Double Pendulum Forecasting with Artificial Neural Networks

Theory of the Artificial Neural Networks, Time Series Analysis and Double Pendulum

Juan José Romero Cruz

August, 2023

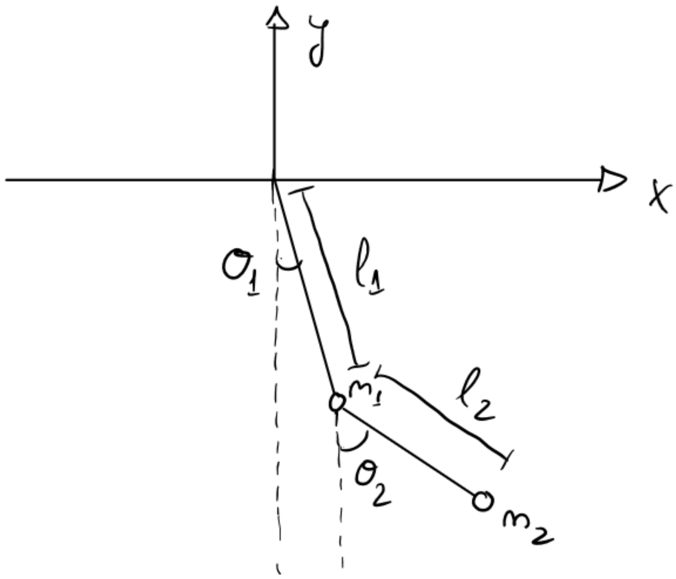
A humble attempt to provide a theoretical support for a naive and beautiful experiment

Contents

1 Introduction

2 The double pendulum

Firstly, It is important to introduce our problem in a purely mechanical point of view, me have the following problem:



As we can see in this gorgeus picture, we have a mass, m_1, hanging by a thread of lenght l_1 and, at the same time, a mass m_2 hangs by another thread of lenght l_2, which is sticked to the mass m_1. We can use cartesian coordinates to obtain the equations but it is better using polar coordinates since we have, as constrain, that the radius is fixed by l_1 and l_2. So, we only have to use the variables θ_1 and θ_2.

I won't write here all the process that has brought me to the set of differential equations but I'll describe the process, which is based in Lagrangian Mechanics.

The two both masses are influenced by gravity, so the Lagrangian is:

$$\begin{aligned} L(\theta_1, \theta_2, \dot{\theta_1}, \dot{\theta_2}) &= T(\dot{\theta_1}, \dot{\theta_2}) - V(\theta_1, \theta_2) \\ T(\dot{\theta_1}, \dot{\theta_2}) &= \frac{1}{2} m_1 (\dot{r_1})^2 + \frac{1}{2} m_2 (\dot{r_2})^2 \\ V(\theta_1, \theta_2) &= m_1 g y_1 + m_2 g y_2 \end{aligned} \tag{2.1}$$

I have found the position vectors of the two both masses, r_1 and r_2. Therefore, the final lagrangian is:

$$\begin{aligned} &L(\theta_1, \theta_2, \dot{\theta_1}, \dot{\theta_2}) = \frac{1}{2} m_1 (l_1 \dot{\theta_1})^2 + \frac{1}{2} m_2 (l_1 \dot{\theta_1})^2 + \frac{1}{2} m_2 (l_2 \dot{\theta_2})^2 + \\ &m_2 l_1 l_2 \dot{\theta_1} \dot{\theta_2} \cos(\theta_1 - \theta_2) + m_1 g l_1 \cos\theta_1 + m_2 g l_2 \cos\theta_2 \end{aligned} \tag{2.2}$$

And now, we can use the Euler-Lagrange equations, which are:

$$\frac{d}{dt}[\frac{\partial L}{\partial \dot{q_i}}] - \frac{\partial L}{\partial q_i} = 0 \tag{2.3}$$

With $q_i = \theta_1, \theta_2$. Finally, after differentiate a lot (wuhuuu), the movement equation for θ_1 is:

$$l_1\ddot{\theta_1} + \mu_2 l_2\ddot{\theta_2}\cos(\theta_1 - \theta_2) + \mu_2 l_2(\dot{\theta_2})^2\sin(\theta_1 - \theta_2) - g\mu_1\sin\theta_1 = 0 \tag{2.4}$$

Where $\mu_1 = \frac{m_1}{m_1+m_2}$, $\mu_2 = \frac{m_2}{m_1+m_2}$. And the movement equation for θ_2 is:

$$l_2\ddot{\theta_2} + l_1\ddot{\theta_1}\cos(\theta_1 - \theta_2) - l_1(\dot{\theta_1})^2\sin(\theta_1 - \theta_2) - g\sin\theta_2 = 0 \tag{2.5}$$

As we can see, these equations are coupled, this means that $\ddot{\theta_1}$ and $\ddot{\theta_2}$ are in the to both equations. If these equations were discoupled, each second derivative would be in one of the equations. This program solves the equations once they are discoupled, but I won't write this.

Finally, the Hamiltonian is:

$$\begin{aligned}H(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{1}{2}(m_1 + m_2)(l_1)^2(\dot{\theta}_1)^2 + \frac{1}{2}m_2(l_2)^2(\dot{\theta}_2)^2 + \\ m_2 l_1 l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2) + m_1 g l_1 cos\theta_1 + m_2 g l_2\cos\theta_2\end{aligned} \tag{2.6}$$

And, the system's energy is:

$$\begin{aligned}E(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2) = \frac{1}{2}m_1(l_1\dot{\theta}_1)^2 + \frac{1}{2}m_2(l_1\dot{\theta}_1)^2 + \frac{1}{2}m_2(l_2\dot{\theta}_2)^2 + \\ m_2 l_1 l_2\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2) - m_1 g l_1\cos\theta_1 - m_2 g l_2\cos\theta_2\end{aligned} \tag{2.7}$$

This is the problem we are going to trackle: we want a model that wouldn't care about the dynamics of this problem, this model will only see the pendulum's evolution a lot of times.

3 Artificial neural networks

4 Using the ANNs for the double pendulum forecasting

We want to know the solution of the system: $x'(t) = f(t,x)$ given a certain time t. Our model will give us a solution, $x(t + \Delta t)$ given $x(t)$. By this way, we will be able to give to the model the previous solution x(t+Δt) in order to obtain the solution $x(t + 2\Delta t)$. By induction we can say that we could calculate x(t+(n+1) Δt) by using as input x(t + n Δn).

As a note, it could have been a good idea having done a dimensionality reduction by using an "autoencoder type structure".

4.1 Train

What we are going to do is something called "self-supervised learning", which is like supervised learning, but the labeled dataset is generated by the computer itself.

Firtsly, remember that we are treatening the following dynamical system:

$$\begin{aligned} &\dot{x}(t) = f(t,x) \\ &\dot{x}, x \in \mathbb{R}^n, t \in [0,T] \end{aligned} \tag{4.1}$$

Therefore, the train follows like this:

1 description

2 description

4.2 Prediction

References

[1] T. Vicsek, A. Czirók et. al,
Novel type of phase transition in a system of self-driven particles,
Physical Review Letters **75** (1995) 1226 [https://doi.org/10.1103/PhysRevLett.75.1226]

[2] P. Degond & S. Motsch,
Continuum limit of self-driven particles with orientation interaction,
Mathematical Models and Methods in Applied Sciences (2007) [https://doi.org/10.1142/S0218202508003005]

[3] P. Degond, A. Frouvelle & S. Merino-Aceituno,
A new flocking model through body attitude coordination,
Mathematical Models and Methods in Applied Sciences (2017), [http://dx.doi.org/10.1142/S0218202517400085]

[4] A. Frouvelle & J. Liu,
Dynamics in a kinetic model of oriented particles with phase transition,
SIAM J. MATH. ANAL. (2012), [10.1137/110823912]

[5] P. Degond, A. Frouvelle & J. Liu,
Macroscopic Limits and Phase Transition in a System of Self-propelled Particles,
J Nonlinear Sci (2013), [10.1007/s00332-012-9157-y]

[6] H. Brezis,
Functional Analysis, Sobolev Spaces and Partial Differential Equations,
Springer, 2010.