FreeSandal

樹莓派, 樹莓派之學習, 樹莓派之教育

STEM 隨筆︰古典力學︰運動學【三．V】

2018-04-16 | 懸鉤子 | 發表迴響

承上篇，如是當知廣義座標

Generalized coordinates

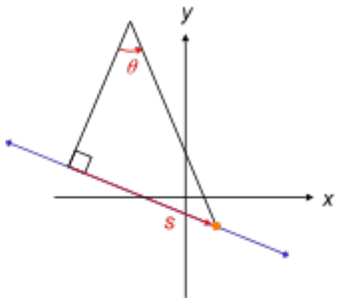
In analytical mechanics, specifically the study of the rigid body dynamics of multibody systems, the term **generalized coordinates** refers to the parameters that describe the configuration of the system relative to some reference configuration. These parameters must uniquely define the configuration of the system relative to the reference configuration.[1] This is done assuming that this can be done with a single chart. The **generalized velocities** are the time derivatives of the generalized coordinates of the system.

An example of a generalized coordinate is the angle that locates a point moving on a circle. The adjective "generalized" distinguishes these parameters from the traditional use of the term coordinate to refer to Cartesian coordinates: for example, describing the location of the point on the circle using x and y coordinates.

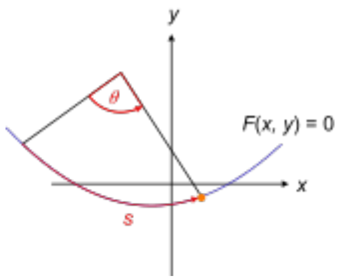
Although there may be many choices for generalized coordinates for a physical system, parameters which are convenient are usually selected for the specification of the configuration of the system and which make the solution of its equations of motion easier. If these parameters are independent of one another, the number of independent generalized coordinates is defined by the number of degrees of freedom of the system.[2][3]

……

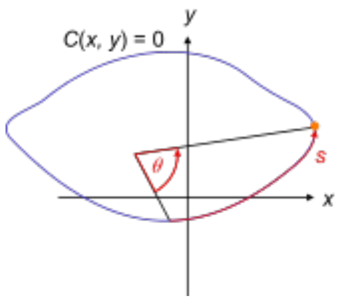
Constraints and degrees of freedom



Open straight path

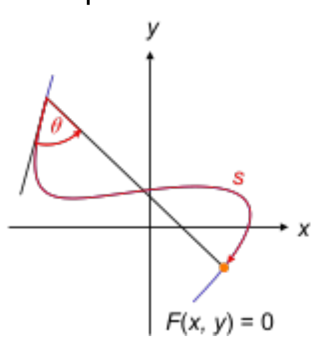


Open curved path $F(x, y) = 0$

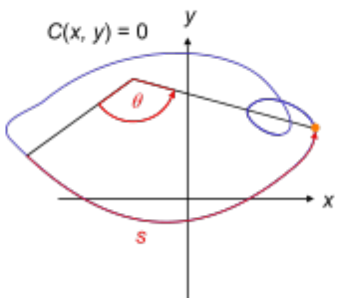


Closed curved path $C(x, y) = 0$

One generalized coordinate (one degree of freedom) on paths in 2D. Only one generalized coordinate is needed to uniquely specify positions on the curve. In these examples, that variable is either arc length s or angle θ. Having both of the Cartesian coordinates (x, y) are unnecessary since either x or y is related to the other by the equations of the curves. They can also be parameterized by s or θ.



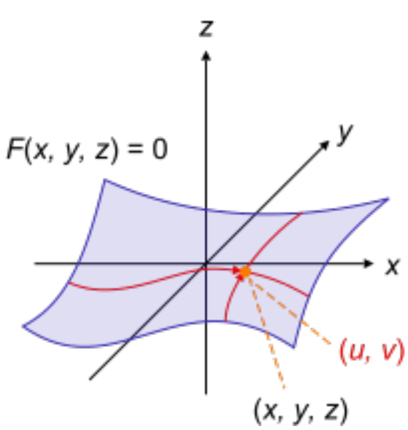
Open curved path $F(x, y) = 0$. Multiple intersections of radius with path.



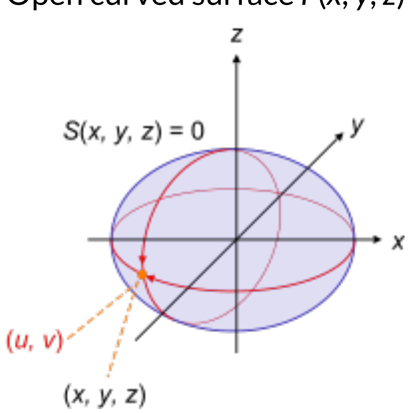
Closed curved path $C(x, y) = 0$. Self-intersection of path.
The arc length s along the curve is a legitimate generalized coordinate since the position is uniquely determined, but the angle θ is not since there are multiple positions for a single value of θ.

Generalized coordinates are usually selected to provide the minimum number of independent coordinates that define the configuration of a system, which simplifies the formulation ofLagrange's equations of motion. However, it can also occur that a useful set of generalized coordinates may be *dependent*, which means that they are related by one or moreconstraint equations.

Holonomic constraints



Open curved surface $F(x, y, z) = 0$



Closed curved surface $S(x, y, z) = 0$

Two generalized coordinates, two degrees of freedom, on curved surfaces in 3d. Only two numbers (u, v) are needed to specify the points on the curve, one possibility is shown for each case. The full threeCartesian coordinates (x, y, z) are not necessary because any two determines the third according to the equations of the curves.

For a system of N particles in 3D real coordinate space, the position vector of each particle can be written as a 3-tuple in Cartesian coordinates:

$$\mathbf{r}_1 = (x_1, y_1, z_1)\,, \quad \mathbf{r}_2 = (x_2, y_2, z_2)\,, \ldots\,, \mathbf{r}_N = (x_N, y_N, z_N)\,.$$

Any of the position vectors can be denoted $\mathbf{r}_k$ where $k = 1, 2, \ldots, N$ labels the particles. A *holonomic constraint* is a *constraint equation* of the form for particle k[4][nb 1]

$$f(\mathbf{r}_k, t) = 0$$

which connects all the 3 spatial coordinates of that particle together, so they are not independent. The constraint may change with time, so time t will appear explicitly in the constraint equations. At any instant of time, when t is a constant, any one coordinate will be determined from the other coordinates, e.g. if x_k and z_k are given, then so is y_k. One constraint equation counts as *one* constraint. If there are C constraints, each has an equation, so there will be C constraint equations. There is not necessarily one constraint equation for each particle, and if there are no constraints on the system then there are no constraint equations.

So far, the configuration of the system is defined by $3N$ quantities, but C coordinates can be eliminated, one coordinate from each constraint equation. The number of independent coordinates is $n = 3N - C$. (InD dimensions, the original configuration would need ND coordinates, and the reduction by constraints means $n = ND - C$). It is ideal to use the minimum number of coordinates needed to define the configuration of the entire system, while taking advantage of the constraints on the system. These quantities are known as **generalized coordinates** in this context, denoted $q_j(t)$. It is convenient to collect them into an n–tuple

$$\mathbf{q}(t) = (q_1(t), q_2(t), \ldots, q_n(t))$$

which is a point in the *configuration space* of the system. They are all independent of one other, and each is a function of time. Geometrically they can be lengths along straight lines, or arc lengths along curves, or angles; not necessarily Cartesian coordinates or other standard orthogonal coordinates. There is one for each degree of freedom, so the number of generalized coordinates equals the number of degrees of freedom, *n*. A degree of freedom corresponds to one quantity that changes the configuration of the system, for example the angle of a pendulum, or the arc length traversed by a bead along a wire.

If it is possible to find from the constraints as many independent variables as there are degrees of freedom, these can be used as generalized coordinates[5] The position vector $\mathbf{r}_k$ of particle *k* is a function of all the *n* generalized coordinates and time,[6][7][8][5][nb 2]

$$\mathbf{r}_k = \mathbf{r}_k(\mathbf{q}(t), t)\,,$$

and the generalized coordinates can be thought of as parameters associated with the constraint.

The corresponding time derivatives of $\mathbf{q}$ are the generalized velocities,

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = (\dot{q}_1(t), \dot{q}_2(t), \ldots, \dot{q}_n(t))$$

(each dot over a quantity indicates one time derivative). The velocity vector $\mathbf{v}_k$ is the total derivative of $\mathbf{r}_k$ with respect to time

$$\mathbf{v}_k = \dot{\mathbf{r}}_k = \frac{d\mathbf{r}_k}{dt} = \sum_{j=1}^{n} \frac{\partial \mathbf{r}_k}{\partial q_j}\dot{q}_j + \frac{\partial \mathbf{r}_k}{\partial t}\,.$$

and so generally depends on the generalized velocities and coordinates. Since we are free to specify the initial values of the generalized coordinates and velocities separately, the generalized coordinates q_j and velocities dq_j/dt can be treated as *independent variables*.

Non-holonomic constraints

A mechanical system can involve constraints on both the generalized coordinates and their derivatives. Constraints of this type are known as non-holonomic. First-order non-holonomic constraints have the form

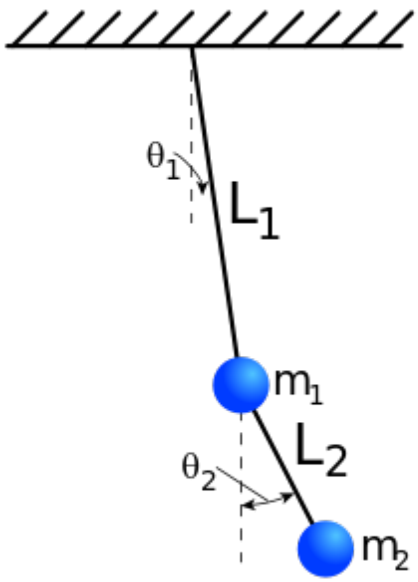
$$g(\mathbf{q}, \dot{\mathbf{q}}, t) = 0\,,$$

An example of such a constraint is a rolling wheel or knife-edge that constrains the direction of the velocity vector. Non-holonomic constraints can also involve next-order derivatives such as generalized accelerations.

在理論以及實務裡的好處也。

固然簡單範例

Double pendulum



A double pendulum

The benefits of generalized coordinates become apparent with the analysis of a double pendulum. For the two masses m_i, i=1, 2, let $\mathbf{r}_i=(x_i, y_i)$, i=1, 2 define their two trajectories. These vectors satisfy the two constraint equations,

$$f_1(x_1, y_1, x_2, y_2) = \mathbf{r}_1 \cdot \mathbf{r}_1 - L_1^2 = 0, \quad f_2(x_1, y_1, x_2, y_2) = (\mathbf{r}_2 - \mathbf{r}_1) \cdot (\mathbf{r}_2 - \mathbf{r}_1) - L_2^2 = 0.$$

The formulation of Lagrange's equations for this system yields six equations in the four Cartesian coordinates x_i, y_i i=1, 2 and the two Lagrange multipliers λ_i, i=1, 2 that arise from the two constraint equations.

Now introduce the generalized coordinates θ_i i=1,2 that define the angular position of each mass of the double pendulum from the vertical direction. In this case, we have

$$\mathbf{r}_1 = (L_1 \sin\theta_1, -L_1\cos\theta_1), \quad \mathbf{r}_2 = (L_1 \sin\theta_1, -L_1\cos\theta_1) + (L_2 \sin\theta_2, -L_2\cos\theta_2).$$

The force of gravity acting on the masses is given by,

$$\mathbf{F}_1 = (0, -m_1 g), \quad \mathbf{F}_2 = (0, -m_2 g)$$

where g is the acceleration of gravity. Therefore, the virtual work of gravity on the two masses as they follow the trajectories $\mathbf{r}_i$, i=1,2 is given by

$$\delta W = \mathbf{F}_1 \cdot \delta\mathbf{r}_1 + \mathbf{F}_2 \cdot \delta\mathbf{r}_2.$$

The variations $\delta\mathbf{r}_i$ i=1, 2 can be computed to be

$$\delta\mathbf{r}_1 = (L_1\cos\theta_1, L_1\sin\theta_1)\delta\theta_1, \quad \delta\mathbf{r}_2 = (L_1\cos\theta_1, L_1\sin\theta_1)\delta\theta_1 + (L_2\cos\theta_2, L_2\sin\theta_2)\delta\theta_2$$

Thus, the virtual work is given by

$$\delta W = -(m_1 + m_2) g L_1 \sin\theta_1 \delta\theta_1 - m_2 g L_2 \sin\theta_2 \delta\theta_2,$$

and the generalized forces are

$$F_{\theta_1} = -(m_1 + m_2) g L_1 \sin\theta_1, \quad F_{\theta_2} = -m_2 g L_2 \sin\theta_2.$$

Compute the kinetic energy of this system to be

$$T = \frac{1}{2} m_1 \mathbf{v}_1 \cdot \mathbf{v}_1 + \frac{1}{2} m_2 \mathbf{v}_2 \cdot \mathbf{v}_2 = \frac{1}{2}(m_1 + m_2) L_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 L_2^2 \dot{\theta}_2^2 + m_2 L_1 L_2 \cos(\theta_2 - \theta_1)\dot{\theta}_1\dot{\theta}_2.$$

Euler–Lagrange equation yield two equations in the unknown generalized coordinates θ_i i=1, 2, given by[14]

$$(m_1 + m_2)L_1^2\ddot{\theta}_1 + m_2L_1L_2\ddot{\theta}_2\cos(\theta_2 - \theta_1) + m_2L_1L_2\ddot{\theta_2}^2\sin(\theta_1 - \theta_2) = -(m_1+m_2)gL_1\sin\theta_1,$$

and

$$m_2L_2^2\ddot{\theta}_2+m_2L_1L_2\ddot{\theta}_1\cos(\theta_2-\theta_1)+m_2L_1L_2\ddot{\theta_1}^2\sin(\theta_2-\theta_1) = -m_2gL_2\sin\theta_2.$$

The use of the generalized coordinates θ_i i=1, 2 provides an alternative to the Cartesian formulation of the dynamics of the double pendulum.

可以方便『紙筆計算』！猶恐複雜推導，易犯無心之錯呦？

已然得手應心者，何不開始學習 SymPy

Lagrange's Method in Physics/Mechanics

`mechanics` provides functionality for deriving equations of motion using Lagrange's method. This document will describe Lagrange's method as used in this module, but not how the equations are actually derived.

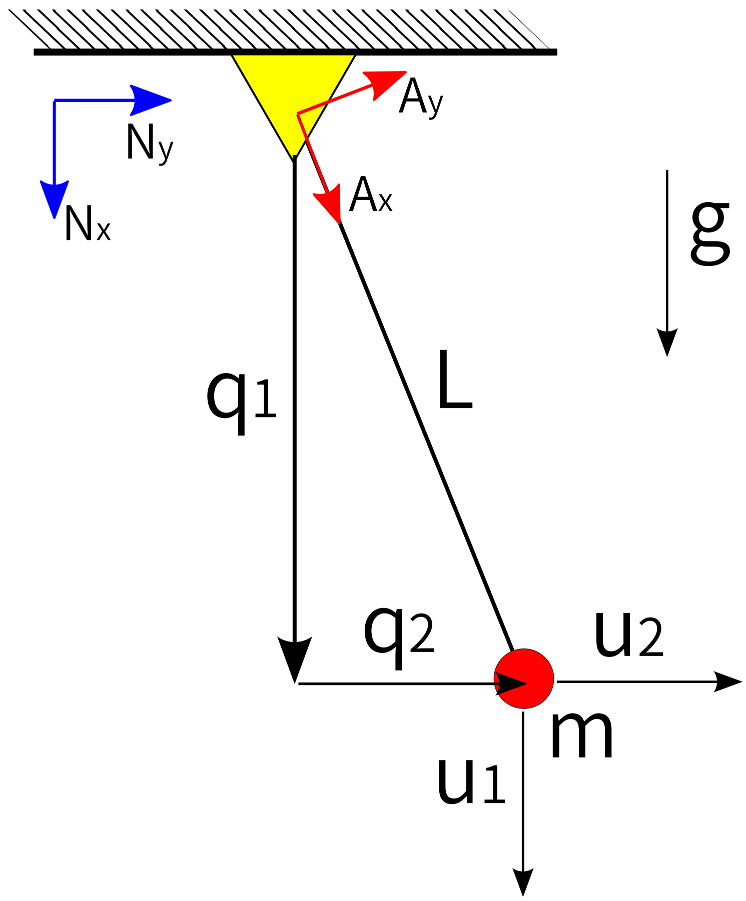
………

的『符號運算』法，將之『自動化』耶？？

特假借

Nonminimal Coordinates Pendulum

In this example we demonstrate the use of the functionality provided in `mechanics` for deriving the equations of motion (EOM) for a pendulum with a nonminimal set of coordinates. As the pendulum is a one degree of freedom system, it can be described using one coordinate and one speed (the pendulum angle, and the angular velocity respectively). Choosing instead to describe the system using the xx and yy coordinates of the mass results in a need for constraints. The system is shown below:



The system will be modeled using both Kane's and Lagrange's methods, and the resulting EOM linearized. While this is a simple problem, it should illustrate the use of the linearization methods in the presence of constraints.

※ 練習

```python
from __future__ import print_function, division
from sympy import init_printing
init_printing(use_latex='mathjax', pretty_print=False)
```

```python
from sympy.physics.mechanics import *
from sympy import symbols, atan, Matrix
q1, q2 = dynamicsymbols('q1:3')
q1d, q2d = dynamicsymbols('q1:3', level=1)
L, m, g, t = symbols('L, m, g, t')
```

```python
N = ReferenceFrame('N')
pN = Point('N*')
pN.set_vel(N, 0)
theta1 = atan(q2/q1)
A = N.orientnew('A', 'axis', [theta1, N.z])
```

```python
P = pN.locatenew('P1', q1*N.x + q2*N.y)
P.set_vel(N, P.pos_from(pN).dt(N))
pP = Particle('pP', P, m)
```

```python
f_c = Matrix([q1**2 + q2**2 - L**2])
```

```python
R = m*g*N.x
```

```python
Lag = Lagrangian(N, pP)
LM = LagrangesMethod(Lag, [q1, q2], hol_coneqs=f_c, forcelist=[(P, R)], frame=N)
lag_eqs = LM.form_lagranges_equations()
Lag, lag_eqs
```

$$\left (\frac{m}{2} \frac{d}{d t} \operatorname{q_{1}}{\left (t \right)}^{2} + \frac{m}{2} \frac{d}{d t} \operatorname{q_{2}}{\left (t \right)}^{2}, \quad \left[\begin{matrix}- g m + m \frac{d^{2}}{d t^{2}} \operatorname{q_{1}}{\left (t \right)} + 2 \operatorname{lam_{1}}{\left (t \right)} \operatorname{q_{1}}{\left (t \right)}\\m \frac{d^{2}}{d t^{2}} \operatorname{q_{2}}{\left (t \right)} + 2 \operatorname{lam_{1}}{\left (t \right)} \operatorname{q_{2}}{\left (t \right)}\end{matrix}\right]\right)$$

```python
op_point = {q1: L, q2: 0, q1d: 0, q2d: 0, q1d.diff(t): 0, q2d.diff(t): 0}
```

```python
lam_op = LM.solve_multipliers(op_point=op_point)
```

```python
op_point.update(lam_op)
A, B, inp_vec = LM.linearize([q2], [q2d], [q1], [q1d],op_point=op_point, A_and_B=True)
```

```python
A
```

$$\left[\begin{matrix}0 & 1\\- \frac{g}{L} & 0\end{matrix}\right]$$

```python
B
```

$$\left[\right]$$

以示之！！