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【二・一】

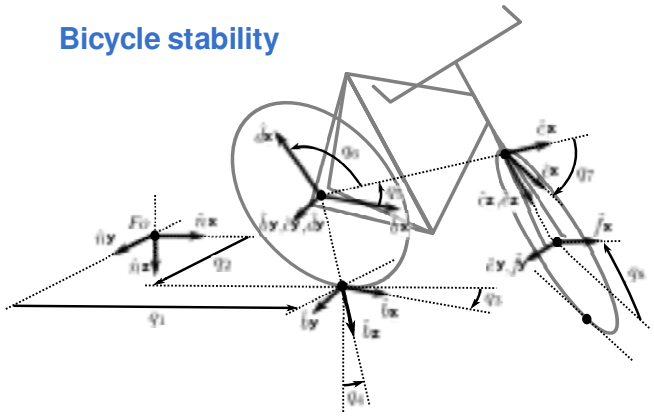
2018-03-26 | 懸鉤子 | 發表迴響

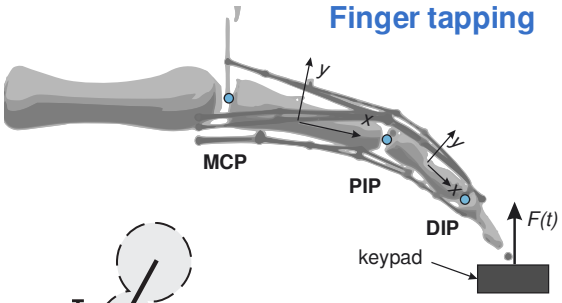
如果還沒讀過

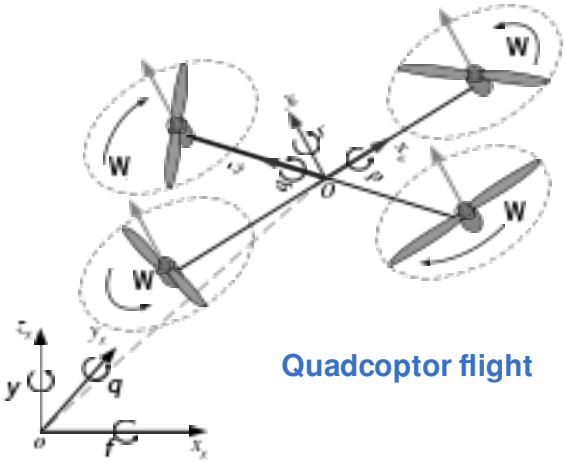
- [14 min] n01_dynamics_overview.ipynb

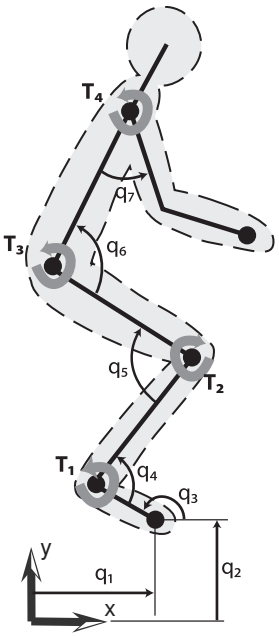
An overview of rigid body dynamics

Rigid body dynamics is concerned with describing the motion of systems composed of solid bodies; such as vehicles, skeletons, robots [1-4]:









This document borrows heavily from [5, 6].

是時候了！

此處擷取其中一段文本，講兩個『參考系定位』之『觀點約定』：

Reference frames

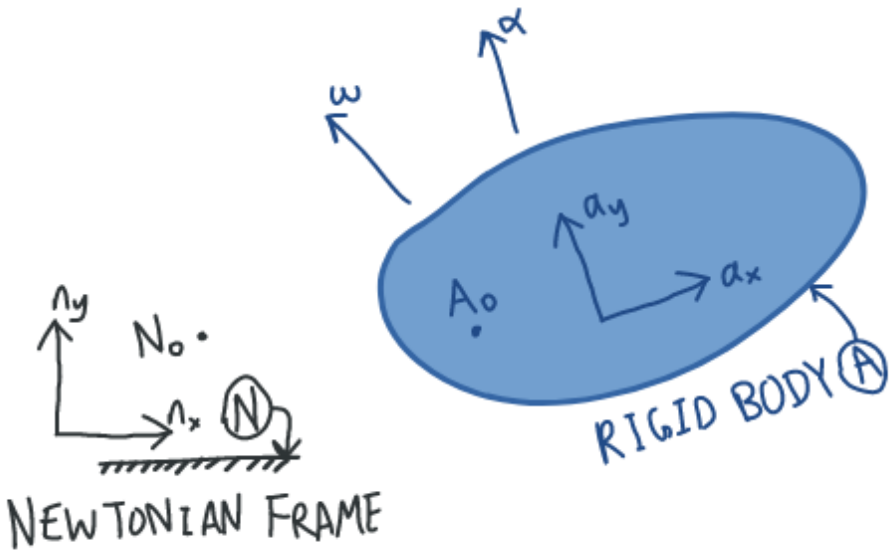
A reference frame (or simply, frame) is a rigid 3D object. We always attach a reference frame to rigid bodies in order to describe their motion. We may also use "empty" reference frames to make a system easier to model.

A reference frame has some *location* in space, but it does *not* have a position. Reference frames contain points, and those *points* have positions.

A reference frame also has an *orientation* in space. To specify its orientation, we choose a vector basis whose orientation is fixed with respect to the reference frame (but there are infinitely many vector bases we *could* label on the frame). In general, we are only interested in the vector bases we attach to reference frames; from here on, we will instead refer to reference frames in the places where we referred vector bases. That is, we express vectors in a reference frame instead of in a vector basis.

A reference frame's location and orientation vary in time. Two important attributes of a reference frame are its **angular velocity** $\vec{\omega}$ and its **angular acceleration** $\vec{\alpha}$; we'll describe these shortly.

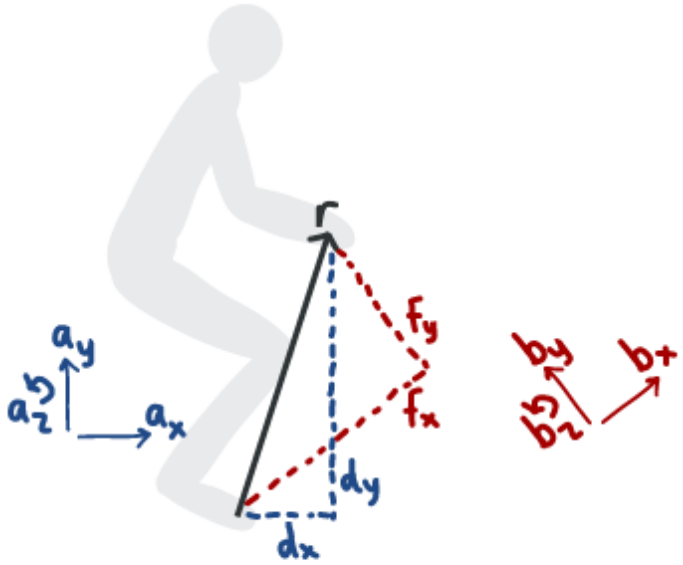
A **Newtonian reference frame** is one in which Newton's second law holds.



Expressing vectors with a vector basis

We have shown you what a vector $\vec{\mathbf{v}}$ looks like, but have yet to express an actual vector mathematically. To do so, we first choose three unit vectors $\hat{\mathbf{a}}_x$, $\hat{\mathbf{a}}_y$, and $\hat{\mathbf{a}}_z$ whose directions we accept as given. Consider the human jumper from above; we choose:

- $\hat{\mathbf{a}}_x$ to point forward,
- $\hat{\mathbf{a}}_y$ to point upwards,
- $\hat{\mathbf{a}}_z$ to point out of the plane (to the subject's right).



These three unit vectors are mutually perpendicular. For pratical reasons, we will always make sure that's the case. If so, the three vectors define a vector basis. We can express the position of the subject's hand from its toes in terms of these three vectors:

$$\vec{\mathbf{r}} = d_x\hat{\mathbf{a}}_x + d_y\hat{\mathbf{a}}_y + 0\hat{\mathbf{a}}_z$$

We call d_x the **measure** of $\vec{\mathbf{r}}$ along $\hat{\mathbf{a}}_x$, and it is equal to $\vec{\mathbf{r}} \cdot \hat{\mathbf{a}}_x$. Note that a vector basis does not have an origin.

We could have chosen a different vector basis, such as $\hat{\mathbf{b}}_x$, $\hat{\mathbf{b}}_y$, $\hat{\mathbf{b}}_z$. Then, we would express $\vec{\mathbf{r}}$ as:

$$\vec{\mathbf{r}} = f_x\hat{\mathbf{b}}_x + f_y\hat{\mathbf{b}}_y + 0\hat{\mathbf{b}}_z$$

Using this alternative vector basis does not change the fact that $\vec{\mathbf{r}}$ is the position of the hand from the toes; it simply changes how we *express* this quantity. It is possible to express a single vector in infinitely many ways, since we can choose to use any valid vector basis. In the next section, we will learn how to relate different vector bases to each other.

Operating on vectors expressed in a basis

Once we express a vector in a vector basis, it is easy to perform operations on it with vectors expressed in the same basis. Consider the two vectors:

- $\vec{\mathbf{a}} = a_x\hat{\mathbf{n}}_x + a_y\hat{\mathbf{n}}_y + a_z\hat{\mathbf{n}}_z$
- $\vec{\mathbf{b}} = b_x\hat{\mathbf{n}}_x + b_y\hat{\mathbf{n}}_y + b_z\hat{\mathbf{n}}_z$

Here are the addition, dot, and cross operations between these two vectors:

$$\vec{\mathbf{a}} + \vec{\mathbf{b}} = (a_x + b_x)\hat{\mathbf{n}}_x + (a_y + b_y)\hat{\mathbf{n}}_y + (a_z + b_z)\hat{\mathbf{n}}_z$$

$$\vec{\mathbf{a}} \cdot \vec{\mathbf{b}} = a_x b_x + a_y b_y + a_z b_z$$

$$\vec{\mathbf{a}} \times \vec{\mathbf{b}} = \det\begin{bmatrix} \hat{\mathbf{n}}_x & \hat{\mathbf{n}}_y & \hat{\mathbf{n}}_z \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{bmatrix}$$

We must specify a vector basis

When a vector is expressed in typical linear algebra notation, information is lost. For example, we don't know the basis in which the following vector is expressed:

$$\vec{\mathbf{v}} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

If we don't know the basis in which v_x, v_y, and v_z are its measures, we cannot add $\vec{\mathbf{v}}$ to another vector, etc. To express a vector in matrix form, we must carry along the basis in which it is expressed. One option for doing so is the following:

$$[\vec{\mathbf{v}}]_n = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}_n$$

The notation $[\vec{\mathbf{v}}]_n$ specifies that $\vec{\mathbf{v}}$ is expressed in the vector basis $\hat{\mathbf{n}}_x$, $\hat{\mathbf{n}}_y$, $\hat{\mathbf{n}}_z$.

```
from sympy.abc import c, d, e, f, g, h, theta
from sympy.physics.vector import ReferenceFrame, dot, cross
```

```
A = ReferenceFrame('A')
```

```
B = A.orientnew('B', 'Axis', (theta, A.z))
```

```
a = c * A.x + d * A.y + e * A.z
b = f * B.x + g * B.y + h * B.z
```

```
a + b
```

$$c\hat{\mathbf{a}}_\mathbf{x} + d\hat{\mathbf{a}}_\mathbf{y} + e\hat{\mathbf{a}}_\mathbf{z} + f\hat{\mathbf{b}}_\mathbf{x} + g\hat{\mathbf{b}}_\mathbf{y} + h\hat{\mathbf{b}}_\mathbf{z}$$

```
dot(a, b)
```

$$c\left(f\cos\left(\theta\right) - g\sin\left(\theta\right)\right) + d\left(f\sin\left(\theta\right) + g\cos\left(\theta\right)\right) + eh$$

```
cross(a, b)
```

$$\left(-eg + h\left(-c\sin(\theta) + d\cos(\theta)\right)\right)\hat{\mathbf{b}}_\mathbf{x} + \left(ef - h\left(c\cos(\theta) + d\sin(\theta)\right)\right)\hat{\mathbf{b}}_\mathbf{y} + \left(-f\left(-c\sin(\theta) +\right.\right.$$

```
(a+b).express(A)
```

$$(c + f\cos(\theta) - g\sin(\theta))\hat{\mathbf{a}}_\mathbf{x} + (d + f\sin(\theta) + g\cos(\theta))\hat{\mathbf{a}}_\mathbf{y} + (e + h)\hat{\mathbf{a}}_\mathbf{z}$$

Rotation matrices (direction cosine matrices)

In almost every problem, we make use of multiple vector bases. The reason is that there is usually a particular basis in which a vector is most conveniently expressed. And, that convenient basis is usually not the same for all vectors we'll deal with. A side effect is that we will often want to change the basis in which a vector is expressed. To do so, we use a rotation matrix (also called a direction cosine matrix). The rotation matrix ${}^aR^b$ allows us to take a vector $\vec{\mathbf{v}}$ expressed in $\hat{\mathbf{b}}_x$, $\hat{\mathbf{b}}_y$, $\hat{\mathbf{b}}_z$ and re-express it in $\hat{\mathbf{a}}_x$, $\hat{\mathbf{a}}_y$, $\hat{\mathbf{a}}_z$:

$$[\vec{\mathbf{v}}]_a = {}^aR^b\,[\vec{\mathbf{v}}]_b$$

The rotation matrix is given by dot products across the two the vector bases:

$${}^aR^b = \begin{bmatrix} \hat{\mathbf{a}}_x \cdot \hat{\mathbf{b}}_x & \hat{\mathbf{a}}_x \cdot \hat{\mathbf{b}}_y & \hat{\mathbf{a}}_x \cdot \hat{\mathbf{b}}_z \\ \hat{\mathbf{a}}_y \cdot \hat{\mathbf{b}}_x & \hat{\mathbf{a}}_y \cdot \hat{\mathbf{b}}_y & \hat{\mathbf{a}}_y \cdot \hat{\mathbf{b}}_z \\ \hat{\mathbf{a}}_z \cdot \hat{\mathbf{b}}_x & \hat{\mathbf{a}}_z \cdot \hat{\mathbf{b}}_y & \hat{\mathbf{a}}_z \cdot \hat{\mathbf{b}}_z \end{bmatrix}$$

Because of the nature of vector bases, this matrix is symmetric and orthogonal. If we instead have a vector in basis a and want to express it in b, we can simply use the inverse of ${}^aR^b$. Since the matrix is orthogonal, its inverse is the same as its transpose.

$${}^bR^a = ({}^aR^b)^{-1} = ({}^aR^b)^T$$

$$[\vec{\mathbf{v}}]_b = {}^bR^a\,[\vec{\mathbf{v}}]_a$$

$$[\vec{\mathbf{v}}]_b = ({}^aR^b)^T\,[\vec{\mathbf{v}}]_a$$

The columns of ${}^aR^b$ are the unit vectors $\hat{\mathbf{b}}_x$, $\hat{\mathbf{b}}_y$, $\hat{\mathbf{b}}_z$ expressed in a:

$${}^aR^b = \begin{bmatrix} [\hat{\mathbf{b}}_x]_a & [\hat{\mathbf{b}}_y]_a & [\hat{\mathbf{b}}_z]_a \end{bmatrix}$$

Successive rotations

We'll usually need to re-express a vector multiple times. Luckily, we can do so by multiplying rotation matrices together:

$$
\begin{aligned}
{}^{d}R^{a} &= ({}^{d}R^{c})({}^{c}R^{b})({}^{b}R^{a}) \\
[\vec{\mathbf{v}}]_{d} &= {}^{d}R^{a}[\vec{\mathbf{v}}]_{a} \\
[\vec{\mathbf{v}}]_{d} &= ({}^{d}R^{c})({}^{c}R^{b})({}^{b}R^{a})[\vec{\mathbf{v}}]_{a}
\end{aligned}
$$

A point of confusion: rotating vs. re-expressing

Sometimes, rotation matrices are used to rotate vectors; that is, cause the vector to point somewhere different. That is NOT how we are using rotation matrices here. Rotating a vector changes the vector itself, while we are only changing how the *same* vector is expressed.

```
B.dcm(A)
```

$$
\begin{bmatrix}
\cos(\theta) & \sin(\theta) & 0 \\
-\sin(\theta) & \cos(\theta) & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

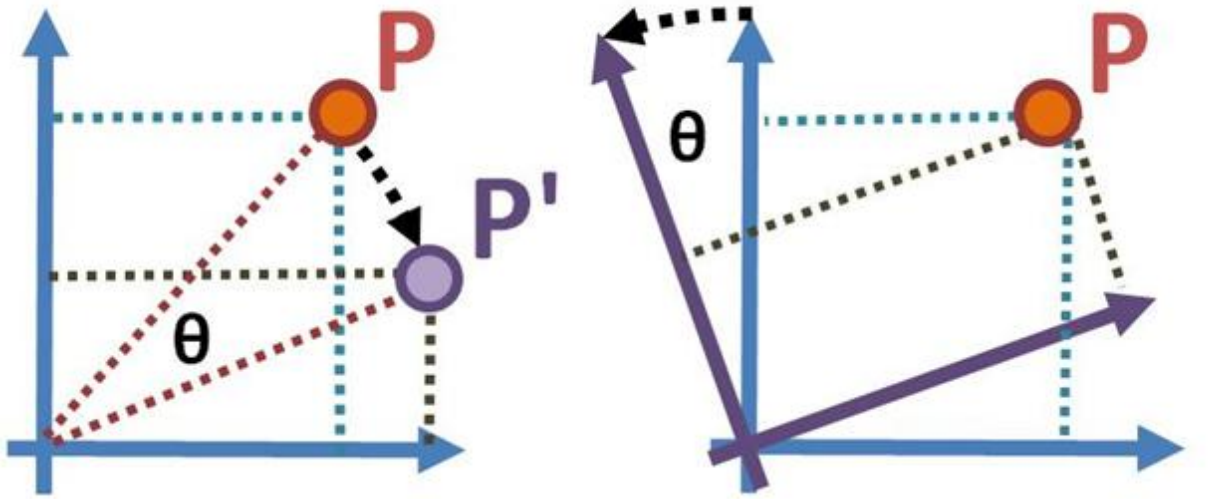
否則困惑恐生也？

Active and passive transformation

In physics and engineering, an **active transformation**, or **alibi transformation**,[1] is a transformation which actually changes the physical position of a point, or rigid body, which can be defined even in the absence of a coordinate system; whereas a **passive transformation**, or **alias transformation**,[2] is merely a change in the coordinate system in which the object is described (change of coordinate map, orchange of basis). By default, by *transformation*, mathematicians usually refer to active transformations, while physicists and engineers could mean either.

Put differently, a *passive* transformation refers to description of the *same* object in two different coordinate systems.[3] On the other hand, an *active transformation* is a transformation

of one or more objects with respect to the same coordinate system. For instance, active transformations are useful to describe successive positions of a rigid body. On the other hand, passive transformations may be useful in human motion analysis to observe the motion of the tibia relative to the femur, that is, its motion relative to a (*local*) coordinate system which moves together with the femur, rather than a (*global*) coordinate system which is fixed to the floor.[3]



In the active transformation (left), a point moves from position P to P' by rotating clockwise by an angle θ about the origin of the coordinate system. In the passive transformation (right), point P does not move, while the coordinate system rotates counterclockwise by an angle θ about its origin. The coordinates of P' in the active case (that is, relative to the original coordinate system) are the same as the coordinates of P relative to the rotated coordinate system.

In [1]:
```
%matplotlib inline
from sympy import *
from sympy.physics.mechanics import *
init_printing(use_latex='mathjax', pretty_print=False)
```

In [2]:
```
from sympy.abc import a, b, c, d, e, f, theta
```

In [3]:
```
A = ReferenceFrame('A')
B = A.orientnew('B', 'Axis', (theta, A.z))
```

In [4]:
```
vector = a * A.x + b * A.y + c * A.z
vector
```

Out[4]:

$$a\hat{\mathbf{a}}_{\mathbf{x}} + b\hat{\mathbf{a}}_{\mathbf{y}} + c\hat{\mathbf{a}}_{\mathbf{z}}$$

In [5]:
```
vector.express(B)
```

Out[5]:

$$(a\cos(\theta) + b\sin(\theta))\hat{\mathbf{b}}_{\mathbf{x}} + (-a\sin(\theta) + b\cos(\theta))\hat{\mathbf{b}}_{\mathbf{y}} + c\hat{\mathbf{b}}_{\mathbf{z}}$$

In [6]:
```
向量 = d * B.x + e * B.y + f * B.z
向量
```

Out[6]:

$$d\hat{\mathbf{b}}_{\mathbf{x}} + e\hat{\mathbf{b}}_{\mathbf{y}} + f\hat{\mathbf{b}}_{\mathbf{z}}$$

In [7]:
```
向量.express(A)
```

Out[7]:

$$(d\cos(\theta) - e\sin(\theta))\hat{\mathbf{a}}_{\mathbf{x}} + (d\sin(\theta) + e\cos(\theta))\hat{\mathbf{a}}_{\mathbf{y}} + f\hat{\mathbf{a}}_{\mathbf{z}}$$

In [8]:
```
vec = vector.express(B).to_matrix(A)
vec
```

Out[8]:

$$\begin{bmatrix} -\left(-a\sin\left(\theta\right)+b\cos\left(\theta\right)\right)\sin\left(\theta\right)+\left(a\cos\left(\theta\right)+b\sin\left(\theta\right)\right)\cos\left(\theta\right) \\ \left(-a\sin\left(\theta\right)+b\cos\left(\theta\right)\right)\cos\left(\theta\right)+\left(a\cos\left(\theta\right)+b\sin\left(\theta\right)\right)\sin\left(\theta\right) \\ c \end{bmatrix}$$

In [9]:
```
simplify(vec)
```

Out[9]:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}$$