A Note on Control Theory

Yixuan Li

March 2024

1 Introduction

In simple words, control theory is when you want to control certain physical quantity (let's say temperature of your room or the speed of your car) near certain value.

The simplest form of control is open-loop control. For example, if you want your car to maintain a certain speed, you just need to press the gas pedal to a specific angle. In Figure 2, u is the angle of the accelerator pedal and x is the speed of the car. But if you want your car to move at a given speed, say, 30 km/h, you may refer to close-loop control.

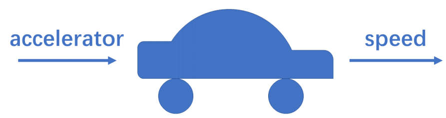

Figure 1: Human Driving

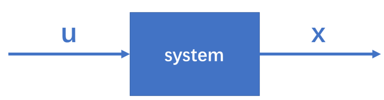

Figure 2: open loop control

Close-loop control is a little more complicated. As shown in Figure 3, the output signal x is transmitted back to the system. Imagine that you want to let your car moving at 30 km/h, you had better look at the speedometer and change the angle of the accelerator pedal according to that.

To make things more clear, we introduce the concept of "controller", shown in Figure 4. In human driving, the controller is the driver while in auto driving, the controller is the machine. Attention! Using the language of reinforcement learning, the controller is the agent and the system is part of the environment. Remember Yukun Cheng once talked about a cyber C.elegans. In that paper, the computer is the controller and the agent while the worm is the system, part of the environment. The math form of Figure 4 is Equation 10.

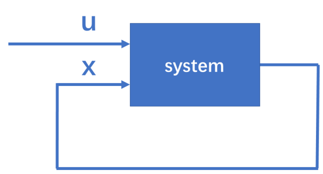

Figure 3: close loop control

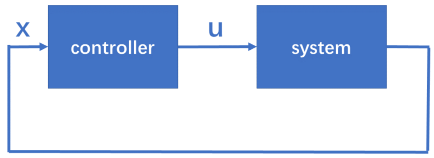

Figure 4: adding controller

In some cases, like fluid mechanics or neuroscience, x has very high dimension, even 10^4. We don't have the time and money to measure every component of x. To figure this out, scientists will just measure part of x. In Figure 4, y is the partial observation of x and $\hat{x}$ is the inferred x by the estimator. Usually, $dim(y) << dim(x)$. In the example of the pendulum, section 4, the Kalman filter will be used as the estimator while the linear quadratic regulator (LQR) will served as the controller. The math form of Figure 5 is Equation 13.

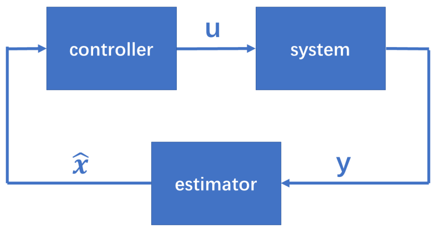

Figure 5: adding estimator

The latter part is organized as follows: section 2 is about the evolution of a system without control. section 3 will tell you how to add the control signal to make the system stabilized at an initially unstable location. Then, section 4 will use the inverted pendulum as an engaging example, which has enormous videos on YouTube (e.g.:Inverted Single Pendulum, Inverted Triple Pendulum). In section 5, we will introduce the concept of data-driven-control and the Willems fundamental lemma. Next, we will talk about how to view a neuron as a controller in section 6, which can be skipped if you don't care about neuroscience. Last but not least, a story between James Clerk Maxwell and the control theory is told in section 7.

2 Without Control

We start with a dynamic system

$$\dot{X} = f(X) \tag{1}$$

, where $f(X)$ is a non-linear function $\mathbb{R}^n \to \mathbb{R}^n$ and X(t) is $\mathbb{R} \to \mathbb{R}^n$.

How do we analyze the behavior of this system? Is it stable or unstable? Usually, a system has 2 kinds of stable state, one is every value keeping the same, the other is periodical change. You see the first kind in a dead neuron, and the second kind in a persistently firing neuron. Mathematicians call the first *fixed point* and the second *limit cycle*. Throughout this note, I will limit myself to the first kind. You may refer to Yukun Cheng's note on the chaos control theory of for the second kind.

When will every value keep the same? Obviously, this means

$$\dot{X} = 0 \tag{2}$$

Looking at (1), we get

$$f(X) = 0 \tag{3}$$

We denote $\bar{X}$ which makes (2) standing, that is, the fixed point of the system.

A fixed point is called *stable* if when the system is slightly deviated from this point under perturbation, it will go back to the point. How do we know if a fixed point is stable or not? Unstable fixed point is like a ball on the top of the hill while stable fixed point is like a ball at the bottom of the bowl. Well, we can do a *local linearization* in n dimensions, just like we analyze the slope of the hill or the bowl in 2D or 3D.

We do a multi-variable Taylor expansion around $\bar{X}$ and only keep the first order

$$f(X) = f(\bar{X}) + \frac{Df}{Dx}|_{\bar{X}}(X - \bar{X}) \tag{4}$$

where $\frac{Df}{Dx}$ is the Jacobian Matrix.

If we set $\bar{X}$ as 0 (by translating the coordinate system) and denote $\frac{Df}{Dx}$ as A, we get a linear equation

$$f(X) = AX \tag{5}$$

Combine with (1), we get

$$\dot{X} = AX \tag{6}$$

We can prove that (see subsection 8.1 in the appendix)

$$X = \sum_{i=1}^{n} \vec{c_i} e^{\lambda_i t} \tag{7}$$

where $\vec{c_i}$ is constant vector and λ_i is the eigenvalue of A.

Obviously, the fixed point is stable if and only if

$$\forall i, Re(\lambda_i) < 0 \tag{8}$$

See subsection 8.2 for local linearization not around the fixed point and section 4 for an intuitional example using the pendulum.

3 With Control, Knowing the System

Now we have came to the really fun part.

I gonna show you that, under certain circumstance, we can make eigenvalue to be any value you like. In other words, we can make the original unstable fixed point to be stable, like what is shown in Figure 6.

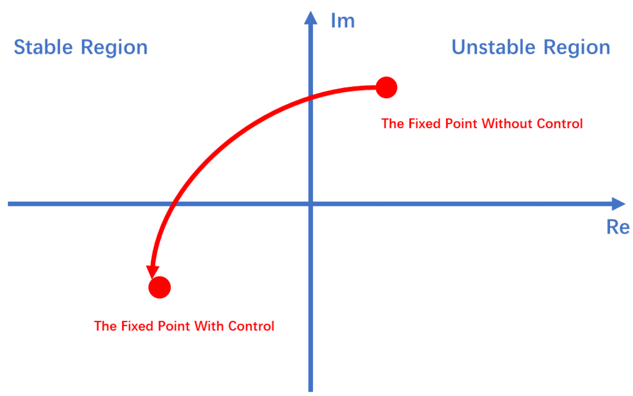

Figure 6: Drive the Fixed Point

To add control, we simply add Bu to the right hand of side of Equation 6:

$$\dot{X} = AX + Bu \tag{9}$$

where $X \in \mathbb{R}^n, u \in \mathbb{R}^q, A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times q}$.

To make things simpler, we shall not consider the estimator at the first place and we assume linear relation of u and X: $u = -KX$.

Put them together:

$$\begin{cases} \dot{X} & = & AX + Bu \\ u & = & -KX \end{cases} \tag{10}$$

where $K \in \mathbb{R}^{q\times n}$.

Equation 10 is the math form of Figure 4.

So

$$\dot{X} = (A - BK)X \tag{11}$$

Mathematicians have proved that, if we define $C := [B,\ AB,\ A^2B, \cdots, A^{n-1}B]$

$$rank(C) = n \Leftrightarrow (A-BK)\ has\ any\ eigenvalues \Leftrightarrow X\ can\ reach\ any\ point\ in\ \mathbb{R}^n \tag{12}$$

And we call the system *controllable* when $rank(C) = n$.

There is an one-line-code in MATLAB to calculate C.

```
C = ctrb(A,B)
```

Attention. In real world, A, B are usually fixed for a given system while K is not. In other worlds, you can only change K, rather than A, B.

There is an one-line-code in MATLAB to calculate K, based on eigenvalues you want and known A, B.

```
K = place(A,B,eigs_wanted)
```

There is a similar concept to controllability, observability.

Consider

$$\begin{cases} \dot{X} &= AX + Bu \\ y &= CX \\ u &= -K\hat{x} \end{cases} \tag{13}$$

where $y \in \mathbb{R}^m, C \in \mathbb{R}^{m \times n}$.

Equation 13 is the math form of Figure 5.

We call a system *observable* if:

$$rank(O) = n \tag{14}$$

where $O := [C,\ CA,\ CA^2, \cdots, CA^{n-1}]^T$.

Roughly speaking, a system is observable means that y conveys all the information of x and u. To be more precise, a system is said to be observable if and only if, for every possible trajectory of state and control vectors (x and u), the current state vector x can be estimated using only the information from outputs y. On the other hand, if the system is not observable, there are state-control trajectories that are not distinguishable by only measuring the outputs.

There is also an one-line-code in MATLAB to calculate O.

```
O = obsv(A,C)
```

4 Pendulum

4.1 Inverted Pendulum

This subsection corresponds to section 2.

A single pendulum is a stick (attention, not a rope) with a ball at the end.

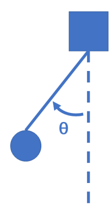

Figure 7: Single Pendulum

By the second law of Newton, we get

$$-mgsin(\theta) = ma + f \tag{15}$$

where f is the friction of the air and f is proportional to $\dot{\theta}$.

Define

$$X := \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \tag{16}$$

Then

$$\dot{X} := \begin{bmatrix} X_2 \\ -sin(X_1) - \epsilon\dot{\theta} \end{bmatrix} \tag{17}$$

where ϵ is a small positive number.

Obviously, this system has 2 fixed points, $(0, 0)$ and $(\pi, 0)$. (In math, this system has infinite fixed points, $(2k\pi, 0)$ and $(\pi + 2k\pi, 0), k \in \mathbb{Z}$, but only twp of them bring new physical contents.)

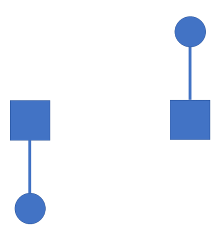

Figure 8: Two Fixed Points

The phase portrait is shown below. In Figure 9, the upper and lower trajectory represents rotation while the circle represents oscillation. In Figure 10, all trajectory will finally fall into fixed points at $(2k\pi, 0), k \in \mathbb{Z}$

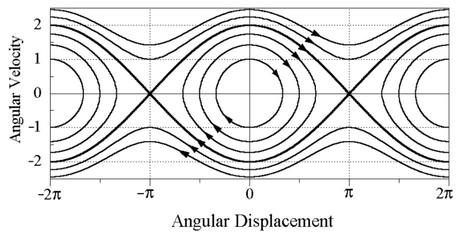

Figure 9: Phase Portrait without Friction[1]

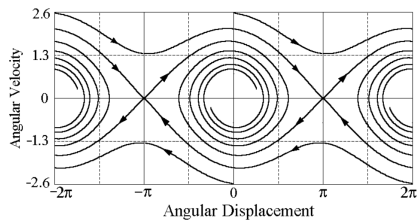

Figure 10: Phase Portrait with Friction[1]

Now, armed with the math tool of section 2, we are ready to analyze the stability of these two fixed points. We denote:

$$\bar{X}_1 = \begin{bmatrix} 0 \\ 0, \end{bmatrix} \bar{X}_2 = \begin{bmatrix} \pi \\ 0 \end{bmatrix} \tag{18}$$

The Jacobian matrix is:

$$\frac{Df}{DX} = \begin{bmatrix} 0 & 1 \\ -cos(X_1) & -\epsilon \end{bmatrix} \tag{19}$$

Then, we implement local linearizations:

$$\frac{Df}{DX}|_{\bar{X}_1} = \begin{bmatrix} 0 & 1 \\ -1 & -\epsilon \end{bmatrix}, \frac{Df}{DX}|_{\bar{X}_2} = \begin{bmatrix} 0 & 1 \\ +1 & -\epsilon \end{bmatrix} \tag{20}$$

It is obvious that, when $\epsilon = 0$, both fixed points are unstable, and when $\epsilon > 0$, $\bar{X}_1$ is stable while $\bar{X}_2$ is unstable.

4.2 Inverted Pendulum on a Cart

This subsection corresponds to section 3.

The codes of this section can be got from here.

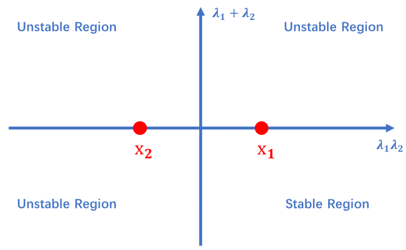

Figure 11: Without Friction

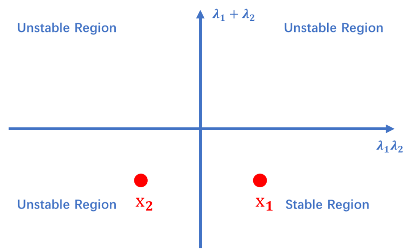

Figure 12: With Friction

4.2.1 Drive the Eigenvalue

Is there a way to make $\bar{X}_2$ stable? Imagine that you use your hands to control the system. When the ball falls to the right, just move your hands slightly to the right. The following graph is a formalized way of it.

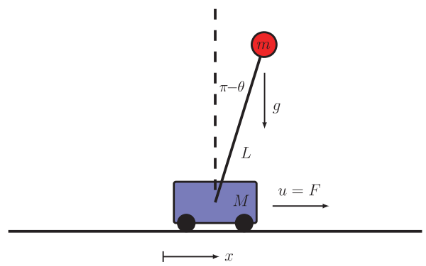

Figure 13: Inverted Pendulum on a Cart[2]

The ODEs of the system is exactly Equation 10, where

$$X = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}, A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\delta}{M} & \frac{bmg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{b\delta}{ML} & -\frac{b(m+M)g}{ML} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{b}{ML} \end{bmatrix} \tag{21}$$

$b = 1$ for the up fixed point, and $b = -1$ for the down fixed point.

You can check the controllability and observability of this system using MATLAB.

```
%% parameters
m = 1;
M = 5;
L = 2;
g = -10;
d = 10; % friction
b = -1; % pendulum up

%% write A B C
A = [0 1 0 0;
    0 -d/M -m*g/M 0;
    0 0 0 1;
    0 -b*d/(M*L) -b*(m+M)*g/(M*L) 0];

eig(A)

B = [0; 1/M; 0; b*1/(M*L)];

C = eye(4);

rank(ctrb(A,B))  % is it controllable?
rank(obsv(A,C))  % is it observable?
```

Then, you can use the built-in function ‘place‘ to drive the eigenvalue to anywhere you like.

```
% p is a vector of desired eigenvalues
p = [-.01; -.02; -.03; -.04]; % not enough
% p = [-.3; -.4; -.5; -.6];  % just barely
% p = [-1; -1.1; -1.2; -1.3]; % good
% p = [-2; -2.1; -2.2; -2.3]; % aggressive
% p = [-3; -3.1; -3.2; -3.3]; % aggressive
% p = [-3.5; -3.6; -3.7; -3.8]; % breaks
K = place(A,B,p);
```

4.2.2 Linear Quadratic Regulator

If you have experimented with eigenvalues, you will find that the smaller the eigenvalues are, the faster the system returns to the fixed point, but the more vulnerable the system becomes.

In general, you can't have both things at the same time. There is a trade-off between the robustness of the system and the time/energy required by the controller. The Linear Quadratic Regulator (LQR) was developed to find this trade-off point.

We define a lost function

$$J := \int_0^{+\infty} (X^T Q X + u^T R u) dt \tag{22}$$

where Q is $\mathbb{R}^{n\times n}$ and R is $\mathbb{R}^{q\times q}$. Q measures the robustness while R measures the control cost.

The intuition of this function is just like enormous quadratic forms in machine learning: it is much easier to calculate derivatives on the quadratic form.

In the pendulum example, we may set:

```
Q = [1 0 0 0;
     0 1 0 0;
     0 0 10 0;
     0 0 0 100];
R = .0001;
```

$Q = diag(1, 1, 10, 100)$ means the deviance of $\dot{\theta}$ is the most unbearable.

Fortunately, there is also an one-line-code in MATLAB to calculate K using LQR.

```
K = lqr(A,B,Q,R);
```

I strongly recommend running the codes by your self, where you will see a very smooth movement of the system using the K of LQR.

4.2.3 Kalman Filter

In the pendulum example, what if we can only do partial measurement? Like the Figure 5?

First, you need to check the observability.

```
A = [0 1 0 0;
     0 -d/M -m*g/M 0;
     0 0 0 1;
     0 -s*d/(M*L) -s*(m+M)*g/(M*L) 0];

B = [0; 1/M; 0; s*1/(M*L)];

C = [1 0 0 0];

rank(obsv(A,C))
```

By changing C, you will find that, the system is observable if and only if we measure x.

After making sure the system is observable, we need to estimate X based on y. Kalman filter is one way to fulfill this task, as shown below.

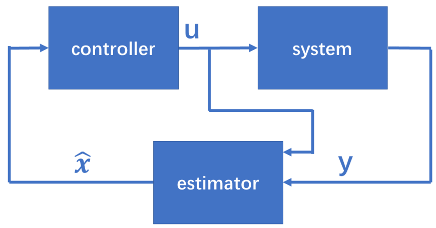

Figure 14: Kalman Filter

The codes are here:

```
%%  Augment system with disturbances and noise
Vd = .1*eye(4);  % disturbance covariance
Vn = 1;        % noise covariance

BF = [B Vd 0*B];  % augment inputs to include disturbance and
    noise

sysC = ss(A,BF,C,[0 0 0 0 0 Vn]);  % build big state space
    system... with single output

sysFullOutput = ss(A,BF,eye(4),zeros(4,size(BF,2)));  % system
    with full state output, disturbance, no noise

%%  Build Kalman filter
[L,P,E] = lqe(A,Vd,C,Vd,Vn);  % design Kalman filter
Kf = (lqr(A',C',Vd,Vn))';    % alternatively, possible to design
     using "LQR" code

sysKF = ss(A-L*C,[B L],eye(4),0*[B L]);  % Kalman filter
    estimator
```

5 With Control, Not Knowing the System

5.1 Overview

Things we have been talking about so far are fantastic, it seems that we can control anything use some one-line-codes of MATLAB. Perfect.

However, if the year of 2022 has got us anything to remember, the world is far from perfect.

Note that, in commands like `K=place(A,B,eigs_wanted)` or `K=lqr(A,B,Q,R)`, you must already know A and B. Is that always possible?

Of course not!

In fluid mechanics, we know the dynamics of the system, but usually we have tens of thousands of dimensions, which leaves us unable to write A and B. In neuroscience, we don't even know the ODEs of the system. Using the words of statisticians, now we have no prior knowledge, we have no choice but to rely on the data.

Scientists of 20 century developed the so-called *data-driven control*, which use the data to infer the differential equations of the system (called *system identification*) and to drive the system to certain fixed point or limit cycle. Data-driven control is also called *machine learning control*.

Some methods like Balanced Truncation (projecting into a lower dimensional space), Balanced Proper Orthogonal Decomposition (producing reduced models for fluids) and Dynamic Mode Decomposition (estimating the leading eigenvalues and eigenvectors) are briefly introduced here,

These methods can be classified to indirect and direct (To clarify, the indirect still involves 2 steps: first you need to identify a model, then a controller is tuned based on such model. The direct, on the other hand, maps the experimental data directly onto the controller, without any model to be identified in between.), iterative and non-iterative, on-line and off-line, according to the Wikipedia article. A neuron seems to perform direct, iterative and on-line data-driven control.

Below, I shall introduce a state-of-the-art (comment from [3]) result for *direct data-driven control*: the Willems fundamental lemma[4].

5.2 The Willems Fundamental Lemma

Here is a thing, the Willems fundamental lemma is proved under 3 conditions: linear, time-invariant, controllability. However, in real life, physicists seem to just use it, even if they know the system is non-linear.

Let us go back to Figure 5 and set $x \in \mathbb{R}^n, u \in \mathbb{R}^q, y \in \mathbb{R}^m$. Now, imagine that you have no prior knowledge of the system. You have no choice but to view it as a black box. You give it an input u, and get an output y from it. You give it an input time series $u(t)$, and get an output time series $y(t)$ from it. Roughly speaking, the Willems fundamental lemma claims that: if your $u(t)$ has enough variance, all possible trajectories of $u(t)$ and $y(t)$ can be obtained from a single trajectory of $u(t)$ and $y(t)$.

To make it stricter, we need to introduce the concept of *persistent excitation*. A time series $u(t), t \in \{1, 2, \cdots, T\}$ is called persistently excited of order k if

$$H_k := \begin{bmatrix} u(1) & u(2) & u(3) & \cdots & u(T-k+1) \\ u(2) & u(3) & \cdots & \cdots & u(T-k+2) \\ u(3) & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u(k) & u(k+1) & \cdots & \cdots & u(T) \end{bmatrix}_{qk \times (T-k+1)} \tag{23}$$

has full row rank, qk. H stands for the Hankel matrix. A necessary condition for $u(t)$ to have persistent excitation of order k is $T \geq (q+1)k - 1$.

If you write codes to test the H matrix, you'll find that, a little variance in $u(t)$ (for example, sampled from `round(N(0,0.3))`) can lead to full row rank.

The Willems fundamental lemma claims that (Theorem 1 in [4]): if $u(t)$ has the order of $n+L$ (i.e., the rank of H is $q(n+L)$), then the observed windows of length L spans the space of all possible windows of length L which the system can produce.

In math form:

$$\begin{bmatrix} \bar{u}_0 \\ \bar{u}_1 \\ \vdots \\ \bar{u}_{L-1} \\ \bar{y}_0 \\ \bar{y}_1 \\ \vdots \\ \bar{y}_{L-1} \end{bmatrix}_{(q+m)L \times 1} = \begin{bmatrix} u(0) & u(1) & \cdots & u(T-L) \\ u(1) & u(2) & \cdots & u(T-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ u(L-1) & u(L) & \cdots & u(T-1) \\ y(0) & y(1) & \cdots & y(T-L) \\ y(1) & y(2) & \cdots & y(T-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ y(L-1) & y(L) & \cdots & y(T-1) \end{bmatrix}_{(q+m)L \times (T-L+1)} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{T-L} \end{bmatrix}_{(T-L+1) \times 1} \tag{24}$$

where the *L.H.S.* is any trajectory of (u, y) that the system can produce and $\vec{g}$ is some constant vector. Of course $\vec{g}$ is different for different *L.H.S.*.

To make Equation 24 more clear, we can rewrite it as:

$$\bar{l} = \begin{bmatrix} \bar{l}_0 & \bar{l}_1 & \cdots & \bar{l}_{T-L} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{T-L} \end{bmatrix} = \sum_{i=0}^{T-L} \bar{l}_i g_i \tag{25}$$

where $\bar{l}$ is any trajectory of (u, y) and $\{\bar{l}_0, \bar{l}_1, \cdots, \bar{l}_{T-L}\}$ are "the observed windows of length L". That is exactly what "the observed windows of length L spans the space of all possible windows of length L which the system can produce" means.

The Willems fundamental lemma has a corollary (Corollary 2 (ii) in [4]): if u is persistently exciting of order $n+1$, (i.e., the rank of H is $q(n+1)$), then:

$$rank\{\begin{bmatrix} u(1) & \cdots & u(T) \\ x(1) & \cdots & x(T) \end{bmatrix}\} = n + q. \tag{26}$$

6 The Neuron as a Direct Data-Driven Controller

section 6 is a note of [5] by Dmitri Lab.

6.1 Why Controller? | The Active Sampling

Why do we need to view the neuron as a controller? The answer to me is the neuron or the whole brain seems to have the ability of *active sampling.*

Neuroscientists used to believe that receiving sensory inputs is a static process. The famous work of Hubel and Wiesel on the retina is also this kind of view. By in fact, the eyes of human can move 3 times in a second (Even if we don't realize it). We are actively sampling, choosing what to see next, instead of just staying there, passively waiting.

In touch and hearing, the active sampling is more obvious that in vision. Below are quotes from Jeff Hawkins' *A Thousand Brains New Theory of Intelligence.*

> If someone places an object onto your open hand, you cannot identify it unless you move your fingers. Similarly, hearing is always dynamic. Not only are auditory objects, such as spoken words, defined by sounds changing over time, but as we listen we move our head to actively modify what we hear.[6]

But up to my knowledge, there is no satisfactory mathematical model for the active sampling. However, this 2024 paper may be a possible solution.

6.2 Notation

Dmitri's paper assumes linearity (see Figure 15:

$$\begin{cases} X_{t+1} &= AX_t + Bu_t \\ y_t &= CX_t \\ u_t &= -K\hat{x}_t \end{cases} \tag{27}$$

where $X \in \mathbb{R}^n, u \in \mathbb{R}^q, A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times q}, y \in \mathbb{R}^m, C \in \mathbb{R}^{m\times n}$.

For non-linear situation, he made an explanation, see subsection 8.2.

This paper implements the Willems fundamental lemma in the following way: set L as 2, T as k and $\hat{x} = y$, drop the row of u_{t+1} , then

$$\begin{bmatrix} \hat{x}_{t+1} \\ \hat{x}_t \\ u_t \end{bmatrix} = \begin{bmatrix} \hat{x}_{1+1} & \cdots & \hat{x}_{k+1} \\ \hat{x}_1 & \cdots & \hat{x}_k \\ u_1 & \cdots & u_k \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \tag{28}$$

The authors let $\{(u_1, \hat{x}_1), (u_2, \hat{x}_2), \cdots, (u_{k+1}, \hat{x}_{k+1})\}$ be the time series in Equation 23, which can be viewed as the data collected in a sufficient long experiment, and let $(u_t, \hat{x}_t)$ be any trajectory of $(u, \hat{x})$ that the system can produce. In other words, future observation-control pairing can be expressed as

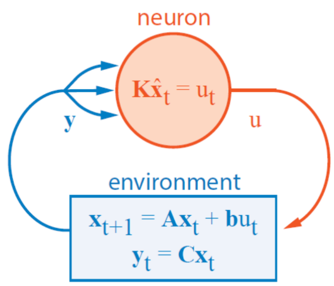

Figure 15: Illustration 1

a linear combination of k historical pairings. Attention please, throughout the paper, future observation-control pairing expressed as a linear combination of historical pairings is equivalent to the successful controlling.

6.3 STDP

It is said that Spike-timing-dependent plasticity (STDP) is the most important discovery in neuroscience in 1990s. Figure 16 is the figure 7 of [7]

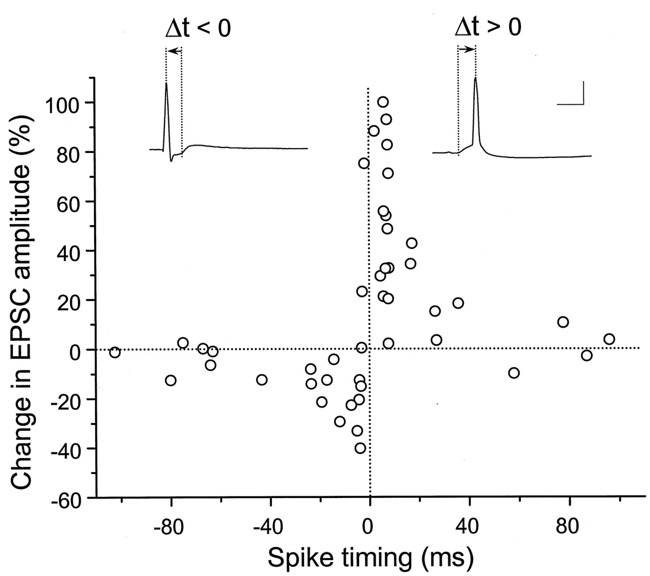

Figure 16: STDP

Dmitri's paper offers a new perspective to explain STDP. Here, the system is the pre-synaptic neuron and the controller is the post-synaptic neuron, like in Figure 4.

First, for simplicity, they set $n = q = 1$ and $\hat{x} = y = x$, so (see Figure 17):

$$x_{t+1} = ax_t + bu_t \tag{29}$$

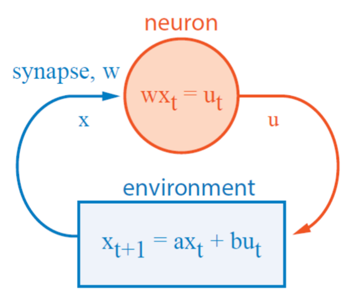

Figure 17: Illustration 2

They utilize LQR to get the optimal u^*:

$$u_t^* = \arg\min_{u_t} q\|x_{t+1}\|^2 + r\|u_t\|^2. \tag{30}$$

Then the optimal weight is:

$$w^* = u_t^*/x_t \tag{31}$$

Note that, x represents the firing rate of the pre-synaptic neuron, u represents the firing rate of the post-synaptic neuron and w stands for the synapse weight. The fixed point of the system is $x = 0$ (i.e. the pre-synaptic neuron is at rest).

$$\begin{bmatrix} 0 \\ \hat{x}_t \\ u_t \end{bmatrix} = \begin{bmatrix} \hat{x}_{1+1} & \cdots & \hat{x}_{k+1} \\ \hat{x}_1 & \cdots & \hat{x}_k \\ u_1 & \cdots & u_k \end{bmatrix} \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \tag{32}$$

After a little of deductions, they got:

$$w^* \propto \sum_{\tau=1}^{k} x_\tau u_\tau - \cos(\widehat{XX_+}) \sum_{\tau=1}^{k} x_{\tau+1} u_\tau, \text{where } \cos(\widehat{XX_+}) = (X_+X_+^T)^{-1}X_+X^T. \tag{33}$$

$\cos(\widehat{XX_+})$ is a constant belongs to $(0, 1)$.

Equation 33 tells us:

- In the natural condition, the system will always return to its fixed point (i.e. the pre-synaptic neuron is at rest).
- In the experimental condition, when the pre-synaptic neuron always fire before the post-synaptic neuron by experimenters, then $w^* > 0$. The smaller the time interval, the bigger w^* is.
- In the experimental condition, when the pre-synaptic neuron always fire after the post-synaptic neuron by experimenters, then $w^* < 0$. The smaller the time interval, the smaller w^* is.
- For the same time interval, the absolute value of w^* of potentiation is bigger than that of depression.

Here is a twist, if $w^* = u_t^*/x_t$ perfectly stands, then the rank of the matrix in Equation 26 will be smaller than 2, then u is not persistently exciting of order 2, then future observation-control pairing can not be expressed as a linear combination of historical pairings. To fix this, the author add a noise term:

$$u_t^* = w^* x_t + \eta_t \tag{34}$$

My comment: The author should have done a simulation and compare their simulation result with the experiment result in Figure 16, but they didn't.

6.4 Feedforward and Feedback Kernel

In figure 3 of [5], the author handled below data in a uniform framework:

1. The blowfly H1 neuron with varying background luminance.
2. Mouse V1 pyramidal neurons responding to different mean injected current waveforms.
3. Salamander retinal ganglion cells exposed to distinct visual stimuli.
4. Drosophila olfactory receptor neurons reacting to varying odorant concentrations.
5. Pyramidal neurons in the rat somatosensory cortex stimulated with current injections of diverse means and variances.

How do they process so many different kind of data using the same pipeline? That is not mystery, let me clarify it.

To do this, the author extended Figure 15 to add a feed-back term. Before, we see the environment as a black box. Now we still remain that way, and just use the data to infer K_{ff} and K_{fb}, instead of trying to get A, B, C. To simplify, they make $n > 1, q = 1, m = 1$ in Equation 27.

Still

$$\begin{cases} X_{t+1} &= AX_t + Bu_t \\ y_t &= CX_t \end{cases} \tag{35}$$

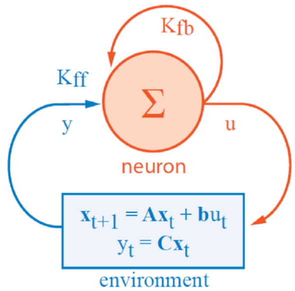

Figure 18: Illustration 3

They let the neuron use the recent past history of u and y (In fact, those are all the things the neuron can exploit) to infer x, i.e.:

$$\hat{x}_t = [y_{t-n}, \cdots, y_{t-1}, u_{t-n}, \cdots, u_{t-1}]^T \tag{36}$$

Still we have $u = K\hat{x}_t$, but now we divide K into two terms: $K = [K_{ff}, K_{fb}]$, so we have:

$$u_t = K_{ff}[y_{t-n} \cdots y_{t-1}]^T + K_{fb}[u_{t-n} \cdots u_{t-1}]^T \tag{37}$$

An intelligent reader might spot that, after being flipped horizontally, the physical content of K_{ff} is the same as the traditional kernel, which is widely used since Hubel and Wiesel. Check subsection 8.4 for this.

It is very easy to calculate K_{ff} and K_{fb} from the experimental data.

Denote

$$U = \begin{bmatrix} u_1 & \cdots & u_t \end{bmatrix} \quad \text{and} \quad \hat{X} = \begin{bmatrix} \hat{x}_1 & \cdots & \hat{x}_t \end{bmatrix} \tag{38}$$

Then

$$U = \begin{bmatrix} K_{ff} & K_{fb} \end{bmatrix} \hat{X} \tag{39}$$

So

$$\begin{bmatrix} K_{ff} & K_{fb} \end{bmatrix} = U\hat{X}^\intercal \left(\hat{X}\hat{X}^\intercal\right)^{-1} \tag{40}$$

In figure 3 of [5], they showed the result K_{ff} and K_{fb} of datasets mentioned above.

My comment: It is interesting to see the the traditional kernel from the perspective of control theory.

6.5 The Experiment of Terry Lab

In 1995, the lab of Terry Sejnowski finished an astonishing experiment:

P.S.: Although the right panel of Figure 19 was also reported in 1976 (figure 4 of [8]), Terry's work gained much more attention than the latter one.

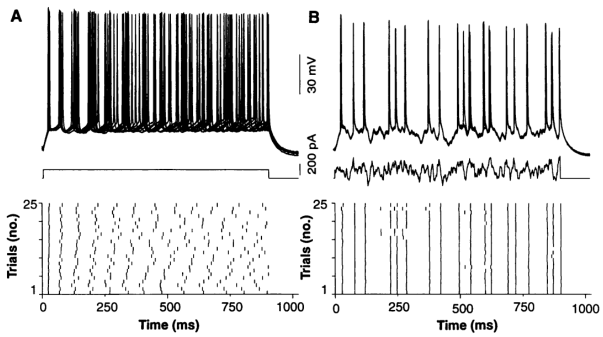

Figure 19: Figure 1 of [4]

Figure 19 is the experiment result of single neuron of the rat brain slice, via current clamp.

Think about the Hodgkin-Huxley model. If you do simulation using computers, you will get the right down panel for both A and B. Why will the neuron lose synchronization in A? Why will the neuron regain synchronization in B by adding noise (according to Dmitri, 50 pA)?

In this 2024 paper, Dmitri came up with that, in simple words, when the input signal is exactly the same, the rank of Hankel matrix in Equation 23 becomes 1, so the neuron lose the ability as a controller, which means that, it can not generate new output based on the history of input/output pairs. And that leads to the fail of synchronization.

To be stricter, Dmitri claimed: if the input signal is not varied enough, then the condition number of the covariance matrix $\hat{X}\hat{X}^T$ of Equation 40 will be very high, which leads to huge error when the neuron is calculating $(\hat{X}\hat{X}^T)^{-1}$, so the neuron's response will be irregular.

At last, Dmitri gave a way to prove his hypothesis via experiments:

> Our hypothesis posits that the singularity at constant current can be empirically validated by measuring spike time variability against noise variance below the threshold reported in [9], and correlating it with the condition number of the time-delay covariance.

My comment: Dmitri used a qualitative explanation, rather than the quantitative. He also didn't show the details of the experiment design. Given that this paper is only an archive version now, they might develop a quantitative explanation at the final version.

6.6 My Comment of the Whole Paper

You might think that this paper is too imaginative to be true. You might also think that this paper is untrustworthy. Both the comments make sense. However, I want you to compare neuroscience with electrodynamics and to recall the period of Michael Faraday. At that time, the field of electrodynamics was also chaotic, with many theories rising and falling, just like the field of neuroscience now. Faraday's field theory was also imaginative to be true and it was declined by great names like Carl Friedrich Gauss, Franz Ernst Neumann and Wilhelm Eduard Weber. **Chaos means fear but flourish while elegance means beauty but death**. After James Clerk Maxwell developed his theories, which was validated by the experiments of Heinrich Hertz, the field of classical electrodynamics was exactly "The building has already been completed; only minor finishing touches remain."

Some theoretical neurosciencetists have already viewed the neuron as the agent in the framework of reinforcement learning. My intuition tells me, their work is equivalent with the work of Dmitri.

A successful theoretical work must predict new experiment phenomena, like the theory of Maxwell predicting electromagnetic wave and the DNA double helix predicting the semi-conservative replication. I am looking forward to the prediction made by Dmitri lab in the final version of this paper.

7 The Story of Maxwell and Control Theory

Much of the content in section 2 and section 3 is developed by a GOAT physicist: James Clerk Maxwell. In 1868, he published a paper named *On Governors*, in which he did a dynamics analysis of the centrifugal governor, a machine used to regulate the velocity of windmills. Because of that, Maxwell is widely regarded as ”the father of control theory”.

Here is a funny story of Maxwell. There is a mathematician, named Edward John Routh, who also did great contribution to the control theory (Routh–Hurwitz stability criterion is named after him). James Clerk Maxwell (1831-1879) and Edward John Routh (1831-1907) were both born in 1831, both studied in the Cambridge University and were both the student of William Hopkins, a very famous mathematician at that time. In 1854, 23-year-old Maxwell and Routh took part in the Mathematical Tripos of Cambridge, which was the one of the most difficult exams all over the world. People who got the first and second highest score would be documented forever (check the list here), where the former would receive the title of ”Senior Wrangler” and the latter would be ”Second Wrangler”. In 1854, Who would win?

It turned out that Maxwell received the second highest score and Routh came to the first. It is said that, Maxwell was very upset about this result, believing his math would never surpass Routh’s. Then, a tearful Maxwell turned to physics thoroughly. The rest of the story is already known to you.

What did I learn from this story? and even from this whole note?

Well, the world is far from perfect, so I am going to make it a better place.

The world is far from perfect and my abilities are limited, so I will find the things I am better at and unite all the people I can unite.

Now, I am also 23-year-old. It is time to decide who I really am.

8 Appendix

8.1 Solve the Linear Dynamic System

Define T as following

$$T = [\xi_1 \cdots \xi_n] \tag{41}$$

where $\xi_1 \cdots \xi_n$ are eigenvectors of A.

Obviously (recall the similarity transformation in the linear algebra)

$$A = TDT^{-1} \tag{42}$$

where $D = diag(\lambda_1, \cdots, \lambda_n)$ and λ_i is the eigenvalue of A

Define Z:

$$Z = T^{-1}X \tag{43}$$

We get:

$$\dot{Z} = DZ \tag{44}$$

which is a *decoupled* equation (that is, no two components of z are related to each other).

That is

$$\begin{bmatrix} \frac{\mathrm{d}z_1}{\mathrm{d}t} \\ \frac{\mathrm{d}z_2}{\mathrm{d}t} \\ \vdots \\ \frac{\mathrm{d}z_n}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \tag{45}$$

So,

$$\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_n(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} z_1(0) \\ z_2(0) \\ \vdots \\ z_n(0) \end{bmatrix} \tag{46}$$

Because of $X = TZ$:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n T_{1i} e^{\lambda_i t} z_i(0) \\ \sum_{i=1}^n T_{2i} e^{\lambda_i t} z_i(0) \\ \vdots \\ \sum_{i=1}^n T_{ni} e^{\lambda_i t} z_i(0) \end{bmatrix} = e^{\lambda_1 t} \begin{bmatrix} T_{11} z_1(0) \\ T_{21} z_1(0) \\ \vdots \\ T_{n1} z_1(0) \end{bmatrix} + e^{\lambda_2 t} \begin{bmatrix} T_{12} z_2(0) \\ T_{22} z_2(0) \\ \vdots \\ T_{n2} z_2(0) \end{bmatrix} + \cdots + e^{\lambda_n t} \begin{bmatrix} T_{1n} z_n(0) \\ T_{2n} z_n(0) \\ \vdots \\ T_{nn} z_n(0) \end{bmatrix} \tag{47}$$

Notice that $T_{11} z_1(0)$ etc. are constant scalars, so we get Equation 7.

The proof is done.

By the way, we can introduce a notation:

$$e^{\Lambda t} = I + \Lambda t + \frac{1}{2!}\Lambda^2 t^2 + \cdots + \frac{1}{n!}\Lambda^n t^n + \cdots \tag{48}$$

Then $Z = e^{Dt}Z(0)$ and $X = e^{At}X(0)$

8.2 Local Linearization Not Around the Fixed Point

In fact, we can do local linearization at any points, not just the fixed point.

$$\begin{cases} \dot{X}_1 = f(\bar{X}) + \frac{Df}{Dx}|_{\bar{X}}(X_1 - \bar{X}) \\ \dot{X}_2 = f(\bar{X}) + \frac{Df}{Dx}|_{\bar{X}}(X_2 - \bar{X}) \end{cases} \tag{49}$$

Then

$$\frac{\mathrm{d}\Delta X}{\mathrm{d}t} = A\Delta X \tag{50}$$

where $\Delta X := X_2 - X_1$ and $A := \frac{Df}{Dx}|_{\bar{X}}$

If we define $Y := \Delta X$, then

$$\dot{Y} = AY \tag{51}$$

In Dmitri's paper, he came up with an idea that the neurons (served as controllers) in certain layer of an artificial or biological neural network were doing local linearization at different points to make the system always being controlled like a linear system. That is, at some point, we can control the ΔX around it. And if X moves too much off it, we can control the ΔX around another point.

8.3 Feedthrough Term

We can also add feedthrough term in Equation 13

$$\begin{cases} \dot{X} & = & AX + Bu \\ y & = & CX + Du \\ u & = & -K\hat{x} \end{cases} \tag{52}$$

But in most real world systems, D is 0.
The most general form is

$$\begin{cases} \dot{X} & = & f(X, u) \\ y & = & g(X, u) \\ \hat{x} & = & h(u, y) \\ u & = & K(\hat{x}) \end{cases} \tag{53}$$

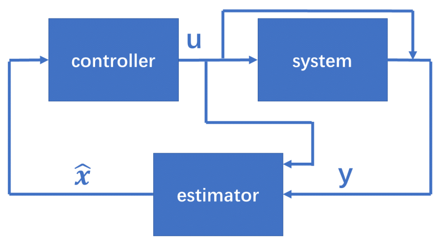

Figure 20: Adding Feed-Through

8.4 The Feedforward Kernel of Dmitri vs Traditional Kernel

In traditional kernel method, we will denote stimulus as S, response as R and assume:

$$R = S * K \tag{54}$$

where $*$ stands for convolution and K is called kernel.

This assumption fits the data of retinal and V1 very well, although fails in more advanced regions like V2 and V4. Check the chapter 2 of [10] for how to use the Wiener series to deduce the spike-trigger-average method to get K.

Rewrite Equation 54 in discrete terms:

$$R_t = \sum_{i=t-n}^{t-1} S_i K_{t-i} \tag{55}$$

Now rethink the first term of the R.H.S. of Equation 37, which can be written as:

$$\tilde{u}_t = K_{ff}[y_{t-n} \cdots y_{t-1}]^T = \sum_{i=t-n}^{t-1} y_i \tilde{K}_i \tag{56}$$

where $\tilde{u}_t$ represents the first part of u_t and $\tilde{K} := K_{ff}$.

Looking at Figure 18, you will find y served as the input to the neuron (i.e.: S in Equation 55) while u served as the output of the neuron (i.e.: R in Equation 55).

So rewrite Equation 56:

$$\tilde{R}_t = \sum_{i=t-n}^{t-1} \tilde{S}_i \tilde{K}_i \tag{57}$$

where $\tilde{R}_t := \tilde{u}_t$ and $\tilde{S}_t := \tilde{y}_t$.

Compare Equation 55 and Equation 57, You will find that the feed-forward kernel of Dmitri and the traditional kernel are horizontally flipped versions of each other.

References

[1] Eugene I Butikov. The rigid pendulum-an antique but evergreen physical model. *European Journal of Physics*, 20(6):429, 1999.

[2] Thomas Peters. *Data-driven science and engineering: machine learning, dynamical systems, and control: by SL Brunton and JN Kutz, 2019, Cambridge, Cambridge University Press, 472 pp.,£ 49.99 (hardback), ISBN 9781108422093. Level: postgraduate. Scope: textbook.* Taylor & Francis, 2019.

[3] Henk J van Waarde. Beyond persistent excitation: Online experiment design for data-driven modeling and control. *IEEE Control Systems Letters*, 6:319–324, 2021.

[4] Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.

[5] Jason Moore, Alexander Genkin, Magnus Tournoy, Joshua Pughe-Sanford, Rob R de Ruyter van Steveninck, and Dmitri B Chklovskii. The neuron as a direct data-driven controller. *bioRxiv*, pages 2024–01, 2024.

[6] Jeff Hawkins. *A thousand brains: A new theory of intelligence.* Basic Books, 2021.

[7] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472, 1998.

[8] Hugh L Bryant and José P Segundo. Spike initiation by transmembrane current: a white-noise analysis. *The Journal of physiology*, 260(2):279–314, 1976.

[9] Zachary F Mainen and Terrence J Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506, 1995.

[10] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems.* MIT press, 2005.