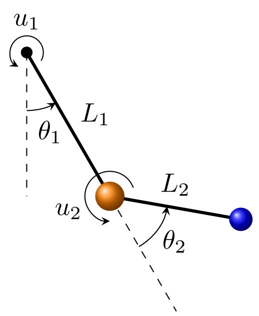
Consider the following undamped double pendulum:



θ_1 = angle of upper rod (measured from the lower vertical)
θ_2 = angle of lower rod (measured relative to θ_1)
m_1 = mass of middle weight
m_2 = mass of end weight
L_1 = length of rigid, massless upper rod
L_2 = length of rigid, massless lower rod
u_1 = input torque on the stationary pivot
u_2 = input torque on the joint connecting the rods

Gravity is straight down in this assignment, with a gravitational constant of $g = 9.81\,\mathrm{m/s^2}$. We define the vector of angles $\theta = [\theta_1 \;\; \theta_2]^T$, which takes its values on the two-dimensional torus $\mathbb{T}^2 = S^1 \times S^1$. We also define the vector of angular velocities $\dot\theta = [\dot\theta_1 \;\; \dot\theta_2]^T$, which takes its values on the plane $\mathbb{R}^2$. Thus the four-dimensional state vector $x = [\theta^T \;\; \dot\theta^T]^T$ takes its values on the state space $X = \mathbb{T}^2 \times \mathbb{R}^2$. If we define the vector of inputs $u = [u_1 \;\; u_2]^T \in \mathbb{R}^2$, then we can write the equations of motion for this system as

$$M(\theta)\ddot\theta + C(\theta,\dot\theta)\dot\theta + \nabla U(\theta) = u\,, \tag{1}$$

where ∇U denotes the gradient of the gravitational potential function

$$U(\theta) = -(m_1+m_2)gL_1\cos(\theta_1) - m_2 g L_2 \cos(\theta_1+\theta_2)\,, \tag{2}$$

and the 2×2 matrix functions $M(\theta)$ and $C(\theta,\dot\theta)$ are given by

$$M(\theta) = \begin{bmatrix} (m_1+m_2)L_1^2 + m_2L_2^2 + 2m_2L_1L_2\cos(\theta_2) & m_2L_2^2 + m_2L_1L_2\cos(\theta_2) \\ m_2L_2^2 + m_2L_1L_2\cos(\theta_2) & m_2L_2^2 \end{bmatrix} \tag{3}$$

$$C(\theta,\dot\theta) = m_2L_1L_2\sin(\theta_2)\begin{bmatrix} -\dot\theta_2 & -\dot\theta_1 - \dot\theta_2 \\ \dot\theta_1 & 0 \end{bmatrix}. \tag{4}$$

1. Show that the matrix function $M(\theta)$ is positive definite for any $\theta \in \mathbb{T}^2$.

2. Show that the matrix function

$$\frac{d}{dt}M(\theta) - 2C(\theta,\dot\theta) \tag{5}$$

is skew-symmetric for all values of θ and $\dot\theta$.

3. Define the total energy (kinetic plus potential) as

$$E(\theta,\dot\theta) = \tfrac{1}{2}\dot\theta^T M(\theta)\dot\theta + U(\theta)\,.$$

Show that the time derivative of the energy along trajectories of the system (1) is

$$\frac{d}{dt}E(\theta,\dot\theta) = \dot\theta^T u = \dot\theta_1 u_1 + \dot\theta_2 u_2 \tag{6}$$

(in particular, energy is conserved when the inputs are zero). Hint: you can derive (6) directly from (1), (5), and the fact that $M(\theta)$ is symmetric, without using the messy definitions in (2), (3), and (4).

4. Suppose you have a single small motor capable of producing torques in the interval $[-0.02, 0.02]$ (in units of $\mathrm{N\,m}$). Suppose also that you have sensors which give you clean measurements of the angular velocities $\dot{\theta}_1$ and $\dot{\theta}_2$. Your goal is to use feedback to damp out the oscillations in the system. You have two choices for attaching your motor to the pendulum: you can attach it to the stationary pivot, so that it produces a torque u_1, or you can attach it to the joint connecting the rods, so that it produces a torque u_2. For each case, use the equation (6) to design a simple feedback control which will tend to decrease the total energy in the system over time. This feedback control can be a static function of the measurements (there is no need to introduce any internal controller states). Using your favorite solver, simulate each controller (one for u_1 and one for u_2), being sure to saturate the control action so that the control signal remains in the interval $[-0.02, 0.02]$. In other words, after you calculate u_i at each time instant, check if it is above 0.02, and if so set it to 0.02; likewise, check if it is below -0.02, and if so set it to -0.02. For these simulations, use parameter values $m_1 = m_2 = 1\,\mathrm{kg}$ and $L_1 = L_2 = 0.5\,\mathrm{m}$. Based on simulations of the two closed-loop systems for various values of initial angles, where would you rather attach your motor to best achieve your goal of damping out the oscillations? You can simulate over a long time period, say $1000\,\mathrm{s}$, plot the energy E versus time, and determine which of the two motor positions usually causes the energy to decrease to a small value like 10^{-15} first.

5. Now consider the undamped pendulum system with zero applied torques $u \equiv 0$. This is an example of a *chaotic system*, one for which tiny changes in the initial state can lead to large changes over time. To illustrate this property, consider starting the pendulum at a specific configuration $\theta_1(0)$ and $\theta_2(0)$ with zero velocities $\dot{\theta}_1(0) = \dot{\theta}_2(0) = 0$, letting go, and then finding the first time T_{flip} at which the lower pendulum flips over. If we let $\phi = \theta_1 + \theta_2$ denote the angle of the lower pendulum measured from the lower vertical, and if $\phi(0)$ is in the interval $(-\pi, \pi)$, then T_{flip} will be the first time for which $|\phi| = 2\pi$ (with $T_{\text{flip}} = \infty$ if this never happens). In this way T_{flip} becomes a function of the initial configuration, and if we assign different colors to different values of T_{flip} we get fractal images like the one in Figure 1. Using your favorite solver with the same parameter values as in part 4, calculate T_{flip} over a grid of initial configurations and generate a fractal plot similar to the one in Figure 1. Some considerations:

   - If the initial energy is less than or equal to the critical value $E_{\text{crit}} = \min\{2(m_1 + m_2)gL_1,\ 2m_2gL_2\}$, then the lower pendulum will never flip. Thus for such initial configurations you can set $T_{\text{flip}} = \infty$ without having to simulate anything.
   - An easy way to calculate T_{flip} is to use a *callback* in the solver to terminate the simulation as soon as the condition $|\phi| = 2\pi$ is satisfied. For example, both Matlab and Julia have solvers with callbacks that can terminate the simulation. With such a callback in place, you can simply set T_{flip} equal to the final simulation time. This works as long as the nominal simulation time is set to some large value, say $5000\,\mathrm{s}$, because in this approach any T_{flip} above that value will be equivalent to $T_{\text{flip}} = \infty$.
   - Figure 1 uses a grid of 300 points along each axis, for a total of 90000 different initial configurations. If you have a slow computer, you might not be able to simulate all of these within a reasonable amount of time, in which case you should use a coarser grid. If your computer has a CPU with multiple cores, you might consider doing these calculations in parallel for a significant speed boost (for example by using `parfor` in Matlab or `pmap` in Julia).
   - The energy E should be conserved in this system, that is, a plot of E versus time should be constant from any initial configuration. However, numerical errors in the solver will cause E to drift over time. Make sure that you choose the error tolerances in the solver small enough so that such drift is negligible over the simulation time horizon, as otherwise you could be getting inaccurate values of T_{flip}.
   - Figure 1 shows a particular range of values for θ_1 and ϕ, but you are free to examine a different range. Just make sure that the range you choose includes some values for which T_{flip} is finite, as otherwise your picture will be quite boring. Also, choose a method for assigning different colors to different values of T_{flip} that results in a picture that you find aesthetically pleasing.
   - Be careful about the relationship between the pixel position in the image (that is, its row and column) and the corresponding initial pair $(\theta_1(0), \phi(0))$. In particular, $\theta_1(0)$ changes along the column and $\phi(0)$ changes along the row, so the pair $(\theta_1(0), \phi(0))$ corresponds to the pair (column,row) instead of (row,column).

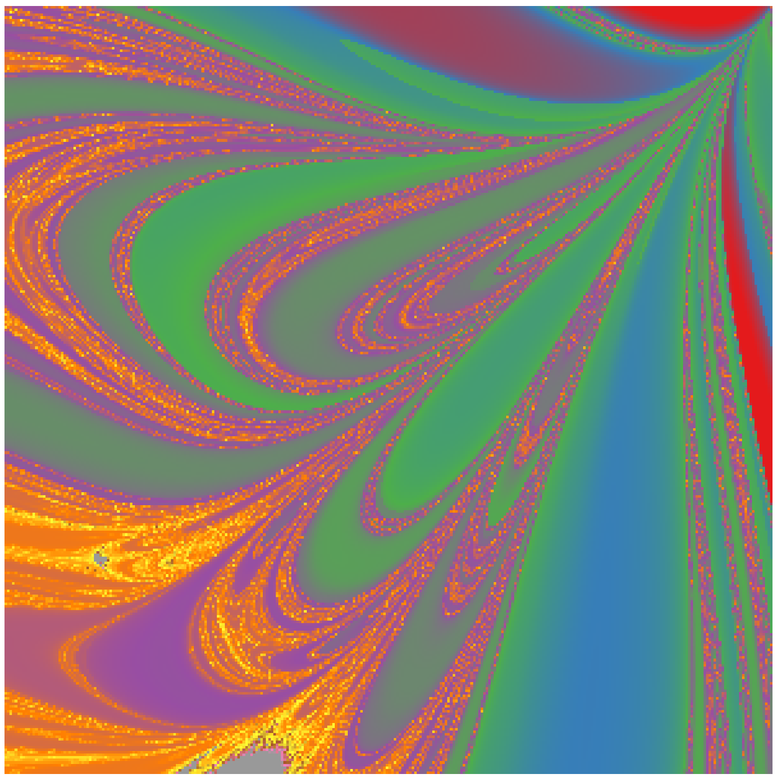


Figure 1: A plot of T_{flip} versus θ_1 on the horizontal axis and ϕ on the vertical axis, each over the interval $(\pi/2, \pi)$.