 (line 1 to 3 of Algorithm 2), the parameter estimation is done in the specified parameter subspace (line 14 to 17 of Algorithm 2).

Note that the physics experiments indicate that the body exhibits different motions with the same initial conditions. Thanks to the stochastic features of this algorithm, various control parameter sequences can be achieved by repeating the motion planning algorithm at the designated iteration times. The parameter estimation algorithm needs to be computed only once in a pre-computation step, and the rigid body simulator (Algorithm 1) with selected control parameter sequences can achieve real-time simulations easily.

7 Results

We implemented the proposed methods using MATLAB R2012a on a standard PC with a Core i7 CPU (3.20 GHz) and 12 GB RAM. The rigid body simulator utilizes a geometric Lie group integrator, and the average simulation time of Algorithm 1 is approximately 20 ms per time step. The turbulent model utilized a typical 32×8 staggered grid, and a semi-Lagrangian method was used to obtain the mean flow. The training data was obtained by the rigid body simulator with different sampling in parameter spaces (approximately 5.7 h for 729 uniform sampling points and 9.3 h for 1311 samples (Figure 5)). With the proposed resampling process, 4913 sampling points were obtained in 4.0 h of resimulations.

A comparison of simulation results with different flow effects is presented in Figure 9 and the supplemental video. The experimental object is an elliptical and planar paper card (4.0 cm major axis, 2.0 cm minor axis, 0.04 cm thickness, and $30°$ initial release angle). The standard rigid body solver handles the body in a vacuum environment without influence from the surrounding flow such that the body moves straight down. To address the nearly singular problem in the previous method [Weissmann and Pinkall 2012] discussed in Section 4.1, the proposed analytic added tensors capture vertical oscillations in potential flow, as shown in Figures 9(b) and 9(c). In contrast to the drag and lift forces, for which there is a relationship between their coefficients and angle of attack [Wu and Popović 2003; Ju et al. 2013], our dynamical model proposes a generalized model of parametric coefficients and distinguishes the translation lift force with rotational lift. Consequently, the elevation of the body's COM occurs at the turning points of the motion trajectory. This phenomenon has been observed for unsteady dynamics of plates and insect wings [Umberto 2005]. Figure 9(f) shows the simulation results considering inertial, viscous, and turbulent effects from the surrounding flow. Evidently, the motion becomes
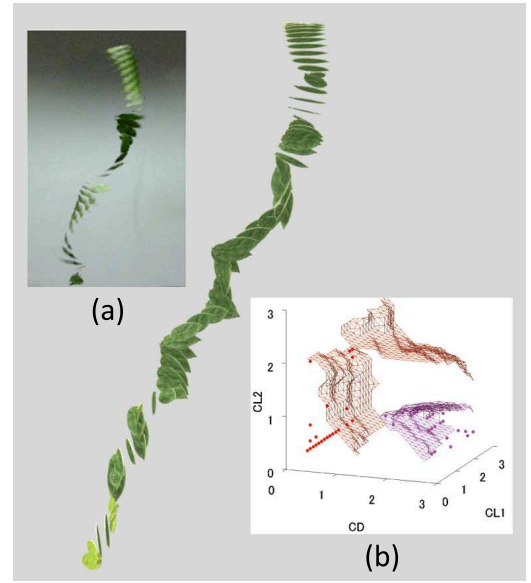

Figure 10: *Comparison of simulation results and ground truth of a leaf falling in air: (a) ground truth from captured video; (b) optimized control parameters in parameter subspaces (SD, red; AR, magenta). All grid surfaces denote boundary surfaces among parameter subspaces.*

sensitive to the flow environment and maintains its inertial property. Compared with the experimental result (Figure 9(g)), the final simulation result (Figure 9(f)) correctly captures the chaotic nature of the card motions

Figure 10 shows a comparison between the simulation results and the captured video of the falling motion of a planar leaf without deformation. The major axis, minor axis, and thickness are 7.3 cm, 4.2 cm, and 0.03 cm, respectively. The captured video suggests that the leaf initially falls straight down and then tumbles from side to side, i.e., $SD \rightarrow AR$. In the parameter estimation process, the motion transition is determined by the proposed motion graph. The proposed method executes energy optimizations in the parameter subspaces of both SD and AR motion patterns as shown in Figure 10(b). The tumbling motions are simulated correctly by the proposed approach, where double tumbling motions are found by considering the turbulent effects, which is clear from the analysis of the periodic oscillations of force coefficients as shown in Figure 11. In contrast to the harmonic oscillations, there are period-two structures corresponding to the vortex-shedding period.

The falling motions of falling leaves in air are shown in Figure 12. All force coefficients from the parameter estimation process have the same initial conditions. Owing to the stochastic features of the proposed methods, the different paths exhibit varied motion transitions among the motion patterns. Because of the influences of turbulence, more complex motions are generated as compared with the basic motion patterns shown in Figure 3. In contrast to the ground truth of captured motions from high-speed camera, the tumbling, fluttering motions and the motion transitions among motion patterns are realistically simulated.

8 Discussion

We proposed a pattern-guided framework for simulating immersed rigid body dynamics by taking into considerations of inertial, viscous and turbulent effects from the surrounding flow. Four motion
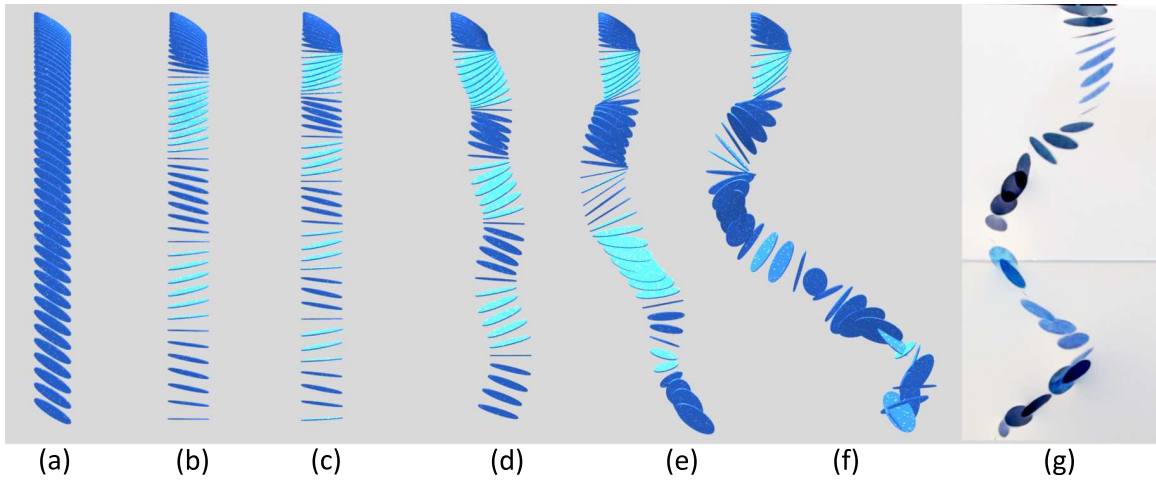
Figure 9: *Flow effects on a falling card in air: (a) no flow effect; (b) inertial effect (previous method); (c) inertial effect; (d) inertial effect and simplified aerodynamics; (e) inertial and viscous effects; (f) the final simulation result with inertial, viscous, and turbulent effects; and (g) experimental result.*
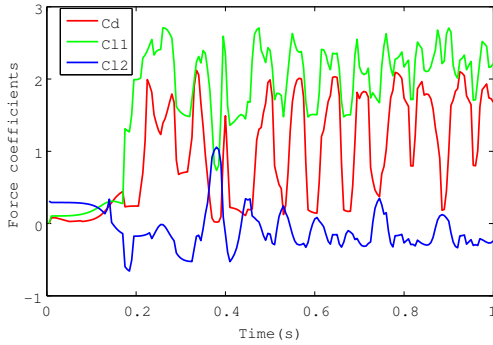

Figure 11: *Force coefficients C_d, C_{l1} and C_{l2} of drag, rotational lift, and translational lift forces, respectively.*

Figure 12: *Comparison between different simulation results of leaves falling from a tree in air under the same initial conditions and the ground truth (top-left subfigure, source: YouTube).*

patterns of the motions are identified from numerical experiments. The relationships among these motion patterns and the control parameter subspaces are clarified. For the parameter estimation of the control parameters in simulations, we proposed a curvature-based motion-planning method based on a motion graph to represent the effect due to vorticity. The proposed framework offers the effective force coefficients and parameter subspaces for realistic and real-time simulations in graphical applications.

Limitations and future work. In the proposed parameter estimation, we considered the influences of control parameters of force coefficients with equal initial conditions. Other relevant parameters may include aspect ratio, mass distribution, initial orientation, initial velocities, and external forces. The proposed framework can be extended to simulations under these conditions with specified motion patterns. The artist control design would benefit from this framework in two way: low-level and high-level controls. The low-level controls with input motion sketches can be embedded into Algorithm 2 by modified the object function as Equation (15) in motion planning, and the high-level controls with designated motion patterns are designed through the fixed pattern initialization and transition matrix of motion patterns.

The external forces, such as wind forces, and collision detection among bodies could be embedded in the proposed dynamical model

as future work. The proposed framework can be embedded into graphical development tools and game engines to enhance simulation realness. Furthermore, the development of the immersed deformable objects and characters with the apparent influence of vorticity, is also a promising future undertaking.

Acknowledgements

References

ANDERSEN, A., PESAVENTO, U., AND WANG, Z. J. 2005. Unsteady aerodynamics of fluttering and tumbling plates. *Journal*

140

of Fluid Mechanics 541 (Oct.), 65–90.

AUGUSTE, F., MAGNAUDET, J., AND FABRE, D. 2013. Falling styles of disks. *Journal of Fluid Mechanics 719* (2), 388–405.

BARAFF, D. 1993. Non-penetrating rigid body simulation. In *EUROGRAPHICS State of the Art Reports*, Eurographics Association.

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid fluid coupling. *ACM Trans. Graph. 26*, 3 (July).

BECKER, M., TESSENDORF, H., AND TESCHNER, M. 2009. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (May), 493–503.

BIESHEUVEL, A., AND HAGMEIJER, R. 2006. On the force on a body moving in a fluid. *Fluid Dynamics Research 38*, 10, 716 – 742.

FIELD, S. B., KLAUS, M., MOORE, M. G., AND NORI, F. 1997. Chaotic dynamics of falling disks. *Nature 388* (July), 252–254.

HE, X., LIU, N., WANG, G., ZHANG, F., LI, S., SHAO, S., AND WANG, H. 2012. Staggered meshless solid-fluid coupling. *ACM Trans. Graph. 31*, 6 (Nov.), 149:1–149:12.

HESS, J., AND SMITH, A. 1967. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences 8*, 0, 1 – 138.

JOHNSTON, B. M., JOHNSTON, P. R., AND ELLIOTT, D. 2013. A new method for the numerical evaluation of nearly singular integrals on triangular elements in the 3d boundary element method. *J. Comput. Appl. Math. 245* (June), 148–161.

JU, E., WON, J., LEE, J., CHOI, B., NOH, J., AND CHOI, M. G. 2013. Data-driven control of flapping flight. *ACM Trans. Graph. 32*, 5 (Oct.), 151:1–151:12.

KOBILAROV, M., CRANE, K., AND DESBRUN, M. 2009. Lie group integrators for animation and control of vehicles. *ACM Trans. Graph. 28*, 2 (May), 16:1–16:14.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Trans. Graph. 21* (July), 473–482.

MIHALEF, V., METAXAS, D., AND SUSSMAN, M. 2009. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Computer Graphics Forum 28*, 2, 229–238.

MOUGIN, G., AND MAGNAUDET, J. 2002. The generalized kirchhoff equations and their application to the interaction between a rigid body and an arbitrary time-dependent viscous flow. *International Journal of Multiphase Flow 28*, 11, 1837 – 1851.

OKAMOTO, T., FUJISAWA, M., AND MIURA, K. T. 2009. An interactive simulation system for flying japanese kites. In *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games*, ACM, New York, NY, USA, Sandbox '09, 47–53.

OZGEN, O., KALLMANN, M., RAMIREZ, L. E., AND COIMBRA, C. F. 2010. Underwater cloth simulation with fractional derivatives. *ACM Trans. Graph. 29*, 3 (July), 23:1–23:9.

PFAFF, T., THUEREY, N., COHEN, J., TARIQ, S., AND GROSS, M. 2010. Scalable fluid simulation using anisotropic turbulence particles. *ACM Trans. Graph. 29*, 6 (Dec.), 174:1–174:8.

POPE, S. B. 2011. Simple models of turbulent flows. *Physics of Fluids 23*, 1, 011301.

POPOVIĆ, J., SEITZ, S. M., AND ERDMANN, M. 2003. Motion sketching for control of rigid-body simulations. *ACM Trans. Graph. 22*, 4 (Oct.), 1034–1054.

RAZAVI, P. 2010. On the motion of falling leaves. *ArXiv e-prints* (July).

REISSELL, L.-M., AND PAI, D. K. 2001. Modeling stochastic dynamical systems for interactive simulation. *Comput. Graph. Forum 20*, 3.

ROBINSON-MOSHER, A., SHINAR, T., GRETARSSON, J., SU, J., AND FEDKIW, R. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph. 27*, 3 (Aug.), 46:1–46:9.

SELINO, A., AND JONES, M. D. 2013. Large and small eddies matter: Animating trees in wind using coarse fluid simulation and synthetic turbulence. *Computer Graphics Forum 32*, 1, 75–84.

SHI, L., YU, Y., WOJTAN, C., AND CHENNEY, S. 2005. Controllable motion synthesis in a gaseous medium. *The Visual Computer 21*, 7, 474–487.

TANABE, Y., AND KANEKO, K. 1994. Behavior of a falling paper. *Phys. Rev. Lett. 73* (Sep), 1372–1375.

UMBERTO, P. 2005. *Unsteady aerodynamics of falling plates*. PhD thesis, Cornell University.

UMETANI, N., KOYAMA, Y., SCHMIDT, R., AND IGARASHI, T. 2014. Pteromys: Interactive design and optimization of free-formed free-flight model airplanes. *ACM Trans. Graph. 33*, 4 (July), 65:1–65:10.

VARSHNEY, K., CHANG, S., AND WANG, Z. J. 2013. Unsteady aerodynamic forces and torques on falling parallelograms in coupled tumbling-helical motions. *Phys. Rev. E 87* (May), 053021.

VELDHUIS, C., BIESHEUVEL, A., AND LOHSE, D. 2009. Freely rising light solid spheres. *International Journal of Multiphase Flow 35*, 4, 312 – 322.

WEI, X., ZHAO, Y., FAN, Z., LI, W., QIU, F., YOAKUM-STOVER, S., AND KAUFMAN, A. E. 2004. Lattice-based flow field modeling. *IEEE Transactions on Visualization and Computer Graphics 10* (November), 719–729.

WEISSMANN, S., AND PINKALL, U. 2012. Underwater rigid body dynamics. *ACM Trans. Graph. 31*, 4 (July), 104:1–104:7.

WU, J.-C., AND POPOVIĆ, Z. 2003. Realistic modeling of bird flight animations. *ACM Trans. Graph. 22*, 3 (July), 888–895.

XIE, H., AND MIYATA, K. 2013. Stochastic modeling of immersed rigid-body dynamics. In *SIGGRAPH Asia 2013 Technical Briefs*, ACM, New York, NY, USA, SA '13, 12:1–12:4.

XIE, H., AND MIYATA, K. 2014. Real-time simulation of lightweight rigid bodies. *The Visual Computer 30*, 1, 81–92.

ZASTAWNY, M., MALLOUPPAS, G., ZHAO, F., AND VAN WACHEM, B. 2012. Derivation of drag and lift force and torque coefficients for non-spherical particles in flows. *International Journal of Multiphase Flow 39*, 0, 227 – 239.

ZHENG, C., AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. *ACM Trans. Graph. 29*, 4 (July), 69:1–69:13.

ZHONG, H., CHEN, S., AND LEE, C. 2011. Experimental study of freely falling thin disks: Transition from planar zigzag to spiral. *Physics of Fluids 23*, 1.

A Calculation of analytic added tensors

Analytic solutions of the Laplace equations only exist for a limited number of regular geometries. We obtain the three axes of the approximated ellipsoid a, b, and c. First, the translational kinetic energy of the surrounding flow along the a-axis is given as follows:

$$E_a = \frac{1}{2} m_f \frac{a_0 u^2}{2 - a_0} \tag{18}$$

where $m_f = \frac{4}{3} \rho_f \pi abc$ is the fluid mass occupied by the body. a_0 is defined by the following form of elliptic integrals:

$$a_0 \equiv abc \int_0^\infty \frac{d\lambda}{(a^2 + \lambda)\sigma} \tag{19}$$

where $\sigma = \sqrt{(a^2 + \lambda)(b^2 + \lambda)(c^2 + \lambda)}$. Then, the rotational kinetic energy is given as follows

$$E_a^r = \frac{1}{2} m_f \frac{(b^2 - c^2)^2(c_0 - b_0)\omega^2}{10(b^2 - c^2) - 5(b^2 + c^2)^2(c_0 - b_0)} \tag{20}$$

Here, b_0 and c_0 are defined in the form of a_0. Owing to the symmetrical expressions for the $b-$ and $c-$axes, E_b, E_b^r, E_c, and E_c^r are computed through the same procedure. From the kinetic energy of flow in different directions, added mass tensors M and J are obtained directly, which are explicitly diagonal for an ellipsoid. These general forms of analytic solutions of added tensors are adaptive to different structures such as cylinders, spheroids, spheres and other structures similar to these geometries.

B Transition matrix measurement

Based on our experimental results from labelling motion patterns in each video of the falling experiments, the transition matrix $\mathbf{P}$ of the transition probabilities is satisfied as follows:

$$\{\mathbf{P}_{ij}\} = \{\frac{n_{ij}}{n_i}\} = \begin{pmatrix} 0.154 & 0.385 & 0.038 & 0.423 \\ 0 & 0.383 & 0.086 & 0.531 \\ 0 & 0 & 0.652 & 0.348 \\ 0 & 0 & 0 & 1.0 \end{pmatrix} \tag{21}$$

where n_{ij} and n_i denote the number of patterns transferred from state $\boldsymbol{Q}_i$ to state $\boldsymbol{Q}_j$ and the total number of states transferred from state $\boldsymbol{Q}_i$, respectively. We model the prior distribution function using a normal distribution of motion patterns $\{SD, AR, AZ, ZZ\}$. From the measured $\mathbf{P}$, the transitions among motion patterns are unidirectional, which increase the effectiveness of motion planning.

C Turbulent model

Considering the turbulent flow generated around a moving rigid body, the fluid velocity of a turbulent flow is decomposed into a mean flow and a fluctuating velocity by Reynolds decomposition with k-ε turbulent model. This model is a semi-empirical model based on energy transport equations [Pfaff et al. 2010], including coupled equations for the Lagrangian derivatives of $D_t k$ and $D_t \varepsilon$.

$$D_t k = G - \varepsilon \tag{22}$$
$$D_t \varepsilon = \lambda(C_1 G - C_2 \varepsilon) \tag{23}$$

Here, $\lambda = \varepsilon/k$ is turbulent frequency with turbulent kinetic energy k, and its dissipation rate ε. C_1 and C_2 are empirical constants ($C_1 = 1.44$, $C_2 = 1.92$). $G = 2v_T \sum_{ij} S_{ij}^2$ represents the generation of turbulent kinetic energy with strain tensor S_{ij} and turbulent viscosity $v_T = 0.09k^2/\varepsilon$. In this work, the mean flow simulation is defined on marker-and-cell grids with the mean falling velocity of the body $U_0 = \sqrt{(\rho_s/\rho_f - 1)gb}$ as inflow velocity. Note that implementation of the turbulent model needs to be computed only once because λ and ε do not depend on the body's state.