

Appendix B

Three-Dimensional Contact Dynamics Formulation

The two-dimensional formulation discussed in Chapter 4 may be extended to three dimensions in a number of ways. All appropriate physical constraints of the three-dimensional contact problem can be found in [99] and hence, will not be repeated here. The exception is that the integration of rotational degrees of freedom needs special attention.

Kinematically, the non-trivial difference between the two- and three-dimensional case is a result of rotations of arbitrary-shaped particles. In the following, we outline one possible approach to integrate the rotational degrees of freedom in three dimensions. Without loss of generality, we consider the dynamic problem without contact elasticity. Unless otherwise noted, all quantities pertain to a single particle in the principal body frame.

The discretization of rotational equilibrium equations using the θ -method gives

$$\mathbf{I} \frac{\boldsymbol{\omega} - \boldsymbol{\omega}_0}{\Delta t} + \text{skew}[\boldsymbol{\omega}] \mathbf{I} \boldsymbol{\omega} = \mathbf{m}_0 \quad (\text{B.1})$$

with the update equation for the (body frame) rotational degrees of freedom as (analogous to equation (4.6)b):

$$\boldsymbol{\omega} = \frac{1}{\theta} \left[\frac{\boldsymbol{\alpha} - \boldsymbol{\alpha}_0}{\Delta t} - (1 - \theta) \boldsymbol{\omega}_0 \right] \quad (\text{B.2})$$

and where \mathbf{I} is the time-independent tensor of (principal) inertia in the body frame, \mathbf{m}_0 are the external moments in the body frame, and $\text{skew}[\boldsymbol{\omega}]$ is the skew-symmetric matrix defined by $\text{skew}[\boldsymbol{\omega}] \boldsymbol{\omega} = 0$. We observe that the rotational equilibrium equations are nonlinear due to the presence of the products of angular velocities. This nonlinearity can be accounted

for within the mathematical programming framework proposed in this paper as follows. We first define the (nonlinear) residual equations as

$$\mathbf{r}(\boldsymbol{\omega}) = \mathbf{m}_0 - \mathbf{I} \frac{\boldsymbol{\omega} - \boldsymbol{\omega}_0}{\Delta t} - \text{skew}[\boldsymbol{\omega}] \mathbf{I} \boldsymbol{\omega} \quad (\text{B.3})$$

A truncated Taylor expansion about the k -th iterate $\boldsymbol{\omega}_k$ then gives the following iterative procedure for the (unconstrained) solution of $\boldsymbol{\omega}$

$$\begin{aligned} \mathbf{J}|_{\boldsymbol{\omega}_k} \delta \boldsymbol{\omega} &= \mathbf{r}(\boldsymbol{\omega}_k) \\ \boldsymbol{\omega}^{k+1} &= \boldsymbol{\omega}^k + \delta \boldsymbol{\omega}, \quad \boldsymbol{\omega}^{k=0} = \boldsymbol{\omega}_0 \end{aligned} \quad (\text{B.4})$$

where $\mathbf{J} = -\frac{\partial \mathbf{r}}{\partial \boldsymbol{\omega}}$. To cast this iterative procedure along with all the physical constraints of the problem as a quadratic programming problem, we note from equation (B.2) that the variation of $\boldsymbol{\omega}$ is simply (since $\boldsymbol{\omega}_0$ and $\boldsymbol{\alpha}_0$ are fixed)

$$\delta \boldsymbol{\omega} = \frac{\delta \boldsymbol{\alpha}}{\theta \Delta t} \quad (\text{B.5})$$

which allows us to rewrite the iterative procedure in equation (B.4) as

$$\begin{aligned} \bar{\mathbf{J}}|_{\boldsymbol{\omega}_k} \delta \boldsymbol{\alpha} &= \mathbf{r}(\boldsymbol{\omega}_k) \\ \boldsymbol{\alpha}^{k+1} &= \boldsymbol{\alpha}^k + \delta \boldsymbol{\alpha}, \quad \boldsymbol{\alpha}^{k=0} = \boldsymbol{\alpha}_0 \end{aligned} \quad (\text{B.6})$$

where $\bar{\mathbf{J}}$ is the effective tensor of inertia, calculated to be

$$\bar{\mathbf{J}} = \frac{1}{\theta \Delta t} \mathbf{J} = \begin{pmatrix} \frac{I_1}{\theta \Delta t^2} & \frac{-\omega_3(I_2 - I_3)}{\theta \Delta t} & \frac{-\omega_2(I_2 - I_3)}{\theta \Delta t} \\ \frac{-\omega_3(I_3 - I_1)}{\theta \Delta t} & \frac{I_2}{\theta \Delta t^2} & \frac{-\omega_1(I_3 - I_1)}{\theta \Delta t} \\ \frac{-\omega_2(I_1 - I_2)}{\theta \Delta t} & \frac{-\omega_1(I_1 - I_2)}{\theta \Delta t} & \frac{I_3}{\theta \Delta t^2} \end{pmatrix} \quad (\text{B.7})$$

We see that equation (B.6)a has an identical structure to equation (4.10) except now the problem needs to be solved incrementally within each time step. The necessary modification to the formulation then entails defining an incremental problem within each time step and replacing the rotational equilibrium equations of the form given by equation (4.10) with equation (B.6)a.

We note several properties of the effective tensor of inertia given by equation (B.7) in relation to the proposed mathematical programming framework. First, the terms are functions of the body frame angular velocities. Second, the effective tensor of inertia is no longer diagonal. In the force-based problem described by equation (4.15), the inverse of the global moment of inertia matrix is required. However, recognizing that the global moment of inertia is now block diagonal, the inverse of the global moment of inertia is another block diagonal matrix, which is composed of the inverse of 3×3 blocks and these can be calculated efficiently. Finally, the non-symmetry of $\bar{\mathbf{J}}$ poses no problem since the associated quadratic term for a non-symmetric matrix \mathbf{Q} can be written as

$$\delta\boldsymbol{\alpha}^T \mathbf{Q} \delta\boldsymbol{\alpha} = \frac{1}{2} \delta\boldsymbol{\alpha}^T (\mathbf{Q} + \mathbf{Q}^T) \delta\boldsymbol{\alpha} \quad (\text{B.8})$$

which implies that a non-symmetric matrix \mathbf{Q} can be replaced by $\frac{1}{2}(\mathbf{Q} + \mathbf{Q}^T)$.

Computationally, a sequence of quadratic programming problems is solved within a time step for the incremental unknowns $(\delta\mathbf{x}, \delta\boldsymbol{\alpha})$ until a prescribed tolerance is reached. We note that we can formulate the translational problem incrementally by simply replacing $\Delta\mathbf{x}$ with $\delta\mathbf{x}$; this is possible because the translational equilibrium equations already have a linear structure. Finally, we need to update the orientation of each particle. Here, a singularity-free quaternion approach [55] may be taken. We denote the quaternion vector representing the orientation of the particle as $\mathbf{z} = (z_i), i = 1, \dots, 4$. The required orientation matrix of each particle can be calculated as a function of quaternion values as

$$\mathbf{A}(\mathbf{z}) = \begin{pmatrix} -z_1^2 + z_2^2 - z_3^2 + z_4^2 & -2(z_1 z_2 - z_3 z_4) & 2(z_2 z_3 + z_1 z_4) \\ -2(z_1 z_2 + z_3 z_4) & z_1^2 - z_2^2 - z_3^2 + z_4^2 & -2(z_1 z_3 - z_2 z_4) \\ 2(z_2 z_3 - z_1 z_4) & -2(z_1 z_3 + z_2 z_4) & -z_1^2 - z_2^2 + z_3^2 + z_4^2 \end{pmatrix} \quad (\text{B.9})$$

In turn, the evolution of the quaternions can be expressed as a singularity-free set of equations as

$$\dot{\mathbf{z}} = \boldsymbol{\Psi} \mathbf{z}, \quad \mathbf{z}^T \mathbf{z} = 1 \quad (\text{B.10})$$

where

$$\mathbf{\Psi} = \frac{1}{2} \begin{pmatrix} 0 & \omega_3 & -\omega_1 & -\omega_2 \\ -\omega_3 & 0 & -\omega_2 & \omega_1 \\ \omega_1 & \omega_2 & 0 & \omega_3 \\ \omega_2 & -\omega_1 & -\omega_3 & 0 \end{pmatrix} \quad (\text{B.11})$$

The general solution is given by

$$\mathbf{z} = \exp \left(\int_{t_0}^t \mathbf{\Psi}[\omega(s)] ds \right) \mathbf{z}_0 \quad (\text{B.12})$$

Here, several choices of updating \mathbf{z} exist. One possible choice is the generalized trapezoidal rule of the form

$$\int_{t_0}^t \mathbf{\Psi}[\omega(s)] ds \approx \Delta t [\theta \mathbf{\Psi} + (1 - \theta) \mathbf{\Psi}_0] =: \Delta t \mathbf{\Psi}_\theta \quad (\text{B.13})$$

where we have used the same θ parameter as in equation (B.2). In this case, one can update \mathbf{z} as

$$\mathbf{z} = \exp(\Delta t \mathbf{\Psi}_\theta) \mathbf{z}_0 \quad (\text{B.14})$$

Efficient algorithms for the computation of matrix exponentials can be found in [140].