Chương 2. Robot di động

TS. Phạm Duy Hưng

Khoa Điện tử - Viễn thông, Trường Đại học Công nghệ - ĐHQGHN

hungpd@vnu.edu.vn

Nội dung chương

- Cơ bản về Robot di động
- Kiến trúc điều khiển phản ứng
- Công cụ mô phỏng
- Phương pháp định vị

Đọc:

[1] Steels, Luc. "When are robots intelligent autonomous agents?." *Robotics and Autonomous systems* 15.1-2 (1995): 3-9.

[2] Brooks, Rodney. "A robust layered control system for a mobile robot." *IEEE journal on robotics and automation* 2.1 (1986): 14-23.

[3] R. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior," *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 1987, pp. 264-271, doi: 10.1109/ROBOT.1987.1088037

1. Cơ bản về robot

Issac Asimov

- ✓ Robot không làm hại người hoặc để con người bị hại.
- ✓ Robot phải tuân theo mệnh lệnh con người trừ khi mâu thuẫn với điều 1.
- ✓ Robot phải tự bảo vệ mình với hành vi tự vệ không mâu thuẫn điều 1, 2.

3D-Tasks:

- ✓**D**irty (Bẩn)
- ✓**D**ull (Tẻ nhạt)
- ✓**D**angerous (Nguy hiểm)

Tự trị (Autonomy)

- **Autonomy**: 'be relatively independent of' (Steels, 1995)
  - ✓**Tự trị thể hiện qua các mức độ:**
    - Năng lượng
    - Cảm nhận và tính toán
    - Điều khiển
  - ✓**Autonomos (tiếng Hy Lạp)** = **Autos** (bản thân) và **nomos** (quy tắc hoặc luật)
  - ✓**Quan hệ giữa Autonomy – automatic:**
    - **Automatic:** self-steering (tự lái), self-regulating (tự điều chỉnh).

      KHÔNG TỰ TẠO RA CÁC LUẬT
    - **Autonomy:** Self-regulating (tự điều chỉnh)+**Self-governing (tự quản).**

      Hệ thống tự động + tự phát triển các luật

"Khả năng của hệ thống robot thực hiện nhiệm vụ một cách độc lập, không cần sự can thiệp liên tục từ con người"

Autonomy

- **Autonomous Agent:** tự điều khiển mà không chịu sự kiểm soát của Agent từ bên ngoài. Agent phải tự có **knowledge** và **motivation** (hàm nghĩa Agent phải tự biết làm gì để thực hiện quyền điều khiển và muốn thực hiện quyền điều khiển theo cách của nó).
- **Mức độ tự trị** khác nhau phụ thuộc vào nhiệm vụ/môi trường robot hoạt động. Môi trường không thể dự đoán trước thực sự là một thách thức: tay máy, robot di động và nút cảm biến.

Ở chừng mực nào đó, các robot hiện nay không được coi là tự trị vì chúng không có mục tiêu và động lực riêng chúng, chỉ có mục tiêu và động lực của những người thiết kế chúng.

Tự trị (Autonomy)

- **Tự trị mức 1:** Con người phải luôn trong tầm nhìn của robot; luôn theo sát robot khi nó hoạt động đề phòng các sự kiện không lường trước.

  Ví dụ: máy kéo nông nghiệp có dẫn đường bằng GPS. Con người được yêu cầu ngồi trong cabin tuy nhiên máy kéo có thể tự lái theo những tuyến đường định trước.

- **Tự trị mức 2:** Người vận hành chuyển vai trò làm người giám sát (từ xa). Robot có thể nằm ngoài tầm nhìn nhưng con người vẫn phải ở trong khu vực làm việc và tiếp tục giám sát robot, xử lý những trường hợp nó cần giải cứu. Thời gian giữa các lần can thiệp tăng lên khoảng 1 giờ. Ở mức tự chủ này, con người có thể làm việc khác trong khi robot hoạt động tuy nhiên cũng chỉ có khả năng giám sát 1 đến 2 robot.

Earthsense: Agricultural Intelligence

Tự trị (Autonomy)

- **Tự trị mức 3:** khả năng triển khai quy mô lớn với nhiều robot. Nhóm robot có đủ khả năng giải quyết các trường hợp phức tạp trong vài ngày để 1 người giám sát có thể giám sát một số robot. Ở mức tự trị này, các bài toán (chẳng hạn hệ thống canh tác trong nông nghiệp) bắt đầu mở rộng quy mô. Tuy vậy, vẫn cần các khu vực kĩ thuật để đổi pin, sửa chữa hoặc giải cứu robot bị mắc kẹt.

Earthsense: Agricultural Intelligence

Tự trị (Autonomy)

- **Tự trị mức 4:** robot thực sự có thể triển khai với quy mô lớn mà không bị ràng buộc bởi các chi phí lao động. Các robot có khả năng tự giải quyết nhiều trường hợp phức tạp, trở lên tự trị đủ để con người cảm thấy không cần thiết phải có mặt trong khu vực làm việc. Các robot có khả năng tự tìm các trạm sạc pin, thay pin, thực hiện các sửa chữa nhỏ và thoát khỏi các tình huống khó khăn (có thể tìm sự giúp đỡ từ một người ở xa). Mức tự trị này yêu cầu có hạ tầng.

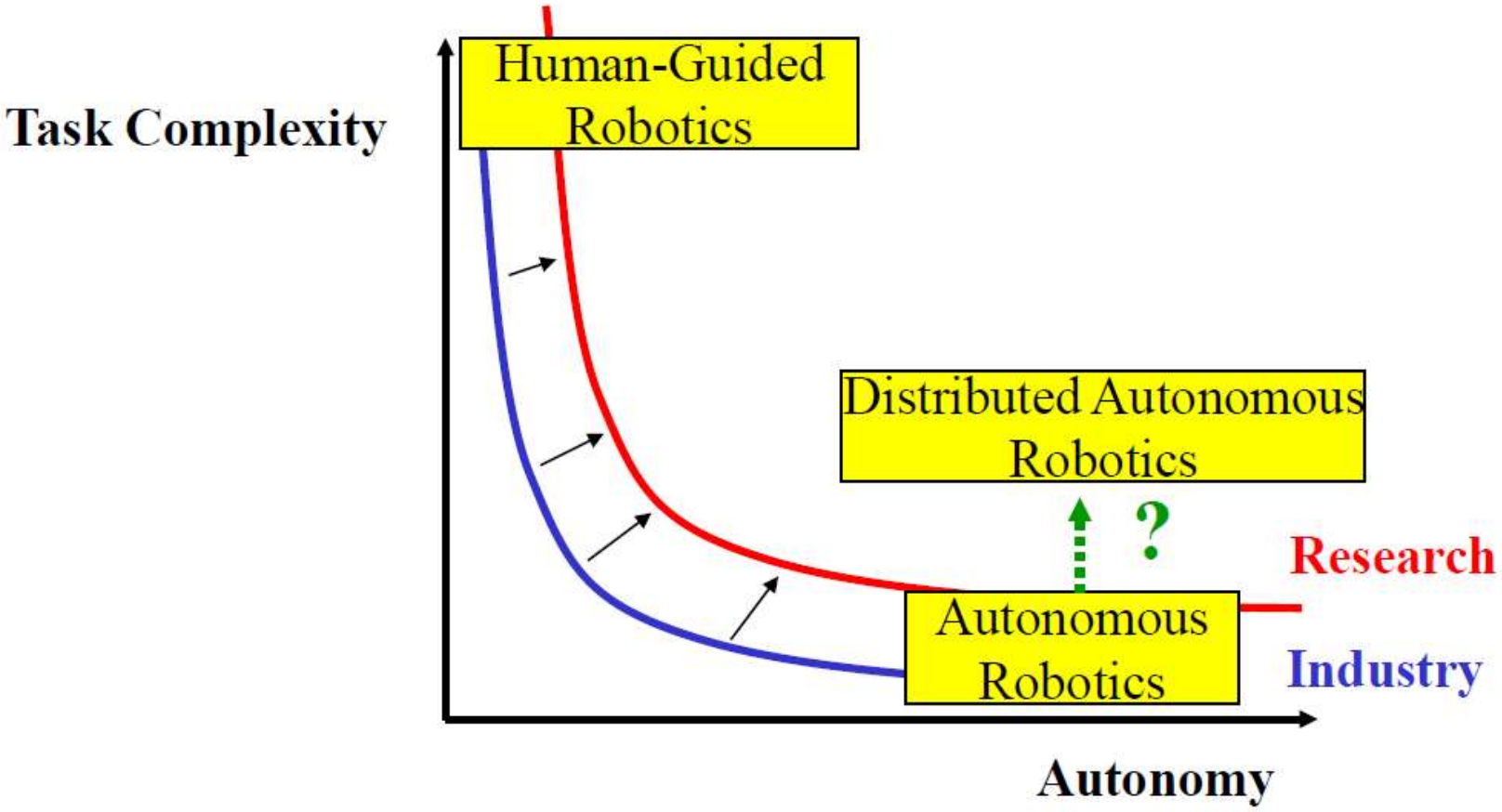
Earthsense: Agricultural Intelligence

Tự trị (Autonomy)

- **Tự trị mức 5:** các robot bắt đầu học hỏi kinh nghiệm để hoàn thiện hoạt động ngoài những gì người thiết kế đã lập trình. Các robot học hỏi lẫn nhau, tại chỗ và từ các nhóm robot khác. Chúng học cách dự đoán các sự kiện ảnh hưởng đến năng lực của chúng và lên kế hoạch một cách chủ động.

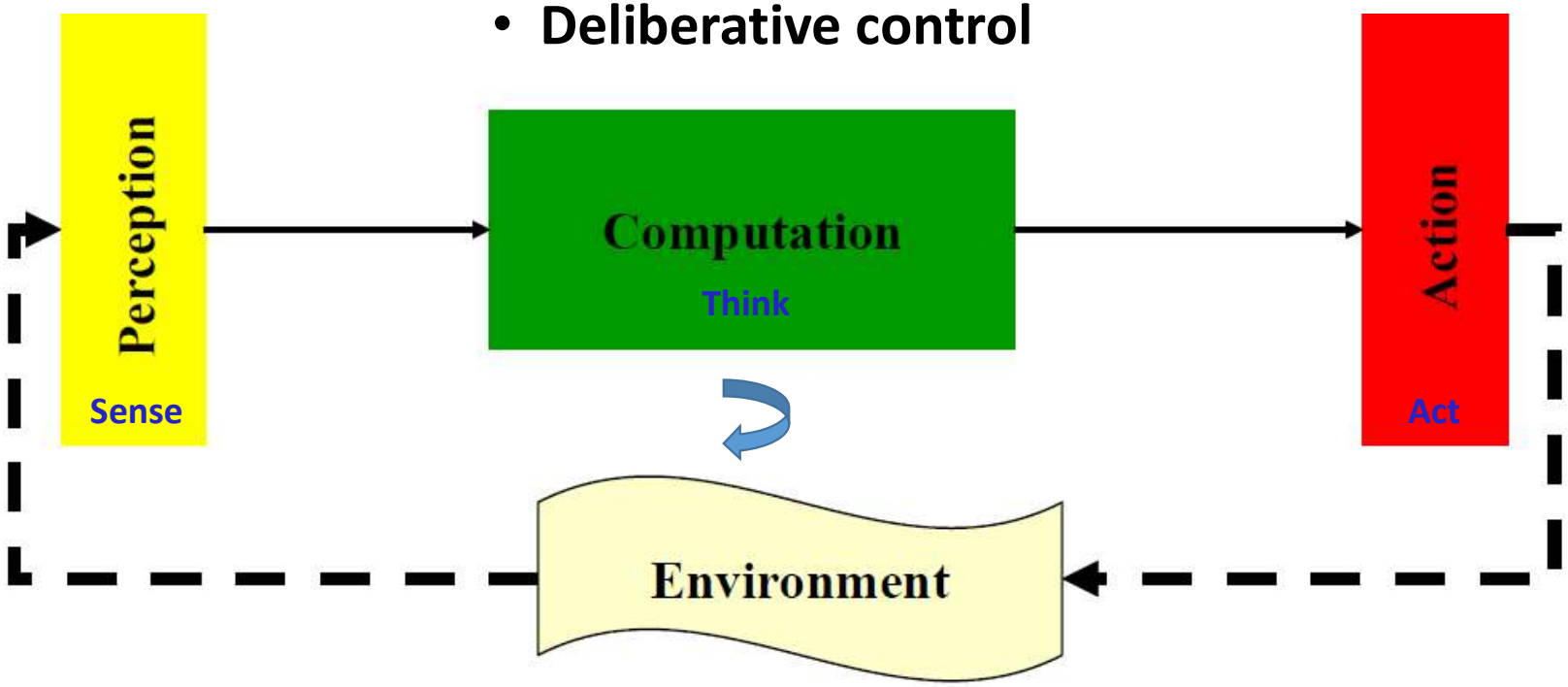
Mức	Miêu tả	Thời gian can thiệp
0	Điều khiển từ xa hoàn toàn bằng tay	N/A
1	Robot trong tầm nhìn (hands off)	5 phút
2	Người điều hành tại chỗ hoặc gần đó (eyes off)	1 giờ
3	1 người điều hành giám sát nhiều robot (mind off)	8 giờ
4	Người giám sát không có mặt tại chỗ (monitoring off)	3 ngày
5	Robots thích ứng và cải thiện khả năng thực thi (development off)	Hoạt động mở rộng

Tự trị (Autonomy)



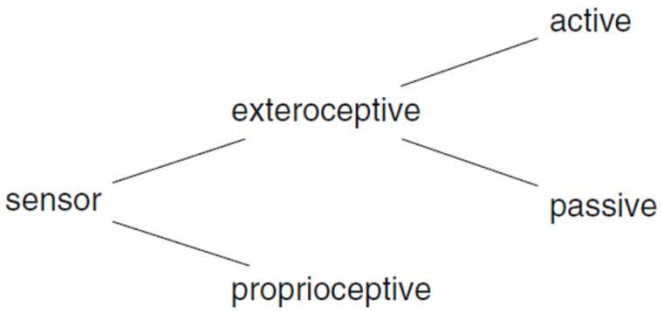
Vòng nhận thức và hành động (Perception-to-action loop)

- **Sensors**
- **Reactive control**
- **Reactive+Memory control**
- **Deliberative control**
- **Actuators**



Cảm nhận - Sensors

- **Proprioceptive** (“body”) và **exteroceptive** (“Environment”)
  - ✓Ví dụ về proprioceptive: đo vận tốc, đo góc khớp
  - ✓Ví dụ về exteroceptive: đo khoảng cách, cường độ sáng, cường độ âm thanh…
- **Passive** (“đo năng lượng xung quanh”) và **active** (“phát năng lượng vào môi trường và đo phản ứng môi trường”)
  - ✓Ví dụ về passive: đầu dò nhiệt, microphones, camera
  - ✓Ví dụ về active: laser rangefinder, IR proximity sensors, ultrasound sonars.



Chấp hành – Actuators

- **Cơ cấu chấp hành:** tạo ra lực tương tác và cơ chế thực hiện các đặc tính động học và động lực học mong muốn.
  - Điện – cơ: DC motor, Stepper motor, Servo.
  - Thủy lực
  - …
- **Cơ cấu vận động (locomotion)** khác nhau cho các mục đích khác nhau
  - ✓ Bánh xe
  - ✓ Chân
  - ✓ Bay

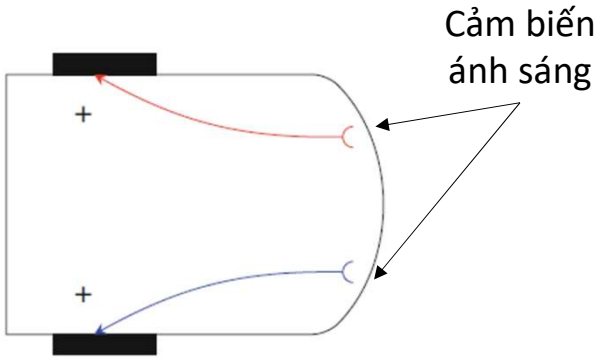
Nghĩ - Computation

- **Thường sử dụng vi điều khiển:** bộ nhớ mở rộng có thể được thêm vào.
- **Chuyển đổi:** ADC và DAC
- **Kiến trúc điều khiển:**
  - ✓Reactive control with/without memory:
    - Điều khiển phản ứng (dựa trên phản xạ reflex-based)
    - Sử dụng ước lượng trạng thái hiện tại để đưa ra các hành động
    - Ví dụ xe Braitenberg
  - ✓Deliberative control:
    - Điều khiển có kế hoạch (dựa trên kế hoạch planning-based)
    - Dự đoán các trạng thái tương lai
    - Chuỗi hành động được lập kế hoạch, tối ưu tham số
    - Ví dụ thuật toán A*, RRTs

Kiến trúc điều khiển phản ứng (Reactive control)

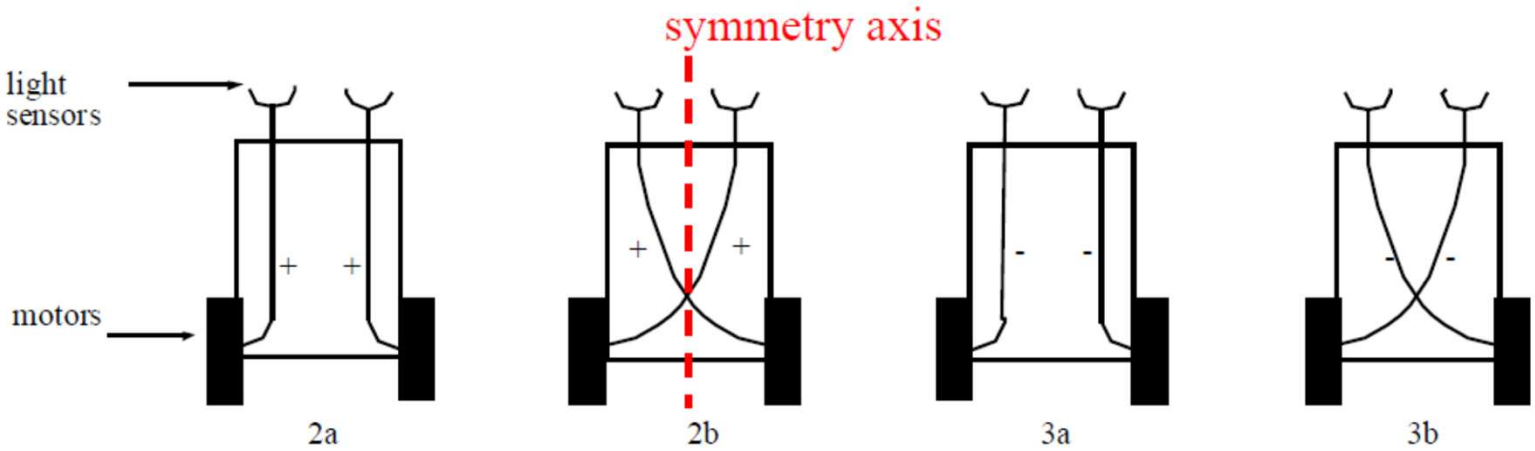
Xe Braitenberg

- **Công nghệ tưởng tượng** của Valentino Braitenberg – nhà khoa học thần kinh.
- MIT Media Lab sau đó đã chế tạo mẫu thử nghiệm.
- **Hành vi phản ứng:** hành động liên quan đến sự xuất hiện của sự kiện, và không phụ thuộc vào trạng thái (bộ nhớ).
- **Kiến trúc gần (proximal architecture):** lối ra điều khiển 'gần' với lối vào cảm biến; sử dụng toán tử tuyến tính/phi tuyến đơn giản trên dữ liệu thô.
- Các cảm biến được nối trực tiếp với mô tơ.



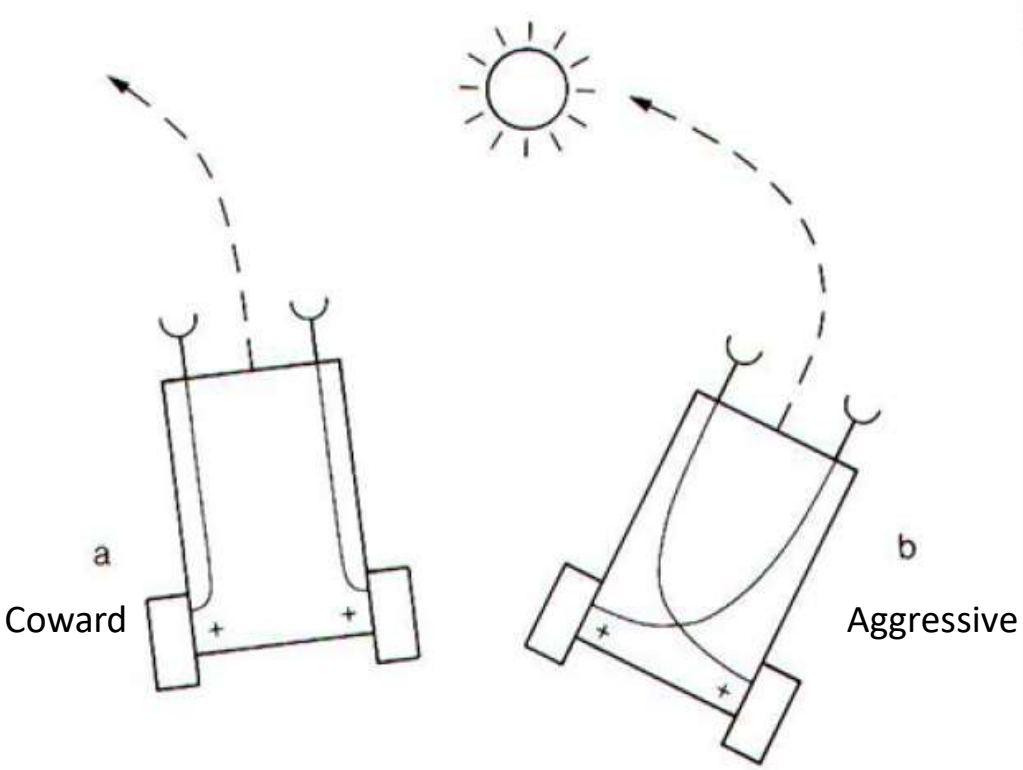
Braitenberg vehicle

Xe Braitenberg

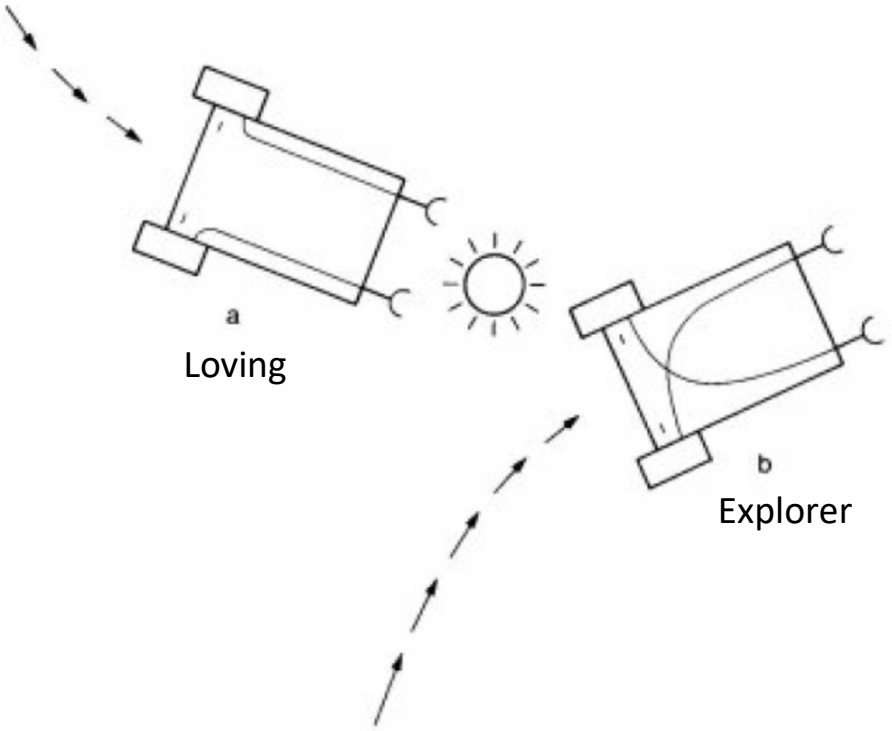


- Làm việc trên sự sai khác giữa các cảm biến
- + Kích thích (excitation); - ức chế (inhibition);
- Bộ điều khiển tuyến tính: **Đầu ra=hệ số có dấu * đầu vào**

Kiến trúc phản ứng với Braitenberg



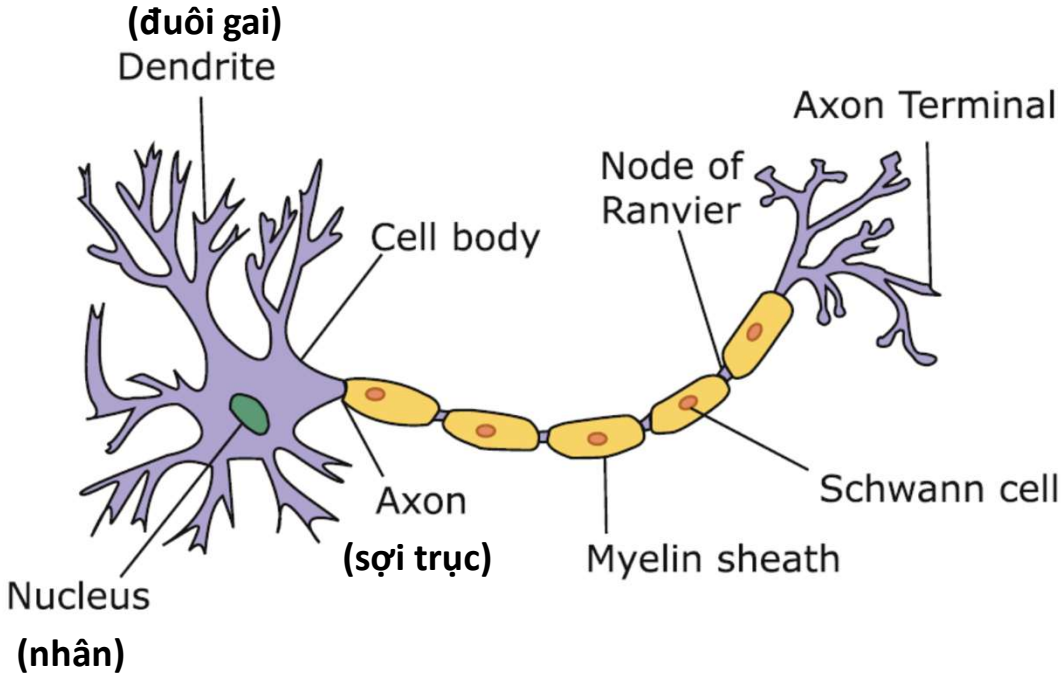
Kích thích



Ức chế

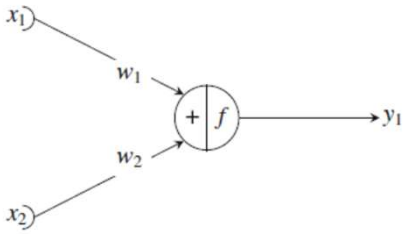
Kiến trúc phản ứng với mạng Nơron nhân tạo

- **Số lượng tế bào thần kinh não:** 100 tỷ (người)>>71 triệu (chuột)
- **Nhiệm vụ của tế bào thần kinh là xử lý và truyền thông tin trong cơ thể.** Xung đầu vào được nhận qua khớp thần kinh vào các đuôi gai, truyền đến nhân - nơi xử lý các xung; và truyền xung đầu ra qua sợi trục.
- **Các khớp thần kinh đóng vai trò điều chỉnh việc truyền tín hiệu, có tính thích nghi => Yếu tố chính tạo nên khả năng nhớ và học tập.**



Cấu trúc 1 nơ ron thần kinh gồm nhân (nucleus) và một sợi trục dài (axom) cho phép kết nối với các nơ ron khác thông qua đuôi gai (dendrite) nhờ khớp thần kinh (synapse).

Kiến trúc phản ứng với mạng Nơron nhân tạo



Mạng nơ ron nhân tạo

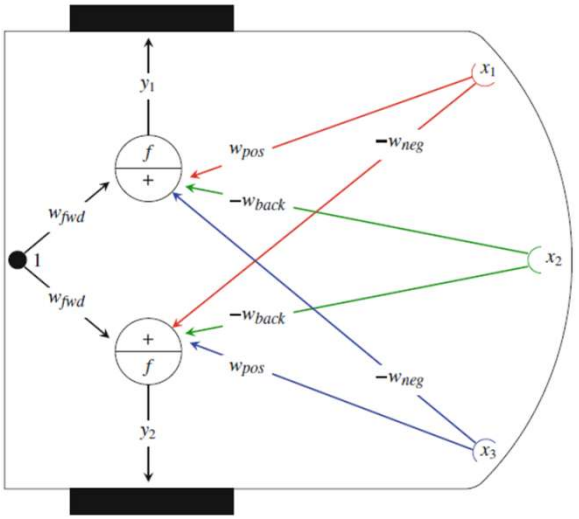
Ký hiệu	Ý nghĩa
f	Hàm lối ra nơ ron
+	Tổng các lối vào
x_i	Lối vào
y_i	Lối ra
w_i	Trọng số lối vào

VD: bài toán tránh chướng ngại vật

Đặc điểm kỹ thuật:

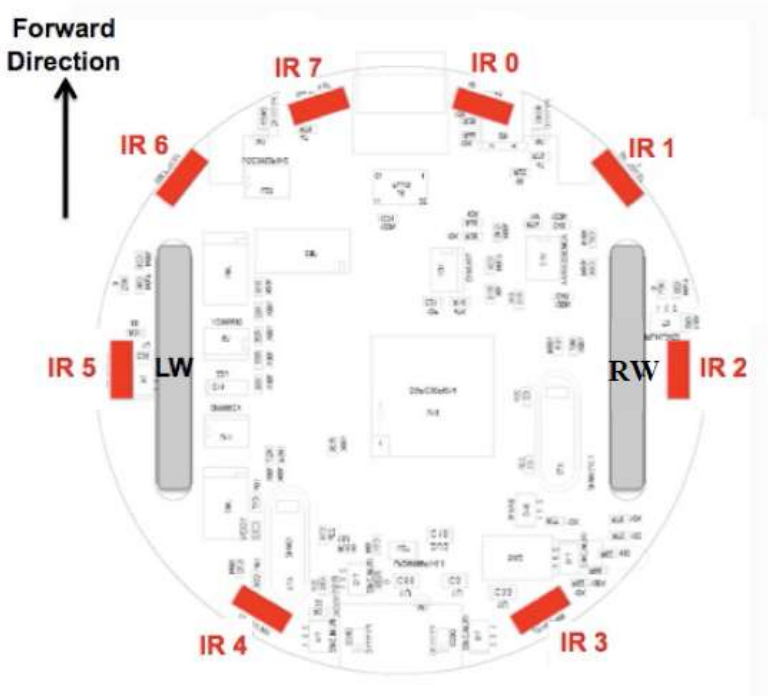
Robot có ba cảm biến hướng về phía trước.

- Robot di chuyển về phía trước trừ khi phát hiện chướng ngại vật.
- Nếu cảm biến trung tâm phát hiện chướng ngại vật, robot sẽ lùi từ từ về phía sau.
- Nếu cảm biến bên trái phát hiện chướng ngại vật, robot sẽ rẽ phải.
- Nếu cảm biến bên phải phát hiện chướng ngại vật, robot sẽ rẽ trái.



Ký hiệu	Ý nghĩa
W_{fwd}	Trọng số chuyển động tiến
W_{back}	Trọng số chuyển động lùi
W_{pos}	Trọng số quay bánh xe dương
W_{neg}	Trọng số quay bánh xe âm

Thực thi cho robot e-puck



- 2 actuators
- 8 proximity sensors
- Tốc độ mô tơ là tổ hợp tuyến tính

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} \alpha_{L0} & \alpha_{L1} & \cdots & \alpha_{L7} \\ \alpha_{R0} & \alpha_{R1} & \cdots & \alpha_{R7} \end{bmatrix} \cdot \begin{bmatrix} d_{IR0} \\ \vdots \\ d_{IR7} \end{bmatrix} + \begin{bmatrix} v_{L0} \\ v_{R0} \end{bmatrix}$$

Bias

Kiến trúc phản ứng dựa trên quy tắc (Rule-based)

QUY TẮC NẾU –THÌ (IF – THEN)

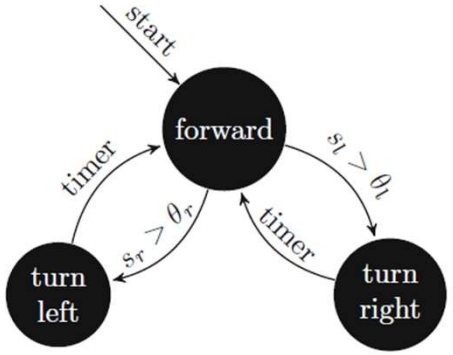
Ví dụ trường hợp tránh vật cản với 3 cảm biến:

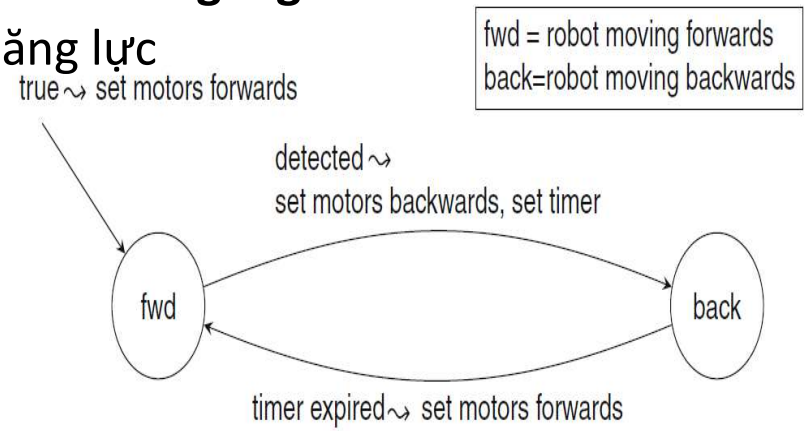
- **Rule 1:**
  - **Nếu** cảm biến tiệm cận bên trái kích hoạt **thì** rẽ phải
- **Rule 2:**
  - **Nếu** cảm biến tiệm cận bên phải kích hoạt **thì** rẽ trái
- **Rule 3:**
  - **Nếu** không cảm biến nào được kích hoạt **thì** đi thẳng

Điều khiển phản ứng với kiến trúc gộp

- **Kiến trúc gộp (Subsumption architecture):**
  - ✓Rodney Brooks 1986, MIT: tiếp cận cắt ngang (theo năng lực) # theo chức năng
  - ✓ Levels and Layers: levels of competence và layers of control.
  - ✓ Mỗi layers là một **mô đun**, ánh xạ trực tiếp từ trạng thái cảm nhận (sensing) thành hành động (action).
  - ✓Các mô đun được biểu diễn bởi trạng thái máy hữu hạn FSM (Finite State Machines) => **A robot control system specification language**
  - ✓Các mô đun hoạt động gối lên nhau => tăng năng lực

FSM cho hành vi tránh va chạm

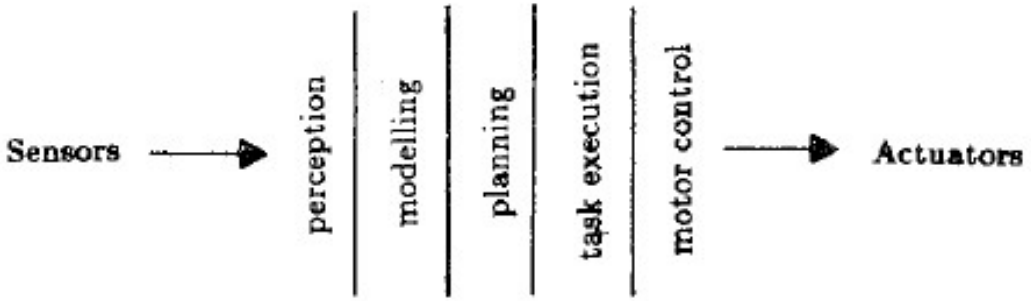




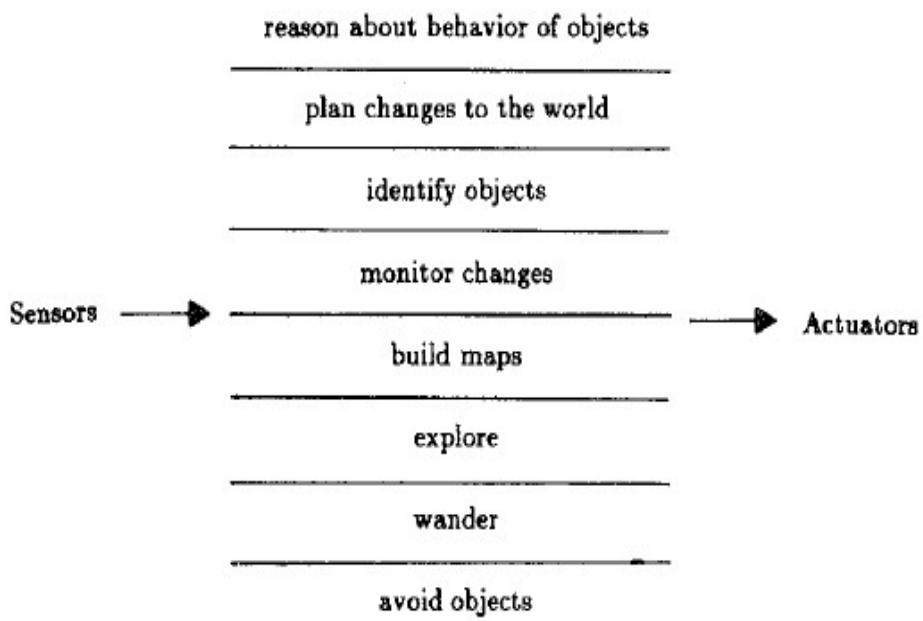
FSM cho xe Braitenberg

Điều khiển phản ứng với kiến trúc gộp

- Kiến trúc gộp (Subsumption architecture):



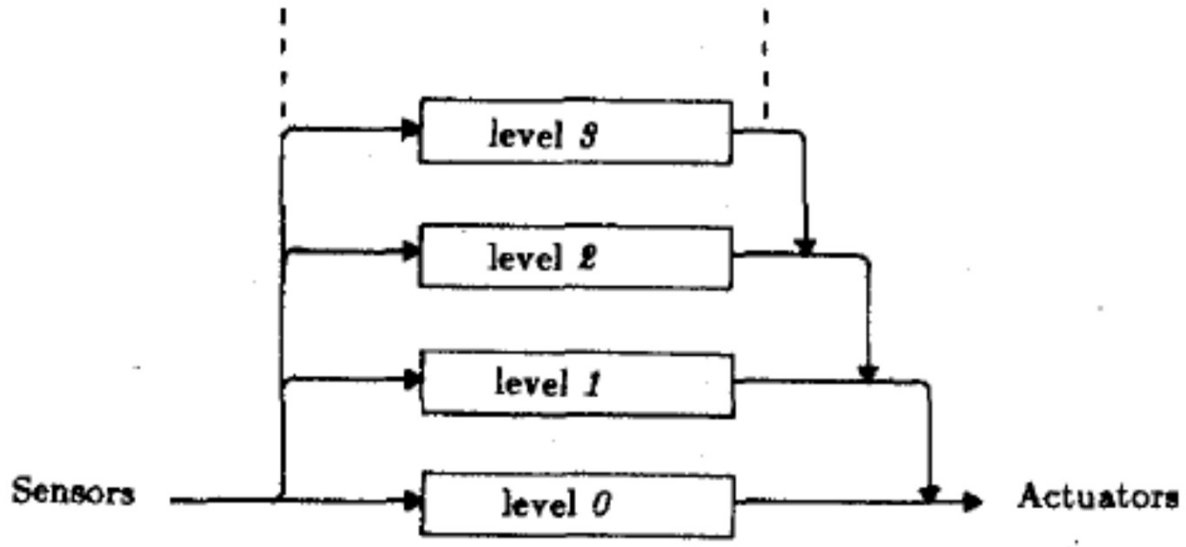
Mô hình cổ điển (nối tiếp); nhấn mạnh vào điều khiển kế hoạch (deliberative control)



Kiến trúc gộp (song song); nhấn mạnh vào điều khiển phản ứng (reactive control)

Điều khiển phản ứng với kiến trúc gộp

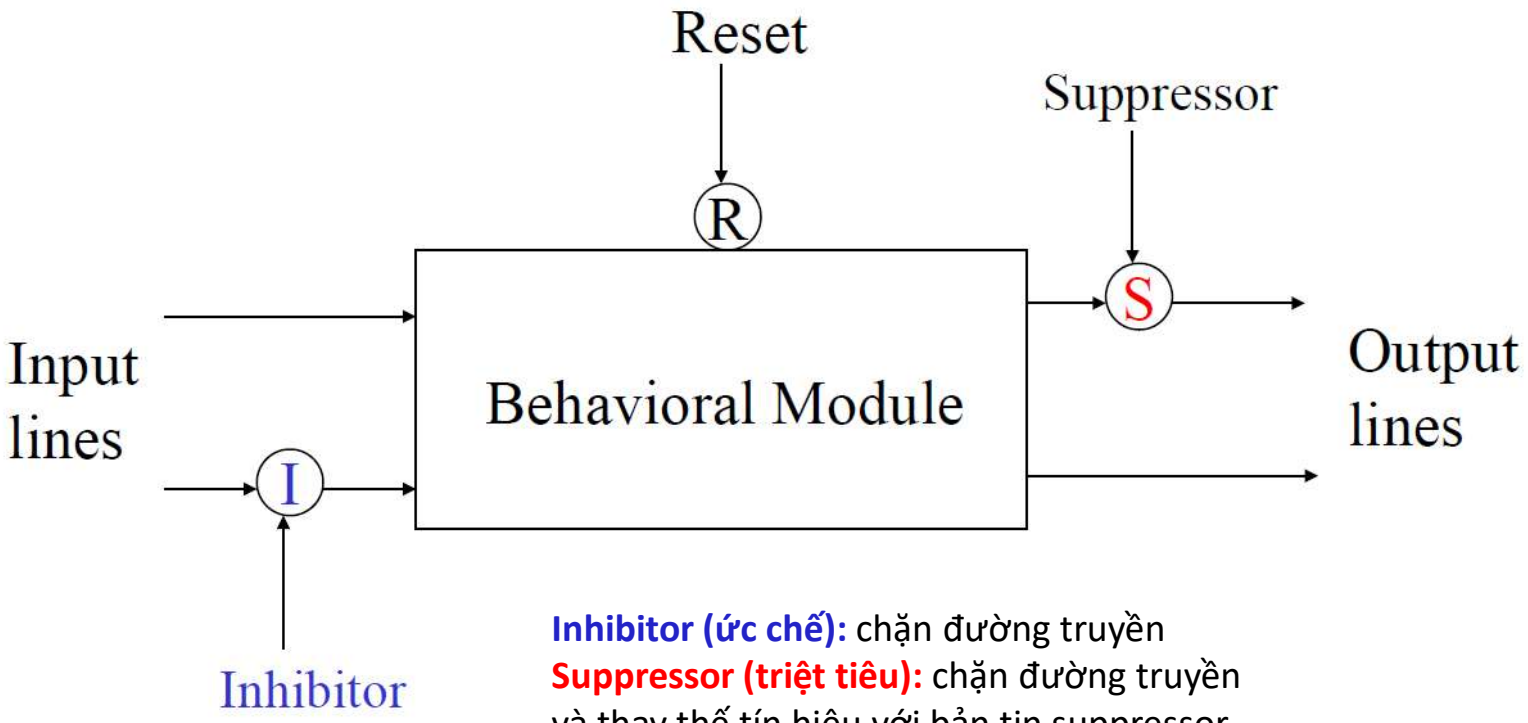
- Kiến trúc gộp (Subsumption architecture):



Cấu trúc đa tầng: cấp cao hơn có thể thay thế vai trò của cấp thấp hơn

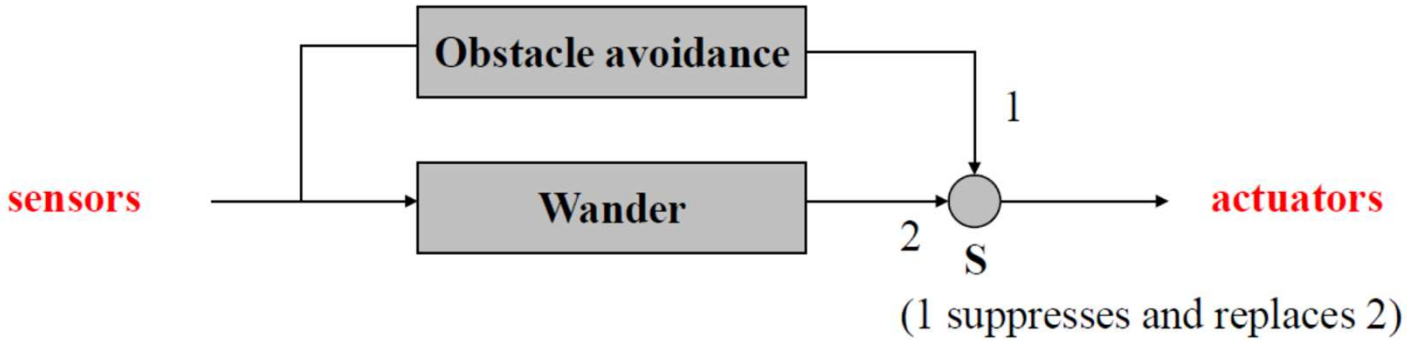
Điều khiển phản ứng với kiến trúc gộp

- Kiến trúc gộp (Subsumption architecture):



Inhibitor (ức chế): chặn đường truyền
Suppressor (triệt tiêu): chặn đường truyền và thay thế tín hiệu với bản tin suppressor

Điều khiển phản ứng với kiến trúc gộp

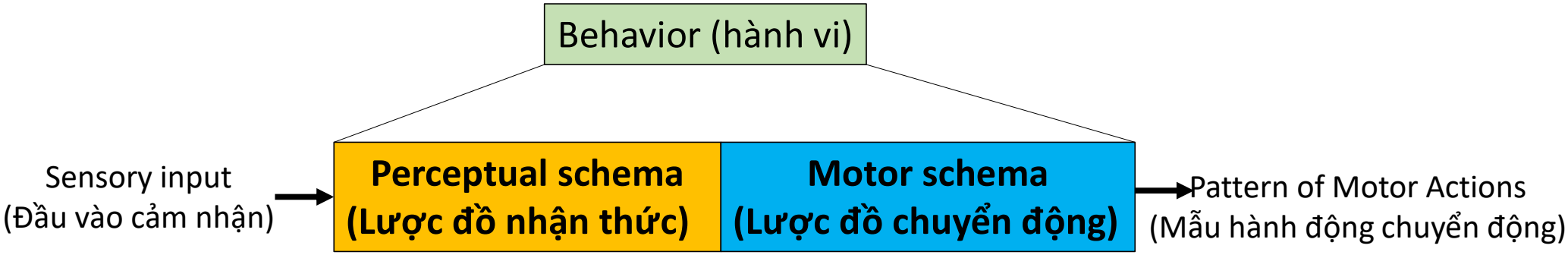


Điều khiển phản ứng với kiến trúc gộp

- ✓Hỗ trợ hành vi song song, mỗi lớp hành vi có thể chạy độc lập và không đồng bộ.
- ✓Thời gian thực thi nhanh
- — Hạn chế trong việc hỗ trợ thiết kế cách mô đun lớp trên độc lập với lớp dưới.

Điều khiển phản ứng với lược đồ chuyển động

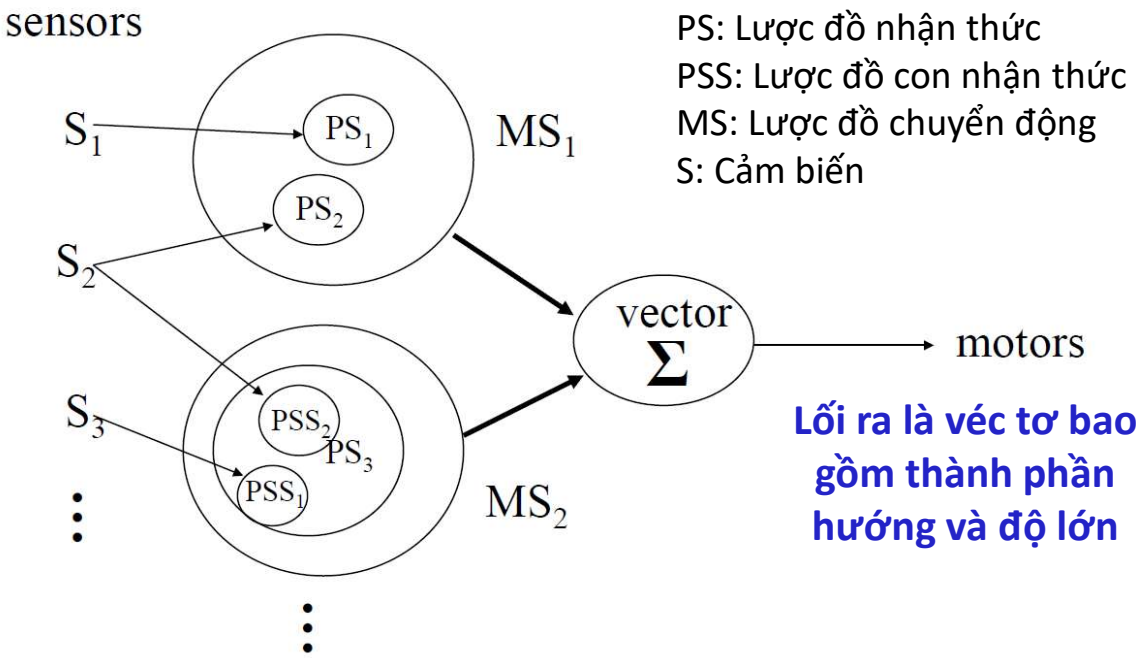
- **Lược đồ chuyển động: Motor Schemas**
  - ✓Ronald Arkin 1987, Georgia Tech



- ✓ Lược đồ là mẫu hành động: khối chức năng trung gian giữa nhận thức và hành động
- ✓ Từ các hành động cơ bản -> tạo ra hành động phức tạp
- ✓ Đáp ứng hành vi được trình bầy dưới dạng véc tơ, tạo ra bằng phương pháp trường thế
- ✓ Không có phân cấp, mỗi hành vi có thể đóng góp ở mức độ khác nhau

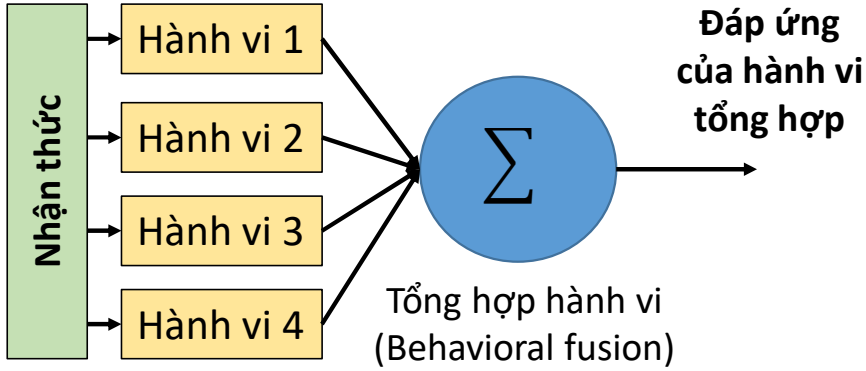
Điều khiển phản ứng với lược đồ chuyển động

- **Lược đồ chuyển động (Motor Schemas)**



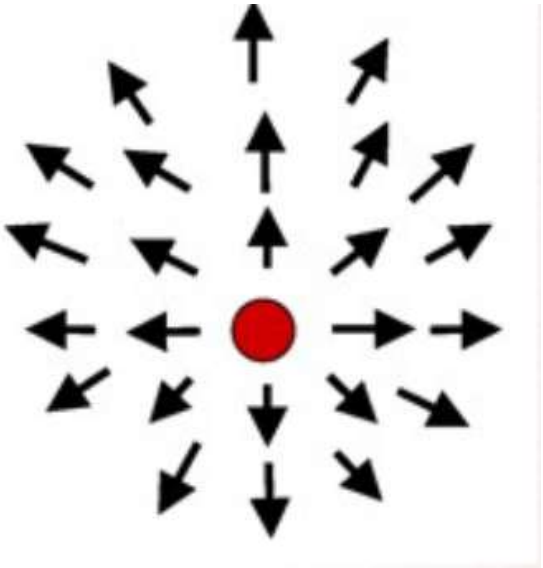
PS: Lược đồ nhận thức
PSS: Lược đồ con nhận thức
MS: Lược đồ chuyển động
S: Cảm biến

Lối ra là véc tơ bao gồm thành phần hướng và độ lớn

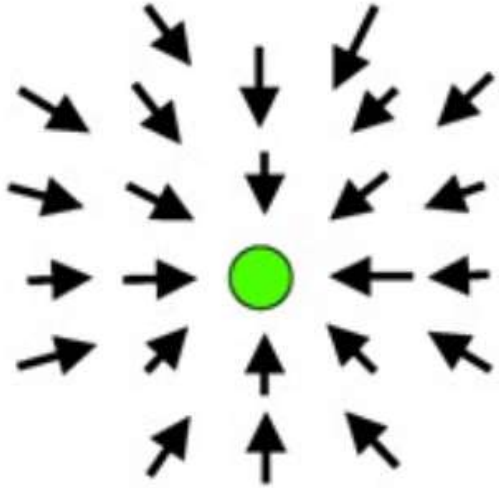


Điều khiển phản ứng với lược đồ chuyển động

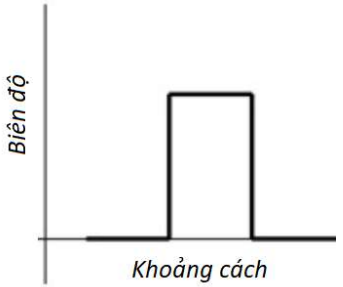
- **Trường thế nhân tạo APF (Artificial Potential Field)**



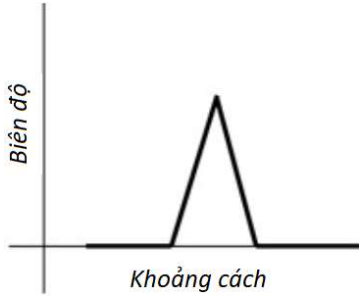
Đẩy (Repulsion)



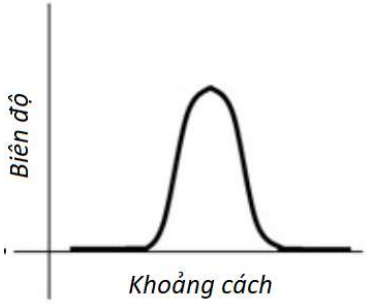
Hút (Attraction)



Dạng hằng số

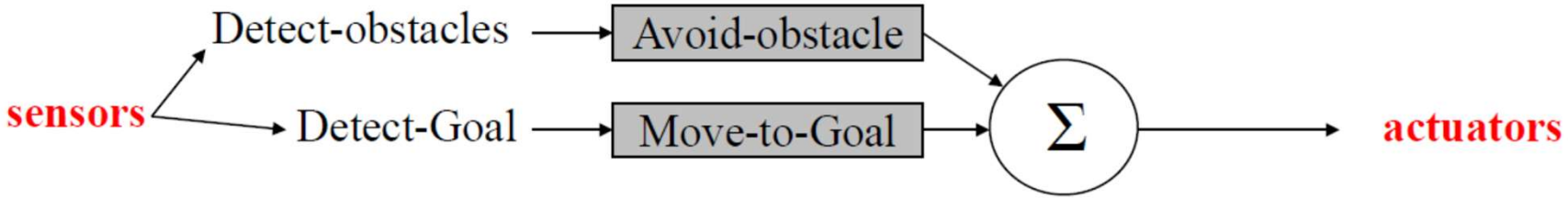


Dạng tuyến tính



Dạng mũ

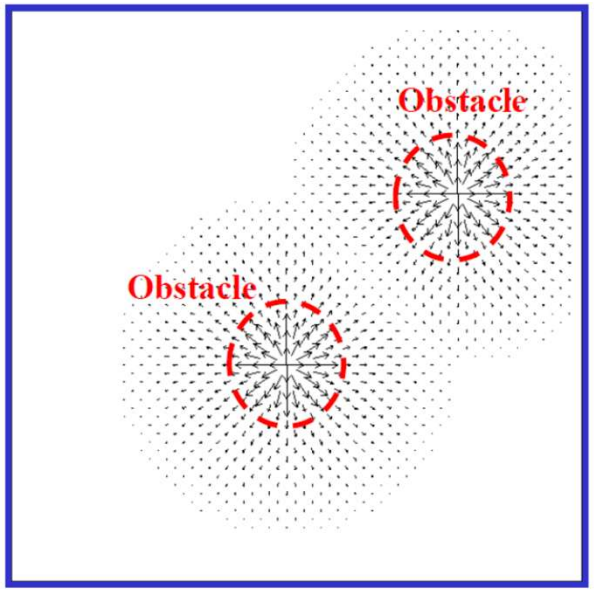
Điều khiển phản ứng với lược đồ chuyển động



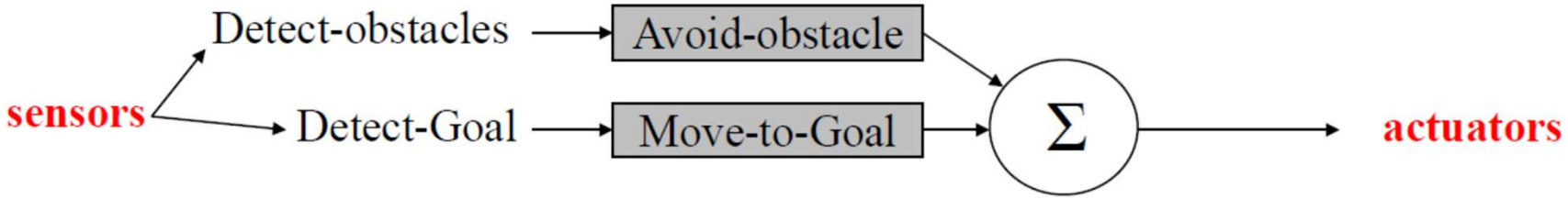
- **Tránh vật cản tĩnh (Avoid-static-Obstacle)**

$$V_{magnitude} = \begin{cases} 0 & for \quad d > S \\ \frac{S-d}{S-R} G & for \quad R < d \le S \\ \infty & for \quad d \le R \end{cases}$$

S: phạm vi ảnh hưởng của chướng ngại vật
R: bán kính của vật cản
G: hệ số
d: khoảng cách từ robot tới tâm vật cản
$V_{magnitude}$: véc tơ dọc theo đường nối giữa robot và vật cản, chiều hướng ra khỏi vật cản



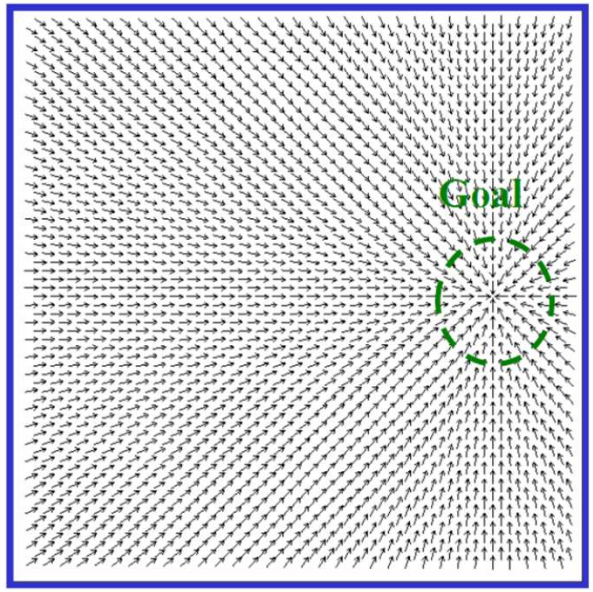
Điều khiển phản ứng với lược đồ chuyển động



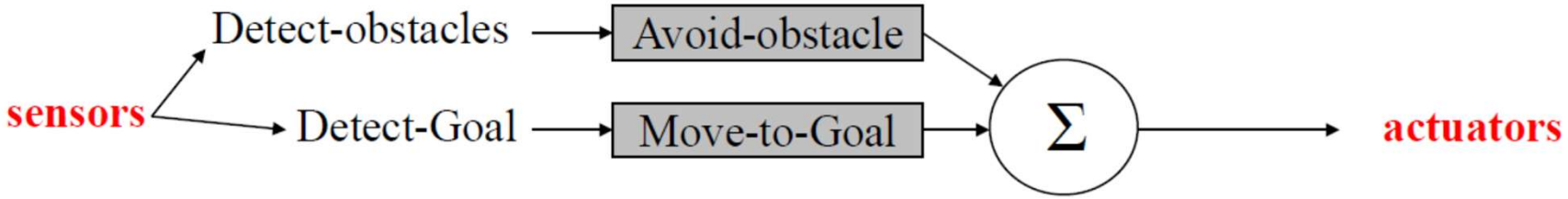
- **Di chuyển đến mục tiêu (Move-to-goal)**

Output = vector = (r, φ)
(Biên độ và hướng)

$V_{magnitude}$= giá trị hệ số cố định
$V_{direction}$= hướng tới mục tiêu mong đợi



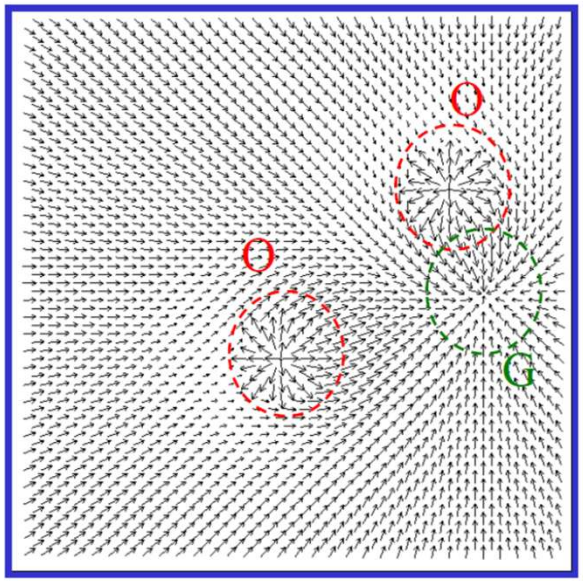
Điều khiển phản ứng với lược đồ chuyển động



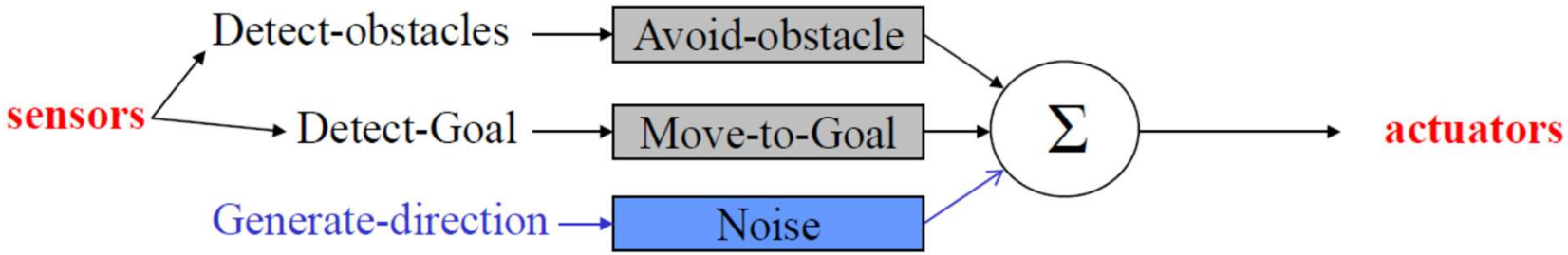
- **Move-to-goal + Avoid obstacle**

Tổng hợp tuyến tính (Tổng có trọng số)

=> Cực tiểu cục bộ (local minima)



Điều khiển phản ứng với lược đồ chuyển động



- ✓ Thêm nhiễu là giải pháp đơn giản để tránh bị mắc kẹt trong cực tiểu cục bộ bằng cách sử dụng các véc tơ ngẫu nhiên. => Random walk (wander)
- ✓ Tiếp cận khác phức tạp hơn: thiết kế các hàm thế cẩn thận để không tạo ra cực tiểu cục bộ.

Điều khiển phản ứng với lược đồ chuyển động

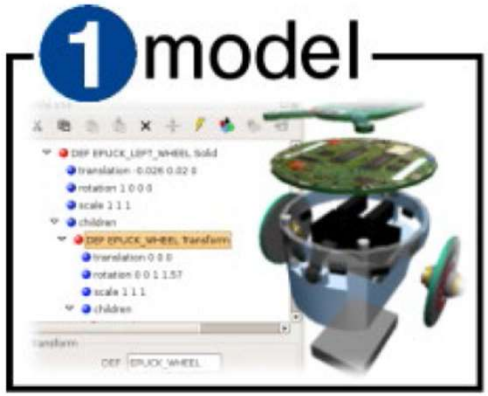
- Đánh giá lược đồ chuyển động
  - ✓Hỗ trợ các hành vi đồng thời một cách tự nhiên
  - ✓Có thể trộn hành vi được tinh chỉnh.
  - — Các vấn đề của phương pháp trường thế (potential field)
  - — Có thể chậm và chi phí tính toán lớn

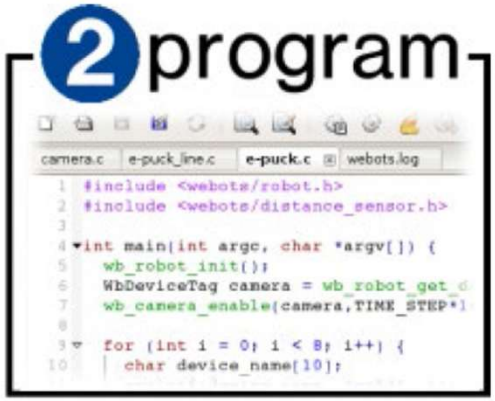
2. Các công cụ mô phỏng được sử dụng trong khóa học

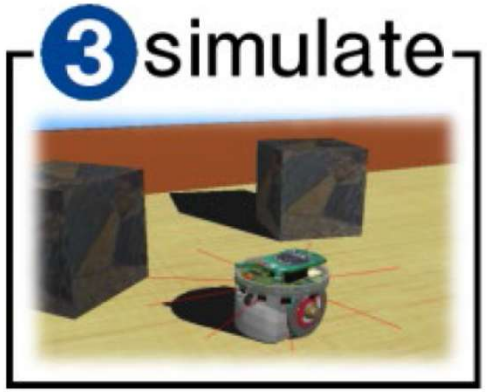
Mô phỏng: tại sao?

- Tạo mẫu phần cứng tốn nhiều thời gian và nói chung đắt hơn
- Linh hoạt trong thiết lập thí nghiệm
- Dễ dàng hơn để giám sát các thí nghiệm (và đánh giá các số liệu)
- Đánh giá các thuật toán trong các thí nghiệm khó thực hiện trong thực tế (như số lượng lớn robot) hoặc thâm chí không thể)như cảm biến không nhiễu).
- Dự đoán kết quả trước khi xây dựng phần cứng

Webots: trình mô phỏng độ chân thực cao





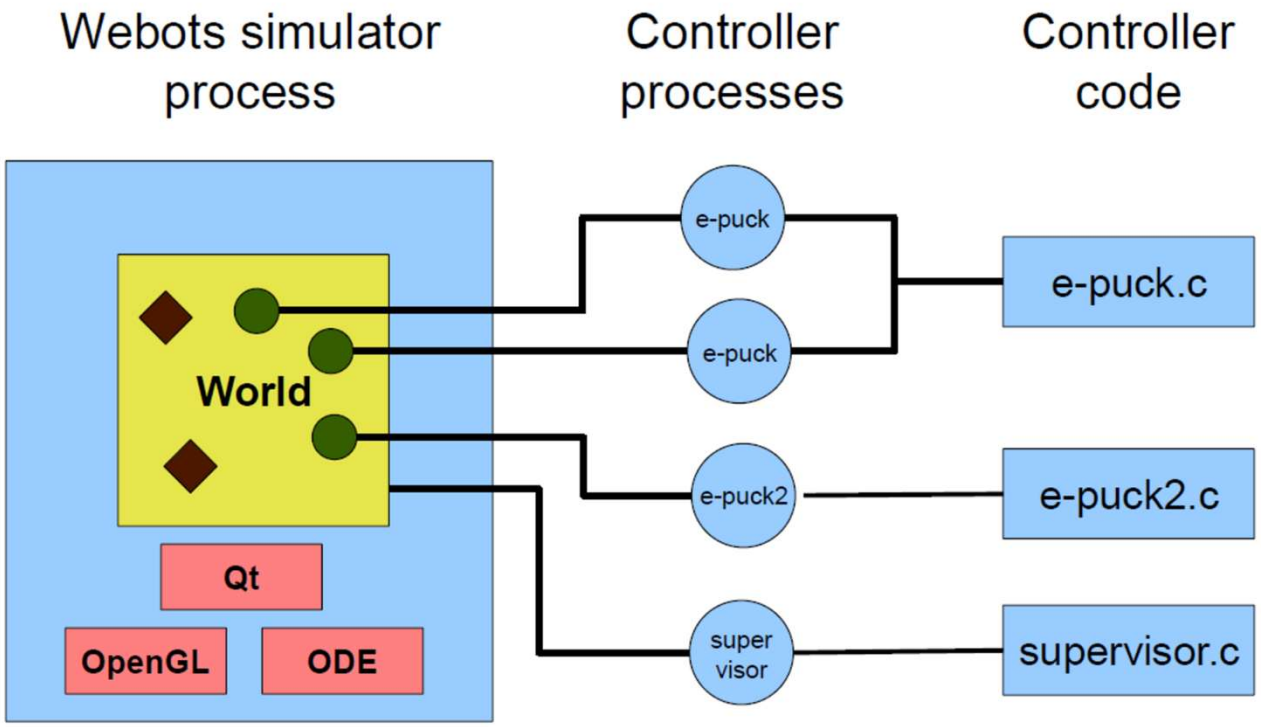


Trong khóa học này, chúng ta chỉ tập trung vào bước 2 và 3.

Tính năng của webots

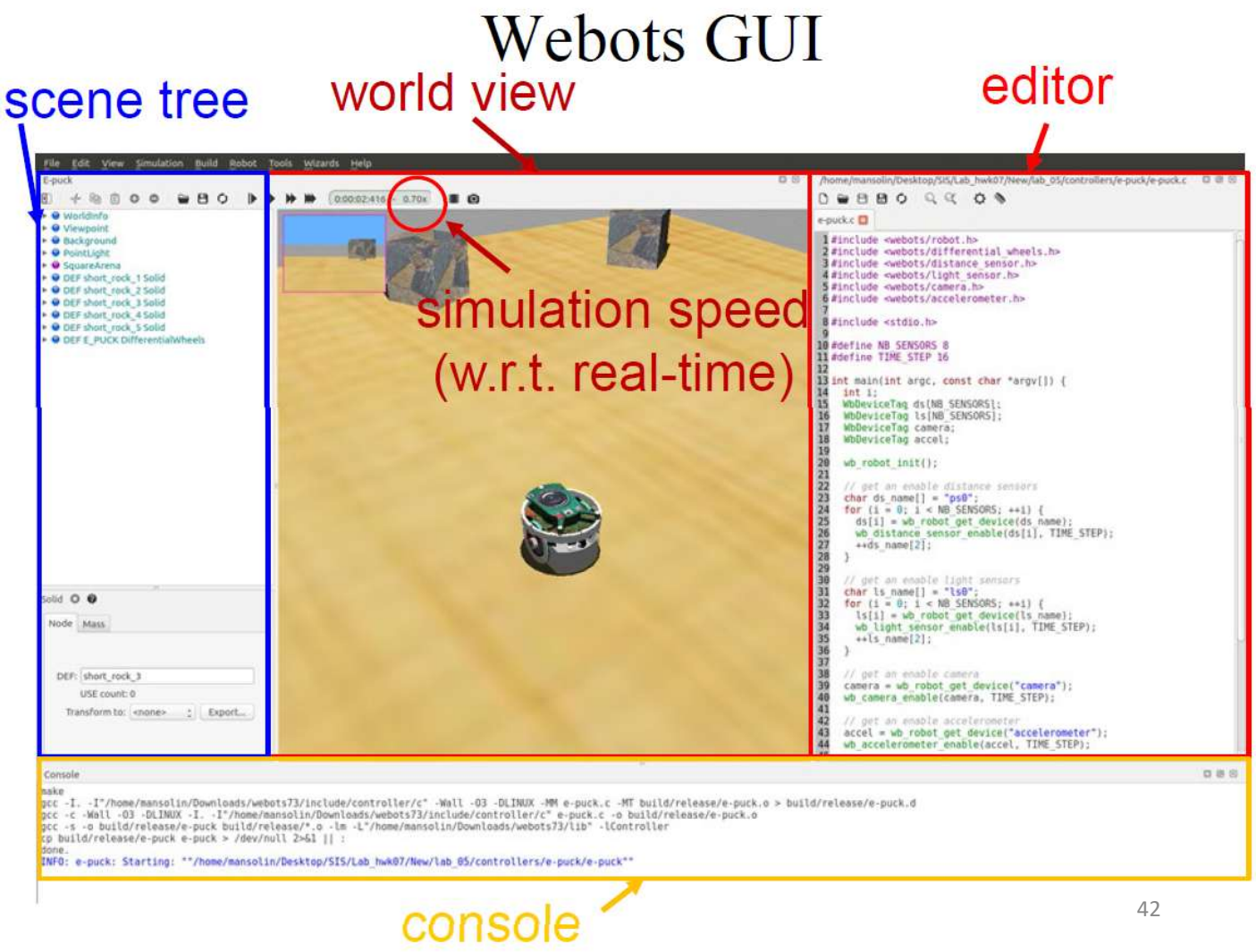
- Từ 1/2019, mã nguồn mở
- Trình mô phỏng chân thật môi trường vật lý
- Có thể điều chỉnh sự cân bằng giữa độ chân thật và chi phí tính toán thông qua mode động học và động lực học.
- Tạo mẫu nhanh; có thể giao tiếp với ROS
- Thường có thể chạy nhanh hơn thơi gian thực
- Các cảm biến có sẵn: cảm biến khoảng cách, cảm biến ánh sáng, camera, gia tốc, cảm biến chạm, cảm biến vị trí, GPS, cảm biến lực,…
- Các cơ cấu chấp hành có sẵn: động cơ, đầu kẹp, …

Nguyên lý của webots



Nhiều robot hơn, mô phỏng sẽ chậm hơn

Webot GUI


Webots GUI
scene tree
world view
editor
simulation speed
(w.r.t. real-time)
console

Bài tập

- BT1: thực thi lược đồ chuyển động cho e-puck sử dụng webots cho bài toán di chuyển đến mục tiêu và tránh va chạm.
- BT2: tránh va chạm nhiều e-puck sử dụng webots