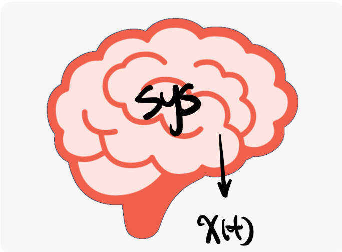


希望通过 sys 中的变量随 t 的变化情况，
来研究 sys 的 temporal dynamics.
sys 可以表示大脑中の任意 scale.

Continuous Dynamic System:

1D.

$$\frac{dx(t)}{dt} = f(x(t))$$

既然研究表征系统の变量 x 的 temporal dynamics，那自然要研究 x 随时间 t 的变化率。而这个变化率又肯定与 x 的当前值有关，即，和 f(x) 有关。

Deterministic 1D Linear Dynamic System

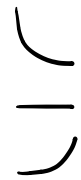
$$\frac{dx(t)}{dt} = \dot{x}(t) = a\cdot x(t) \xrightarrow[\text{Solution}]{\text{Analytic}} x(t) = x_0\cdot e^{at} \xrightarrow{\text{dynamics}}$$

$a \in \mathbb{R}$

$a<0$

$a=0$

$a>0$



$$\frac{dx(t)}{dt} = \dot{x}(t) = \lambda\cdot x(t) \xrightarrow[\text{Solution}]{\text{Analytic}}$$

$\lambda = a+bi \in \mathbb{C}$

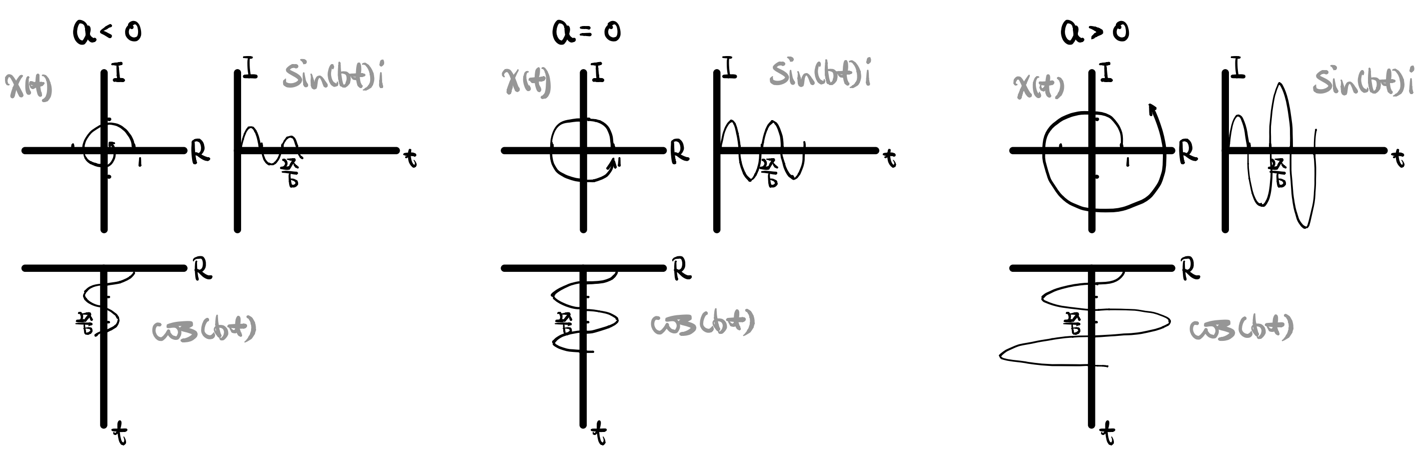
$x(t) \in \mathbb{C}$

$$\begin{aligned} x(t) &= x_0\cdot e^{\lambda t} \\ &= x_0\cdot e^{(a+bi)t} \\ &= x_0\cdot e^{at}\cdot e^{bti} \\ &= \underbrace{x_0\cdot e^{at}}_{②}\cdot \underbrace{(\cos(bt)+\sin(bt)\cdot i)}_{①} \end{aligned}$$

② 向量的模长随时间的变化

① 以 $b=2\pi f$ 的角频率在复平面逆时针旋转的模为1の向量.

dynamics



所以，这个在复空间表征的系统 x(t) 的 temporal dynamic 由 $\lambda = a+bi$ 来衡量.

a：与实数の情况相同，表示 x(t) 的幅度

b：表示 x(t) 的振荡角频率（频率 $= \frac{b}{2\pi}$）

问：在什么情况下，可以将系统用 $\dot{x} = \lambda x$ 表示？

暂时不记得了，但在电路系统中可能行。这里の关键是，当系统表示成此形式时，a, b 对系统动态の表示

2D. or n D.

$$\frac{d\vec{x}(t)}{dt} = f(\vec{x}(t)) \longrightarrow \begin{pmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{pmatrix} = \begin{pmatrix} & A & \end{pmatrix}_{2\times 2} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{2\times 1}$$

Deterministic
2D
Linear
Dynamic
System

$$\dot{\vec{x}} = A \cdot \vec{x}$$

Analytic solution

对A进行特征值分解

$$A \cdot (\vec{v}_1, \vec{v}_2) = \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} (\vec{v}_1, \vec{v}_2)$$

Behavior of this system depends on the **eigenvalues** and **eigenvectors** of **A**

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$$

Analytic solution:

$$\mathbf{x}(t) = \mathbf{v}_1^T \mathbf{x}_0 \mathbf{v}_1 \exp(\lambda_1 t) + \mathbf{v}_2^T \mathbf{x}_0 \mathbf{v}_2 \exp(\lambda_2 t)$$

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \underbrace{\vec{v}_1^T \cdot \vec{x}_0}_{\text{const 1}} \cdot e^{\lambda_1 t} \cdot \vec{v}_1 + \underbrace{\vec{v}_2^T \cdot \vec{x}_0}_{\text{const 2}} \cdot e^{\lambda_2 t} \cdot \vec{v}_2$$

$$= C_1 \cdot e^{\lambda_1 t} \cdot \vec{v}_1 + C_2 \cdot e^{\lambda_2 t} \cdot \vec{v}_2$$

dynamics

将 $\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ 的 dynamics 看作是向量在 2D 空间中的变化。由公式可知, $\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ 的变化可以看作是在基 $\vec{v}_1$, $\vec{v}_2$ 上投影变化的"总效果".

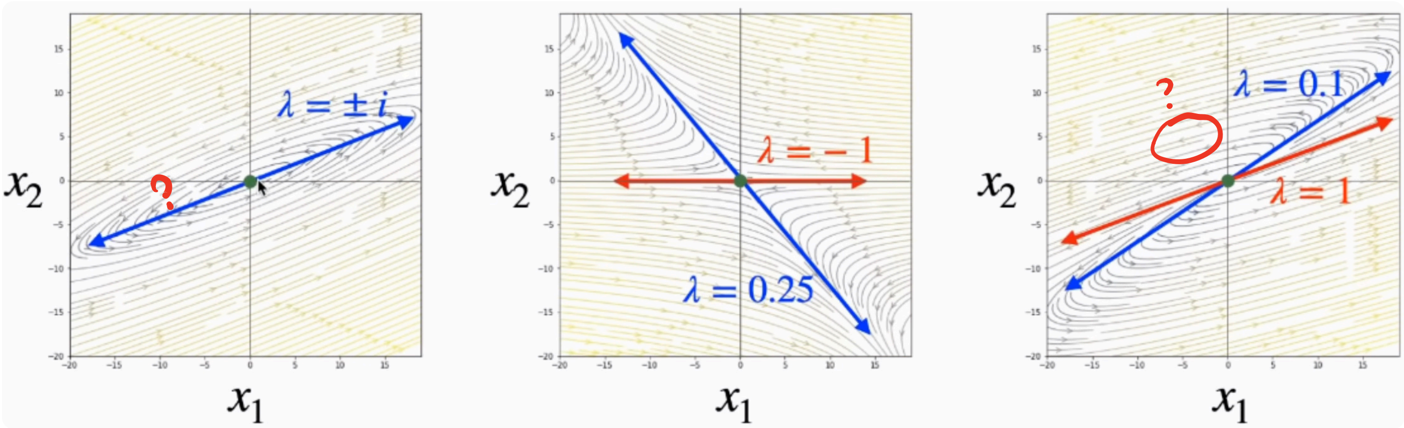
$\lambda_1 > 0$, 则 $\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ 在 $\vec{v}_1$ 的投影 $c_1 \cdot e^{\lambda_1 t}$ 呈增加趋势 $(e^{\lambda_1 t} > 1)$

$\lambda_1 = 0$, $c_1 \cdot e^{\lambda_1 t}$ 呈平稳趋势 $(e^{\lambda_1 t} = 1)$

$\lambda_1 < 0$, $c_1 \cdot e^{\lambda_1 t}$ 呈减小趋势 $(e^{\lambda_1 t} < 1)$

(λ_2 同理)

$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$ 的 dynamics 由 $\vec{v}_1$ $\vec{v}_2$ 基上的总效果来确定.

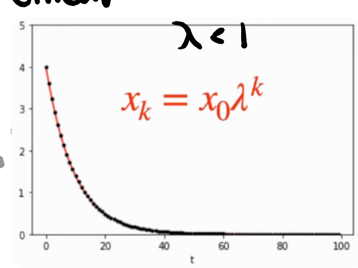


Discrete Dynamic System:

① k时刻值 唯一の决定 k+1时刻值 , deterministic & linear

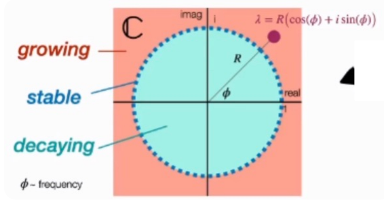
当$\lambda \in R$:

$$x_{k+1} = \lambda \cdot x_k \xrightarrow[\text{solution}]{\text{analytic}} x_{k+1} = x_0 \cdot \lambda^k \xrightarrow{\text{dynamics}}$$



当 $\lambda = a+bi \in \mathbb{C}$:

$$x_{k+1} = \lambda \cdot x_k \xrightarrow[\text{solution}]{\text{analytic}} x_{k+1} = x_0 \cdot \lambda^k \xrightarrow{\text{dynamics}}$$



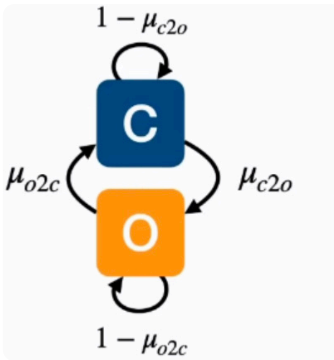
$\|\lambda\| > 1$，随着 $k\uparrow$，$\|\lambda^k\|\uparrow$，系统无限制增大.

② $k, k-1, \cdots, k-n$ 时刻值 唯一的决定 $k+1$ 时刻状态，deterministic & linear

$$x_{k+1} = \lambda_1 x_k + \lambda_2 x_{k-1} + \cdots + \lambda_{n+1} x_{k-n}$$

在 tutorial 中，这种情况出现在 autoregression 中.

③



※ *State Vector **x** at timestep k:*
Probability of
being in C or O state

$$\begin{bmatrix} C \\ O \end{bmatrix}_{k+1} = \mathbf{A} \begin{bmatrix} C \\ O \end{bmatrix}_k = \begin{bmatrix} 1-\mu_{c2o} & \mu_{o2c} \\ \mu_{c2o} & 1-\mu_{o2c} \end{bmatrix} \begin{bmatrix} C \\ O \end{bmatrix}_k$$

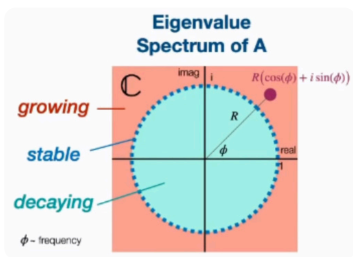
the present state determines the probability of transitions, which means that transitions to the next state are not deterministic, in fact they have some element of chance to it. All we can do is specify the probability, but we can't specify exactly what's going to happen next.

This is our probabilistic model of closing and opening ion channels, and take the eigendecomposition of this 'A' matrix here, what you're going to get are, because this is a two by two system, you're going to get two eigenvalues and two eigenvectors. Okay? **Your goal is to discover which of these eigenvalues corresponds to the stable solution.** In other words, the one that equals one. **And if you look at the corresponding eigenvector that tells you something about the direction in which this solution is stable.**

$$\mathbf{AV} = \mathbf{VD} \qquad \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \qquad \mathbf{V} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$$

Take eigendecomposition of A — **Eigenvalues** — **Eigenvectors**

见 Tutorial 2



注意！ 虽然这里公式与 tutorial 1 中的 two-dimensional deterministic linear dynamical system 形式看起来相同，但是两者的含义是不同的！

- **two-dimensional deterministic linear dynamical system** 中 x1 x2 是**不同的**变量在 t 时刻的确定值。这两个值通过线性变换（两个变量交互），唯一的决定了该时刻两个变量的变化率。这是一个由当前的状态得出接下来状态的概率分布。这是 deterministic、同时 linear 的过程。

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix} a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

- 在本节中，[C,O]_k.Transpose = S_k 表示系统在 k 时刻**状态 S** 的两种可能性。其通过状态转换矩阵 A 得出的，是不确定的 k+1 时刻**状态 S** 的两种可能性。这是一个 non-deterministic、但 linear 的过程。

$$\begin{bmatrix}C\\ O\end{bmatrix}_{k+1} = \mathbf{A}\begin{bmatrix}C\\ O\end{bmatrix}_k = \begin{bmatrix}1-\mu_{c2o} & \mu_{o2c}\\ \mu_{c2o} & 1-\mu_{o2c}\end{bmatrix}\begin{bmatrix}C\\ O\end{bmatrix}_k$$

注意！ 这节的关键是引出了离散的线性动态系统。虽然上面的例子是 non-deterministic，但 discrete linear system 完全可以是 deterministic。即，

- k 时刻的状态通过线性权重唯一的决定了 k+1 时刻的状态；

$$s_{k+1} = a \cdot s_k$$

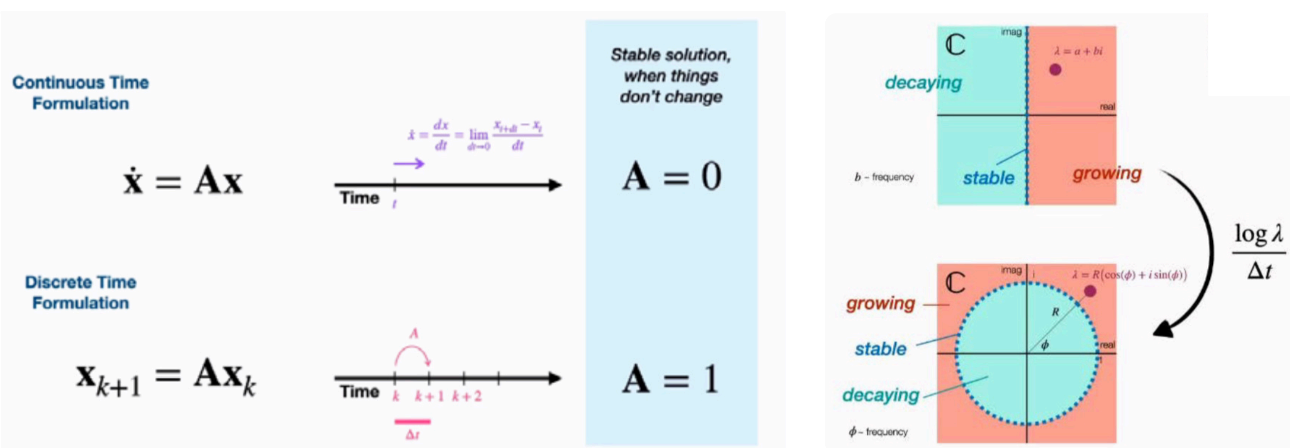
- k 时刻的不同变量的状态通过状态转移矩阵 A 唯一的决定了 k+1 时刻的状态；

$$\begin{bmatrix}s_1\\ s_2\end{bmatrix}_{k+1} = [A]\cdot\begin{bmatrix}s_1\\ s_2\end{bmatrix}_k \text{ or } S_{k+1} = [A]\cdot S_k$$

- k 时刻及之前若干时刻的状态唯一的决定了 k+1 时刻的状态；

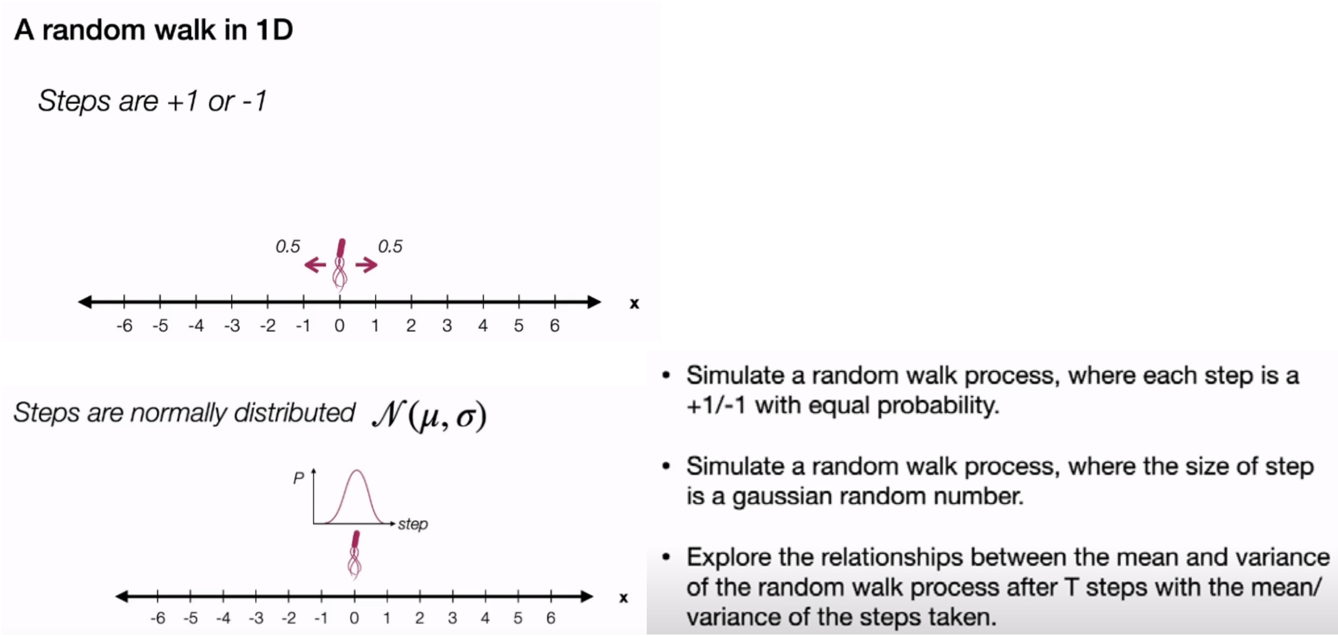
$$S_{k+1} = a_1\cdot S_k + a_2\cdot S_{k-1} + a_3\cdot S_{k-2} + a_4\cdot S_{k-3}$$

Continuous & Discrete Linear System 比较



TUTORIAL 3

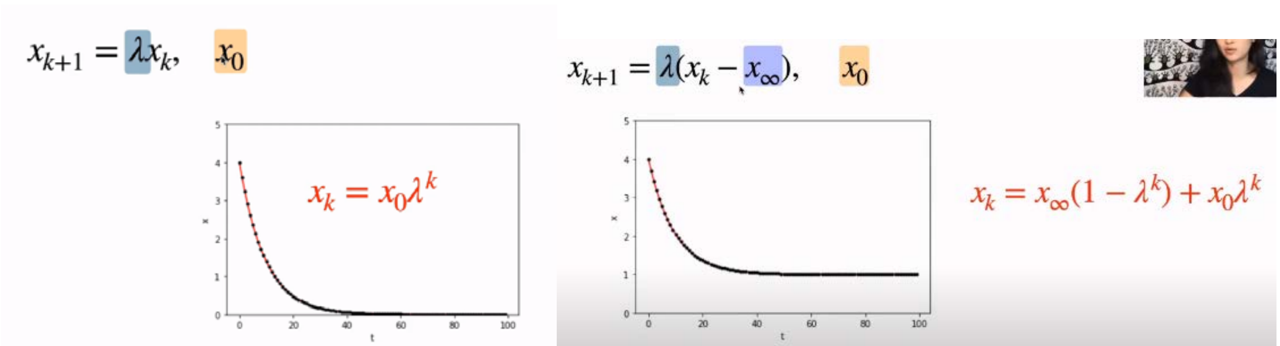
Ornstein-Uhlenbeck, OU process, that has aspects of **deterministic as well as stochastic** dynamical systems. So it's really exciting to be putting them together.



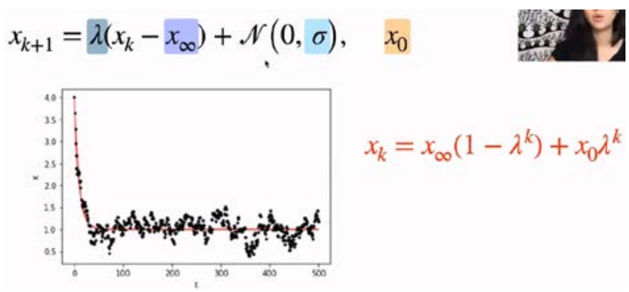
---

in particular what you should have seen is that the variance of this distribution of solutions increases as a function of time in a way that is proportional to the variance of the underlying steps that you're taking.???

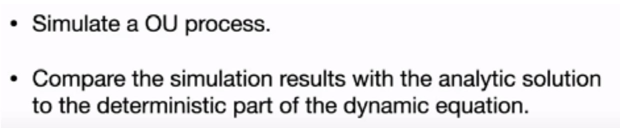
If you glue it together, this random walk process with the deterministic dynamical system exactly like the type that we did in tutorial 1, what you end up with is an Ornstein-Uhlenbeck process that has aspects of this deterministic behavior with a little bit of randomness injected to it. **This OU process is a really popular model for many things. Perhaps most relevantly in neuroscience, it has been useful for modeling decision making and short-term memory.**



What we're going to do is just add one more term to this equation, that is not deterministic. In particular. we're going to specify another term here where the mapping from xk to xk plus 1, has a bit of process noise. That is a Gaussian random number with mean, 0 and standard deviation, sigma.



注意：上图中，红色的公式都是 analytic solution。

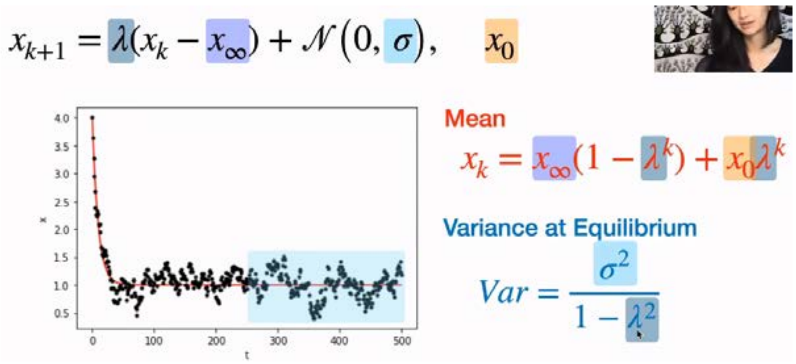


What you saw in the previous part of this tutorial, is that **the mean of this OU process follows the analytic solution for the deterministic part of it.** This is what I'm plotting as the red line here.

What about the variance? Well as it turns out, unlike **a random walk process, where the variance grows basically without bounds for large times T.**

In this particular case, we have two processes that are competing with each other. And so the random walk process is trying to increase the variance of the solution and it does. But then, we have this deterministic part, this lambda here, which is less than one, that's trying to bring it back. It has a decaying effect, right? So it's trying to bring it back towards x infinity. And so the balance of these two respective components, these two parameters, determines the variance of the ultimate solution for the OU process.

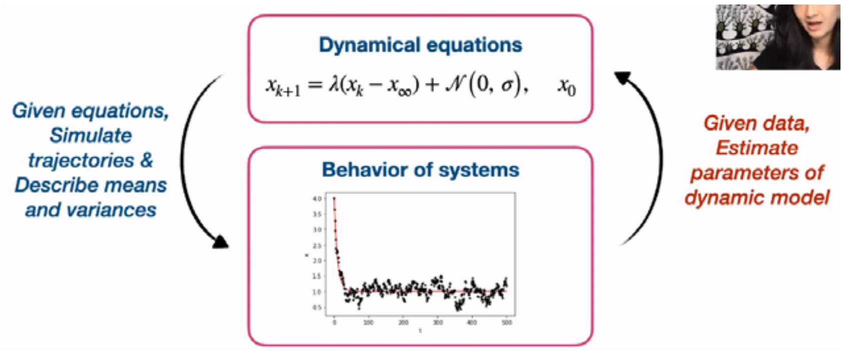
In particular, we can write the variance of the solution at equilibrium(平衡) after a long time, as the balance between sigma, which is the standard deviation of the random walk process, and lambda, which is this decaying process that is trying to bring it back together.



We looked at a bunch of different types of dynamical equations. These are governing equations that tell us what happens in time as the state of the system goes from one step to the next. And we've looked at a bunch of these types of dynamical equations, and **we simulated and looked at the behaviors of systems by describing their trajectories in time and by looking at how their means and variances change in time.** Okay, so that's great. And that's really great for especially for building intuitions of the kinds of behaviors that you can possibly get out of these types of equations. **But what happens if you don't have dynamical equations, but instead you have data?** Okay, so it's actually really relevant especially for experimental neuroscience. You don't start with governing equations of the brain, you start with data. And so **if you start with data and you have at least some notion of the type of dynamical equations that might be governing that behavior, what you want to do is the opposite. You don't want to simulate the system, you want to estimate the parameters of the dynamic model if you've given data,** and so that's the topic of this last tutorial.

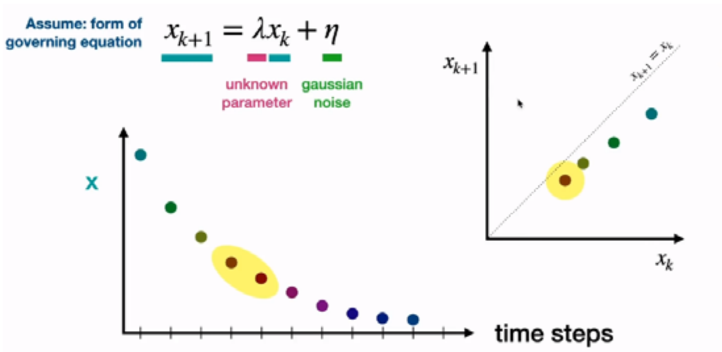
之前三节是在已知动态系统的公式及参数的情况下，对系统随时间的表现进行研究；
但在实际的实验中，刚好是反过程！我们有系统随时间的表现数据，以及对系统公式的假设，但是我们不知到公式的参数！这就是一个机器学习问题了！如何由动态系统的数据，来去估计动态系统的参数?

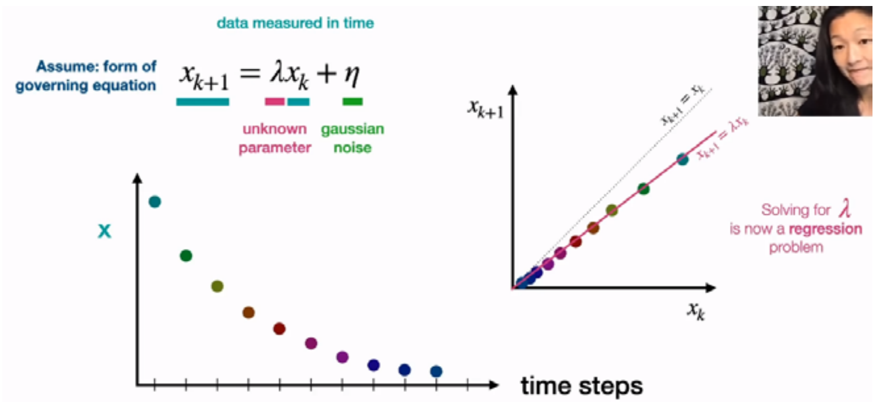
TUTORIAL 4，参数估计

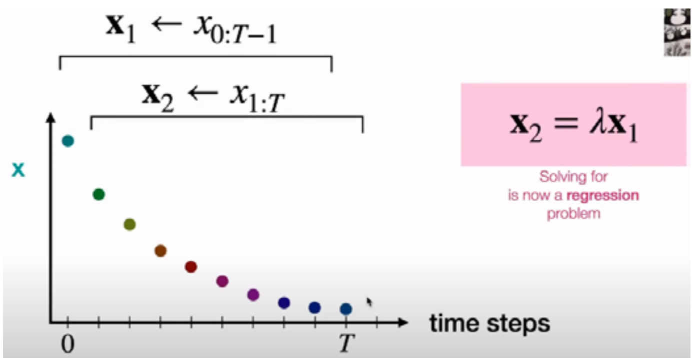


let's go back to the same equation that we talked about in tutorial 3. Where we have something that is a dynamical equation where there's X at time k plus 1 is some parameter times X of k, and there's some noise involved. Now this noise for now I'm not going to think it's not that important, but you can kind of just assume that it's there.

1) 情况一：k 时刻的状态通过线性权重唯一的决定了 k+1 时刻的状态







So now we can solve this equation here as a regression problem, and the solution should give us the underlying governing equation for the coefficient lambda.

2）情况二：k 时刻及之前若干时刻的状态唯一的决定了 k+1 时刻的状态