

# **Обчислювальні принципи роботи мозку та автономні дрони**

Осауленко В'ячеслав

Гостьова лекція. КПІ

12.10.23

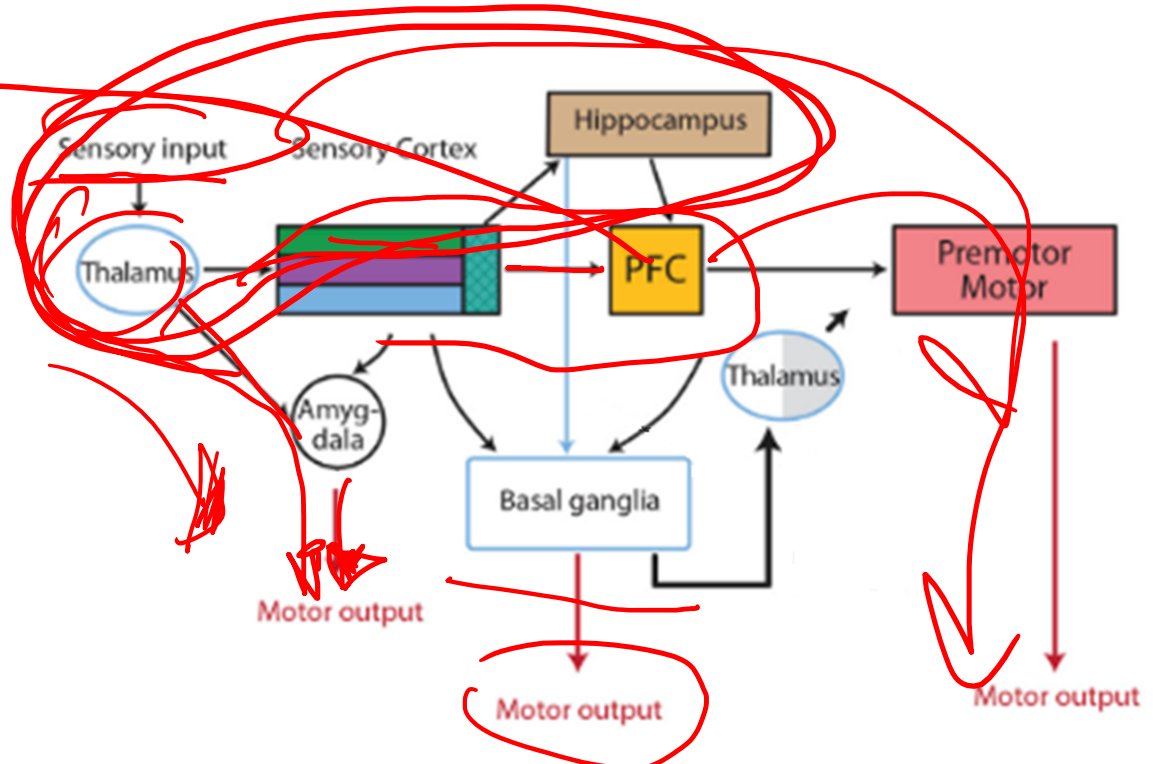
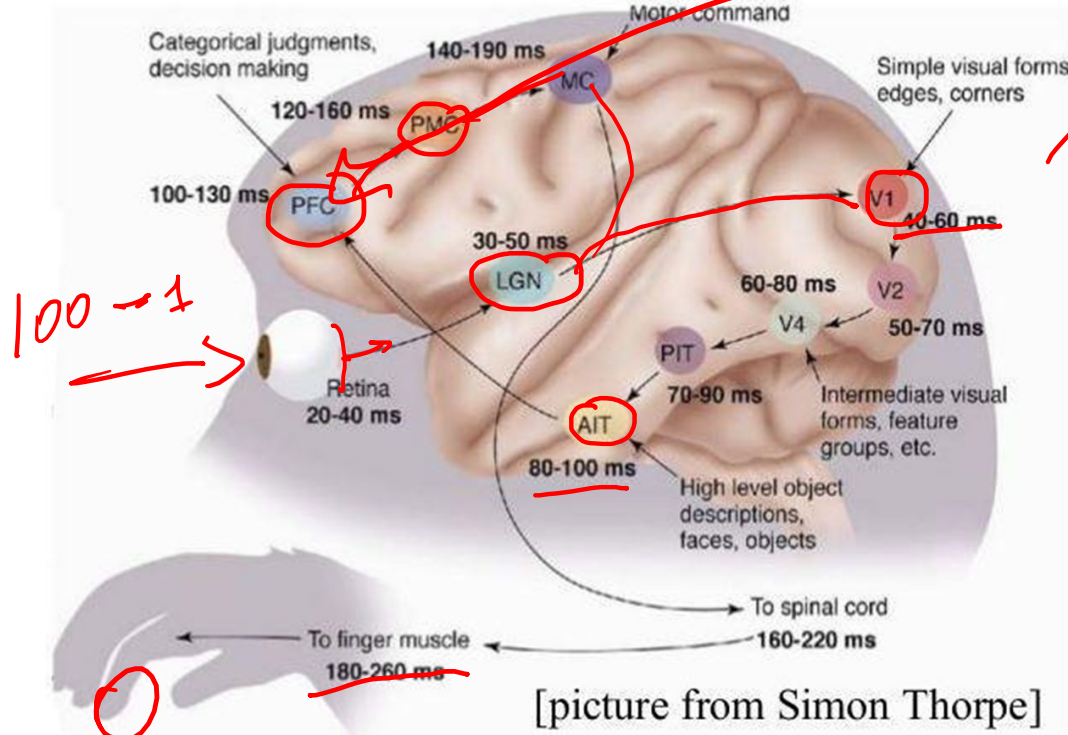
# Лекція. План

Огляд про мозок та дрони

Обчислювальні принципи, порівняння

Дрони: ідеї, перспективи

# Мозок. Загальна архітектура

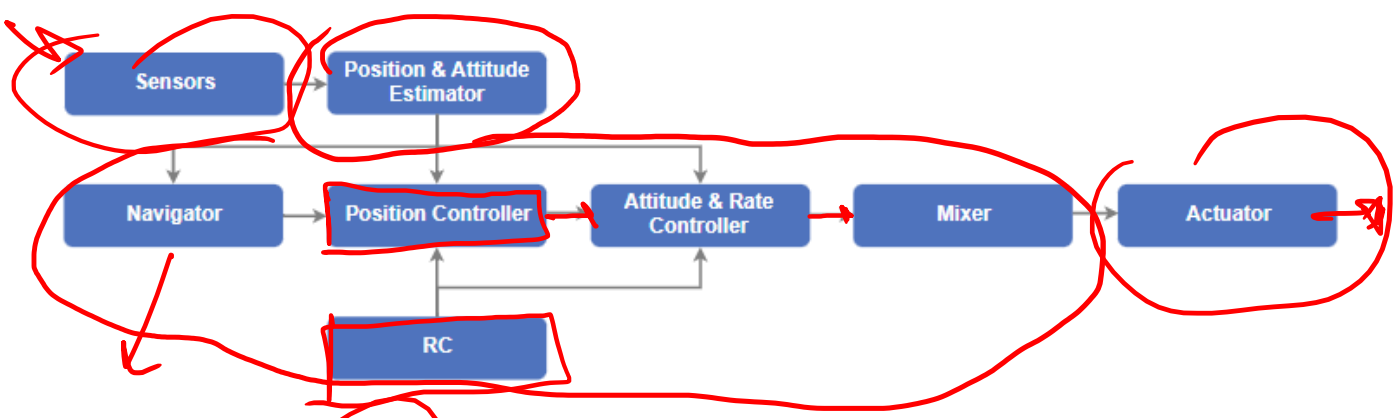


Lisman, J. (2015). The challenge of understanding the brain: where we stand in 2015. *Neuron*, 86(4), 864-882.

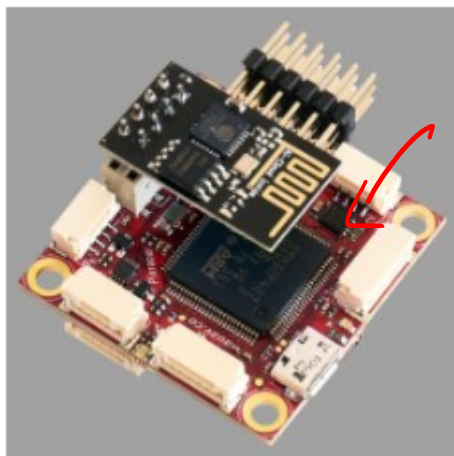
# Дрони

*ArduPi Not*

## Flight Stack (PX4)



QAV-R 5" KISS ESC Racer (Pixracer)

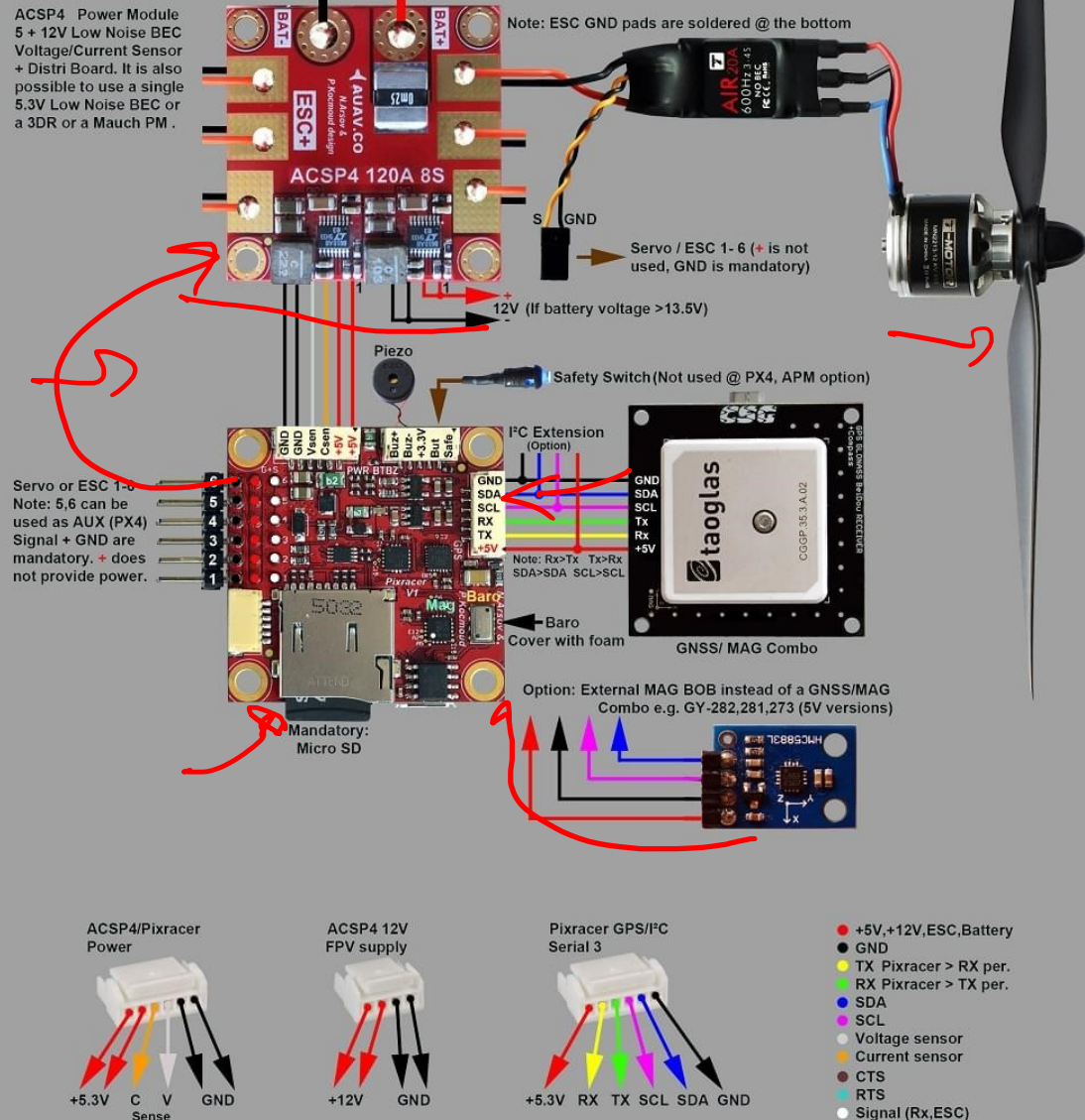


*→ Pixracer*

<https://docs.px4.io/main/en/concept/architecture.html>

# Pixracer

Setup proposal bottom side



ACSP4/Pixracer Power

ACSP4 12V FPV supply

Pixracer GPS/I/C Serial 3

+5.3V C V GND Sense

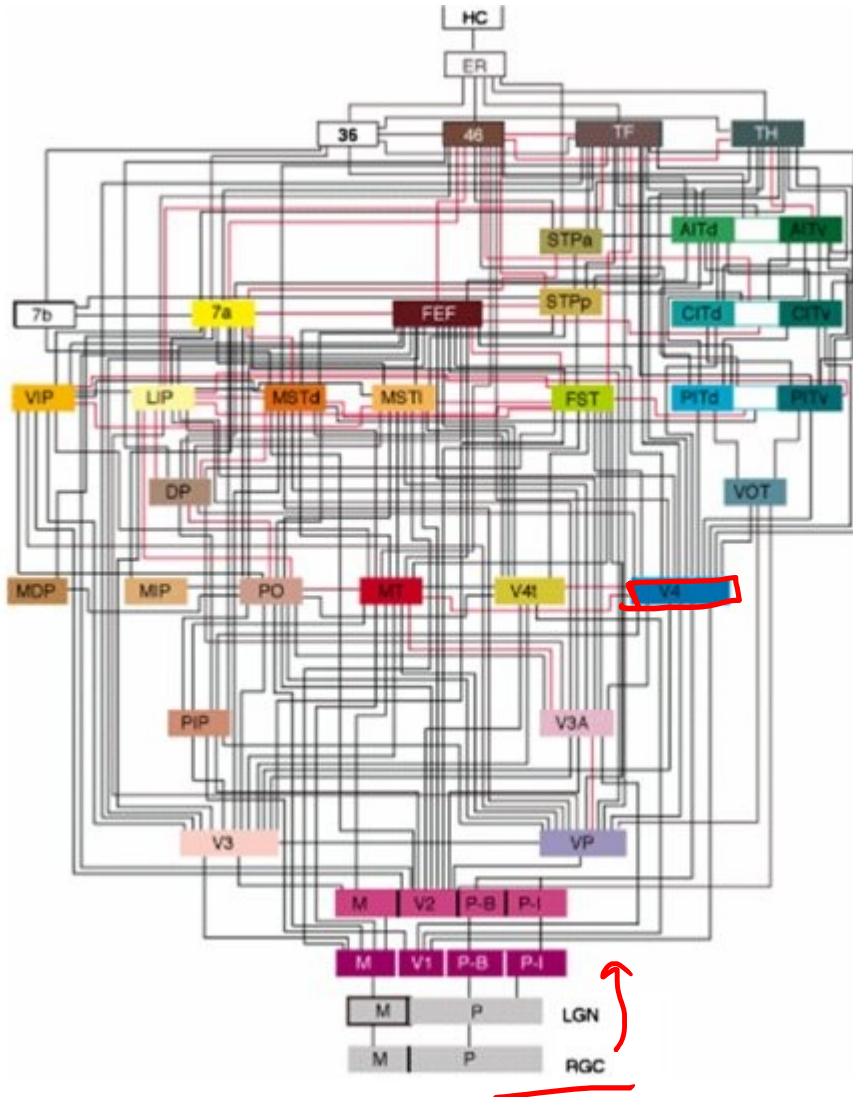
+12V GND

+5.3V RX TX SCL SDA GND

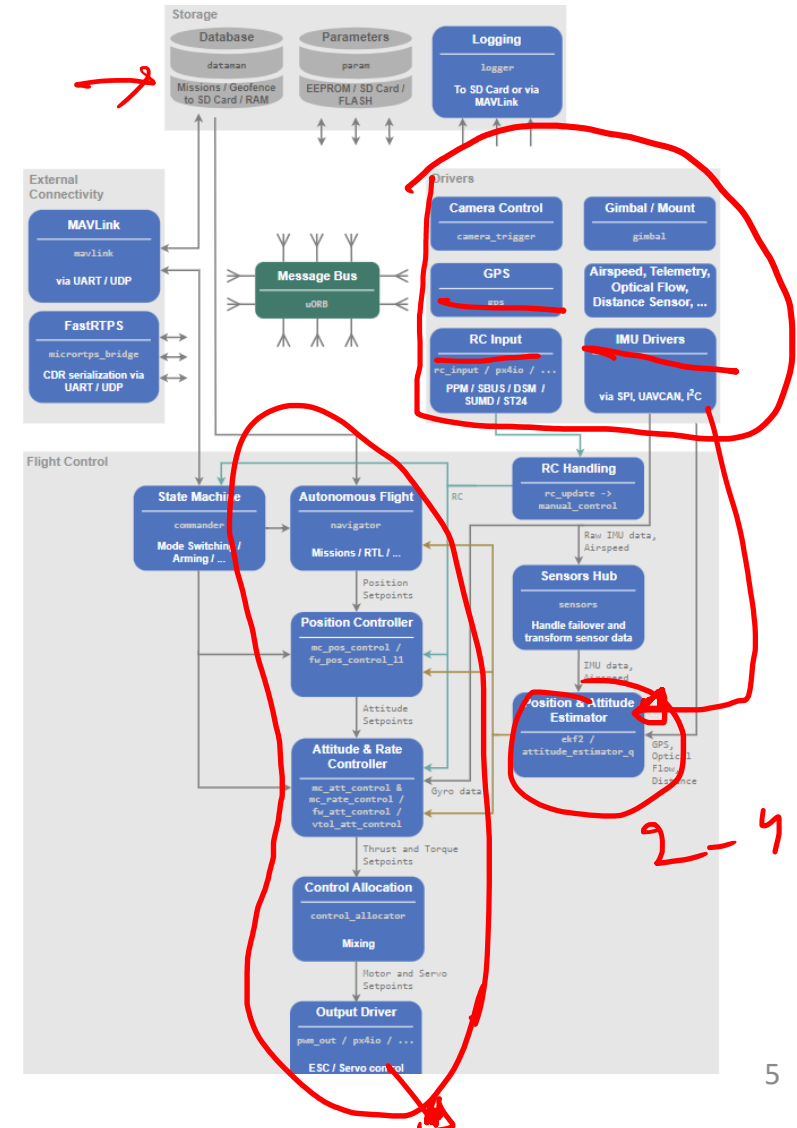
Important: The Pixracer is already coming with all wires, ready to go for ACSP4, 3DR DF-13 externals such as GPS, Telemetry. So this is meant for wiring of external 3rd party hardware. Keep in mind colors don't matter. Before connect, check pin assignments carefully.

# Насправді все складніше

Hierarchical organization of the monkey visual system (from Felleman & Van Essen, 1991).




# High-Level PX4 Software Architecture




# Обчислювальні принципи (мозку та дронів)

1. Модель
2. Передбачення
3. Контроль

 principle  
/ˈprɪnsɪpl/  
*noun*

1. a fundamental truth or proposition that serves as the foundation for a system of belief or behaviour or for a chain of reasoning  
"the basic principles of justice"

Similar: [truth](#) [proposition](#) [concept](#) [idea](#) [theory](#) [postulate](#) [assumption](#) 

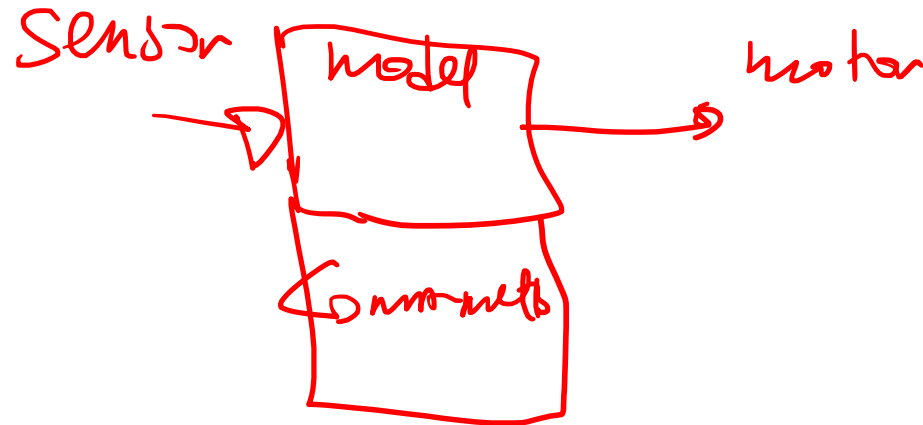
2. a general scientific theorem or law that has numerous special applications across a wide field.

# Принцип 1. Побудова моделі (дрони)

Варіант 1. Мат модель.

Варіант 2. Навчання з даних

Варіант 3. Посередині. (Ідентифікація систем)



# Принцип 1. Побудова моделі (дрони)

Варіант 1. Мат модель.

$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p}_{WB} \\ \dot{q}_{WB} \\ \dot{v}_{WB} \\ \dot{\omega}_B \end{bmatrix} = \mathbf{f}_{dyn}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} v_W \\ q_{WB} \cdot \begin{bmatrix} 0 \\ \omega_B/2 \end{bmatrix} \\ \frac{1}{m} q_{WB} \odot \mathbf{T}_B + \mathbf{g}_W \\ \mathbf{J}^{-1} (\boldsymbol{\tau}_B - \omega_B \times \mathbf{J} \omega_B) \end{bmatrix}$

$\mathbf{T}_B = \begin{bmatrix} 0 \\ 0 \\ \sum T_i \end{bmatrix}$  and  $\boldsymbol{\tau}_B = \begin{bmatrix} d_y(-T_0 - T_1 + T_2 + T_3) \\ d_x(-T_0 + T_1 + T_2 - T_3) \\ c_\tau(-T_0 + T_1 - T_2 + T_3) \end{bmatrix}$

$\mathbf{x}_{k+1} = \mathbf{f}_{RK4}(\mathbf{x}_k, \mathbf{u}_k, \delta t)$

$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$

$\mathbf{x}_0$   
 $\mathbf{u}$

$Mg = F + mg$

The diagram shows a quadcopter drone with four rotors. The world coordinate system  $O_{World}$  has axes  $x_W$  (red),  $y_W$  (green), and  $z_W$  (blue). The body coordinate system  $O_{Body}$  has axes  $x_B$  (red),  $y_B$  (green), and  $z_B$  (blue). Thrust vectors  $T_0, T_1, T_2, T_3$  are shown as red arrows pointing upwards from each rotor. Gravity  $g_W$  is shown as a red arrow pointing downwards. Handwritten red annotations include:
 

- A circle around the  $\tau$  term in the dynamics equation.
- A circle around the  $\frac{1}{m} q_{WB} \odot \mathbf{T}_B + \mathbf{g}_W$  term.
- A circle around the  $\mathbf{T}_B$  matrix.
- A circle around the  $\mathbf{x}_{k+1} = \mathbf{f}_{RK4}(\mathbf{x}_k, \mathbf{u}_k, \delta t)$  equation.
- A circle around the  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  equation.
- Arrows pointing from  $\mathbf{x}_0$  and  $\mathbf{u}$  to the right.
- A note  $Mg = F + mg$  with an arrow pointing to the gravity term.



# Принцип 1. Побудова моделі (дрони)

Варіант 3. Посередині. (Ідентифікація систем)



$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{p}_{WB} \\ \dot{q}_{WB} \\ \dot{v}_{WB} \\ \dot{\omega}_B \end{bmatrix} = \mathbf{f}_{dyn}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} v_W \\ \mathbf{q}_{WB} \cdot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_B/2 \end{bmatrix} \\ \frac{1}{m} \mathbf{q}_{WB} \odot \mathbf{T}_B + \mathbf{g}_W \\ \mathbf{J}^{-1} (\boldsymbol{\tau}_B - \boldsymbol{\omega}_B \times \mathbf{J} \boldsymbol{\omega}_B) \end{bmatrix}$$

$$\mathbf{T}_B = \begin{bmatrix} 0 \\ 0 \\ \sum T_i \end{bmatrix} \quad \text{and} \quad \boldsymbol{\tau}_B = \begin{bmatrix} d_y(-T_0 - T_1 + T_2 + T_3) \\ d_x(-T_0 + T_1 + T_2 - T_3) \\ c_\tau(-T_0 + T_1 - T_2 + T_3) \end{bmatrix}$$

$$\mathbf{x}_{k+1} = \mathbf{f}_{RK4}(\mathbf{x}_k, \mathbf{u}_k, \delta t).$$

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{f}_{\mathcal{F}}(\mathbf{x}, \mathbf{u}) + \mathbf{f}_{\mathcal{D}}(\mathbf{x}, \mathbf{u}),$$

$$\mathbf{f}_{\mathcal{D}}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{f}_{\mathcal{D}_\theta}(\mathbf{x}, \mathbf{u}) \\ \mathbf{f}_{\mathcal{D}_\psi}(\mathbf{x}, \mathbf{u}) \end{bmatrix},$$

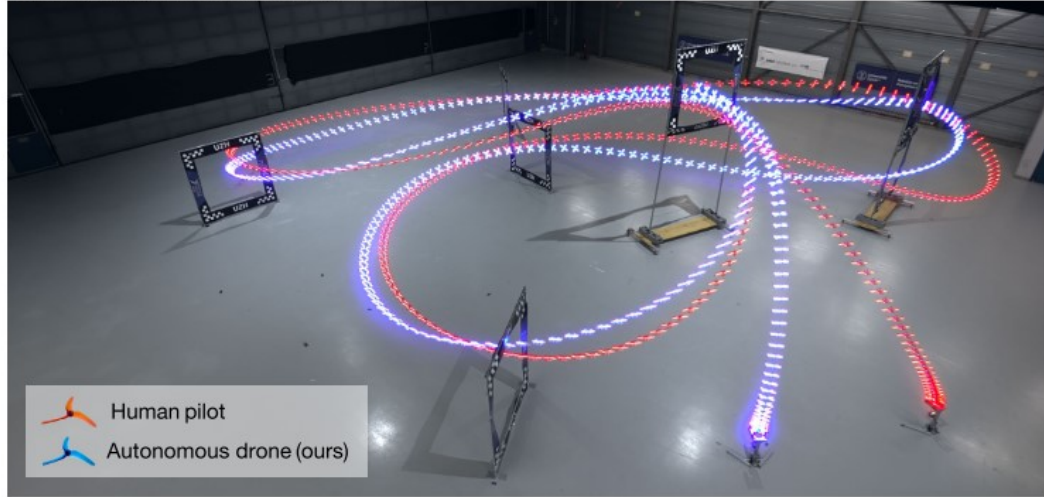
		Model Configuration									
		[None]	MLP								(CNN)
Architecture	Layers	-	2	2	5	5	12	12	20	50	18
	Neurons	-	16	128	16	128	32	512	512	512	-
Parameter Count		0	354	17K	12K	67K	12K	182K	500K	13M	12M
		Control Freq. [Hz]									
Naive	ARM-CPU		403	20	280	6	66	2	<1	-	-
RTN-MPC	ARM-CPU	562	148	135	118	102	85	67	11	-	<1
	Jetson-GPU		109	107	88	84	63	61	46	-	9
Naive	i7-CPU		2228	116	1139	31	168	11	<1	<<1	-
RTN-MPC	i7-CPU	4262	1096	1071	885	784	588	507	91	39	4
	RTX3000-GPU		781	770	586	598	363	363	232	117	63

Salzmann, T., et al. (2023). Real-time Neural MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms

# Принцип 1. Побудова моделі (дрони)

## Варіант 2. Навчання з даних

a Drone racing: human versus autonomous



Перегони на дронах

b Head-to-head competition



c Human champions

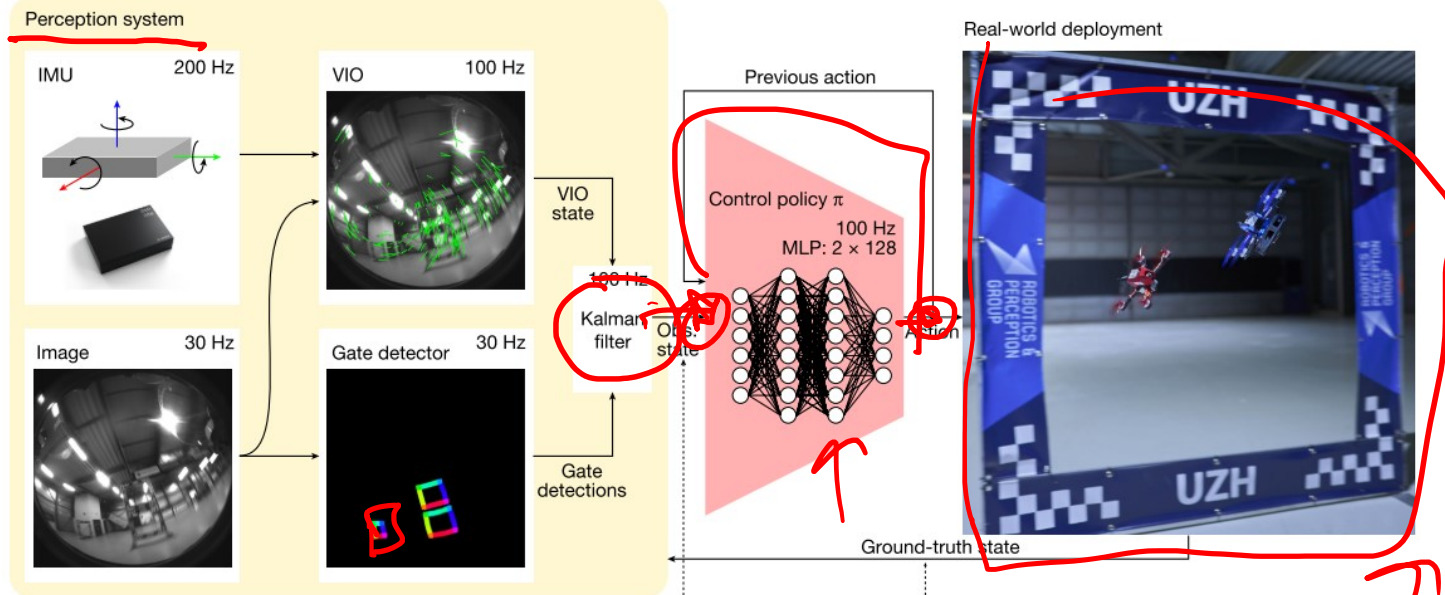


Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., & Scaramuzza, D. (2023). Champion-level drone racing using deep reinforcement learning.

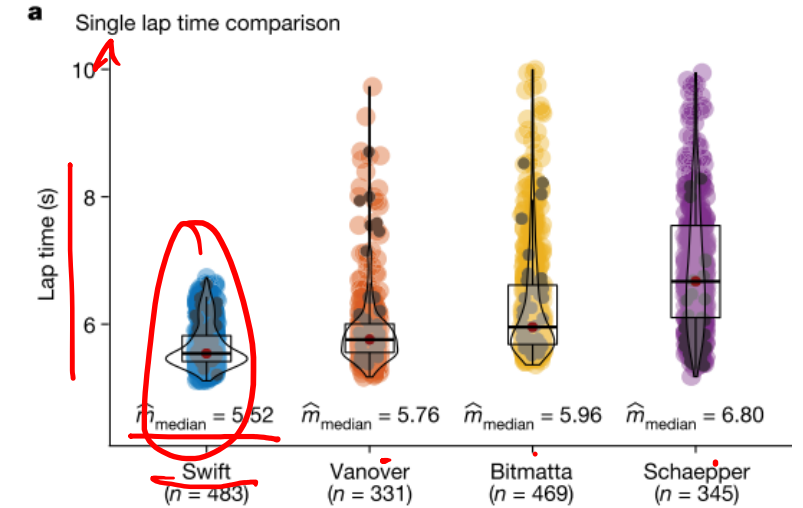
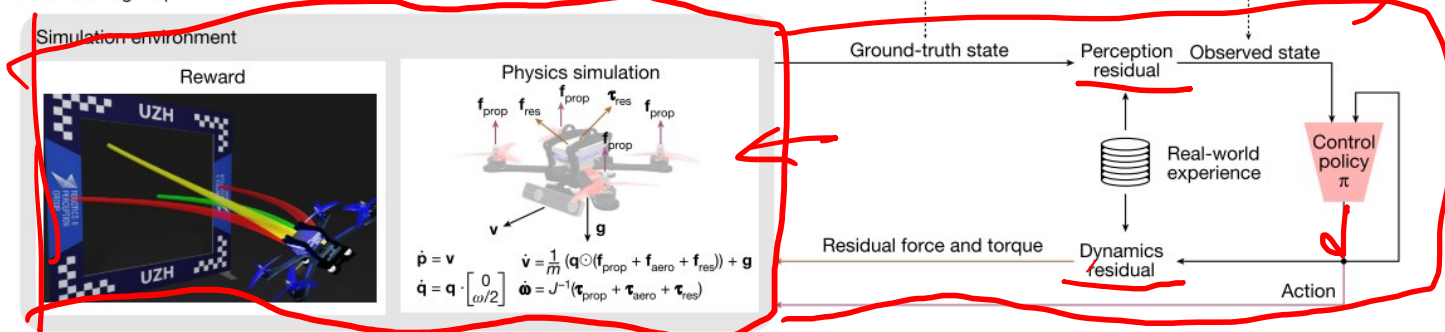
# Принцип 1. Побудова моделі (дрони)

## Варіант 2. Навчання з даних

a Real-world operation



b RL training loop



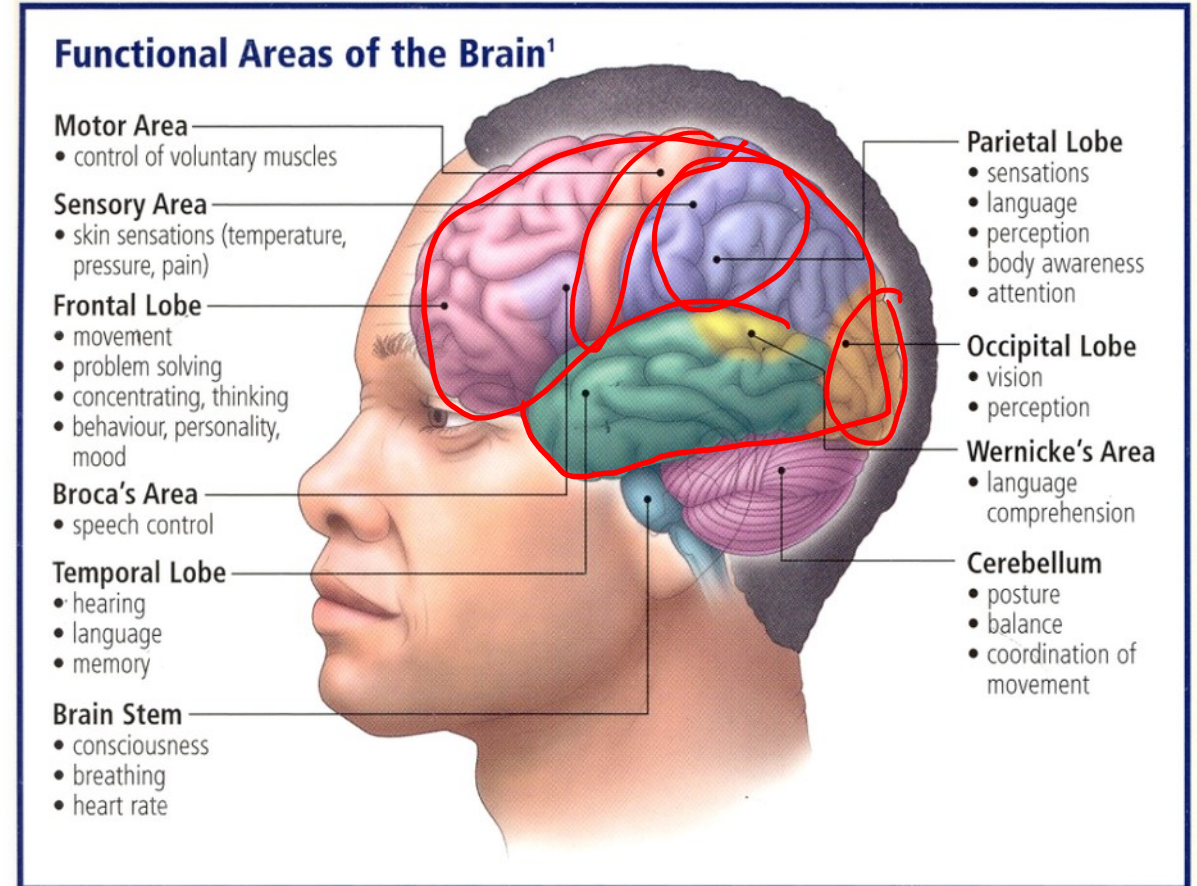
Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., & Scaramuzza, D. (2023). Champion-level drone racing using deep reinforcement learning.

<https://www.youtube.com/watch?v=HGULBBAo5IA>

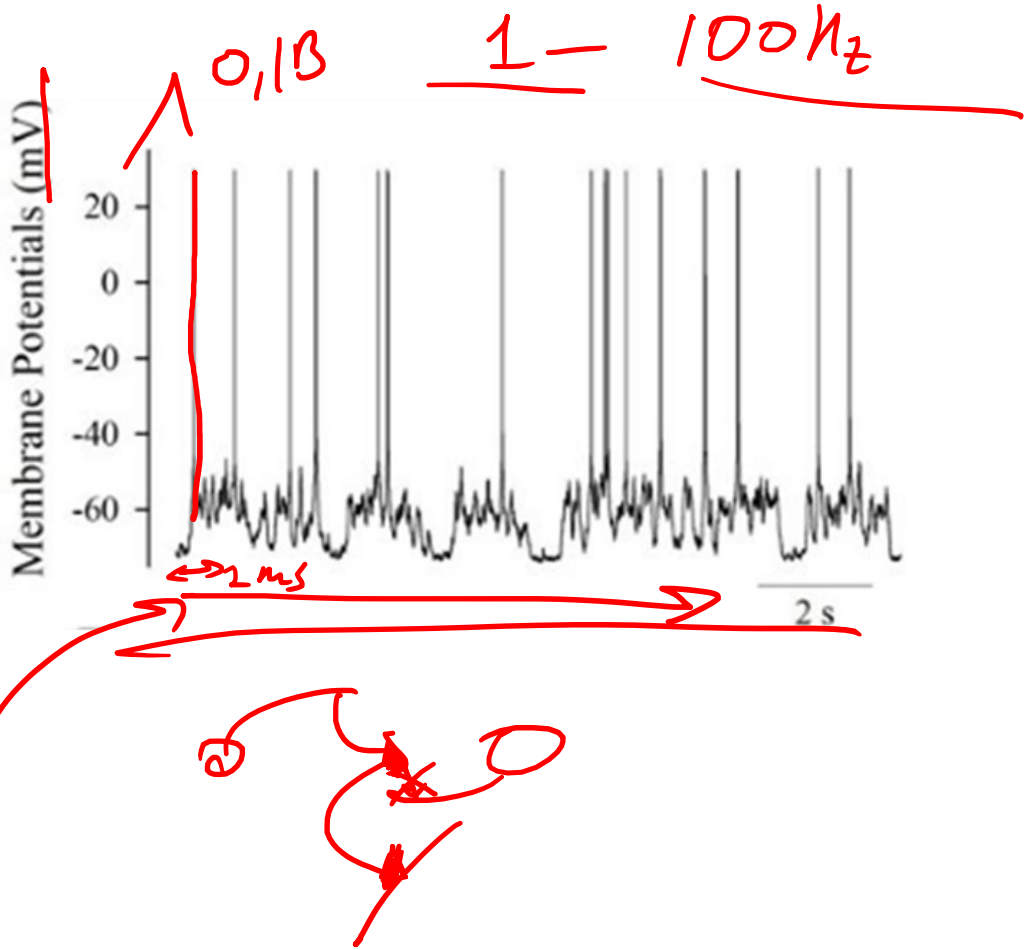
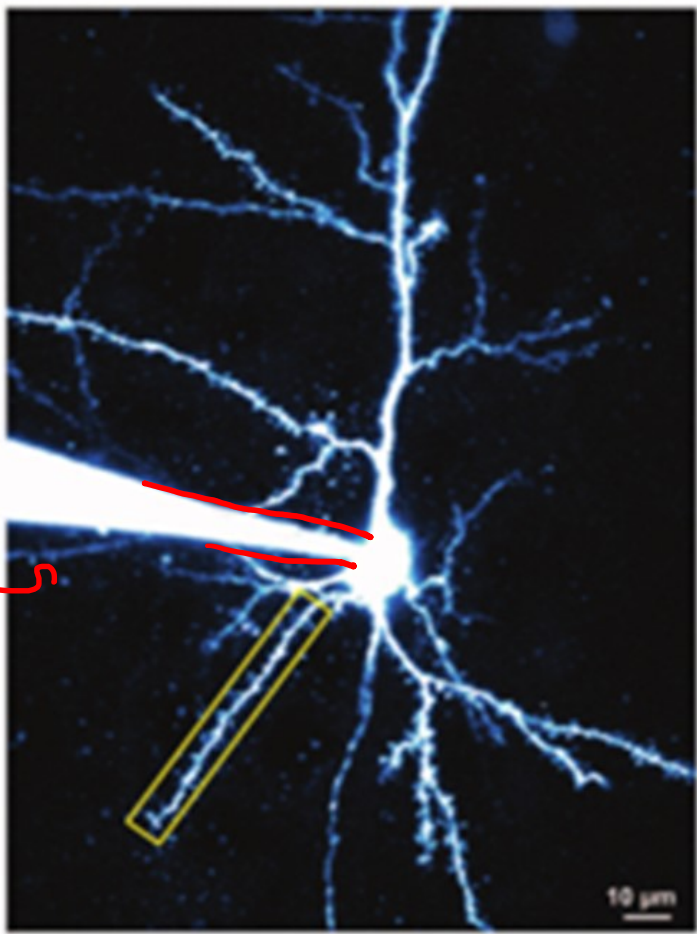
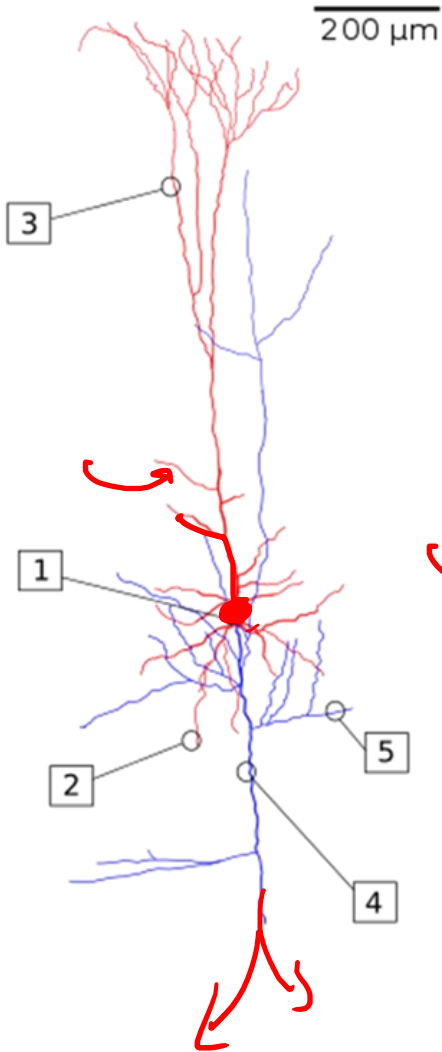
# Принцип 1. Побудова моделі (мозок)

- Причинно-наслідкові зв'язки
- Пам'ять
- Абстракції
- Генералізація
- Композиція ←
- Стиснення ↘
- Увага
- Послідовності

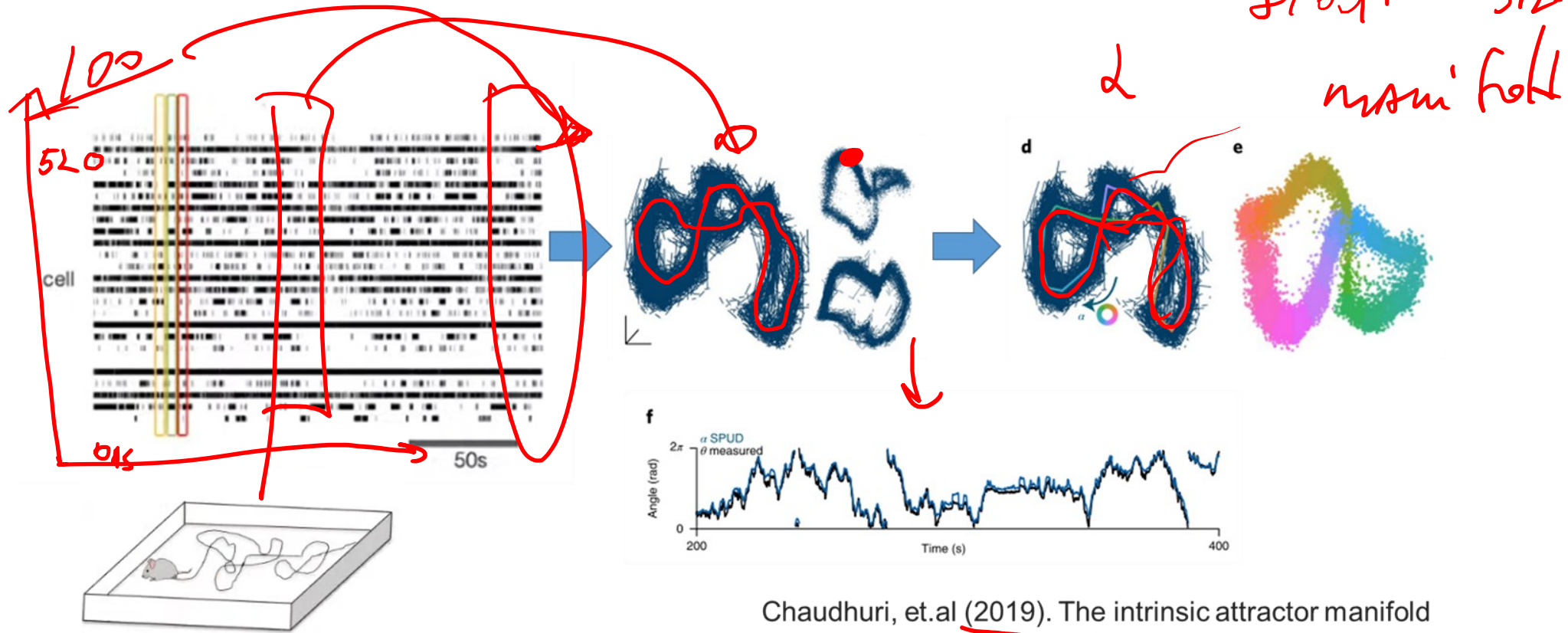
S →  
U → S



# Нейрон



# Кодування кута повороту



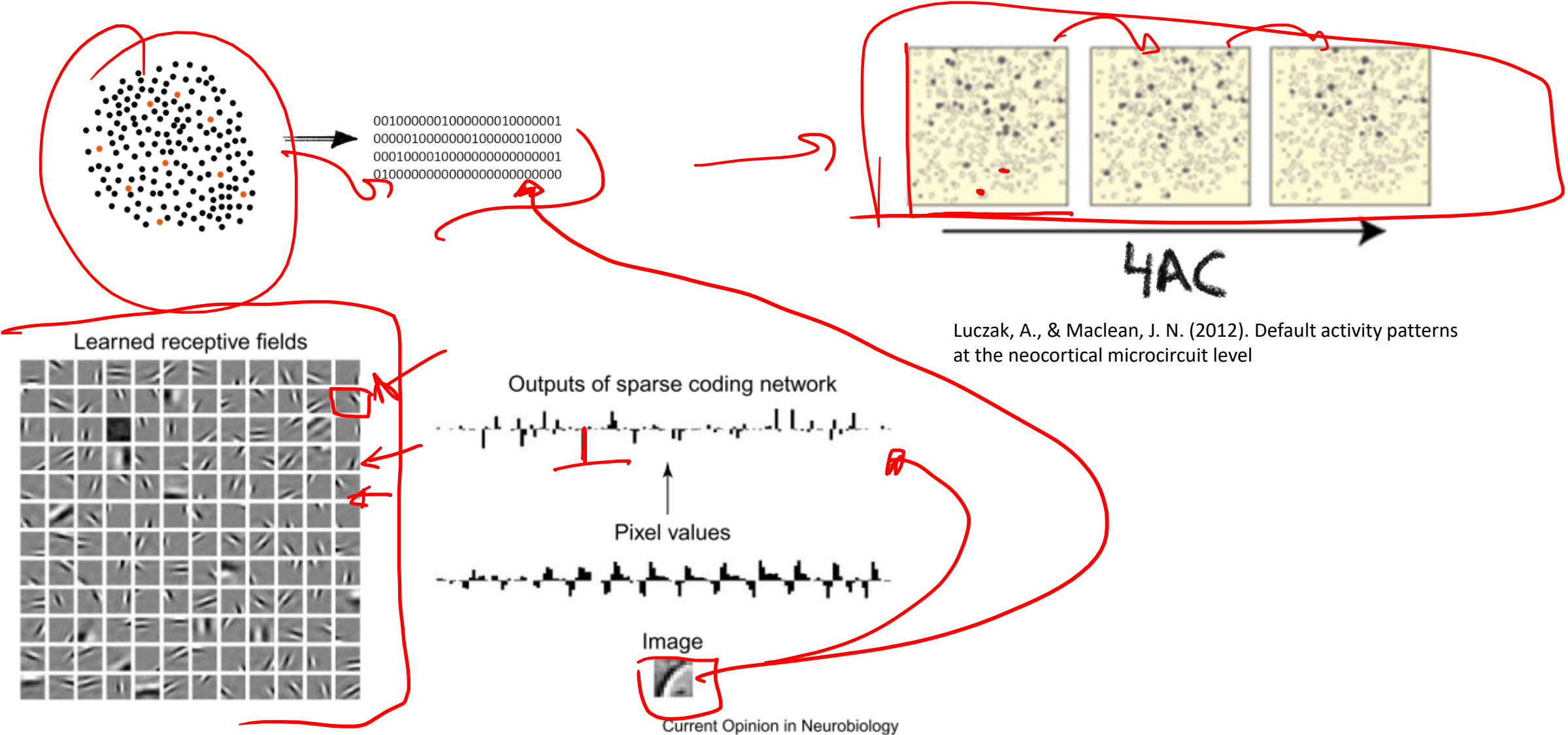
float on manifold

“Antero-dorsal thalamic nucleus and the postsubiculum of mice”

Chaudhuri, et.al (2019). The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep.

Більше у [video](#)

# Розріджене кодування



Olshausen, B. A., & Field, D. J. (2004). Sparse coding of sensory inputs. In *Current Opinion in Neurobiology*.

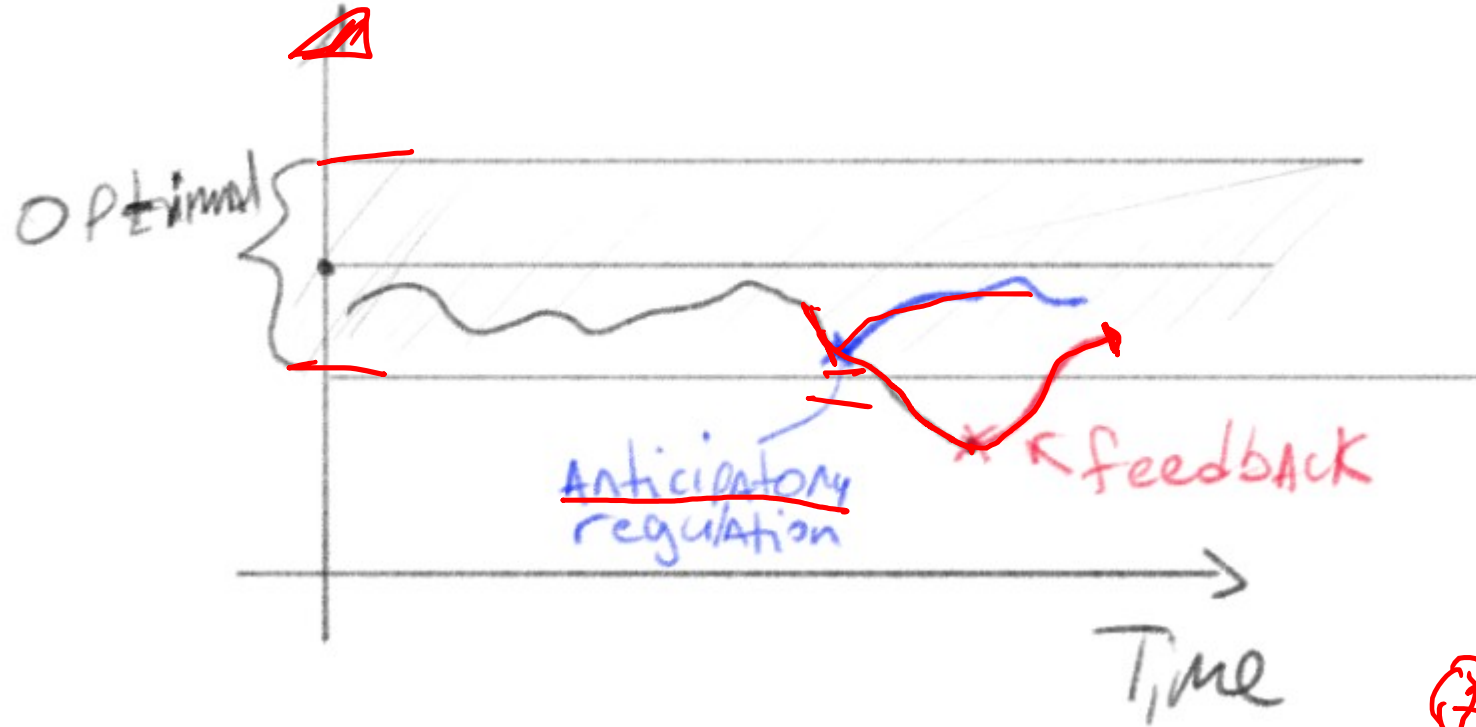




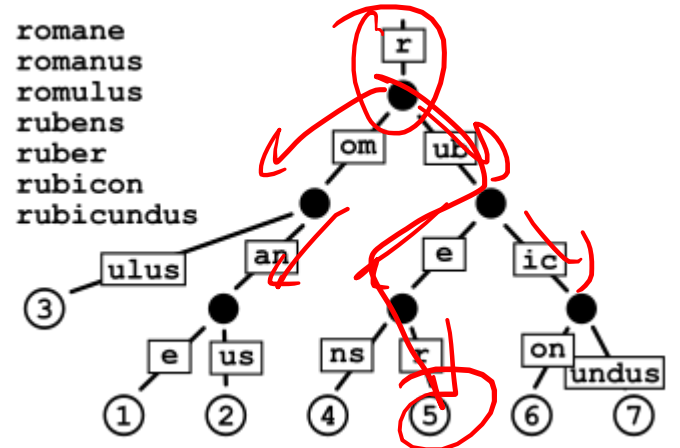


# Принцип 2. Передбачення (мозок)

- Anticipatory regulation vs feedback

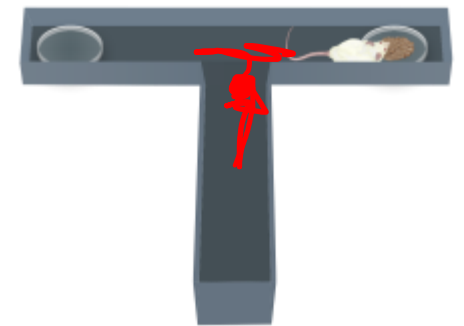


- 1 romane
- 2 romanus
- 3 romulus
- 4 rubens
- 5 ruber
- 6 rubicon
- 7 rubicundus

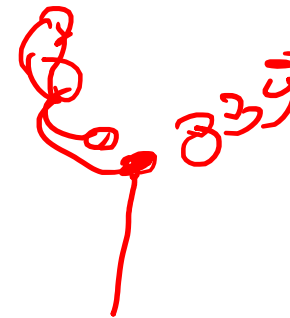


Radix tree

Sequence completion



T maze



# Принцип 2. Передбачення (дрони)

➤ PID control (feedback)

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

$$q(t) = x^* - x$$

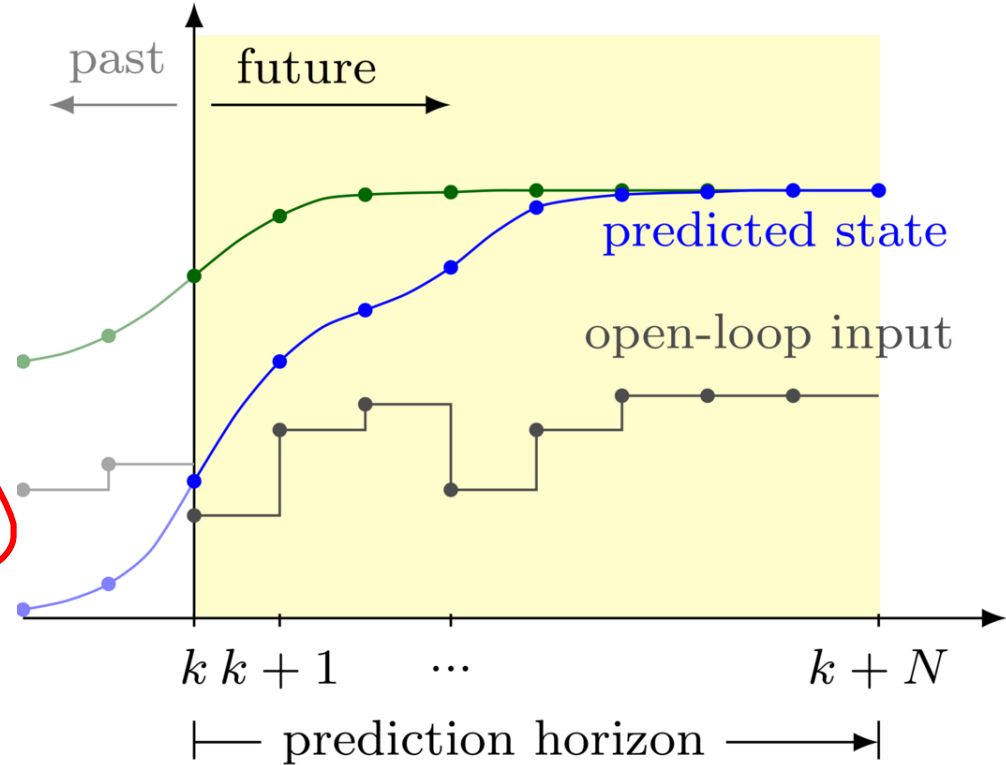
➤ Model Predictive Control (MPC)

$$\dot{x} = f(x, u)$$

➤ Trajectory optimization

<http://underactuated.csail.mit.edu>

$$x_t \rightarrow x_{t+N}$$



#### D. MPC Formulation

In its most general form, MPC stabilizes a system subject to its dynamics  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$  along a reference  $\mathbf{x}^*(t), \mathbf{u}^*(t)$ , by minimizing a cost  $\mathcal{L}(\mathbf{x}, \mathbf{u})$  as in:

$$\begin{aligned} & \min_{\mathbf{u}} \int \mathcal{L}(\mathbf{x}, \mathbf{u}) & (9) \\ \text{subject to} & \quad \dot{\mathbf{x}} = \mathbf{f}_{dyn}(\mathbf{x}, \mathbf{u}) & \quad \mathbf{x}(t_0) = \mathbf{x}_{init} \\ & \quad \mathbf{r}(\mathbf{x}, \mathbf{u}) = 0 & \quad \mathbf{h}(\mathbf{x}, \mathbf{u}) \leq 0 \end{aligned}$$

where  $\mathbf{x}_0$  denotes the initial condition and  $\mathbf{h}, \mathbf{r}$  can incorporate (in-)equality constraints, such as input limitations.

For our application, and as most commonly done, we specify the cost to be of quadratic form  $\mathcal{L}(\mathbf{x}, \mathbf{u}) = \|\mathbf{x} - \mathbf{x}^*\|_Q^2 + \|\mathbf{u} - \mathbf{u}^*\|_R^2$  and discretize the system into  $N$  steps over time horizon  $T$  of size  $dt = T/N$ . We account for input limitations by constraining  $0 \leq \mathbf{u} \leq u_{max}$ , and optionally include the GP predictions within the system dynamics.

$$\begin{aligned} & \min_{\mathbf{u}} \mathbf{x}_N^\top Q \mathbf{x}_N + \sum_{k=0}^N \mathbf{x}_k^\top Q \mathbf{x}_k + \mathbf{u}_k^\top R \mathbf{u}_k & (10) \\ \text{subject to} & \quad \mathbf{x}_{k+1} = \mathbf{f}_{RK4}(\mathbf{x}_k, \mathbf{u}_k, \delta t) \\ & \quad \mathbf{x}_0 = \mathbf{x}_{init} \\ & \quad u_{min} \leq \mathbf{u}_k \leq u_{max} \end{aligned}$$

where  $\mathbf{f}_{RK4}$  can be extended to the corrected dynamics  $\mathbf{f}_{cor}$ .

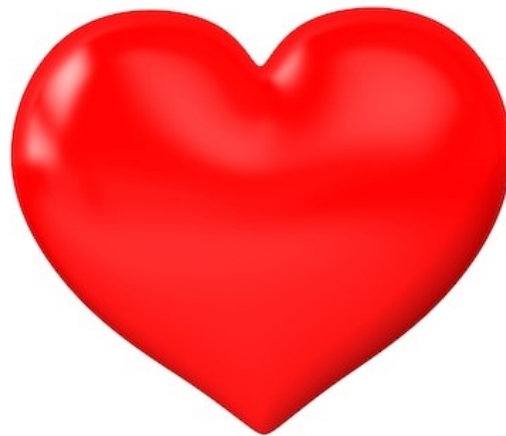
To solve this quadratic optimization problem we construct it using a multiple shooting scheme [32] and solve it through a sequential quadratic program (SQP) executed in a real-time iteration scheme (RTI) [32]. All implementations are done using ACADOS [33] and CasADi [34].

## Принцип 2. Передбачення (мозок)

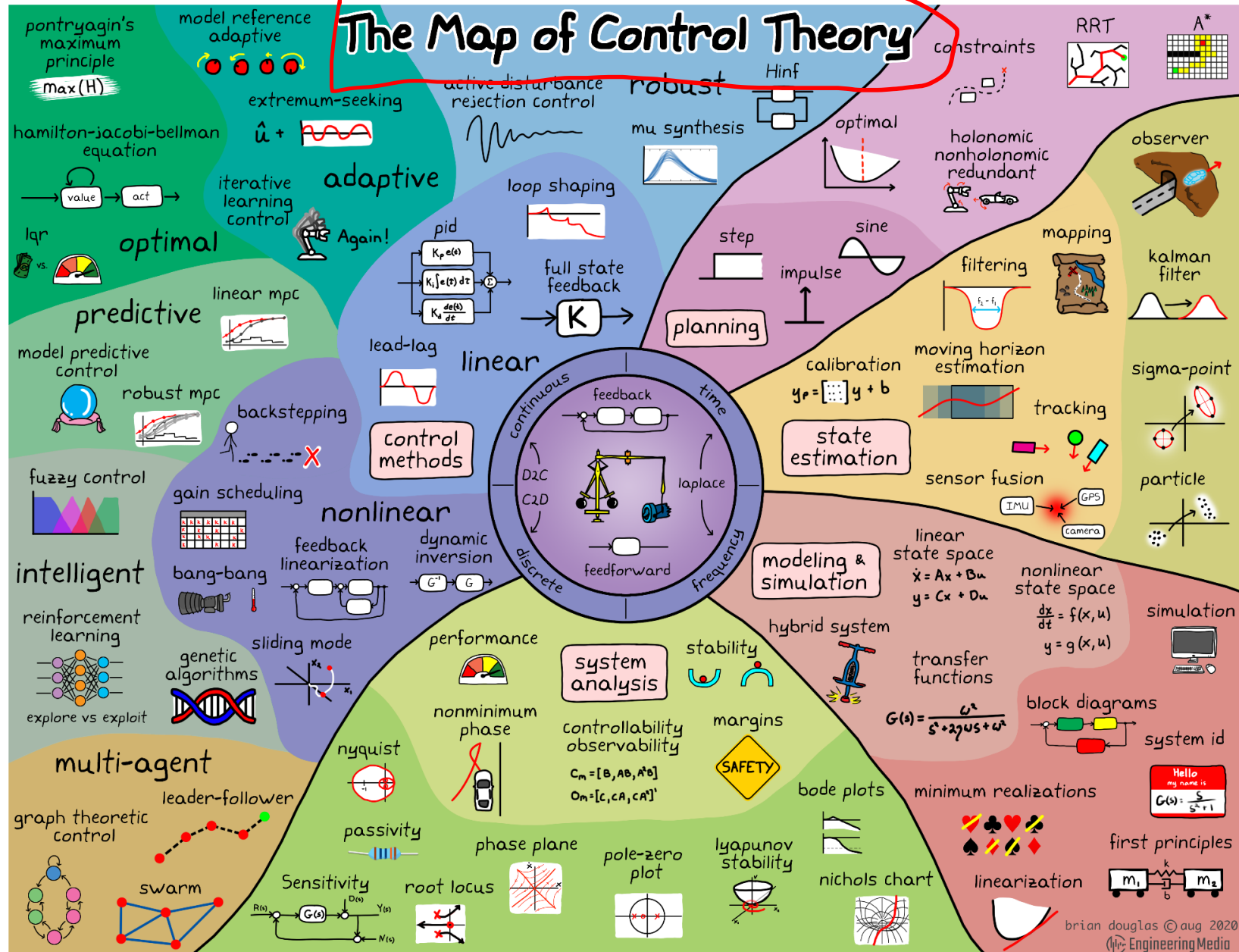
Здійснюється швидко на основі складної моделі,  
що вибудовується з досвіду,  
та зберігає інформацію на різних рівнях абстракції,  
композиційно та ієрархічно.

$$x^o = f(x, y)$$

MLP



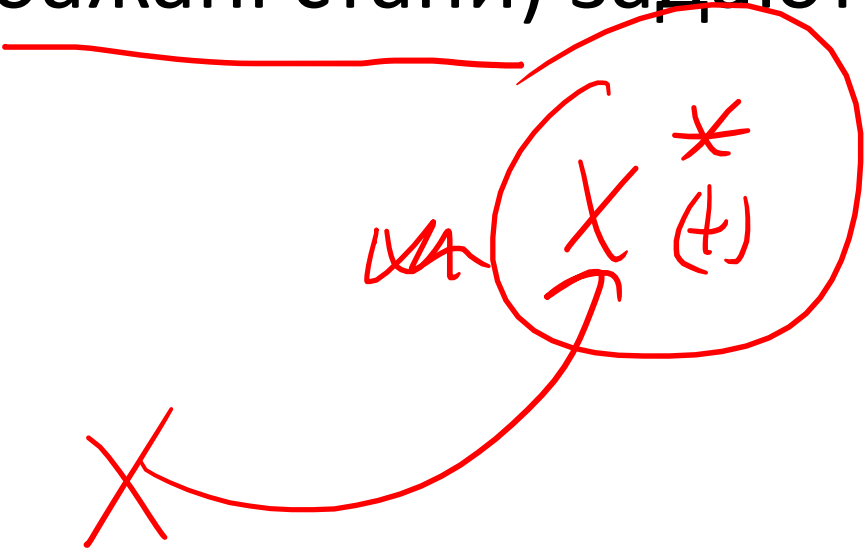
# Принцип 3. Контроль середовища (дрони)



<https://engineeringmedia.com/map-of-control>

# Принцип 3. Контроль середовища (дрони)

Сетпойнти (бажані стани) задають люди



# Принцип 3. Контроль середовища (мозок)

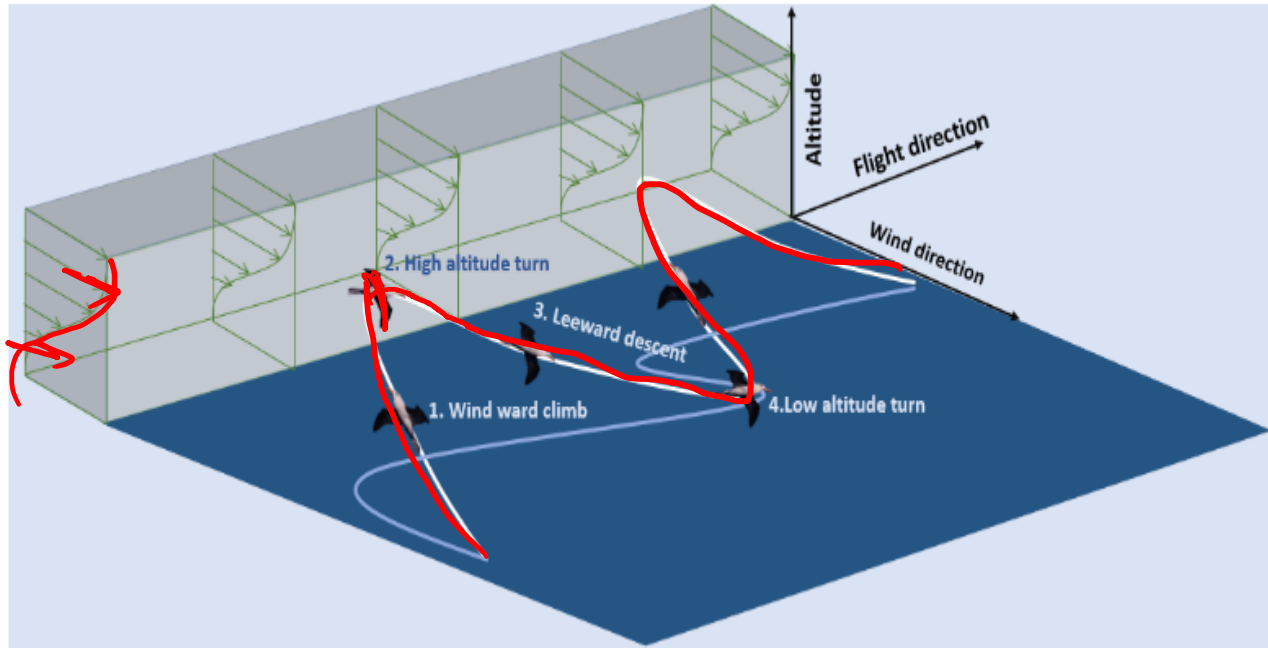
- Ієрархічні контролери
- Контроль через симуляцію
- Competence view
- Broad set of reusable skills
- Звідки беруться бажання?



AI cake by LeCun



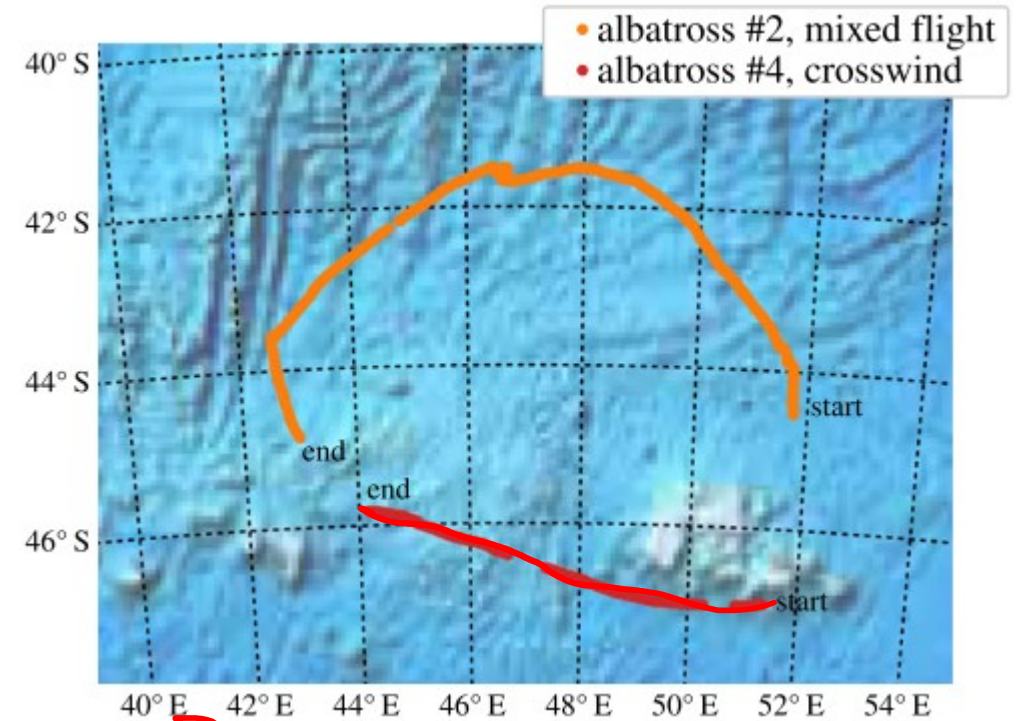
# Динамічне планерування



Eisa, S. A., & Pokhrel, S. (2023). Analyzing and Mimicking the Optimized Flight Physics of Soaring Birds: A Differential Geometric Control and Extremum Seeking System Approach with Real Time



$\chi(x)$



650 km, 9 h, approximately crosswind flight performed in 8–15 m/s winds

Bousquet, G. D., Triantafyllou, M. S., & Slotine, J. J. E. (2017). Optimal dynamic soaring consists of successive shallow arcs.

# Дрони. Перспективи

1. Краща інтеграція комп'ютерного зору
2. Оптимізація польоту (фізика, батарея, дальність ..)
3. Більш енергоефективні алгоритми
4. Пам'ять про минулі польоти
5. Краща карта середовища
6. Рої дронів

# Підсумок

1. Дрони досягли вражаючого прогресу
2. Але їм ще далеко до можливостей мозку
3. Потрібні нові підходи, подібні до біологічних
4. Щоб створити розумні машини.

# Підсумок

«Область якраз в тому стані, щоб в ній працювати»

[Russ Tedrake](#)

**UNDERACTUATED ROBOTICS**  
*Algorithms for Walking, Running, Swimming, Flying, and Manipulation*

nature communications

Perspective <https://doi.org/10.1038/s41467-023-37180-x>

**Catalyzing next-generation Artificial Intelligence through NeuroAI**

# Контакты

- В'ячеслав Осауленко
- [t.me/viosaulyenko](https://t.me/viosaulyenko)
- [osaulyenko.v.m@gmail.com](mailto:osaulyenko.v.m@gmail.com)

