ME314 Final Project

Wanlin Yang

1 Introduction

The dynamic system is inspired by Angry Bird, in which birds are colliding with a bunch of bricks. This simulation is a simplified "Angry Bird" that a flying ball is hitting a rotating cube in the air, shown as Figure 1.

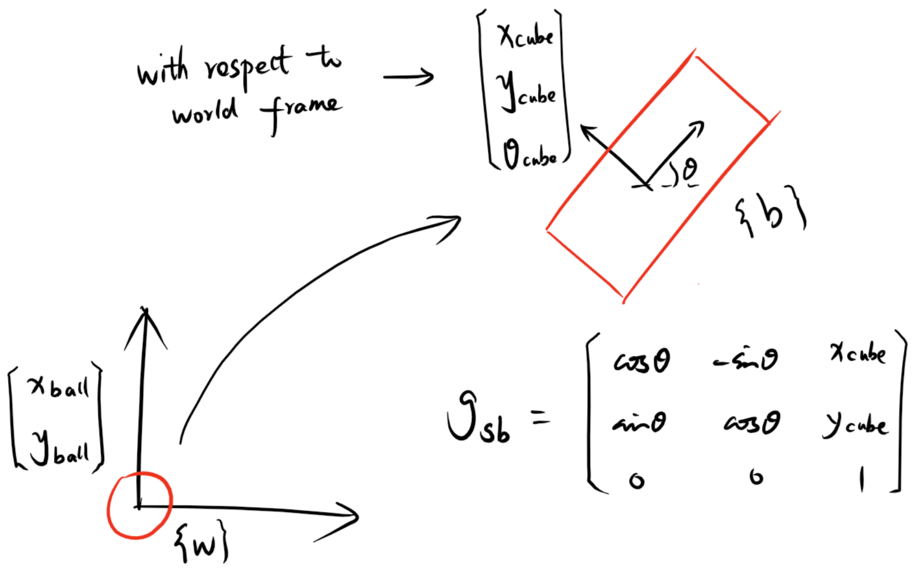

Figure 1: Coordinates of the dynamic system

The dynamic model has 5 degrees of freedom, 2 of the flying ball $(x_b(t), y_b(t))^T$, and 3 of the falling rotating cube $(x_c(t), y_c(t), \theta_c(t))^T$. The ball is ejected by a slingshot following Hooke's law. The ball and the cube will collide in the air, then departing from each other, finally falling on the ground and bouncing up separately.

2 Equations

The system has 5 dynamic parameters varying with time:

$$q = [x_b, y_b, x_c, y_c, \theta_c]^T$$

The Lagrangian:

$$\begin{aligned} KE &= \frac{1}{2}m_b(\dot{x_b}^2 + \dot{y_b}^2) + \frac{1}{2}(V^b)^T \begin{bmatrix} m_c I_{3\times3} & 0 \\ 0 & \mathcal{I} \end{bmatrix} V^b \\ PE &= m_b \cdot g \cdot y_b + m_c \cdot g \cdot y_c \\ L &= KE - PE \end{aligned}$$

where m_b, m_c are mass of ball and cube respectively. $V^b = [v, \omega]^T$, ($V^b \in \mathbb{R}$) is the body velocity of cube.

Force produced by the slingshot follows Hooke's law:

$$F = k(x_b - x_d) + k(y_b - y_d)$$

where (x_d, y_d) is the eject position. The ball is accelerated by the slingshot from $(0, 0)$ with zero velocity.

3 Implementation

Constrain during colliding process: Position of the flying ball will be transferred into body frame $\{b\}$ by g_{sb} matrix:

$$g_{sb} = \begin{bmatrix} \cos\theta & -\sin\theta & x_c \\ \sin\theta & \cos\theta & y_c \\ 0 & 0 & 1 \end{bmatrix}$$

Then the program will judge whether the ball has contacted the edge of the cube. Once colliding, these two objects will apply impact equation with respect to $\{b\}$ frame, and departure velocities are calculated out. These velocities will be transferred back into $\{w\}$ frame after collision.

After colliding: The two objects are assumed to never meet again. For convenience, the coordinates after collision are expressed as $q_b = (x_b, y_b)^T$, $q_c = (x_c, y_c, \theta_c)^T$, and Euler-Lagrange equations of them are calculated separately.

Bouncing of ball and cube from ground: In that the Euler-Lagrange and Hamiltonian equations of the ball and cube are not related after colliding, the impact equations of them are calculated independently. Impact of the ball is pretty simple, while the constraint of the cube is more sophisticated. Constraints of each vertex are derived separately, once one of them equals zero, it means that one vertex of the cube has contacted with the ground. Then apply the constraint of this vertex to impact equations, and the cube will bounce up.

4 Result

The code works efficiently, with no more than 5 seconds before animation. The animation shows the reasonable processes: ejection, colliding, departure, bouncing up independently. And after each impact, the states of the objects, including the linear velocities and angular velocity, are changed obviously.