Homework 2.3

Alden Wu (A17361768)
CSE 291 (C00) SP24

May 9, 2024

2.3.1 Equation of Motion

Consider a triple pendulum where the first rod is attached to the origin, the second is attached to the first, etc. Let g be the acceleration due to gravity, ℓ be the length of each rod, and m be the mass of each rod. The system has three degrees of freedom: θ_1, θ_2, and θ_3, where θ_n is the angle of the n-th rod from the downwards vector (e.g. $-e_2 \in \mathbf{R}^2$ in world coordinates). Let $\theta := (\theta_1, \theta_2, \theta_3)^\top$, and $\omega := \dot{\theta}$. We have the Lagrangian $L(t, \theta, \omega) = K(t, \theta, \omega) - U(t, \theta, \omega)$.

Let K_n be the kinetic energy of the n-th rod, so $K = K_1 + K_2 + K_3$. K_1 can be described using only ω_1, since the first rod is spinning about one end (fixed). The moment of inertia of a rod spinning about one end is $I_1 = m\ell^2/3$. Since this end is fixed to the origin, it has no linear kinetic energy.

$$K_1(t, \theta, \omega) = \frac{1}{2} I \omega_1^2 = m\ell^2 \cdot \frac{1}{6}\omega_1^2 \tag{1}$$

For K_2 and K_3, this strategy no longer works. Instead, let x_n be the center (of mass) of the n-th rod, and $v_n := \|\dot{x}_n\|$. The moment of inertia of a rod spinning about its center is $I = m\ell^2/12$. So $K_n = mv_n^2/2 + I\omega_n^2/2$. With some basic trigonometry/geometry, observe

$$x_2 = \begin{pmatrix} \ell \sin\theta_1 + \frac{1}{2}\ell \sin\theta_2 \\ -\ell\cos\theta_1 - \frac{1}{2}\ell\cos\theta_2 \end{pmatrix} \qquad x_3 = \begin{pmatrix} \ell \sin\theta_1 + \ell\sin\theta_2 + \frac{1}{2}\ell \sin\theta_3 \\ -\ell\cos\theta_1 - \ell\cos\theta_2 - \frac{1}{2}\ell\cos\theta_3 \end{pmatrix}$$

$$\dot{x}_2 = \ell\begin{pmatrix} \cos\theta_1\omega_1 + \frac{1}{2}\cos\theta_2\omega_2 \\ \sin\theta_1\omega_1 + \frac{1}{2}\sin\theta_2\omega_2 \end{pmatrix} \qquad \dot{x}_3 = \ell\begin{pmatrix} \cos\theta_1\omega_1 + \cos\theta_2\omega_2 + \frac{1}{2}\cos\theta_3\omega_3 \\ \sin\theta_1\omega_1 + \sin\theta_2\omega_2 + \frac{1}{2}\sin\theta_3\omega_3 \end{pmatrix}$$

$$\begin{aligned} v_2^2 &:= \|x_2\|^2 = \ell^2\Big(\omega_1^2 + \frac{1}{4}\omega_2^2 + \cos(\theta_1 - \theta_2)\omega_1\omega_2\Big) \\ v_3^2 &:= \|x_3\|^2 = \ell^2\Big(\omega_1^2 + \omega_2^2 + \frac{1}{4}\omega_3^2 + 2\cos(\theta_1 - \theta_2)\omega_1\omega_2 \\ &\qquad + \cos(\theta_1 - \theta_3)\omega_1\omega_3 + \cos(\theta_2 - \theta_3)\omega_2\omega_3\Big) \end{aligned}$$

So now we have K_2 and K_3:

$$\begin{aligned} K_2(t,\theta,\omega) &= \frac{1}{2}mv_2^2 + \frac{1}{2}I\omega_2^2 \\ &= m\ell^2\left(\frac{1}{2}\omega_1^2 + \frac{1}{6}\omega_2^2 + \frac{1}{2}\cos(\theta_1-\theta_2)\omega_1\omega_2\right) \end{aligned} \tag{2}$$

$$\begin{aligned} K_3(t,\theta,\omega) &= \frac{1}{2}mv_3^2 + \frac{1}{2}I\omega_3^2 \\ &= m\ell^2\bigg(\frac{1}{2}\omega_1^2 + \frac{1}{2}\omega_2^2 + \frac{1}{6}\omega_3^2 + \cos(\theta_1-\theta_2)\omega_1\omega_2 \\ &\qquad + \frac{1}{2}\cos(\theta_1-\theta_3)\omega_1\omega_3 + \frac{1}{2}\cos(\theta_2-\theta_3)\omega_2\omega_3\bigg) \end{aligned} \tag{3}$$

Combining (1), (2), and (3),

$$\begin{aligned} K(t,\theta,\omega) = m\ell^2\bigg(&\frac{7}{6}\omega_1^2 + \frac{2}{3}\omega_2^2 + \frac{1}{6}\omega_3^2 + \frac{3}{2}\cos(\theta_1-\theta_2)\omega_1\omega_2 \\ &+ \frac{1}{2}\cos(\theta_1-\theta_3)\omega_1\omega_3 + \frac{1}{2}\cos(\theta_2-\theta_3)\omega_2\omega_3\bigg) \end{aligned} \tag{4}$$

For U, observe again with some basic trigonometry/geometry that

$$U(t,\theta,\omega) = -mg\ell\left(\frac{5}{2}\cos\theta_1 + \frac{3}{2}\cos\theta_2 + \frac{1}{2}\cos\theta_3\right) \tag{5}$$

Now that we have computed $K(t,\theta,\omega)$ and $U(t,\theta,\omega)$, observe that U is independent of ω (as we would expect it to be). Therefore the Euler-Lagrange equation is

$$\frac{d}{dt}\frac{\partial K}{\partial\omega} = \frac{\partial K}{\partial\theta} - \frac{\partial U}{\partial\theta}$$

To simplify notation, we divide the Euler-Lagrange equation by $m\ell^2$ and transpose:

$$\frac{1}{m\ell^2}\left(\frac{d}{dt}\frac{\partial K}{\partial\omega}\right)^\top = \frac{1}{m\ell^2}\left(\frac{\partial K}{\partial\theta}\right)^\top - \frac{1}{m\ell^2}\left(\frac{\partial U}{\partial\theta}\right)^\top \tag{6}$$

Now for some tedious computations. Differentiating (4) and (5),

$$\frac{1}{m\ell^2}\left(\frac{\partial U}{\partial\theta}\right)^\top = \frac{g}{\ell}\begin{pmatrix} \frac{5}{2}\sin\theta_1 \\ \frac{3}{2}\sin\theta_2 \\ \frac{1}{2}\sin\theta_3 \end{pmatrix} \tag{7}$$

$$\frac{1}{m\ell^2}\left(\frac{\partial K}{\partial\theta}\right)^\top = \begin{pmatrix} -\frac{3}{2}\sin(\theta_1-\theta_2)\omega_1\omega_2 - \frac{1}{2}\sin(\theta_1-\theta_3)\omega_1\omega_3 \\ \frac{3}{2}\sin(\theta_1-\theta_2)\omega_1\omega_2 - \frac{1}{2}\sin(\theta_2-\theta_3)\omega_2\omega_3 \\ \frac{1}{2}\sin(\theta_1-\theta_3)\omega_1\omega_3 + \frac{1}{2}\sin(\theta_2-\theta_3)\omega_2\omega_3 \end{pmatrix} \tag{8}$$

$$\frac{1}{m\ell^2}\left(\frac{d}{dt}\frac{\partial K}{\partial \omega}\right)^\top = \begin{pmatrix} \begin{array}{c} \frac{7}{3}\dot{\omega}_1 - \frac{3}{2}\sin(\theta_1-\theta_2)(\omega_1-\omega_2)\omega_2 + \frac{3}{2}\cos(\theta_1-\theta_2)\dot{\omega}_2 \\ -\frac{1}{2}\sin(\theta_1-\theta_3)(\omega_1-\omega_3)\omega_3 + \frac{1}{2}\cos(\theta_1-\theta_3)\dot{\omega}_3 \\ \hline \frac{4}{3}\dot{\omega}_2 - \frac{3}{2}\sin(\theta_1-\theta_2)(\omega_1-\omega_2)\omega_1 + \frac{3}{2}\cos(\theta_1-\theta_2)\dot{\omega}_1 \\ -\frac{1}{2}\sin(\theta_2-\theta_3)(\omega_2-\omega_3)\omega_3 + \frac{1}{2}\cos(\theta_2-\theta_3)\dot{\omega}_3 \\ \hline \frac{1}{3}\dot{\omega}_3 - \frac{1}{2}\sin(\theta_1-\theta_3)(\omega_1-\omega_3)\omega_1 + \frac{1}{2}\cos(\theta_1-\theta_3)\dot{\omega}_1 \\ -\frac{1}{2}\sin(\theta_2-\theta_3)(\omega_2-\omega_3)\omega_2 + \frac{1}{2}\cos(\theta_2-\theta_3)\dot{\omega}_3 \end{array} \end{pmatrix}_{3\times 1}$$

$$= \underbrace{\begin{pmatrix} \frac{7}{3} & \frac{3}{2}\cos(\theta_1-\theta_2) & \frac{1}{2}\cos(\theta_1-\theta_3) \\ \frac{3}{2}\cos(\theta_1-\theta_2) & \frac{4}{3} & \frac{1}{2}\cos(\theta_2-\theta_3) \\ \frac{1}{2}\cos(\theta_1-\theta_3) & \frac{1}{2}\cos(\theta_2-\theta_3) & \frac{1}{3} \end{pmatrix}_{3\times 3}}_{A} \begin{pmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{pmatrix}_{3\times 1}$$
$$+ \underbrace{\begin{pmatrix} -\frac{3}{2}\sin(\theta_1-\theta_2)(\omega_1-\omega_2)\omega_2 - \frac{1}{2}\sin(\theta_1-\theta_3)(\omega_1-\omega_3)\omega_3 \\ -\frac{3}{2}\sin(\theta_1-\theta_2)(\omega_1-\omega_2)\omega_1 - \frac{1}{2}\sin(\theta_2-\theta_3)(\omega_2-\omega_3)\omega_3 \\ -\frac{1}{2}\sin(\theta_1-\theta_3)(\omega_1-\omega_3)\omega_1 - \frac{1}{2}\sin(\theta_2-\theta_3)(\omega_2-\omega_3)\omega_2 \end{pmatrix}_{3\times 1}}_{b} \tag{9}$$

Using the results from (7), (8), and (9), we finally have from (6) the desired equation of motion:

$$\begin{aligned} \dot{\theta} &= \omega \\ \dot{\omega} &= A^{-1}\left(\frac{1}{m\ell^2}\left(\frac{\partial K}{\partial \theta}\right)^\top - \frac{1}{m\ell^2}\left(\frac{\partial U}{\partial \theta}\right)^\top - b\right) \end{aligned} \tag{10}$$

Since (7), (8), and (9) are actually independent of m, the above equation of motion is also independent of m. So the choice of m does not affect the motion of a triple pendulum system.

2.3.2 Numerical Integration

The numerical integrator used in the provided video was RK4 with the initial conditions

$$\begin{aligned}\theta_1(0) &= 1.5708 \text{ rad}\\ \theta_2(0) &= 0.46 \text{ rad}\\ \theta_3(0) &= 1.8 \text{ rad}\\ \omega_n(0) &= 0 \text{ rad/s}\end{aligned}$$

and the constant parameters

$$\begin{aligned}\ell &= 1 \text{ m}\\ g &= 9.80665 \text{ m/s}^2\\ \Delta t &= 1/24 \text{ s} \approx 0.42 \text{ s}\end{aligned}$$

The algorithm was implemented in Houdini VEX. Note A^{-1} in (10) is computable using the VEX built-in function `invert`, since $A \in \mathbf{R}^{3\times 3}$ and VEX has matrix type `matrix3`. VEX also has the type `matrix` for $\mathbf{R}^{4\times 4}$, so we could even simulate a quadruple pendulum system in VEX fairly easily (the derivation for the equation of motion would be quite a bit longer, though mostly the same in spirit). For simulating an n-pendulum where $n \geqslant 5$, we would have to resort to Python, such as with `scipy.linalg.solve` (since A is symmetric, we can pass `assume_a='sym'`).

Additionally, a green/red bar above the triple pendulum was included in the video. The bar represents the total energy $K + U$ of the system (U is generally negative, so the size of the bar is technically $K + (U + U_0)$, where U_0 is some baseline quantity to guarantee $U + U_0 \geqslant 0$ J). The size of the bar fluctuates slightly over the duration of the video. There is a noticeable overall decrease in size by the end of the video, illustrating the dissipation from using RK4.

2.3.3 VEX Code

```
float cos12 = cos(theta1 - theta2);
float cos13 = cos(theta1 - theta3);
float cos23 = cos(theta2 - theta3);
float sin12 = sin(theta1 - theta2);
float sin13 = sin(theta1 - theta3);
float sin23 = sin(theta2 - theta3);

matrix3 A = set(
    7/3.f,          3/2.f * cos12,  1/2.f * cos13,
    3/2.f * cos12,  4/3.f,          1/2.f * cos23,
    1/2.f * cos13,  1/2.f * cos23,  1/3.f
);

vector b = set(
    -3/2.f * sin12 * (omega1 - omega2) * omega2 - 1/2.f * sin13 * (omega1 - omega3) * omega3,
    -3/2.f * sin12 * (omega1 - omega2) * omega1 - 1/2.f * sin23 * (omega2 - omega3) * omega3,
    -1/2.f * sin13 * (omega1 - omega3) * omega1 - 1/2.f * sin23 * (omega2 - omega3) * omega2
);

vector dK_dtheta = set(
    -3/2.f * sin12 * omega1 * omega2 - 1/2.f * sin13 * omega1 * omega3,
    3/2.f * sin12 * omega1 * omega2 - 1/2.f * sin23 * omega2 * omega3,
    1/2.f * sin13 * omega1 * omega3 + 1/2.f * sin23 * omega2 * omega3
);

vector dU_dtheta = g/l * set(
    5/2.f * sin(theta1),
    3/2.f * sin(theta2),
    1/2.f * sin(theta3)
);

float dtheta1_dt = omega1;
float dtheta2_dt = omega2;
float dtheta3_dt = omega3;
vector domega_dt = invert(A) * (dK_dtheta - dU_dtheta - b);
```

Listing 1: VEX implementation of f, where $\dot{y} = f(y)$ and $y = (\theta_1, \omega_1, \theta_2, \omega_2, \theta_3, \omega_3)^\top$. The code is to be used by a "Detail Wrangler" inside of a "Solver."

```
function float[] add(const float a[], b[])
{
    float c[];
    resize(c, max(len(a), len(b)));
    for (int i = 0; i < max(len(a), len(b)); i++)
        c[i] = a[i] + b[i];
    return c;
}

function float[] mult(const float k, a[])
{
    float b[];
    resize(b, len(a));
    for (int i = 0; i < len(a); i++)
        b[i] = k * a[i];
    return b;
}

float yPrev[];
yPrev[0] = f@theta1;
yPrev[1] = f@omega1;
yPrev[2] = f@theta2;
yPrev[3] = f@omega2;
yPrev[4] = f@theta3;
yPrev[5] = f@omega3;

float k1[] = f(yPrev, f@l, f@g);
float k2[] = f(add(yPrev, mult(f@dt/2, k1)), f@l, f@g);
float k3[] = f(add(yPrev, mult(f@dt/2, k2)), f@l, f@g);
float k4[] = f(add(yPrev, mult(f@dt, k3)), f@l, f@g);
float yNext[] = add(yPrev, mult(f@dt/6, add(k1, add(mult(2, k2), add(mult(2, k3),
     k4)))));

f@theta1 = yNext[0];
f@omega1 = yNext[1];
f@theta2 = yNext[2];
f@omega2 = yNext[3];
f@theta3 = yNext[4];
f@omega3 = yNext[5];
```

Listing 2: VEX implementation of the RK4 algorithm with y as an array of arbitrary size, since VEX's largest vector type is `vector4`, while the triple pendulum system requires six variables. The code is to be used alongside some function `f`, e.g. the implementation for f in 1.