



UNIVERSITÉ
LAVAL

GLO-4001/7021

INTRODUCTION À LA ROBOTIQUE

MOBILE

Plan de cours

Présentation générale

Historique

Révision mathématique

Automne 2017

Philippe Giguère

Plan de cours

Informations générales

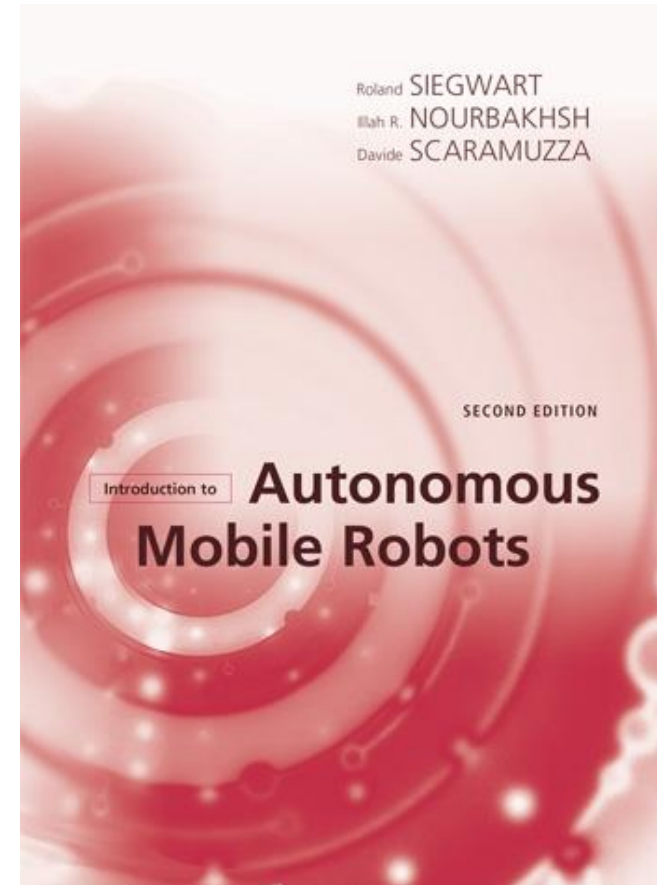
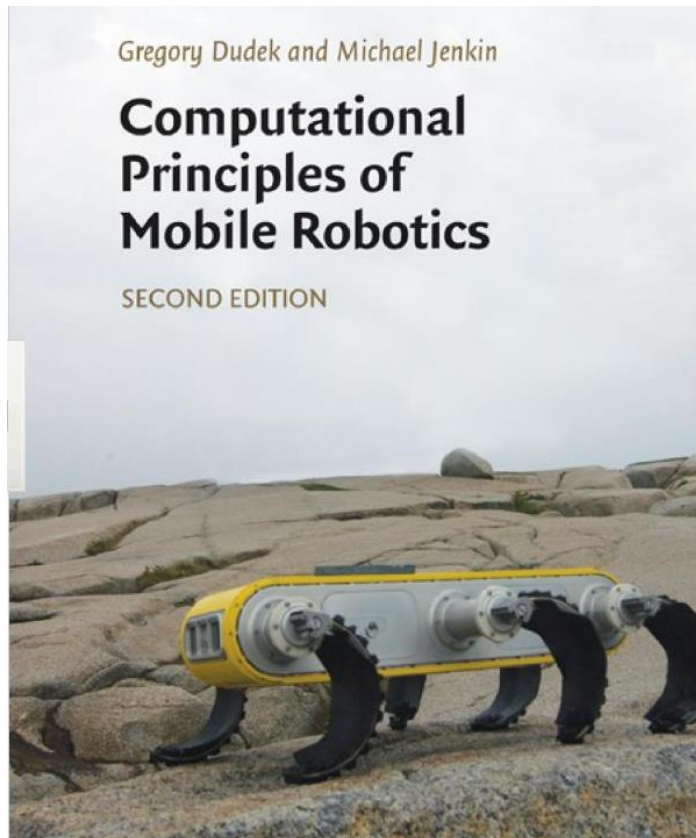
Site web du cours

<http://www.ift.ulaval.ca/~pgiguere/cours/IntroRobotique/index.html>

Disponibilités : mercredi 14h00 à 16h00 PLT-3976

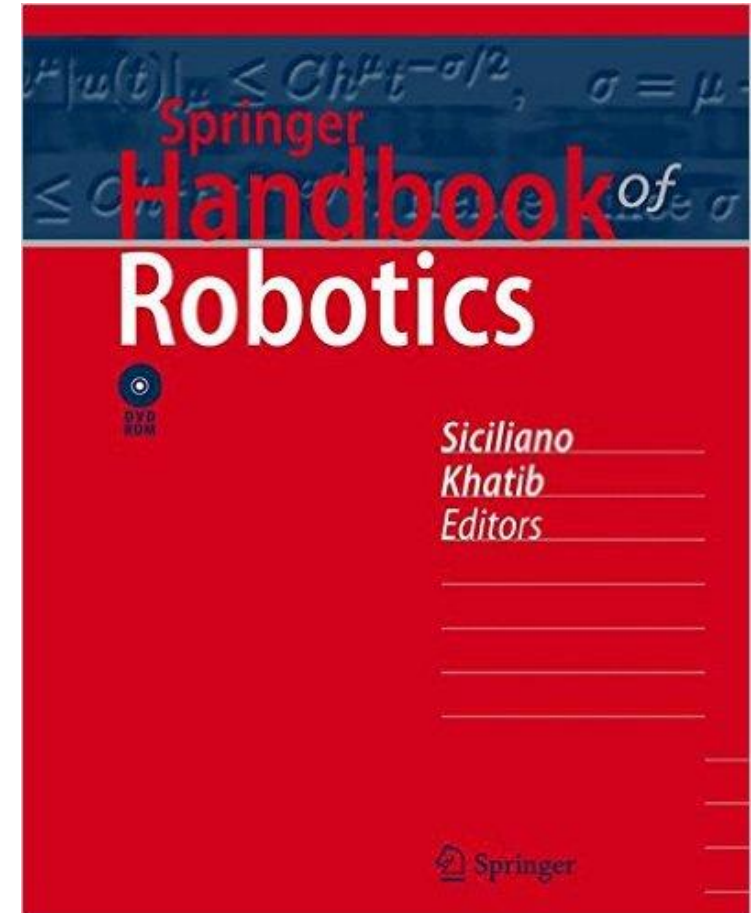
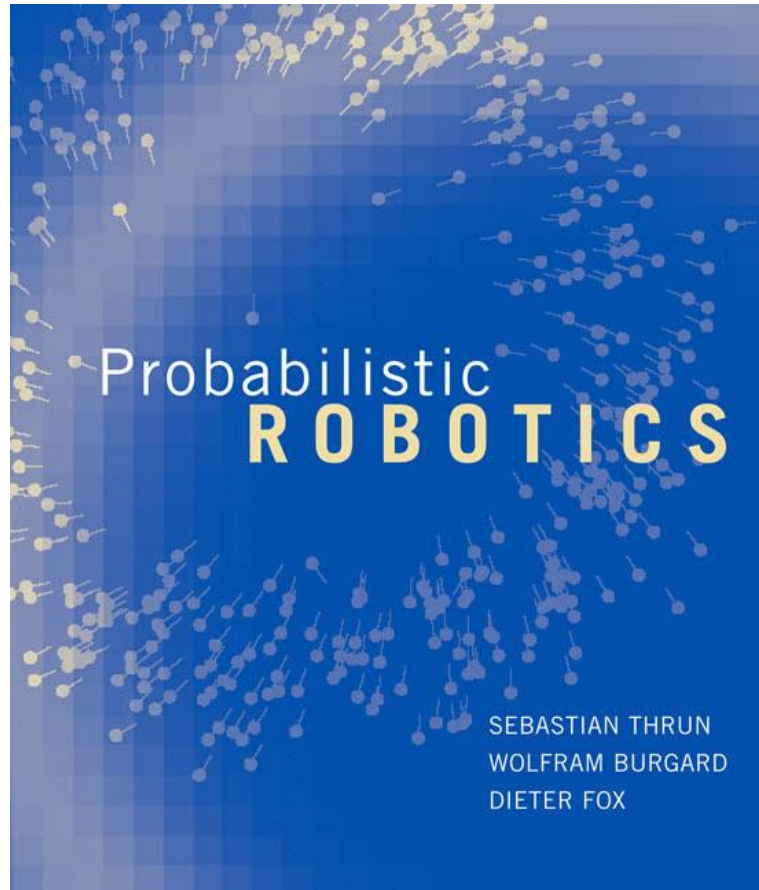
Manuel du cours : 2 proposés

- Manuel utilisé par le passé



- Disponible sur Amazon
- Je le connais moins

Autres références



Version électronique téléchargeable
via librairie U. Laval

Évaluations

Examen intra : **mardi 24 octobre**. 1 feuille au plomb, recto-verso

Examen final : **mardi 12 décembre**. 2 feuilles au plomb, recto-verso
(dont celle de l'intra)

2 travaux pratiques

Présentation orale de 15 min. pour GLO-7021, vers fin nov, 6%.

Projet

	GLO-4001	GLO-7021
Taille des équipes	1-4	1-3
Rapport écrit	17 %	17 % (incluant résumé 3-5 articles)
Objectif	Démontrer l'assimilation des concepts du cours	Assimilation et pousser plus loin que ce qui est vu en classe
Portée	Démo / reproduction de résultats	Cas idéal : mènerait à une publication scientifique

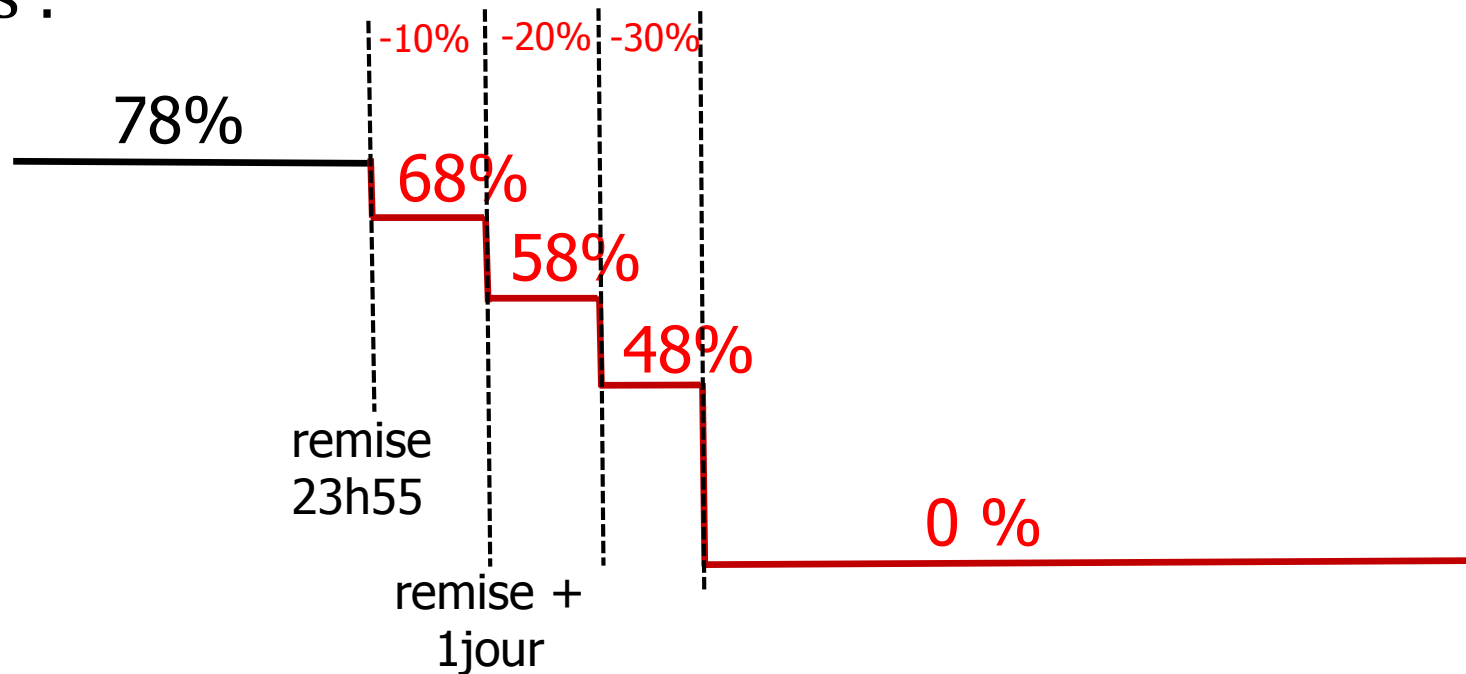
Proposition de projet : **1 novembre** Rapport final : **22 décembre**

Démo filmée : pour éviter le stress du « live »...

Prêt des équipements en fonction de la taille des équipes

Rapport du TP et retards

- Pour étudiants 2^{ème} / 3^{ème} cycle, rapport en latex → pdf
 - *MikTex* sur Windows, *pdflatex* sur Ubuntu
 - `Overleaf` ou `sharelatex` en ligne.
- Toujours fournir le code source
- Retards :



Politique sur le plagiat

- Veuillez en prendre connaissance

Politique sur le plagiat et la fraude académique

Règles disciplinaires

Tout étudiant qui commet une infraction au Règlement disciplinaire à l'intention des étudiants de l'Université Laval dans le cadre du présent cours, notamment en matière de plagiat, est passible des sanctions qui sont prévues dans ce règlement. Il est très important pour tout étudiant de prendre connaissance des articles 28 à 32 du Règlement disciplinaire. Celui-ci peut être consulté à l'adresse suivante:

http://www.ulaval.ca/sg/reg/Reglements/Reglement_disciplinaire.pdf

Plagiat

Tout étudiant est tenu de respecter les règles relatives au plagiat. Constitue notamment du plagiat le fait de:

- i) copier textuellement un ou plusieurs passages provenant d'un ouvrage sous format papier ou électronique sans mettre ces passages entre guillemets et sans en mentionner la source;
- ii) résumer l'idée originale d'un auteur en l'exprimant dans ses propres mots (paraphraser) sans en mentionner la source;
- iii) traduire partiellement ou totalement un texte sans en mentionner la provenance;
- iv) remettre un travail copié d'un autre étudiant (avec ou sans l'accord de cet autre étudiant);
- v) remettre un travail téléchargé d'un site d'achat ou d'échange de travaux scolaires.

Robots (2017) + capteurs

Kobuki



Kangaroo



gyroscopes
(interne)

tous accessibles à partir de *matlab* ou *python*

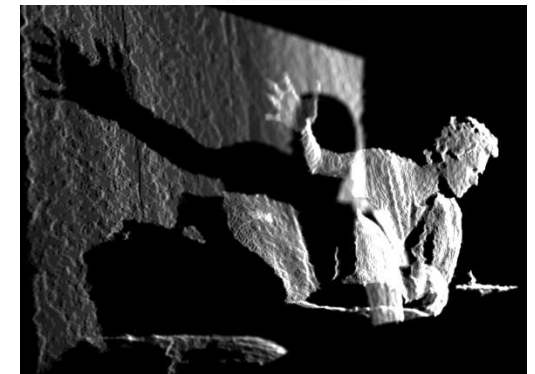


scanneur
laser 2D

odométrie
(11.7 impuls./mm)

télémètre infrarouge

2x



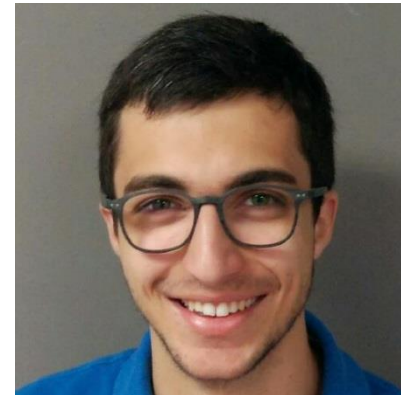
Kinect

Laboratoires du vendredi

- ~9 séances d'expérimentations
 - familiariser avec les capteurs et le robot
 - application concrète et visible des algorithmes
- 14 robots
- Vendredi 8h30-10h20 *ou* 10h30 à 12h20
 - idéalement vous répartir moitié-moitié



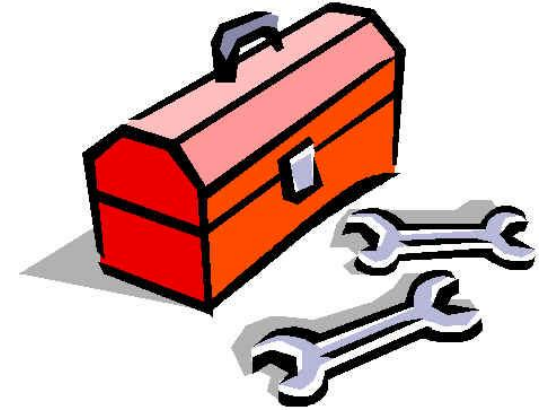
David Landry



Alexandre Gariépy

3 facettes

(1) Boîte à outils des connaissances, technologies et algorithmes utilisés en robotique mobile



algorithmes

capteurs

actionn.

environnement

(2) Développer une compréhension **verticale** et **holistique** d'un système robotisé

(3) Expérimenter!



JORGE CHAM © 2007

WWW.PHDCOMICS.COM

Pourquoi la robotique mobile?

- ... ne pas manquer le bateau...
- Prochaine révolution technologique? (*Industrie 4.0*)
 - machine à vapeur
 - chemin de fer
 - radio/télévision
 - informatique
 - réseautique (Internet)

Google car (Waymo)



on construit 60 millions
voiture par an...

Pourquoi la robotique mobile?

- Waymo (Google)
- Zoox
- Tesla
- Uber ATG
- Nissan
- Toyota Research Institute
- Bosch R&D (Palo Alto)
- Groupe Volkswagen (Audi)
- Ford
- nuTonomy
- Argo AI
- AutoX
- Cruise Automation
- Lyft
- NIO
- et +...

Pourquoi la robotique mobile?

- Kiva system (inspiré de RoboCup, 775 M\$ Amazon)



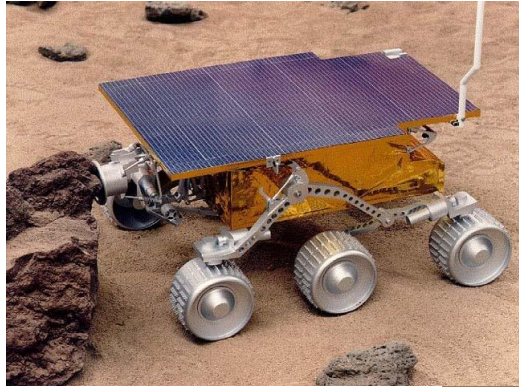
Pourquoi la robotique mobile?

- Endroits dangereux pour les humains
 - Centrales nucléaires Fukushima en 2011



Pourquoi la robotique mobile?

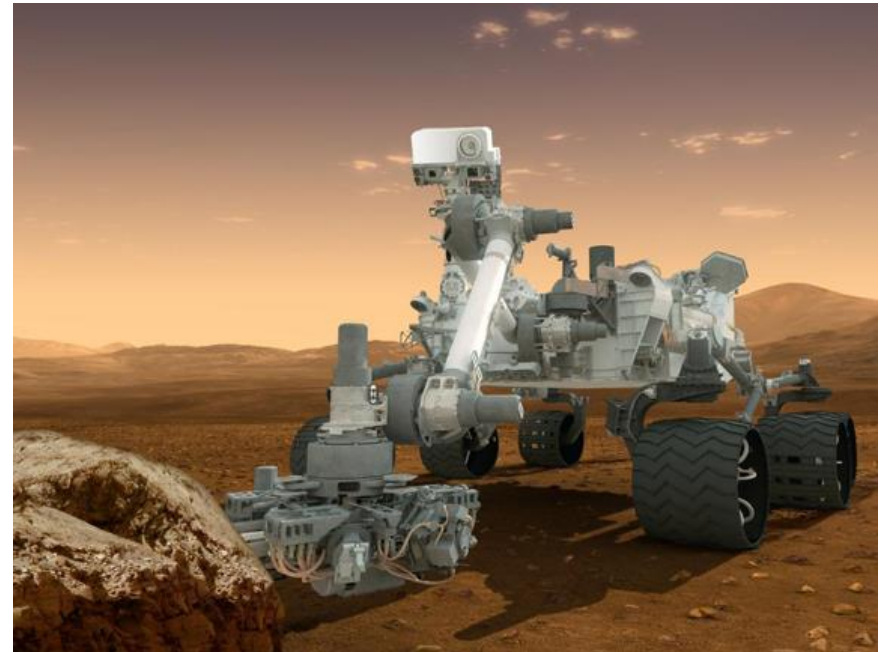
- Exploration spatiale



Sojourner
(1997)



Spirit + Opportunity
(2003)



Curiosity (2012)

Pourquoi la robotique mobile?

- Entretien ménager



Roomba de iRobot



Neato Robotics



Asus

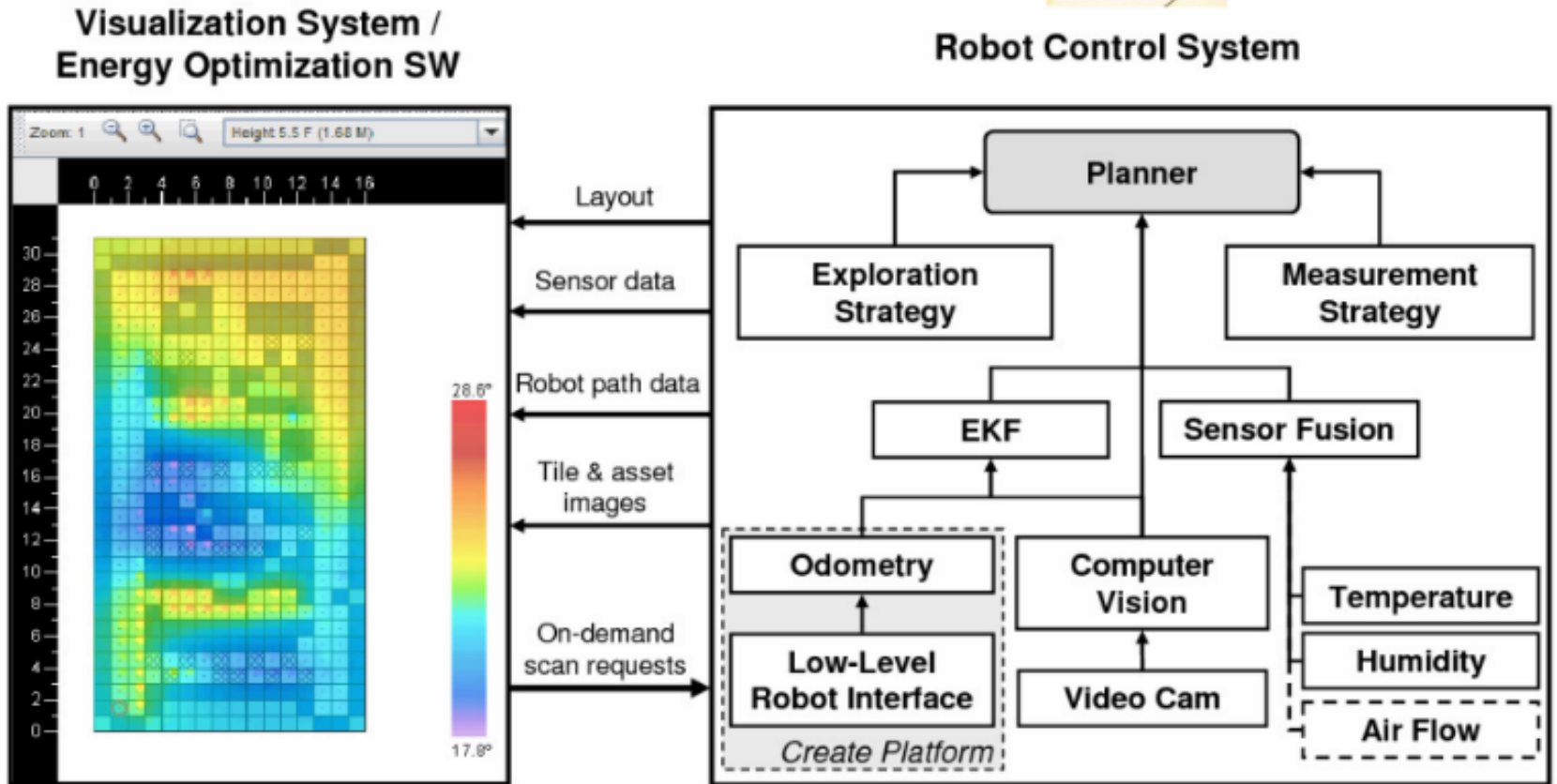


LawnBott

Automower
d'Husqvarna



IBM Roomba-Based Robot Measures Data Centre Heat



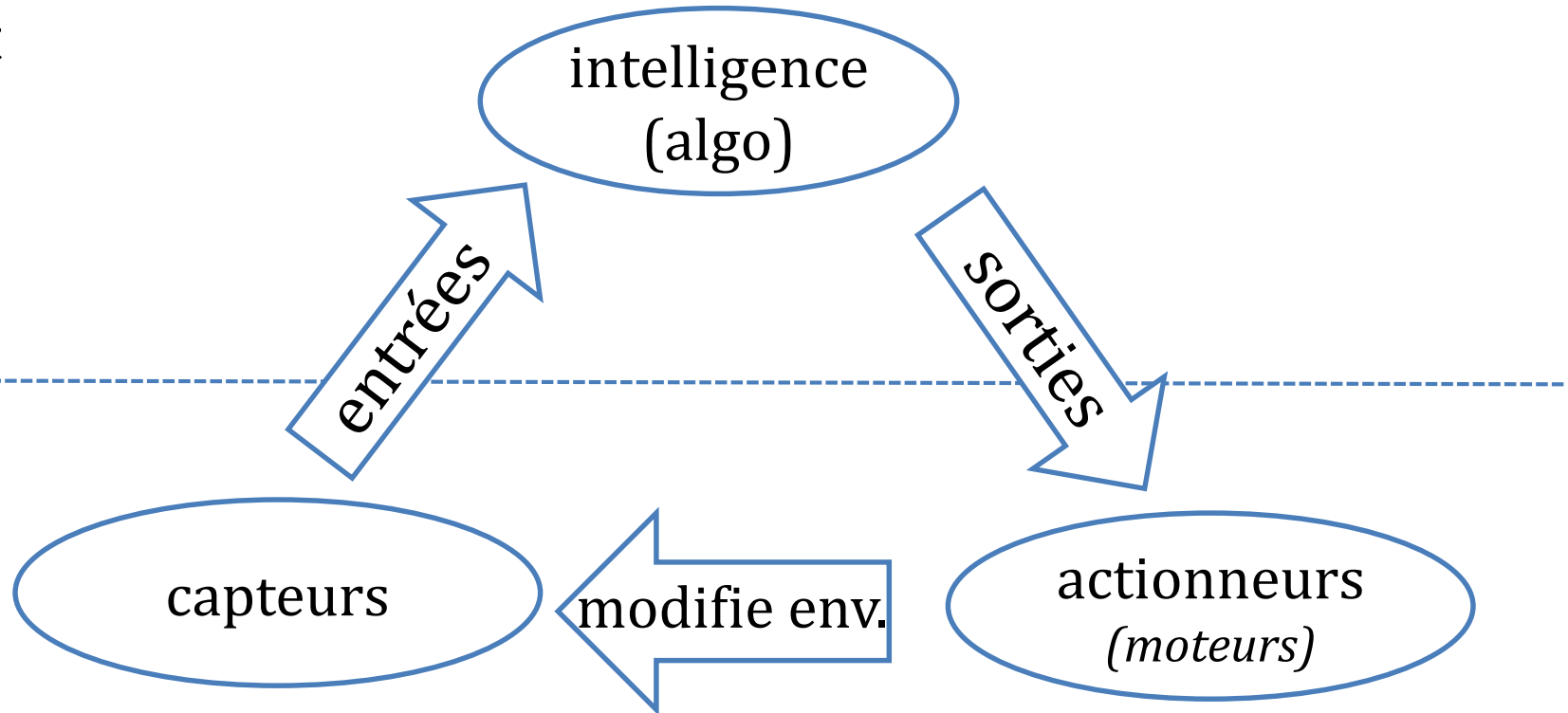
Vous devriez comprendre ce diagramme à la fin du cours

Vue générale de la robotique mobile

Boucle *sense-think-act*

Abstrait

Réalité



- batteries
- communication (radio)

Tout est étroitement imbriqué

- **Bons capteurs (\$)**
 - facile d'interpréter l'information (perception)
 - algorithmes traitement moins poussés
 - e.g. caméra 3D *Kinect* vs. 1 caméra RGB
- **Bons actionneurs/design mécanique**
 - « intelligence mécanique »
 - réduit les besoins en capteurs et intelligence

ROBOTIQ



RHex

Expérimentation en robotique

- Permet de voir si la théorie tient la route!
 - Nos approximations sont-elles valides?
 - Est-ce que nos modèles (capteurs) sont valides?
 - L'algorithme est-il robuste?
 - A-t-on pensé à tout?
 - Qu'en est-il des changements de lieu d'opération, d'éclairage, de température, etc.
 - « *when the rubber meets the road* »

En théorie, il n'y a pas de différence entre la théorie et la pratique, mais en pratique, il y en a une.

Robotique vs. Robotique Mobile

- Travail dans un espace beaucoup plus grand
 - quelques m^3 pour un bras robotisé vs. plusieurs millions de m^3 pour un véhicule routier autonome

Apprentissage automatique + robotique

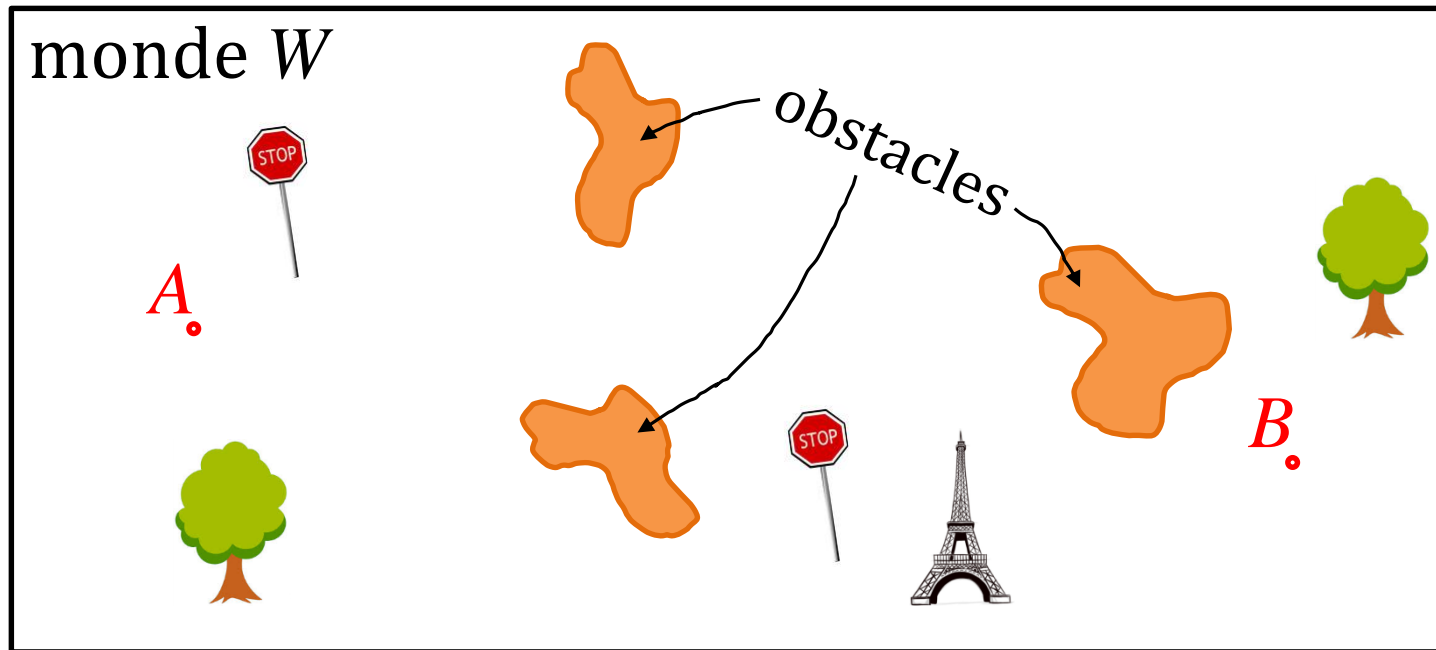
- Pour améliorer les performances du système

Quadrocopter Pole Acrobatics



Problématique de la robotique mobile

Aller de A à B



- Décliné en plusieurs sous-objectifs

Sous-problème 1

Où suis-je?

$$X_t = ?$$

- **A**: En me basant sur mes commandes u ?
- **B**: En me basant sur un capteur? deux capteurs? dix capteurs? Les plus fiables seulement? (fusion d'information)
- En combinant **A** et **B**?

Sous-problème 2

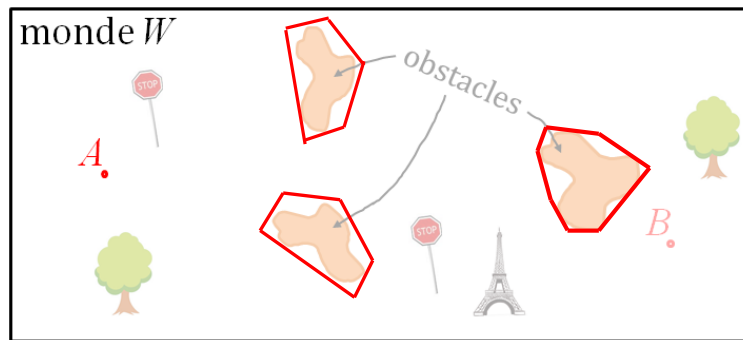
Comment me déplacer?

- Plusieurs designs mécaniques possibles
- Roues, pattes, chenilles, ailes, hélices
- Quelles sont les modèles (équations) de déplacement selon les commandes u ?

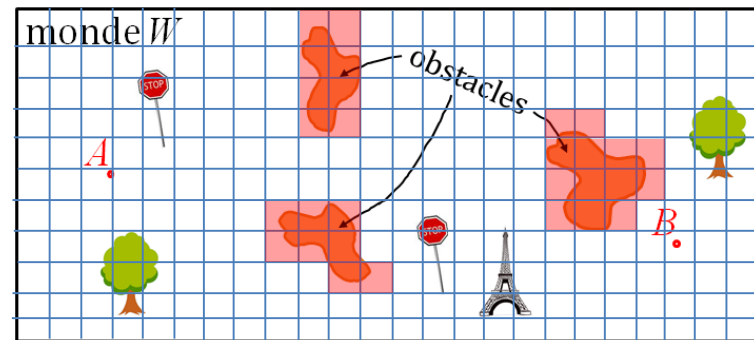
Sous-problème 3

Carte du monde

- Comment représenter le monde W ? (*problème universel de la représentation de la connaissance en IA*)



obstacles : polygones?



obstacles : grille d'occupation

- Choix intimement relié avec la tâches à accomplir
- Compromis entre : **mémoire - temps de calcul - précision**

Sous-problème 4

Quelle trajectoire suivre entre A et B?

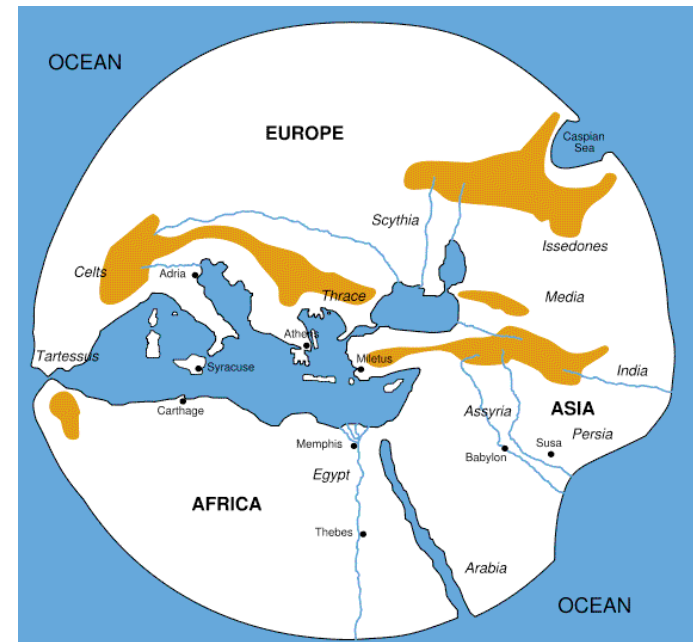
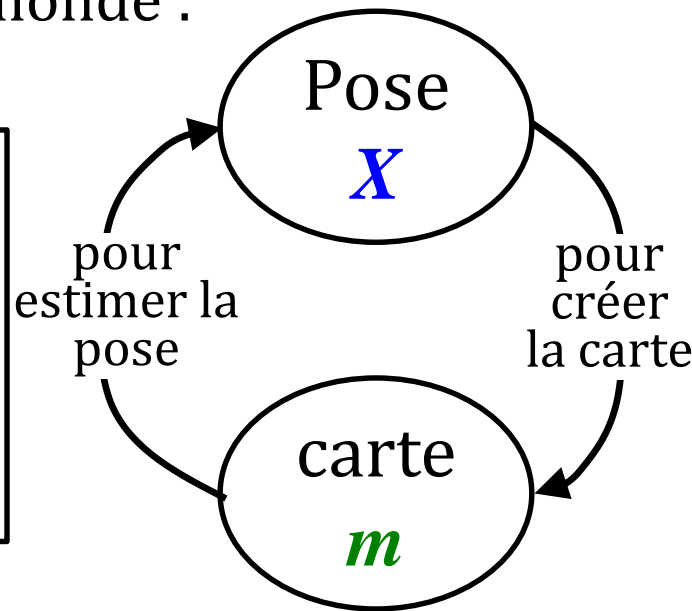
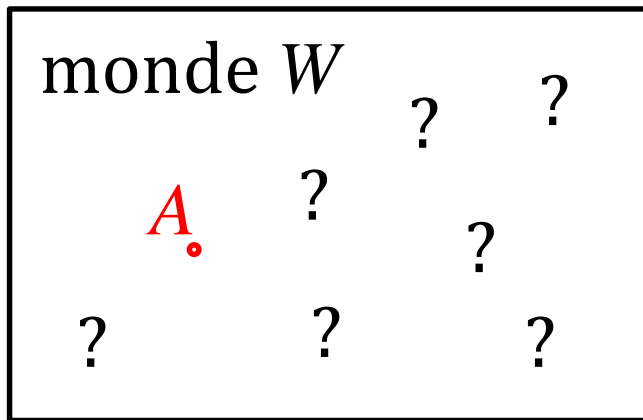
- Planification (*path planning*)
- Basé sur la carte du monde W
- Contente souvent d'une solution sous-optimale
- Critères supplémentaires à minimiser :
 - le temps
 - le risque
 - l'énergie (*en vélo on évite les côtes...*)
 - la probabilité de se perdre (navigation *côtière/coastal navigation*)
 - complexité des instructions (*GPS qui retourne 300 virages?*)

Sous-problème 5

Comment construire une carte?

On part sans carte du monde :

tabula rasa



carte du monde, Grèce antique

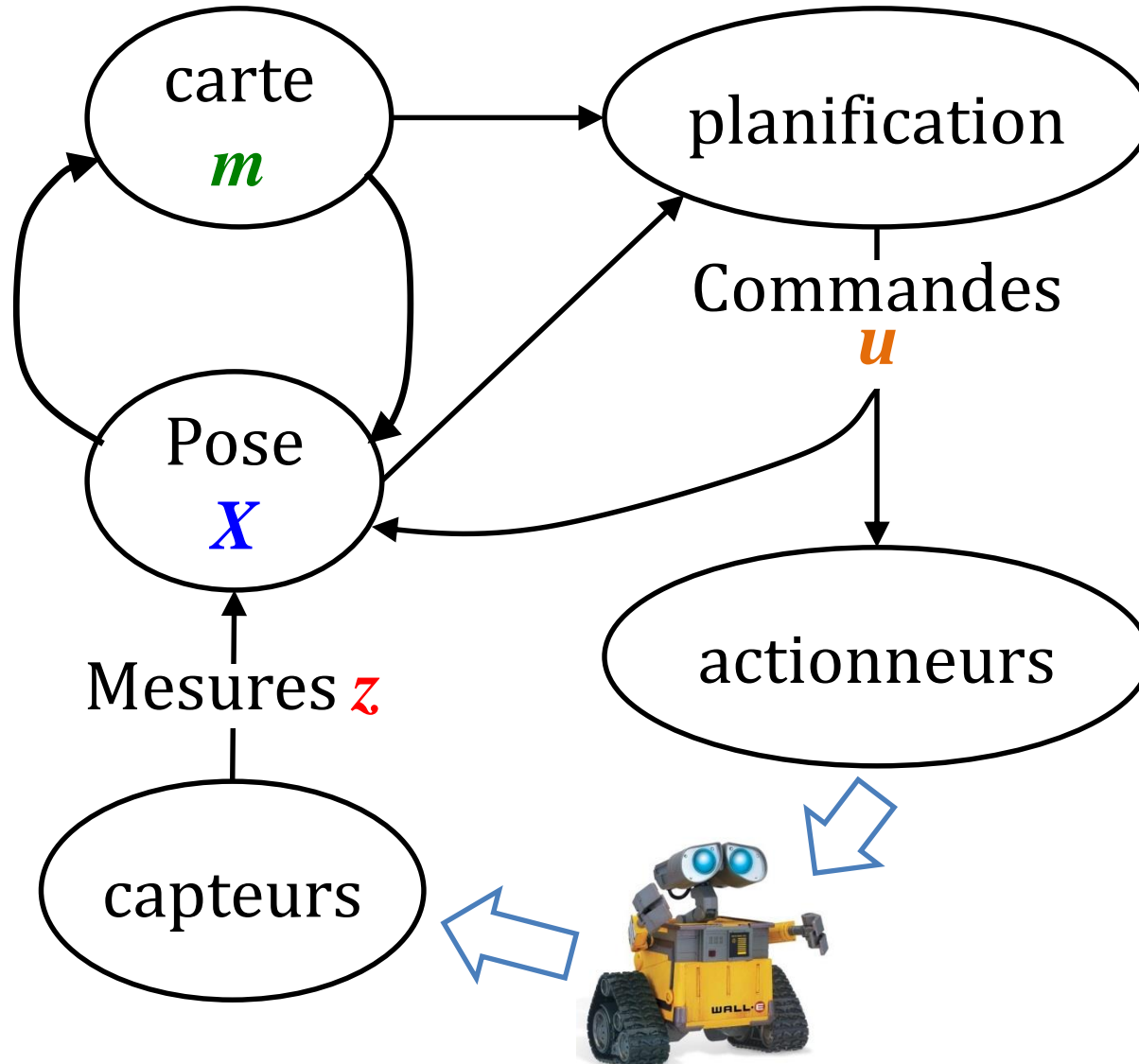
- SLAM : Simultaneous Localization and Mapping

Sous-problème 6

Suis-je déjà passé par là?

- Pour ne pas dupliquer les lieux dans la carte
- Fermeture de boucle pour SLAM
- Reconnaissance de lieux par vision

Vue d'ensemble



Paradoxe de Moravec

- La découverte par les chercheurs en **intelligence artificielle** et en **robotique** que le raisonnement de haut niveau est beaucoup plus facile à reproduire et simuler par un programme informatique que les aptitudes **sensorimotrices** humaines.
- Selon Moravec, la [théorie de l'évolution](#) permet également d'expliquer ce paradoxe. Les tâches **sensorimotrices**, en tant que fonctionnalités biologiques anciennes, ont été perfectionnées par les mécanismes évolutifs durant des millions d'années. Les facultés de **raisonnement**, apparues très récemment sur le plan biologique, ne se sont pas encore aussi perfectionnées

Ce qui vous semble simple est souvent difficile pour un robot

Paradoxe de Moravec



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

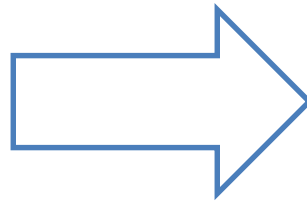
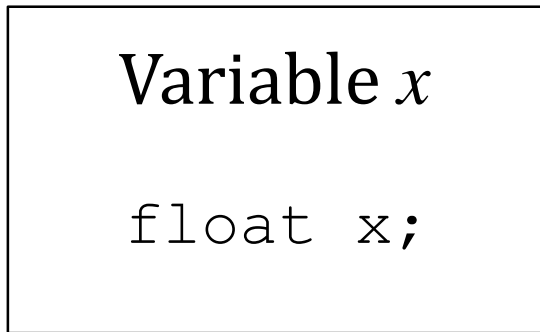
IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.



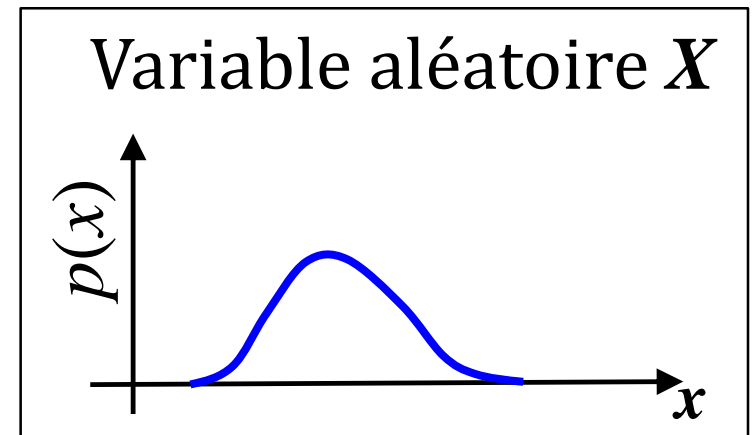
Gestion de l'incertitude : probabilités

- Changement de paradigme :
déterministe

représentation



stochastique
(en partie aléatoire)



- Devra connaître *(approximer)* les **distributions** pour :
 - les mesures z des capteurs
 - les commandes u (déplacements)

Approches probabilistes : succès des 15 dernières années en robotique mobile



Outils mathématiques

- Robotique mobile combine plusieurs outils
 - trigonométrie
 - algèbre linéaire (matrice)
 - calcul différentiel (Jacobienne) et intégral
 - Probabilités et statistiques
- Sera l'occasion de les appliquer de manière concrète

« ...Et su mon établi, y a MES outils. Et ça, mes outils, c'est sacré. O.K. là? Bon. ON TOUCHE PAS À MES OUTILS! Moé-même j'y ai jamais touché. » Yvon Deschamps, « Les adolescents »

Exemple d'application des outils math.

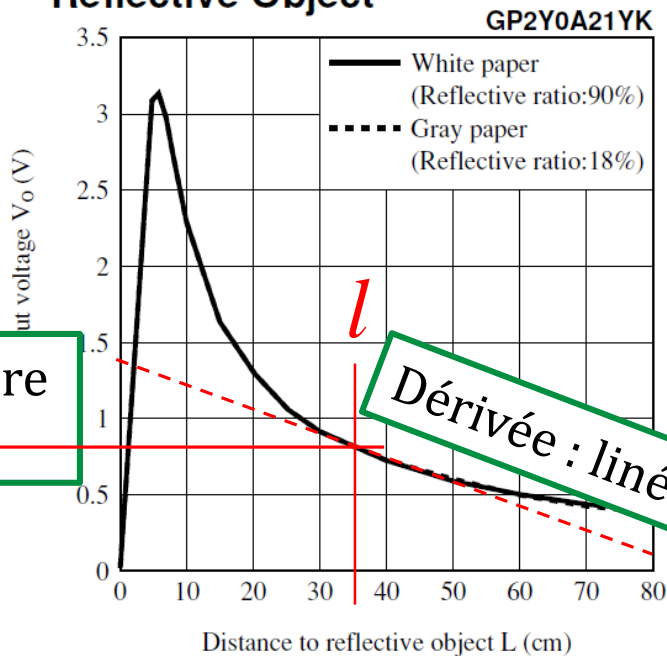
- Objet en chute libre, départ à hauteur h_0 et angle θ_0 , mesures bruitées de distance l , vitesse angulaire ω . Cherche la hauteur $h(t)$ et l'angle $\theta(t)$.

Probabilités : Filtre Kalman Étendu

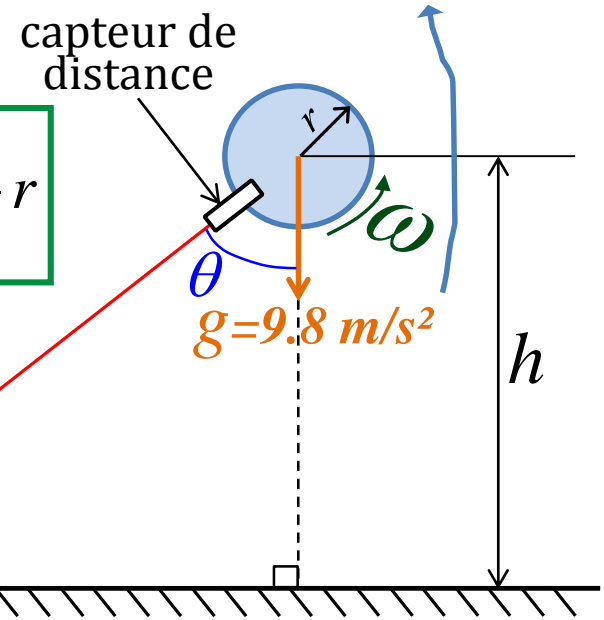
Algèbre linéaire :
$$x(k+1) = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{1}{2} \Delta T^2 \\ \Delta T \end{bmatrix} g$$

Intégrale :
$$h(t) = \iint \text{accélération}$$

Fig.5 Analog Output Voltage vs. Distance to Reflective Object



Trigo :
$$l = \frac{h}{\cos(\theta)} - r$$



(Projet de 2013)

Contenu du cours

Contenu (1)

- Introduction et historique
- Révision (au fil des cours) :
 - algèbre linéaire (matrices et vecteurs)
 - trigonométrie
 - dérivée et intégration
 - probabilités

Contenu (2)

- Capteurs standards en robotique
 - encodeurs d'angle
 - sonar
 - **télémètres infrarouge (labo!)**
 - **télémètres laser (labo!)**
 - GPS (*Global Positioning System*)
 - centrale inertielle (accéléromètres et **gyros**) **(labo!)**
 - magnétomètres

Contenu (3)

- Capteurs visuels
 - **caméra simple** (modèle *pin-hole*) **(labo!)**
 - paire de caméras stéréo
 - **caméra 3D Kinect de Microsoft (labo!)**
 - caméra « temps de vol » (phase)
 - marqueurs fiduciaires
 - **descripteurs SIFT, SURF, ORB (labo!)**
 - descripteurs quantifié : approche bag-of-words
- Odométrie visuelle

Contenu (4)

- Estimation robuste (*RANSAC*)
- Locomotion (à pattes, **à roues**) **(labo!)**
- Cinématique directe et cinématique inverse

Contenu (5)

- Représentation de l'espace et planification
 - Cartographie (métrique, topologique, topométriques, grilles d'occupation)
 - Navigation :
 - graphes de visibilité et de Voronoï
 - champs de potentiels
 - rapidly exploring random trees RRT
- **Calage de nuage de points ICP (labo!)**

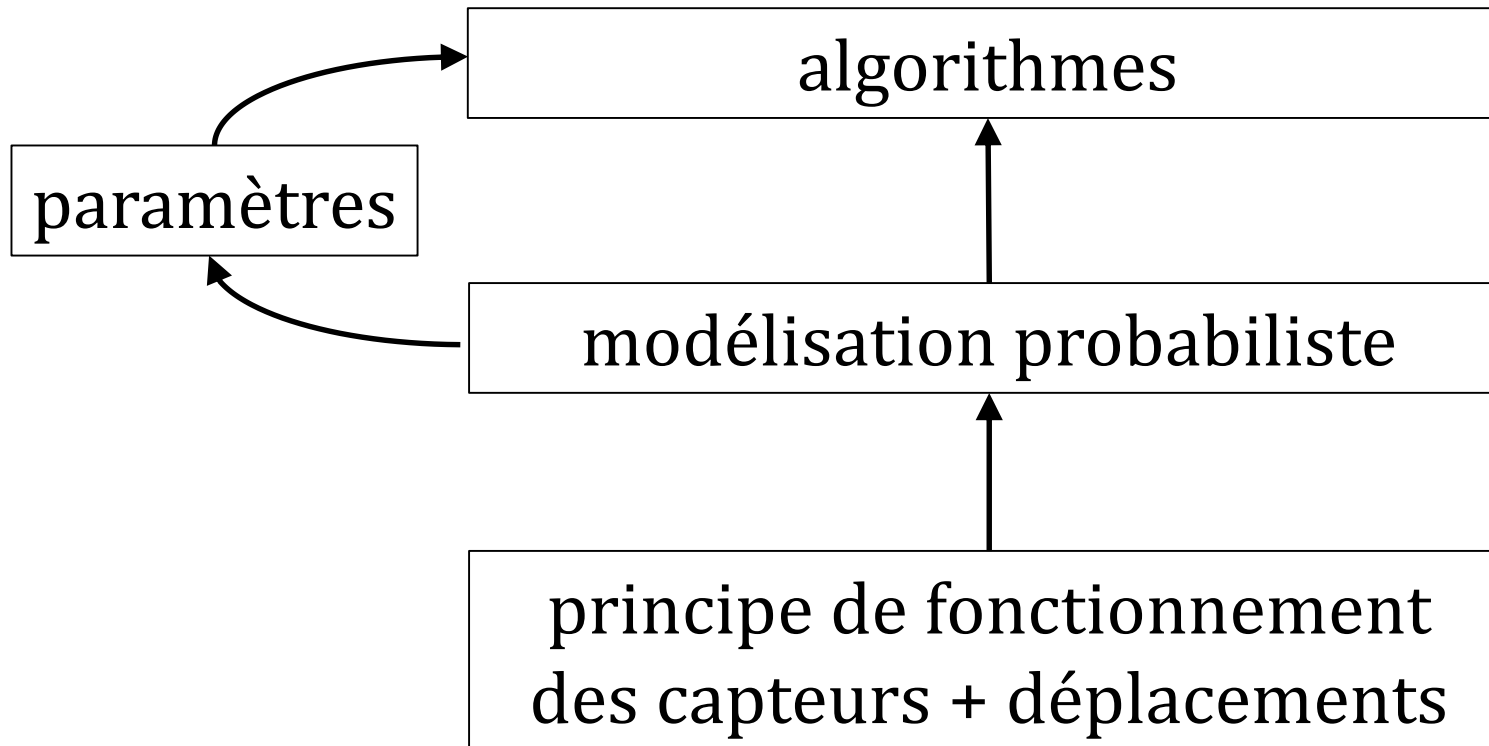
Contenu (6)

- Fusion de capteurs
 - modélisation
- Estimation d'état (Localisation) :
 - filtrage Bayésien
 - **filtre de Kalman (labo!)**
 - filtre de Kalman étendu
 - filtre de Kalman non-parfumé
 - filtre à particules
 - filtre Rao-Blackwellisé

Contenu (7)

- Simultaneous Localization and Mapping (SLAM)
 - Problématique du *loop closure*
 - Reconnaissance de lieu par vision : Visual SLAM
 - par filtre EKF
 - par filtre Rao-Blackwellisé : FastSLAM
 - par optimisation non-linéaire de graphe : GraphSLAM
 - Approches non-métriques
- Teach-and-Repeat

Autre point de vue



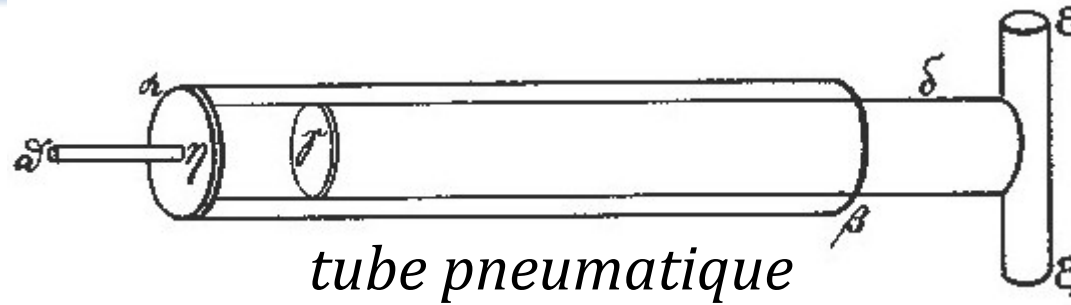
Historique

Talos : 400 av. J.-C.

- Homme géant de bronze
- 3 tours/jours
- Côte de Crète : 1 046 km
- Vitesse moyenne: 130 Km/h



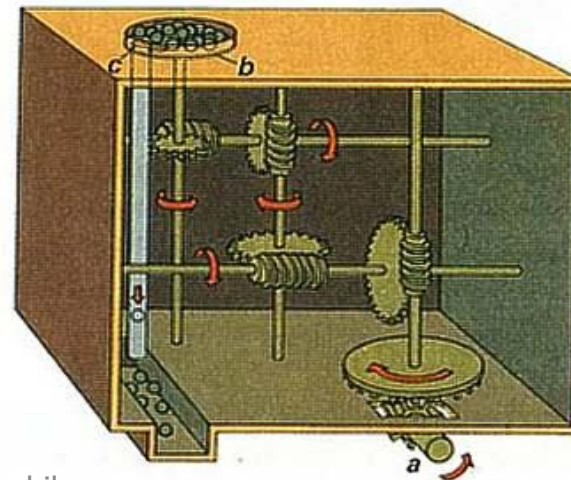
Héron d'Alexandrie 10-70 ap. J.-C.



- Ingénieur et mathématicien grec
- *Traité des automates* **Αυτόματα**
- Un des premiers capteurs: l'odomètre



L'éolipyle
« boule d'Éole »

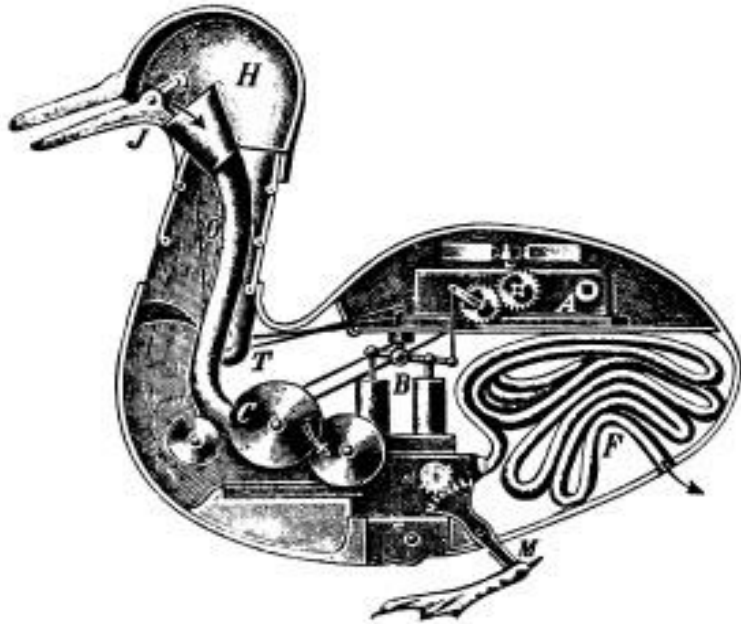


Automates de Léonard de Vinci (1452-1519)

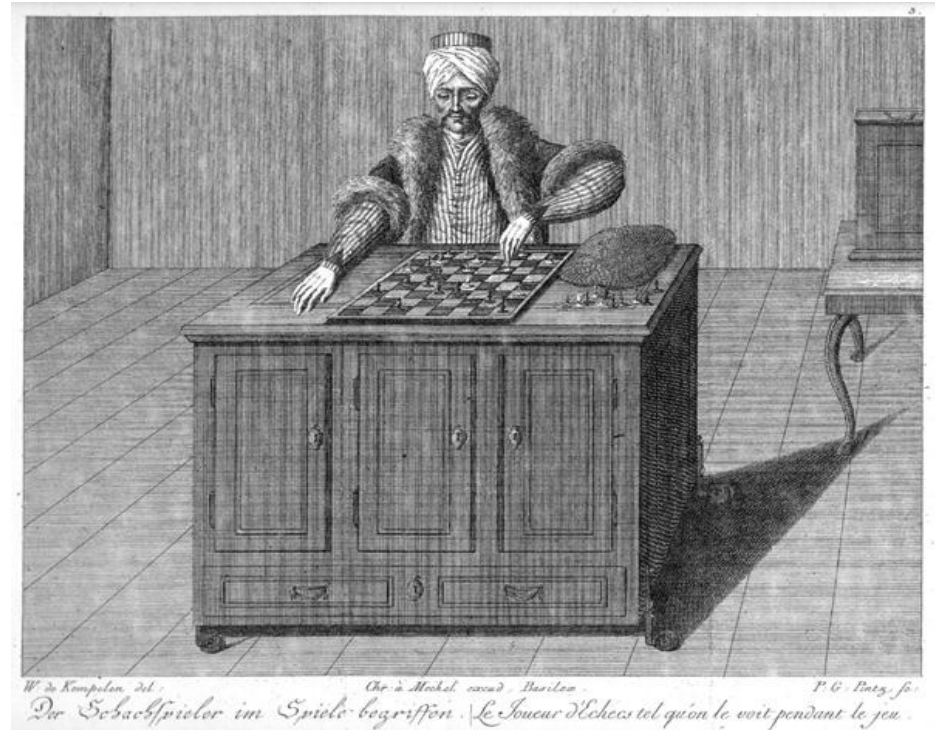


automate programmable

Automates “modernes”

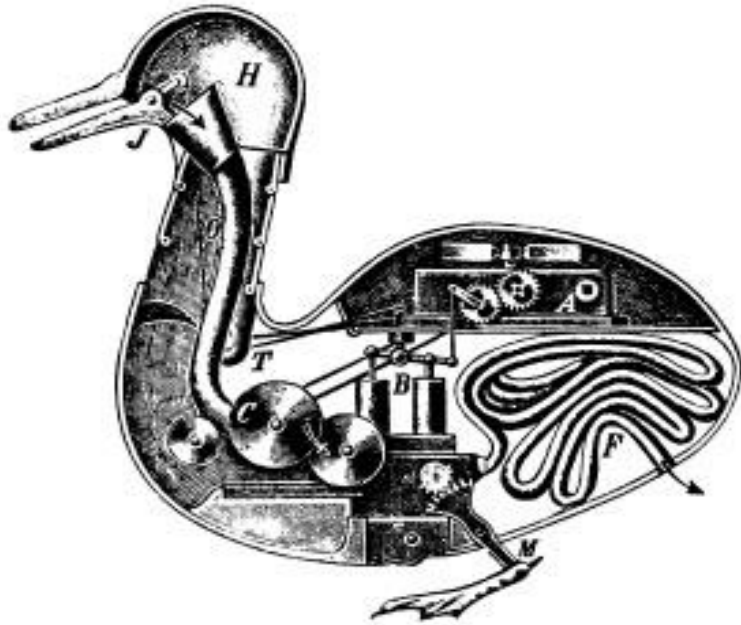


« Canard Digérateur » de
Jacques de Vaucanson
1793

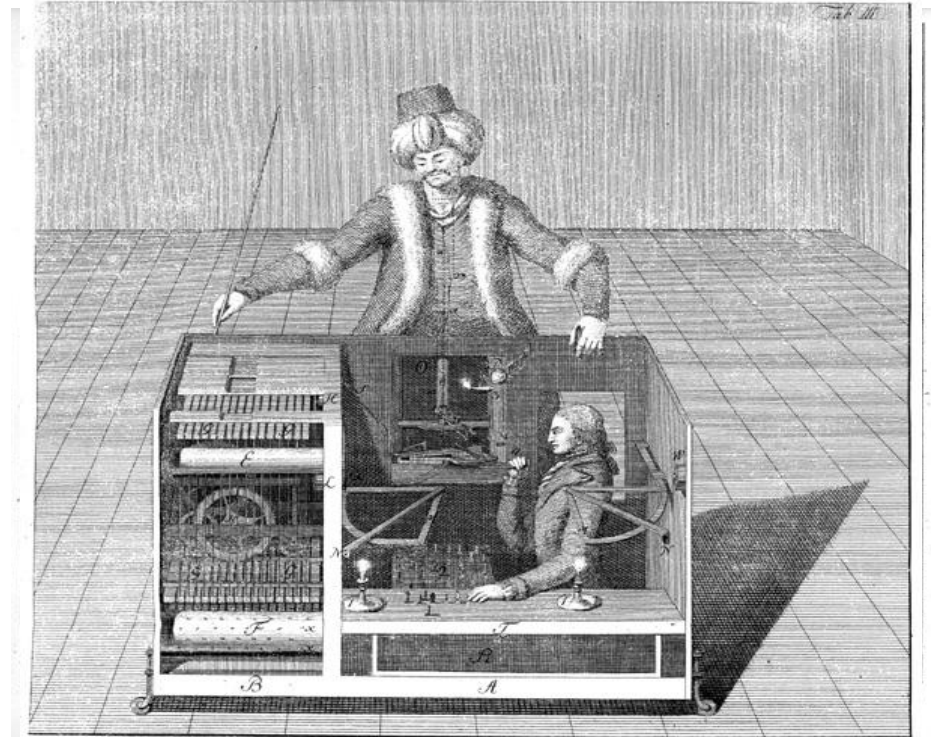


« Le turc mécanique », 1770
*Gagnant contre Napoléon
Bonaparte, Benjamin Franklin,
Catherine de Russie*

Automates “modernes”



« Canard Digérateur » de
Jacques de Vaucanson
1793



« Le turc mécanique », 1770
*Supercherie découverte en 1820
seulement!*

Amazon Mechanical Turk

- Pour faire du « crowdsourcing » de tests en intelligence artificielle

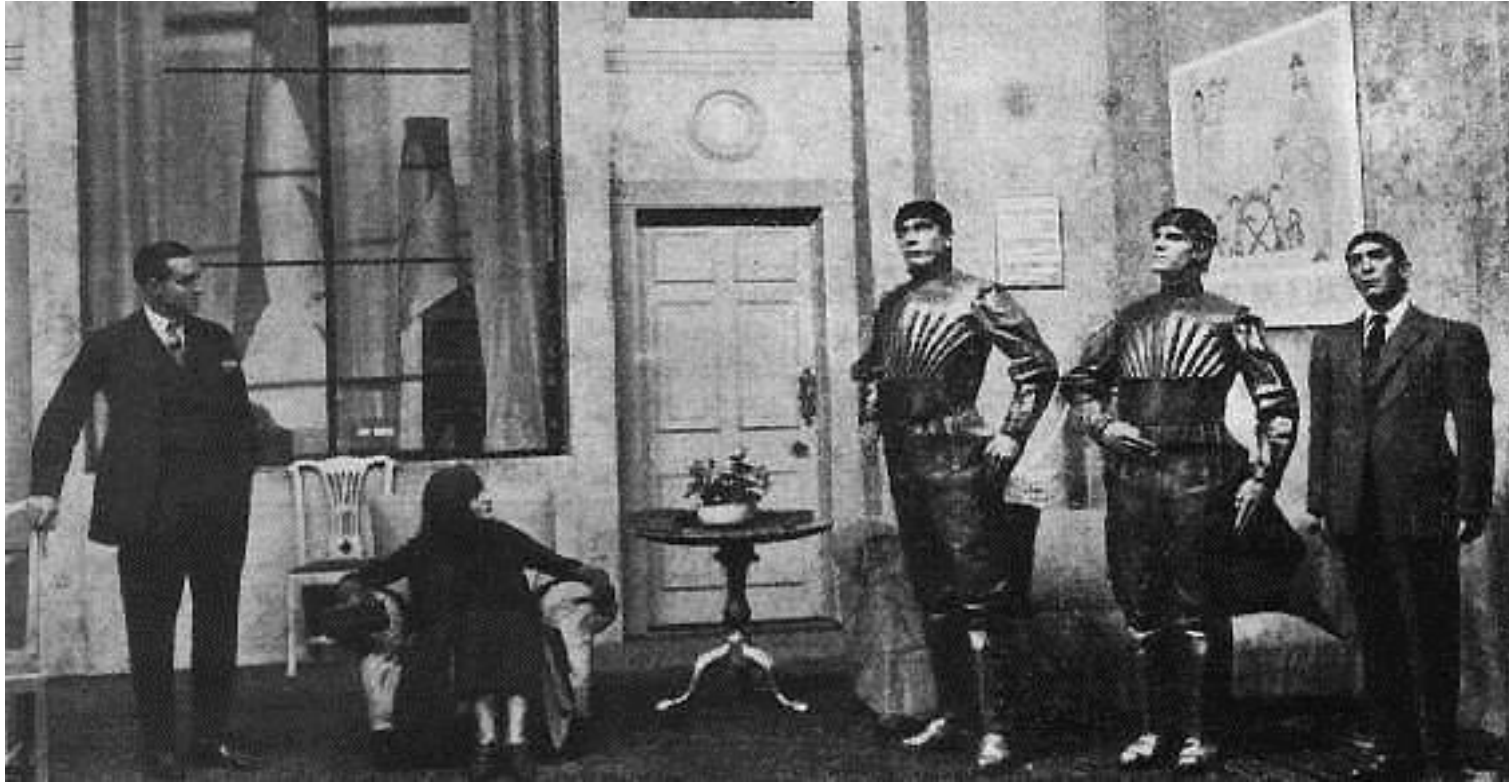


Automates servant le thé, 19^{ème} siècle, Japon



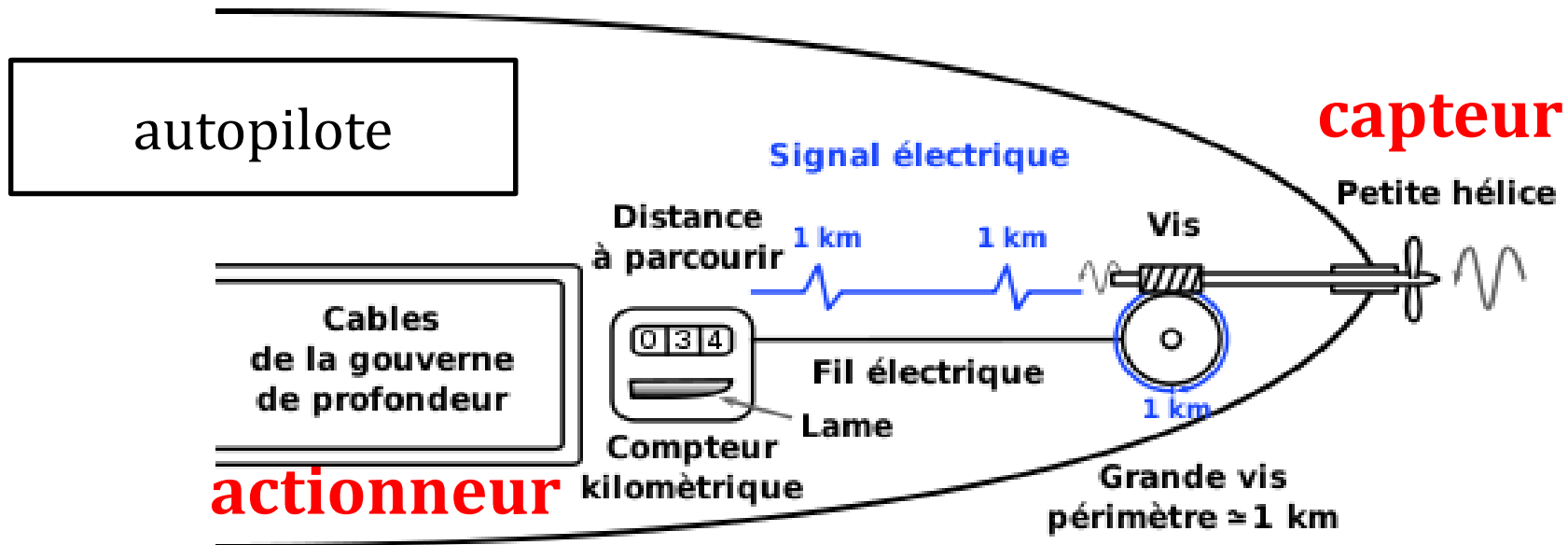
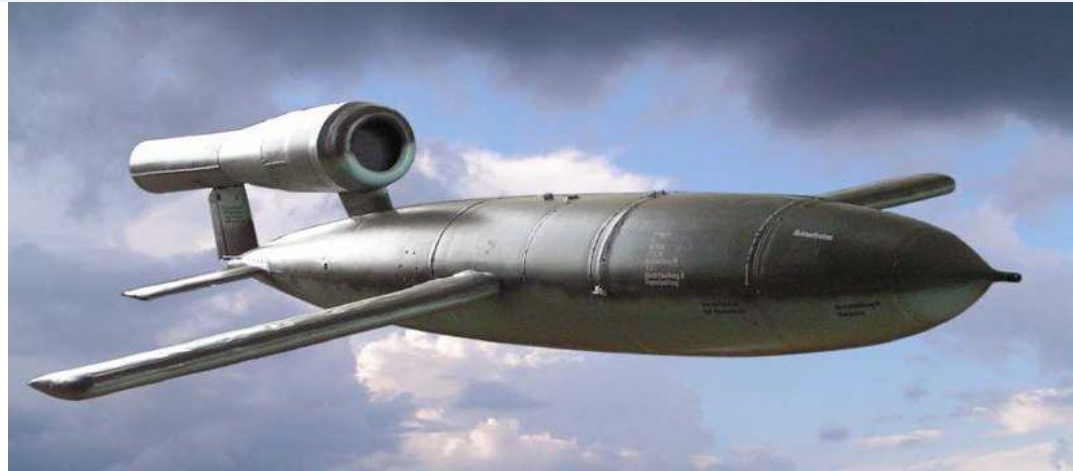
Invention du terme “Robot” (1923)

- “*Rossum's Universal Robots*” par Karel Čapek.
- Tchèque *robota* → travail forcé



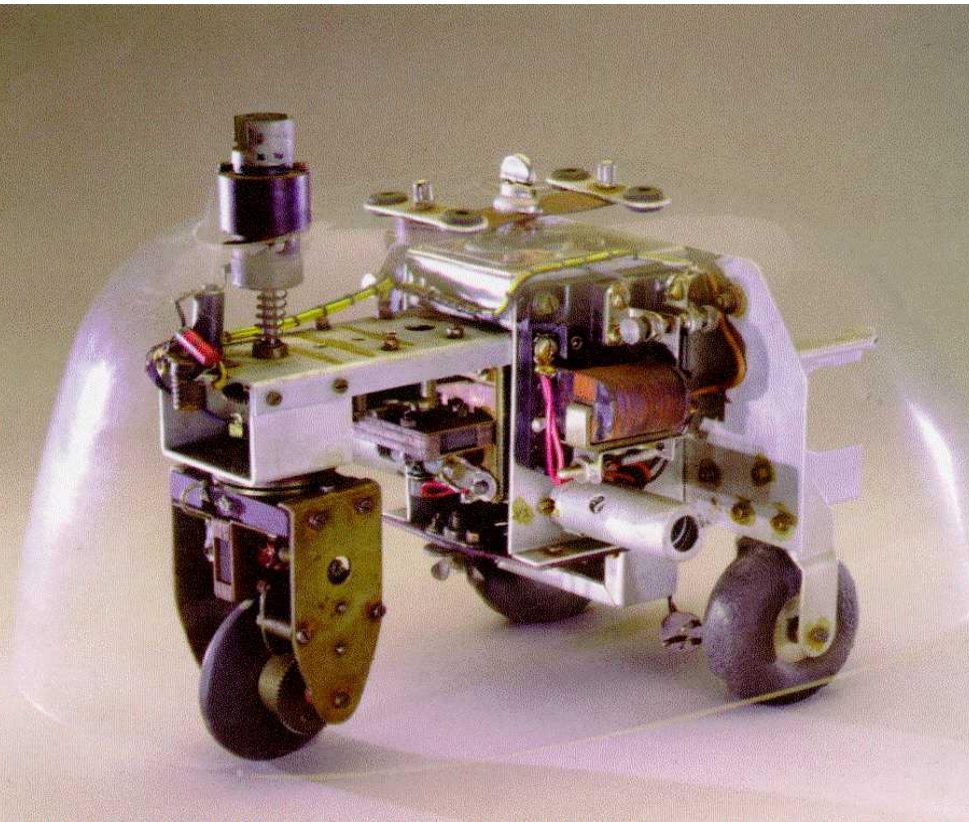
- Après plus de 2 000 ans, le concept d’être artificiel intelligent est finalement baptisé!

Missile V1 (1944)

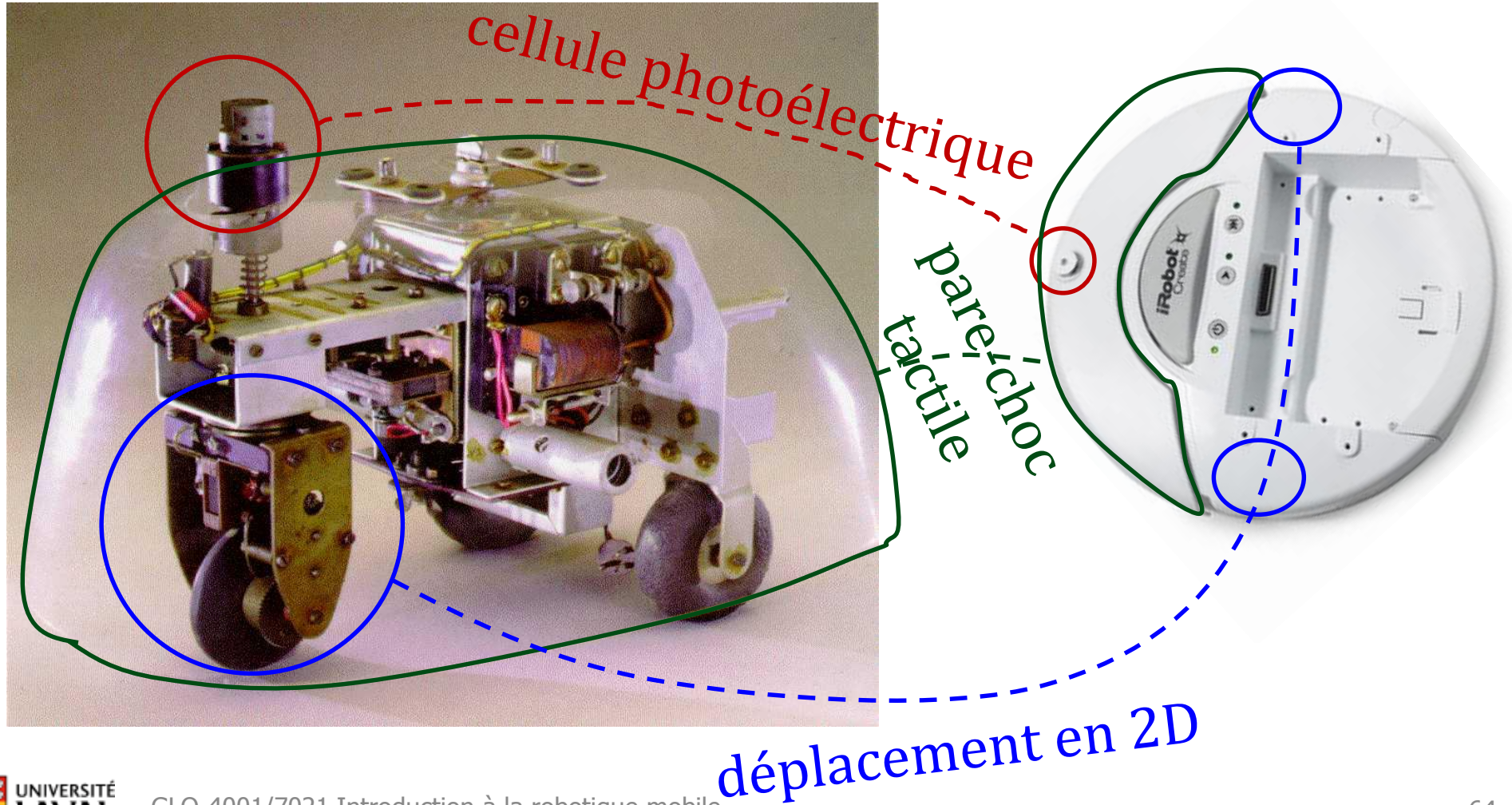


Robots mobiles (1950)

- *La Tortue (turtoise)* de Walter

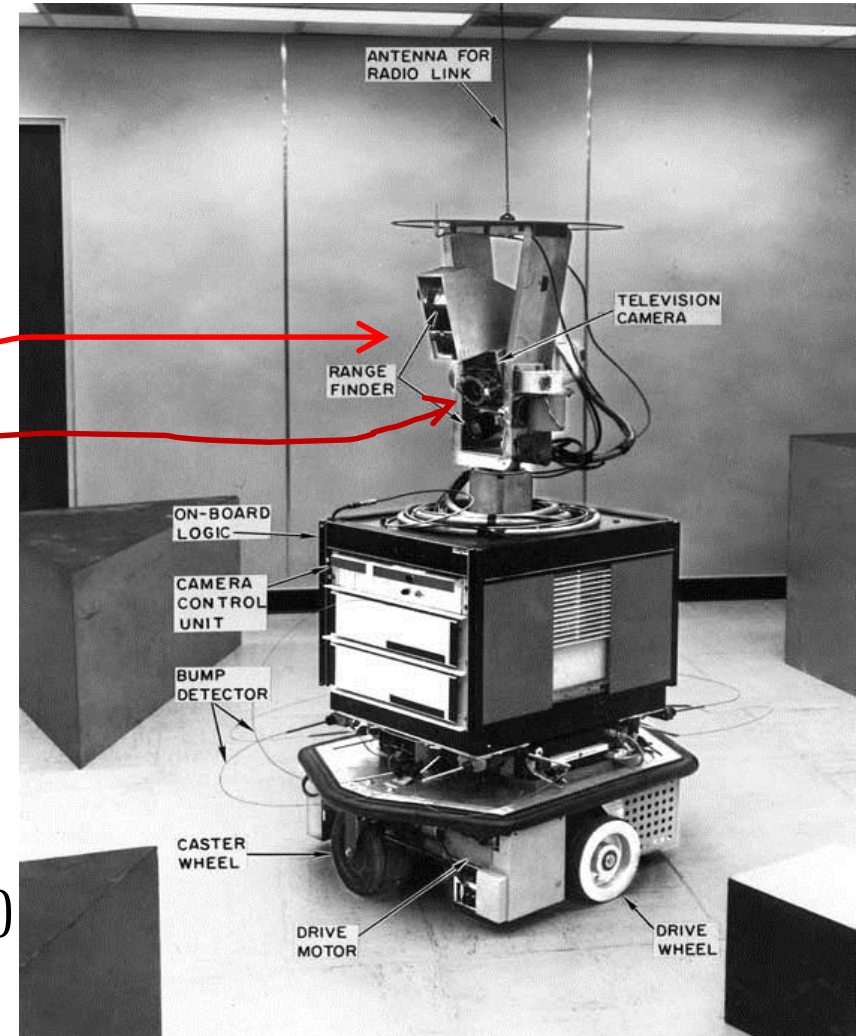


Un précurseur du Roomba?



Shakey (1966 -1972)

- **Shakey**
 - Stanford Research Institute/SRI
 - premier robot “autonome” : **IA**
- **Capteurs**
 - pare-chocs tactiles
 - télémètre optique
 - caméra pivotante
- Monde simplifié (*cube world*)
- Reconnaissance objet
- Trajectoire vers l'objet
- Action simples (pousser objet)
- Planification avec *STRIPS* (symbolique)
- Ordinateur avec 192 Ko mémoire

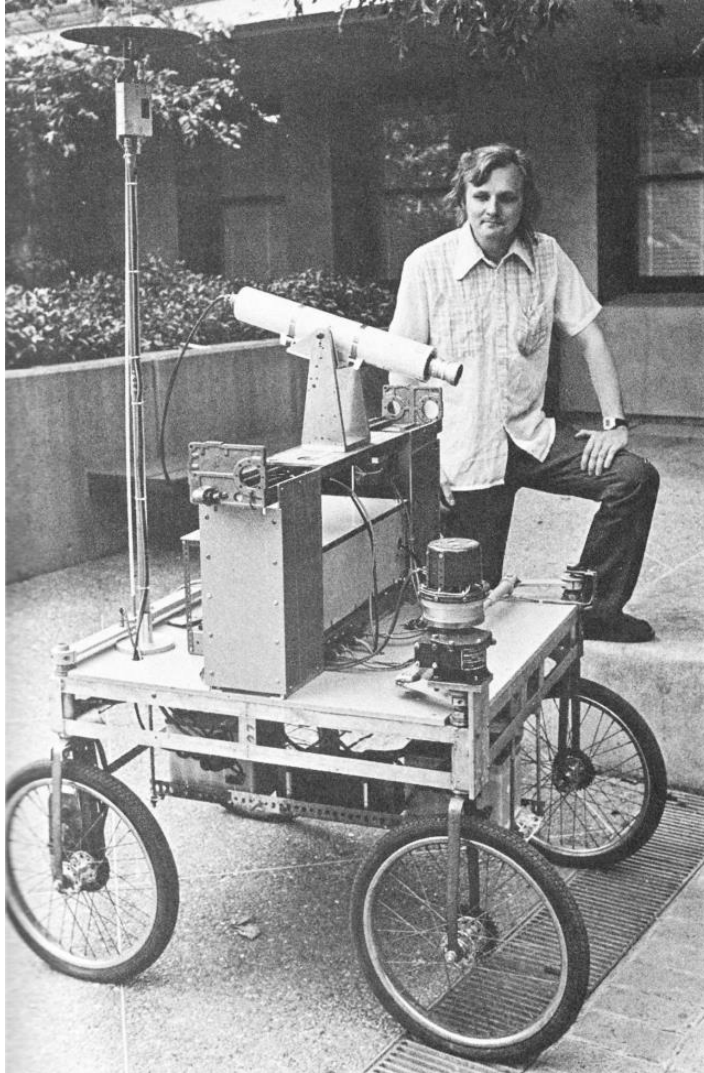


Shakey (1966 -1972)

- Démontre planification, algorithmes de vision, localisation via repères visuels (coins de plancher)



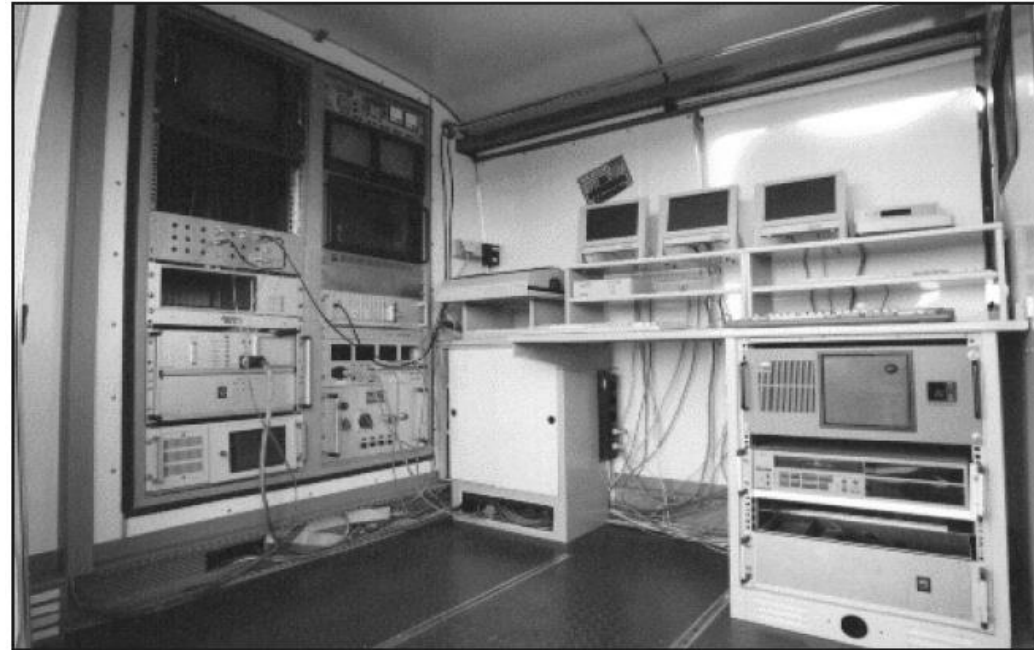
Stanford Cart (1973-1979)



- Développé par Hans Moravec
- Imagerie stéréo par caméra glissante
- Évaluation automatique de la distance vers les obstacles
- Suivi de lignes blanches au sol
- Environ **15 minutes** pour chaque mètre de déplacement

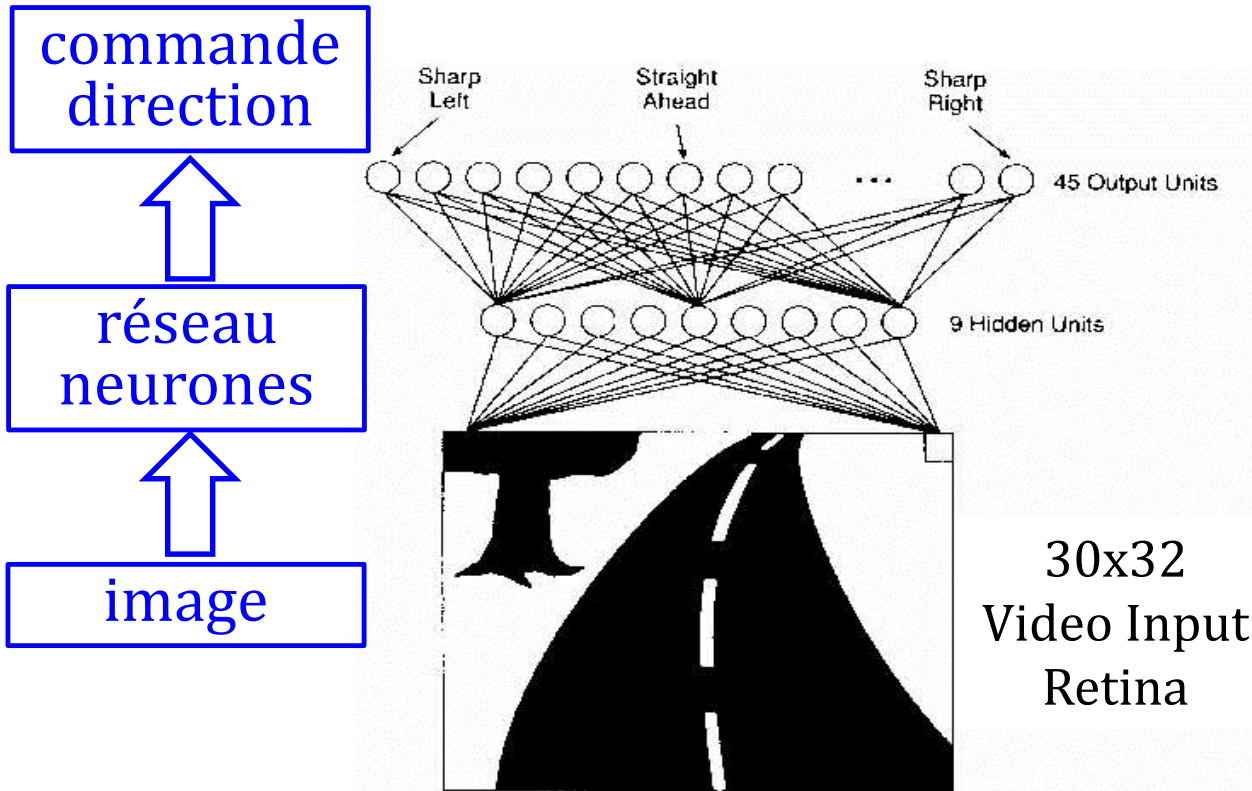
Véhicule robotisé (fin 80)

- *VaMoRs*: conduite sur autoroute
- Suivi des lignes blanches avec **filtre de Kalman**
- Que faire si ligne blanche disparaît?



ALVINN (1992)

Autonomous Land Vehicle In a Neural Network



115 km/h sur 150 km autoroute

Multilayer Network Control Unit for NavLab
(From Kanade et al., 1994. © 1994 ACM. Courtesy of the authors.)

Navlab 5 du CMU (mi-90)

- Conduite autonome sur 2797/2849 miles (98.2%) d'autoroute : *traversée des États-Unis*
- Accélérateur et frein commandé manuellement.



DARPA Grand Challenge (2004)

- But: conduite autonome dans le désert sur trajet de 240 *km*
 - Prix: 1 million \$
 - Meilleure équipe : 11.8 *km*!





DARPA Grand Challenge (2005)

- Prise 2, sur 212 *km*
 - 5 équipes finissent la course!



DARPA Urban Challenge (2007)

- Conduite autonome urbaine sur 96 *km*.

BOSS de
Carnegie
Mellon
University



Chris Urmson
Director, Self-Driving Cars
Google [x]

« Google Car » (2010)



Ford aussi... (13 déc. 2013)



http://www.motorauthority.com/news/1089080_ford-testing-autonomous-driving-with-fusion-hybrid-research-vehicle

2015 Darpa Robotics Challenge (DRC)

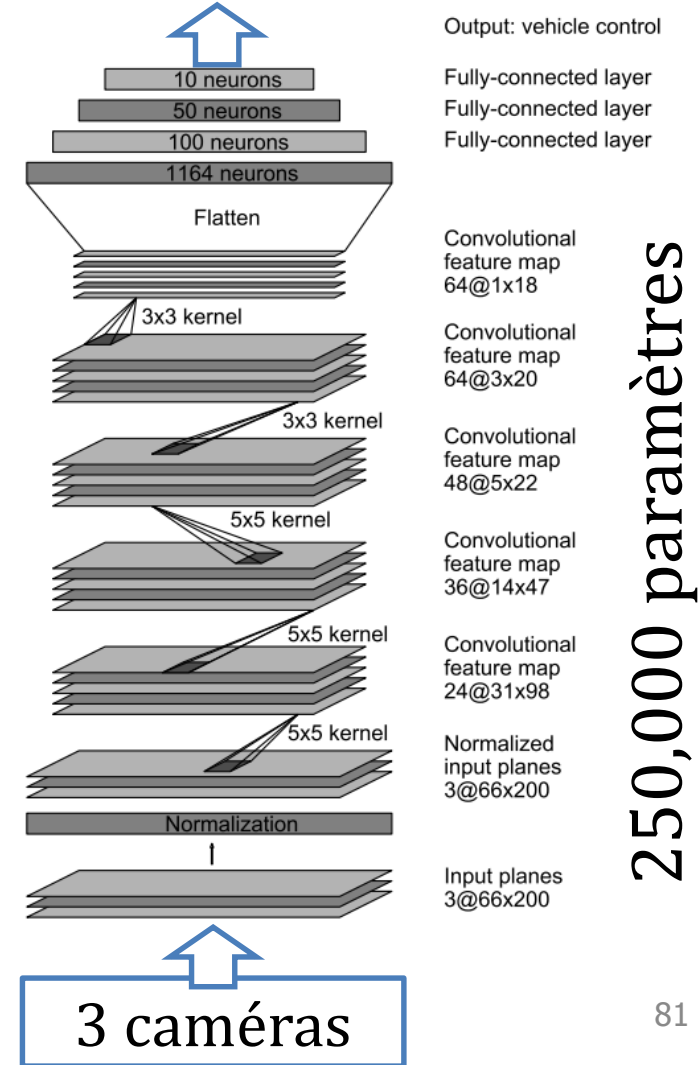
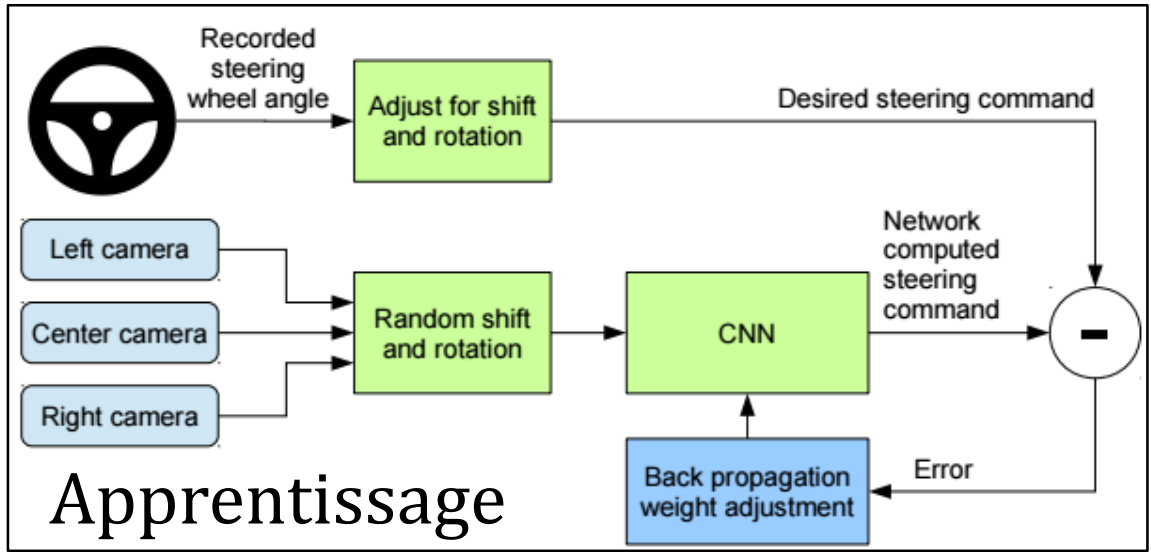
- 1. Drive a utility vehicle at the site.
- 2. Travel dismounted across rubble.
- 3. Remove debris blocking an entryway.
- 4. Open a door and enter a building.
- 5. Climb an industrial ladder and traverse an industrial walkway.
- 6. Use a tool to break through a concrete panel.
- 7. Locate and close a valve near a leaking pipe.
- 8. Connect a fire hose to a standpipe and turn on a valve.



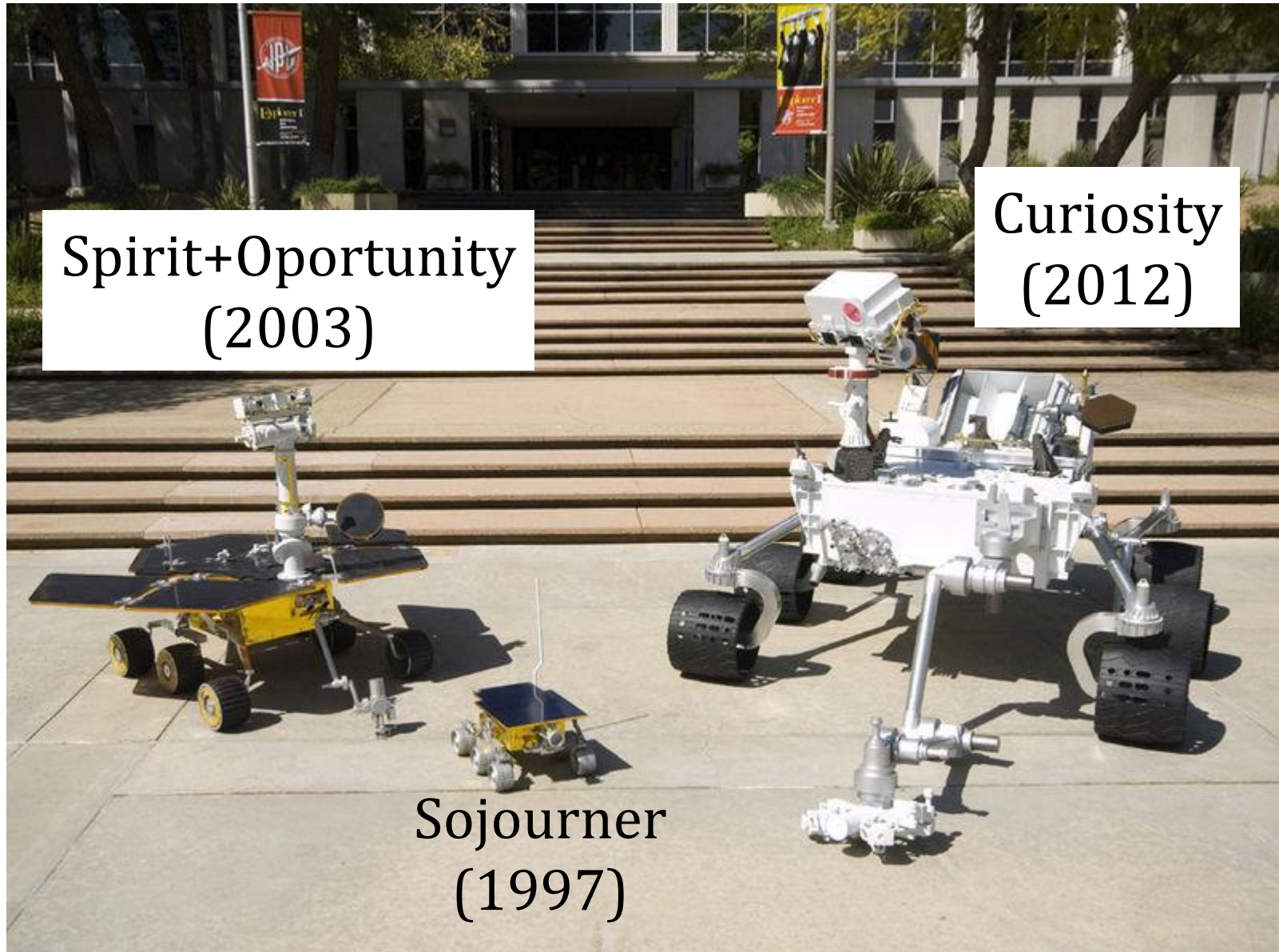
Nvidia end-to-end learning (2016)

- Basé sur les réseaux de neurones profonds convolutionnels (CNN)

Actionneurs (volant, pédales)



Exploration Mars



Spirit+Opportunity
(2003)

Curiosity
(2012)

Sojourner
(1997)

Exploration Mars

MER (2003) Mars Exploration Rover

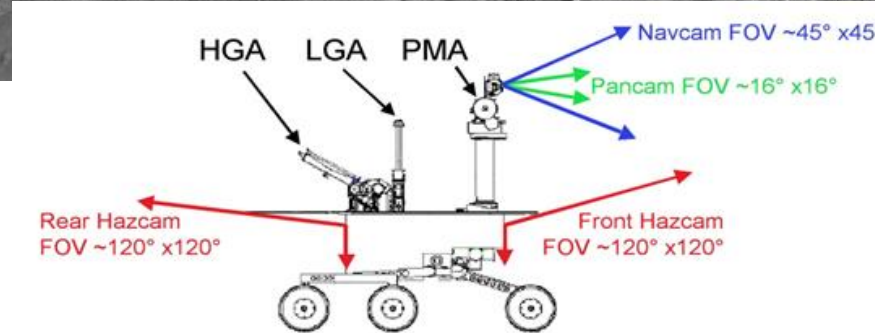
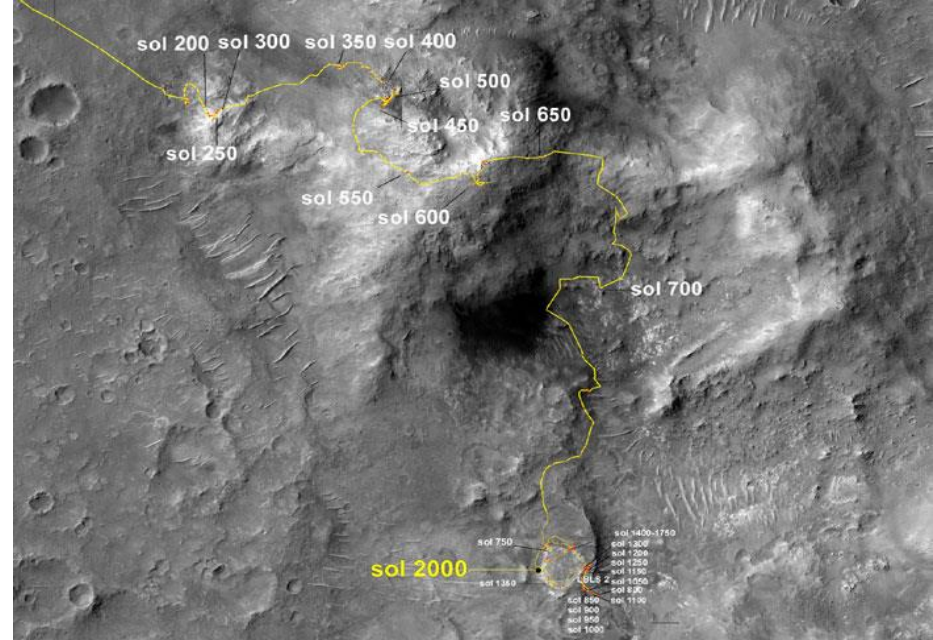
- Vitesse : 50 mm/s
- Distance totale:
 - ~~Spirit~~ : 7,7 km.
 - Opportunity : 45,0 km.
- Processeur
 - RAD 6000 (RISC 32 Bits, 20 MIPS)
 - 128 Mo



i386 - i486

MSL (2012) Mars Science Laboratory

- Vitesse : 25 mm/s
- Processeur *PowerPC*
 - RAD750 (200 MHz, 266 MIPS)
 - 256 Mo



Processeurs peu puissants :
algorithmes efficaces!

Révision des mathématiques

Concepts utilisés

- Trigonométrie et géométrie
- Coordonnées polaires vs. cartésiennes
- Matrices
- Linéarisation avec séries de Taylor
- Statistiques et probabilité (plus tard..)

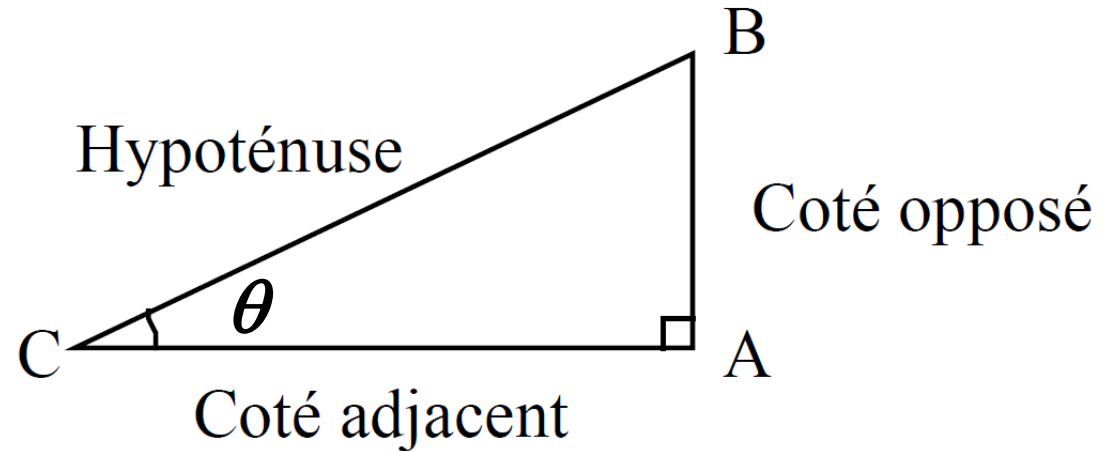
Trigonométrie et géométrie

sin, cos, tan

$$\sin \theta = \frac{\textit{opposé}}{\textit{hypothénuse}}$$

$$\cos \theta = \frac{\textit{adjacent}}{\textit{hypothénuse}}$$

$$\tan \theta = \frac{\sin \theta}{\cos \theta} = \frac{\textit{opposé}}{\textit{adjacent}}$$



Cherchez toujours le triangle rectangle dans les problèmes

sin, cos, tan

- Centaines d'identités trigonométriques

- http://en.wikipedia.org/wiki/List_of_trigonometric_identities

$$\sec \theta = \frac{1}{\cos \theta}, \quad \csc \theta = \frac{1}{\sin \theta}, \quad \cot \theta = \frac{1}{\tan \theta} = \frac{\cos \theta}{\sin \theta}.$$

puissances

$$\cos^2 \theta + \sin^2 \theta = 1$$

$$1 + \tan^2 \theta = \sec^2 \theta$$

signes

$$\sin(-\theta) = -\sin \theta$$

$$\cos(-\theta) = +\cos \theta$$

$$\tan(-\theta) = -\tan \theta$$

multiples

$$\sin 2\theta = 2 \sin \theta \cos \theta$$

$$\cos 2\theta = \cos^2 \theta - \sin^2 \theta$$

$$\tan 2\theta = \frac{2 \tan \theta}{1 - \tan^2 \theta}$$

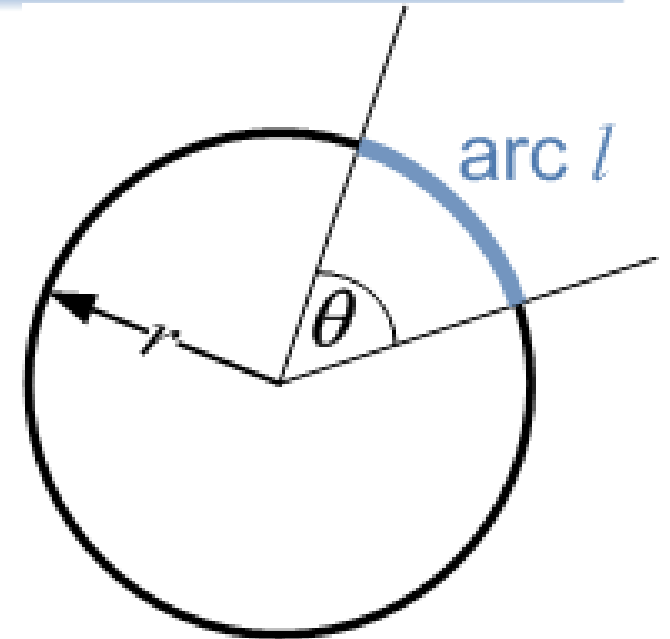
Seront incluses dans les examens

radian

- radian est une mesure d'angle

$$\theta = \text{arc}/\text{rayon} = l/r$$

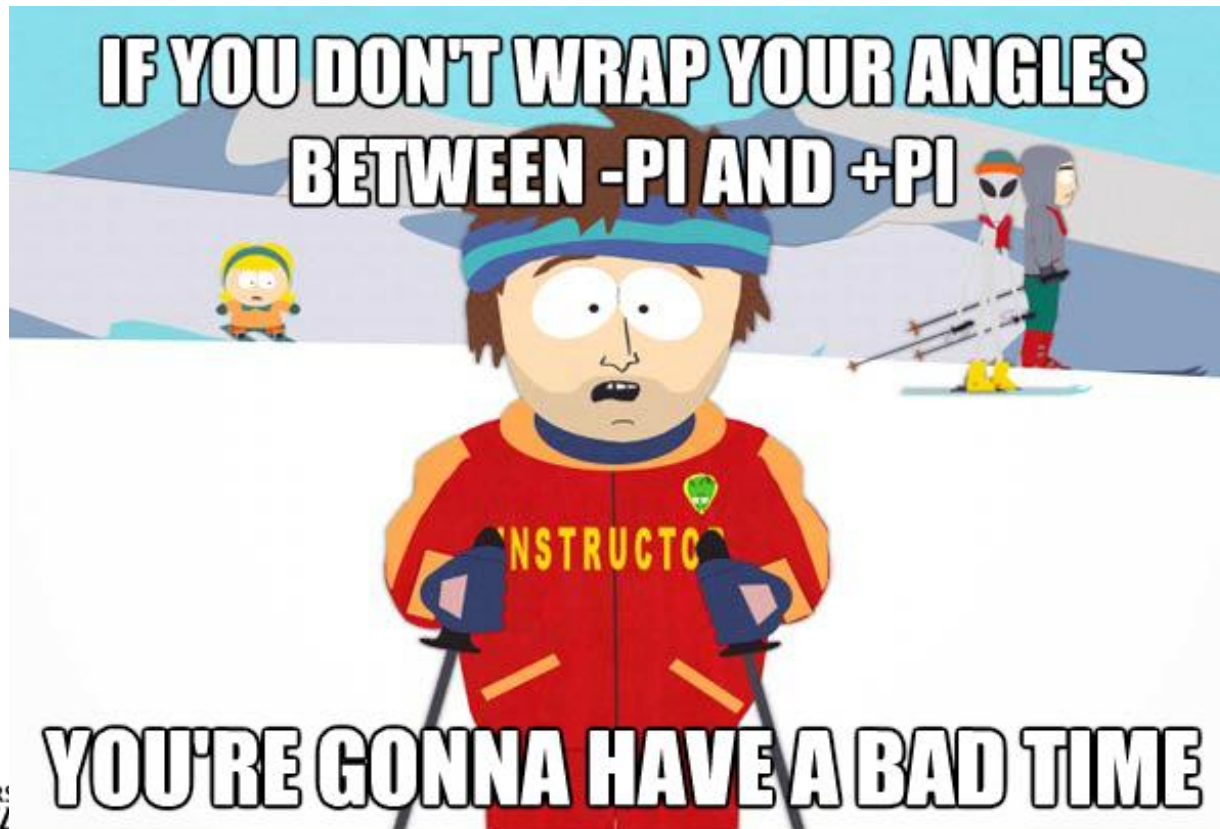
- $360^\circ = 2\pi \rightarrow 1 \text{ rad} \approx 57.3^\circ$



Attention aux calculs d'angles!

- Toujours conserver $-\pi < \theta \leq \pi$ rad!
- Exemple : $\pi + \frac{\pi}{2} = \frac{3}{2}\pi = -\frac{\pi}{2}$

matlab
`wrapToPi ()`



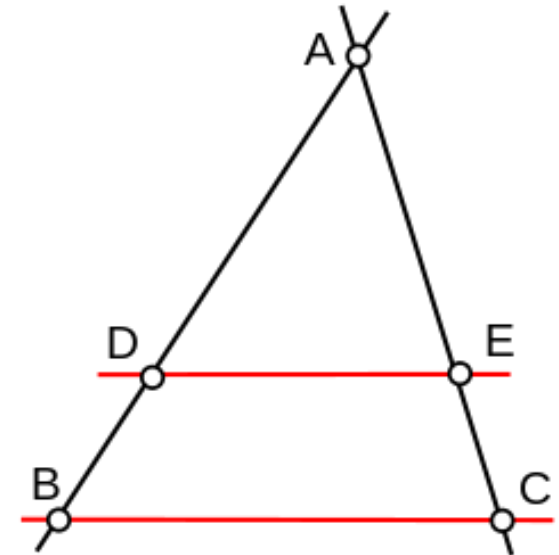
matlab
méfiez-vous des
fonction **std**,
var, etc... sur
des distributions
d'angles

Théorème de Thalès

- Triangle ABC
- Droite (DE) est parallèle à la droite (BC)
- On a :

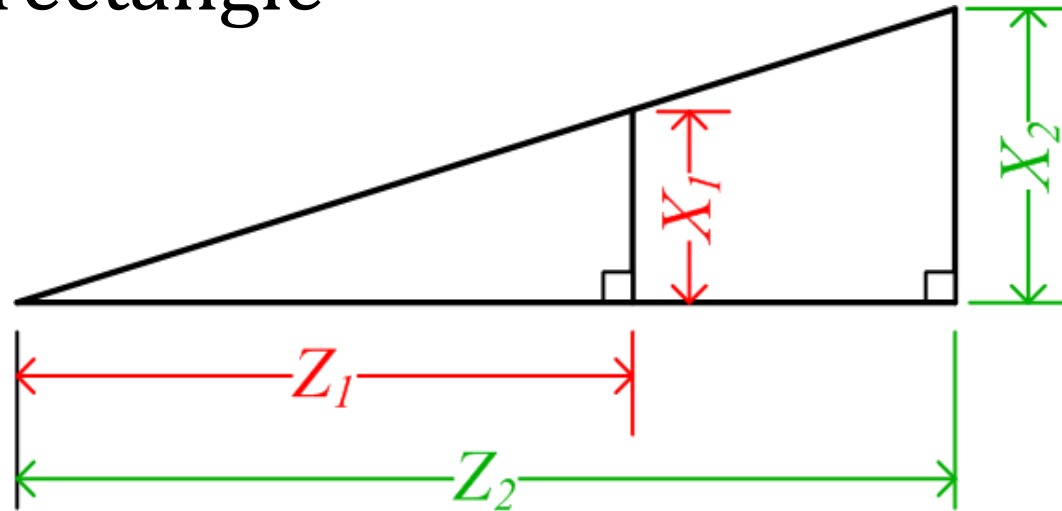
$$\frac{AD}{AB} = \frac{AE}{AC} = \frac{DE}{BC}$$

- On conserve les proportions
- ADE et ABC sont des triangles semblables (mêmes angles)



Théorème de Thalès

- Triangle rectangle



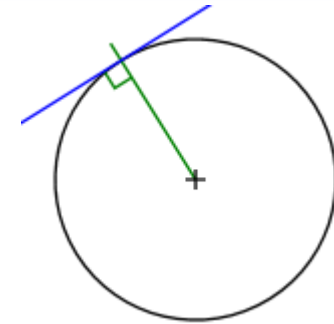
(on conserve les proportions)

$$\frac{X_1}{Z_1} = \frac{X_2}{Z_2}$$

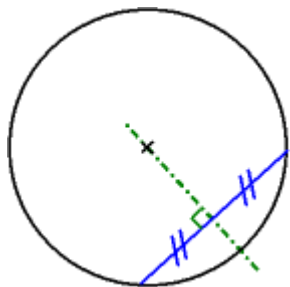
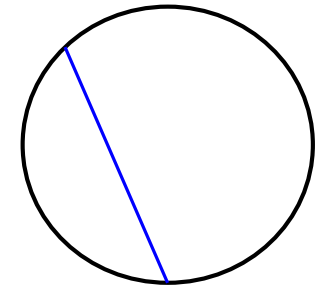
(utilisé par le modèle des caméras type sténopé)

Géométrie d'un cercle : définitions

- La **tangente** est perpendiculaire au **rayon**



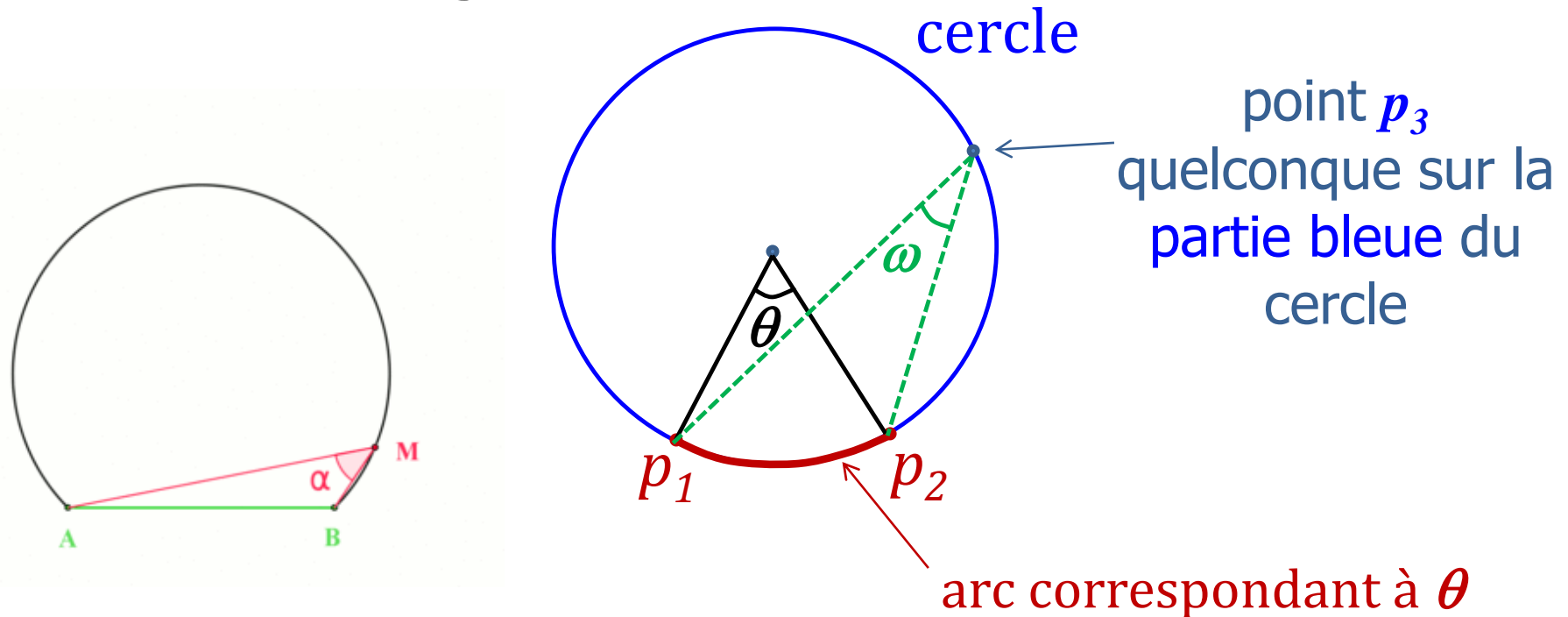
- Une **corde** est un segment de droite dont les extrémités touchent au cercle



- La **médiatrice** d'une **corde** passe par le centre du cercle

Théorème de l'angle inscrit et de l'angle au centre

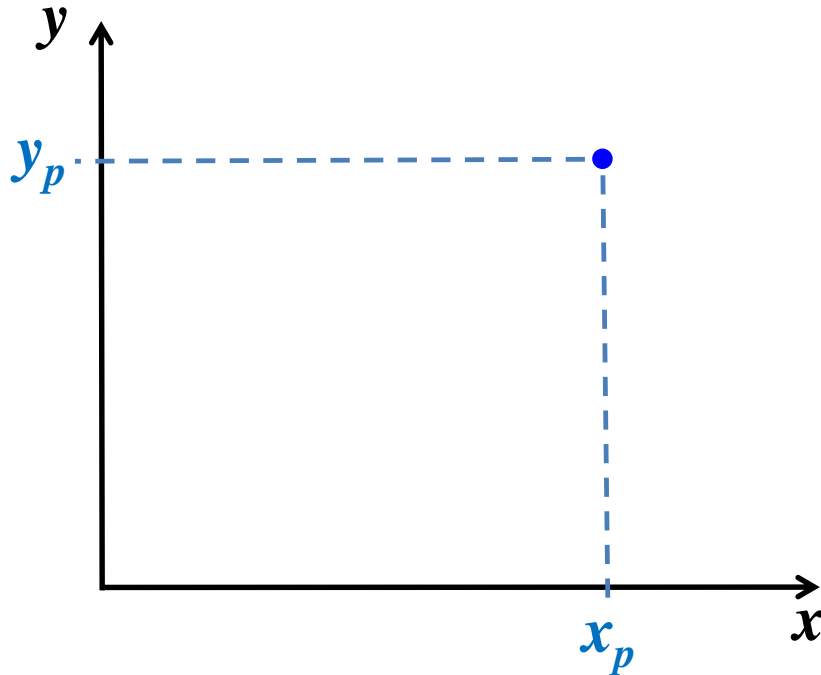
- Quel est l'angle inscrit ω ?



- $\omega = \theta/2$ (peu importe où se trouve p_3 en dehors de l'**arc**!)
(utilisé pour localisation par triangulation)

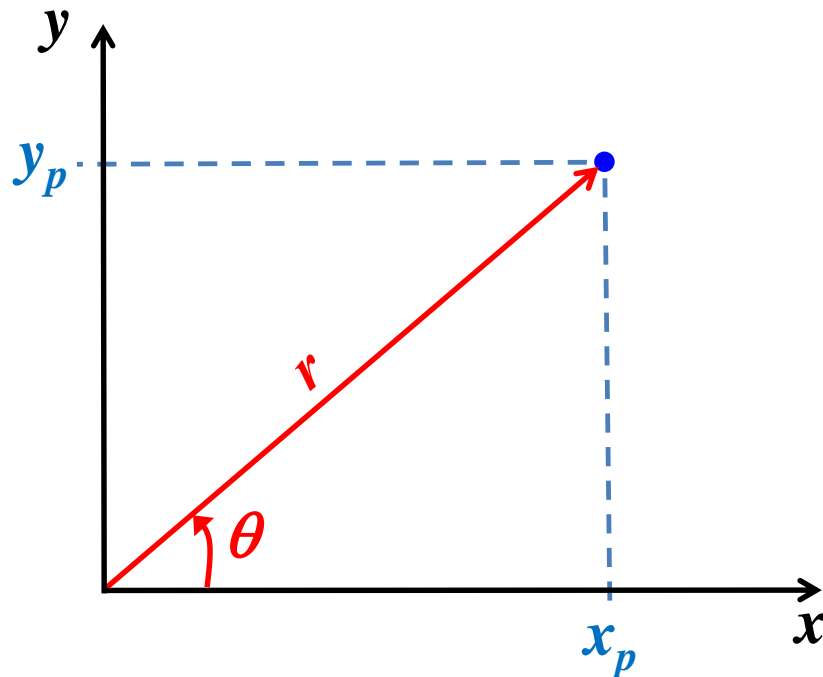
Coordonnées cartésiennes

- Pour représenter un endroit en 2D : (x_p, y_p)



Coordonnées polaires

- Pour représenter un endroit en 2D : (r, θ)



$$\begin{aligned} x_p &= r \cos \theta \\ y_p &= r \sin \theta \end{aligned}$$

dans les deux cas, 2 paramètres

(sera utile car les scans laser sont naturellement en coord. polaires)

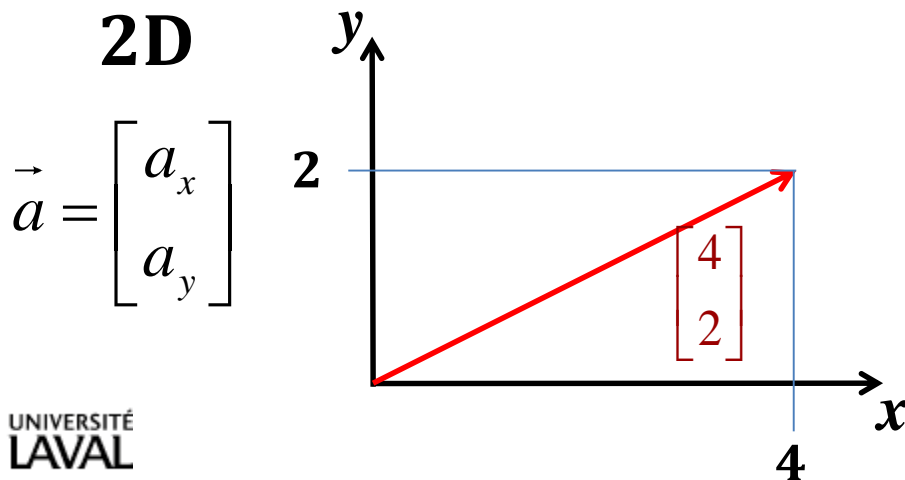
Algèbre linéaire

(et un peu de matlab!)

Vecteur

- Contenant de réels de dimension $n \times 1$
$$\vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

- Quantité géométrique représentant une magnitude et une direction dans l'espace



matlab

```
>> a = [1; 2]
a =
     1
     2

>> a = [1 2]'
a =
     1
     2
```

(nous servira à stocker l'état d'un système, en partie) 98

Vecteur : norme

- Magnitude d'un vecteur (norme-2 ou norme euclidienne)

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$$

- Exemple

$$\vec{a} = [2 \quad 3 \quad 4]^T, \quad \|\vec{a}\| = \sqrt{2^2 + 3^2 + 4^2} = \sqrt{29} = 5.3852\dots$$

```
>> a = [2 3 4]';
```

```
>> norm(a)
```

```
ans =
```

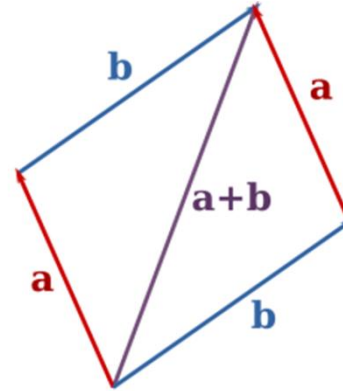
```
5.3852
```

matlab

Vecteur : addition

- Additionne chaque composante à la fois

$$\vec{a} + \vec{b} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_n + b_n \end{bmatrix}$$



- Commutatif : $a+b = b+a$
- Soustraction suit la même logique

```
>> a = [2 3 4]';  
>> b = [0.1 0.2 0.3]';  
>> a+b  
ans =  
  
    2.1000  
    3.2000  
    4.3000
```

matlab

Vecteur : produit scalaire (*dot product*)

- Deux vecteurs en entrée, 1 valeur numérique (un scalaire) en sortie

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

- Exemple

$$\vec{a} = [2 \ 3 \ 4]^T, \vec{b} = [5 \ 6 \ 7]^T \rightarrow \vec{a} \cdot \vec{b} = 2 \times 5 + 3 \times 6 + 4 \times 7 = 56$$

```
>> a = [2 3 4]';  
>> b = [5 6 7]';  
>> dot(a,b)
```

```
ans =  
    56
```

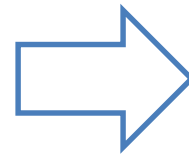
matlab

$\mathbf{a}' * \mathbf{b}$ aussi

Vecteur : angle

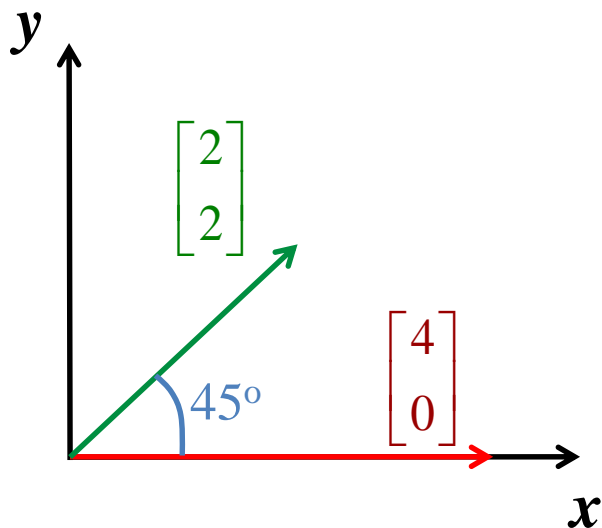
- On peut retrouver l'angle θ entre deux vecteurs grâce à la relation suivante :

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos \theta$$



$$\theta = \arccos \left(\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \right)$$

- Valable en 2D, 3D, etc...



```
>> a = [2 2]';  
>> b = [4 0]';  
>> t = acos(dot(a,b)/(norm(a)*norm(b)))  
t =  
    0.7854  
>> rad2deg(t)  
ans =  
    45.0000
```

matlab

Matrices

- Dimension $n \times 1$: vecteur colonne

$$\vec{a} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Matrice carrée : $n \times n$

majuscule

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

minuscule

- Matrice taille quelconque : $m \times n$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

2 x 3

matlab

```
>> a = [1; 2]
      a =
         1
         2
```

```
>> A = [1 3; 2 4]
      A =
         1     3
         2     4
```

Opérations sur matrices

- Addition $A+B$

$$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 1+5 \\ 1+7 & 0+5 & 0+0 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 8 & 5 & 0 \end{bmatrix}$$

– commutatif : $A+B = B+A$

- Multiplication avec scalaire $2A$

$$2 \cdot \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 8 & 2 \cdot -3 \\ 2 \cdot 4 & 2 \cdot -2 & 2 \cdot 5 \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 10 \end{bmatrix}$$

- Transpose $m \times n \rightarrow n \times m$

– colonne \leftrightarrow rangée

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -6 & 7 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \\ 2 & -6 \\ 3 & 7 \end{bmatrix}$$

matlab

$A+B$

$2 * A$

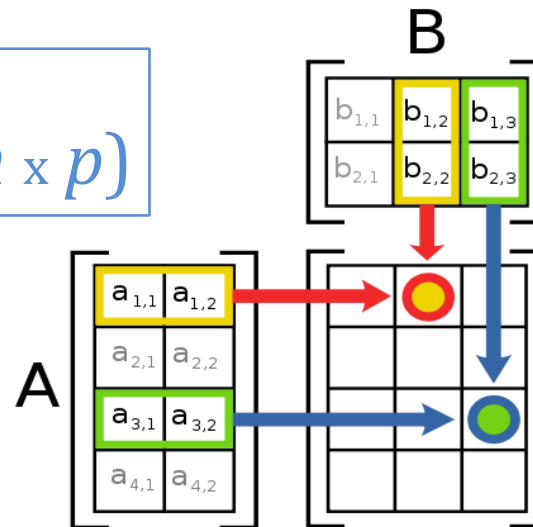
A'

Opérations sur matrices

- Multiplication AB
 - nombre de colonnes de A = nombre de rangées de B

tailles
 $(m \times n) * (n \times p) = (m \times p)$

**Important à savoir
pour l'examen**



matlab

$A * B$

- généralement pas commutatif $AB \neq BA$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 3 \end{bmatrix} \quad \text{mais} \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 0 & 0 \end{bmatrix}$$

Matrices

- Matrice identité $AI_n = I_m A = A$ si A est $m \times n$
(matrice carrée)

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

matlab

```
>> eye(4)
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

- Diagonale

$$\begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix}$$

- Triangulaire

inférieure $\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}$ (lower) and $\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$ supérieure (upper triangular matrix)

Inverse d'une matrice

- Inverse d'une matrice carrée A $n \times n$

$$AB = I_n$$

- Si B existe, A est dite inversible
- L'inverse est noté A^{-1}
- L'inverse est unique : $AA^{-1} = A^{-1}A = I$

matlab *backslash*

```
>> A*inv(A)
ans =
    1.0000         0
    0.0000    1.0000
```

```
>> A\A
ans =
    1         0
    0         1
```


Inverse d'une matrice

matlab

```
>> X = [1 3; 2 4]
```

```
X =
```

```
    1    3  
    2    4
```

```
>> inv(X)
```

```
ans =
```

```
 -2.0000    1.5000  
  1.0000   -0.5000
```

Matrice singulière

- Matrice carrée A est **singulière** si n'a pas d'inverse

$$AB = I_n$$

dans ce cas-ci, B n'existe pas!

Matrice : système d'équations linéaires

- Soit le système d'équation suivant :

$$4x_1 + 3x_2 + 7x_3 = y_1$$

$$5x_1 + 2x_2 = y_2$$

$$4x_2 + 3x_3 = y_3$$

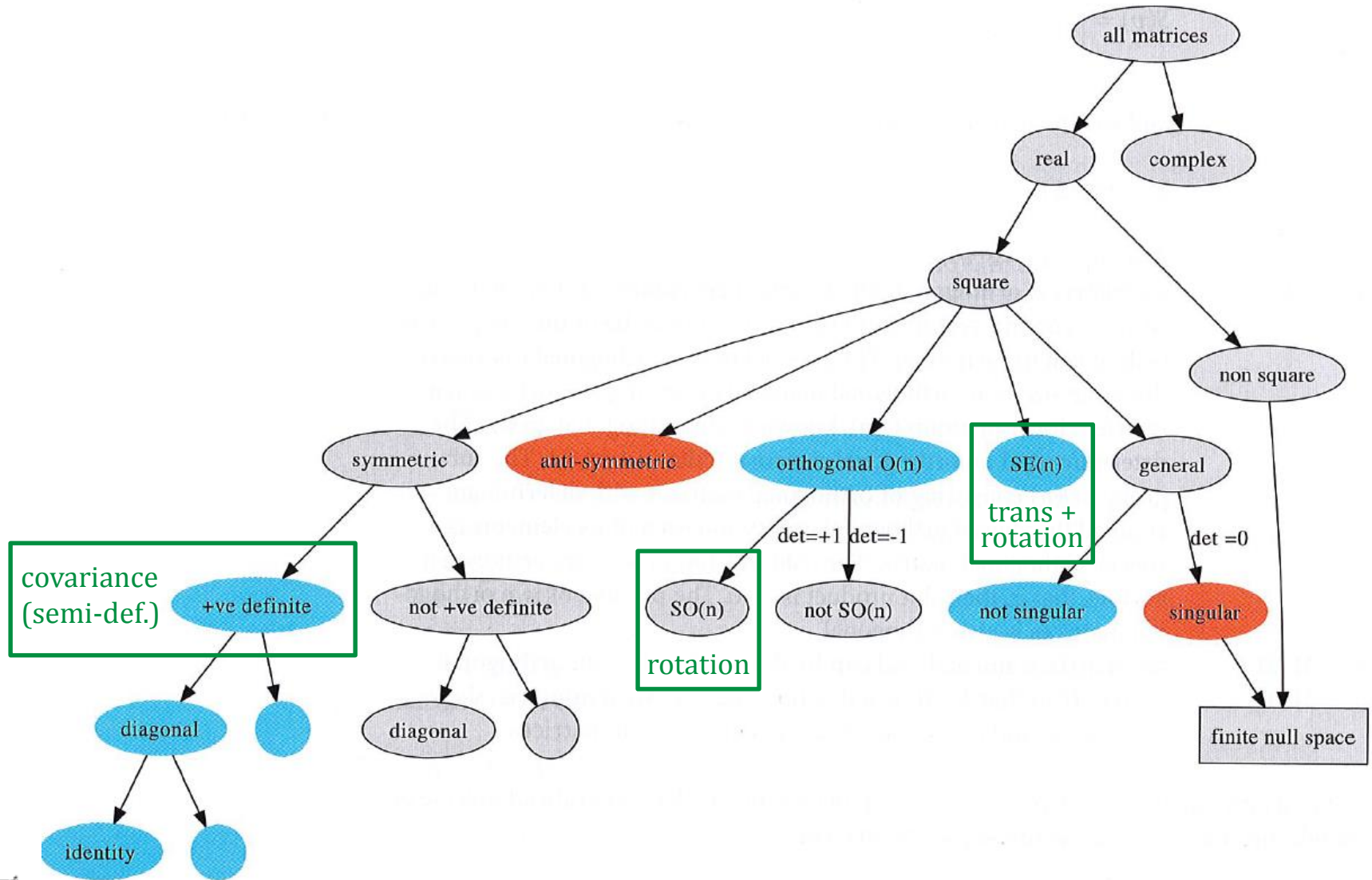
- On peut représenter sous la forme $A\mathbf{x}=\mathbf{y}$:

$$\begin{bmatrix} 4 & 3 & 7 \\ 5 & 2 & 0 \\ 0 & 4 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

*(Filtre de Kalman,
Kalman étendu)*

A (va encoder notre système)

Taxonomie de matrices utiles



Dérivée

Définition de la dérivée

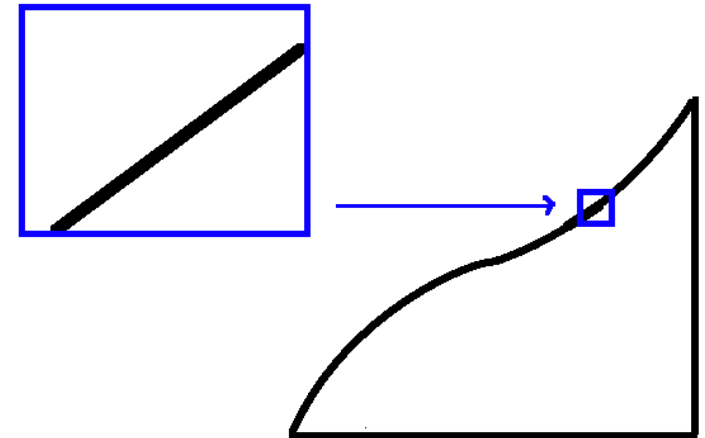
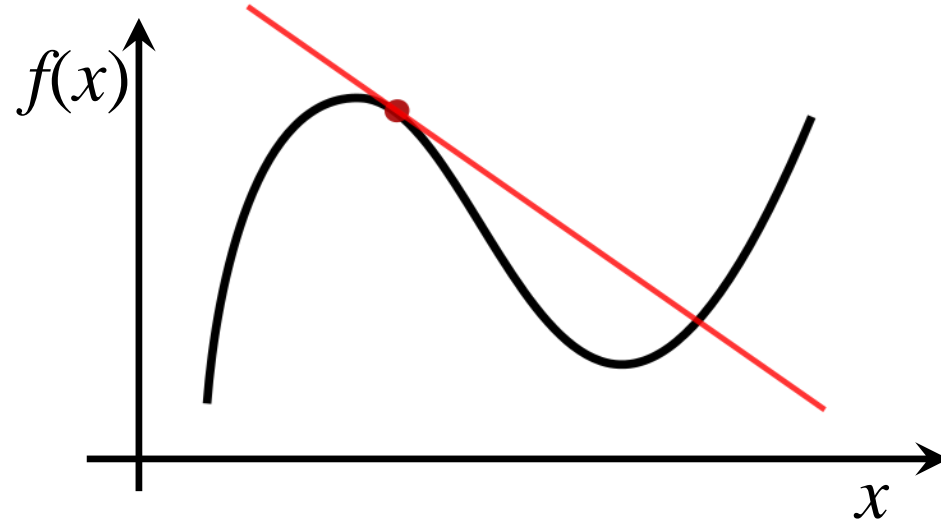
- **Pente** d'une fonction $f(x)$ au point x

- Définition :

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

(Pensez $\frac{\Delta y}{\Delta x}$!)

- Une fonction continue est approximativement linéaire lorsque « zoomé »



Dérivée simple, double

- Exprimée

$$\frac{d}{dx} f(x) = f', \quad \frac{d^2}{dx^2} f(x) = f''$$

dérivée dans le temps

$$\frac{d}{dt} f(t) = \dot{f}, \quad \frac{d^2}{dt^2} f(t) = \ddot{f}$$

Règles des dérivées utiles

- Fonction constante : $f(x) = a$, alors $f'(x) = 0$
- Fonction puissance : $f(x) = x^n$, alors $f'(x) = nx^{n-1}$
- *(pour les autres, utilisez un site en ligne! ;)*
- Somme de deux fonctions $[f(x) + g(x)]' = f'(x) + g'(x)$
- Dérivée des fonctions composées

$$f(g(x))' = f'(g(x))g'(x)$$

- Dérivée partielle par rapport à x_i pour $f(x_1, x_2, \dots, x_m)$
 - On considère les autres x_j comme constants

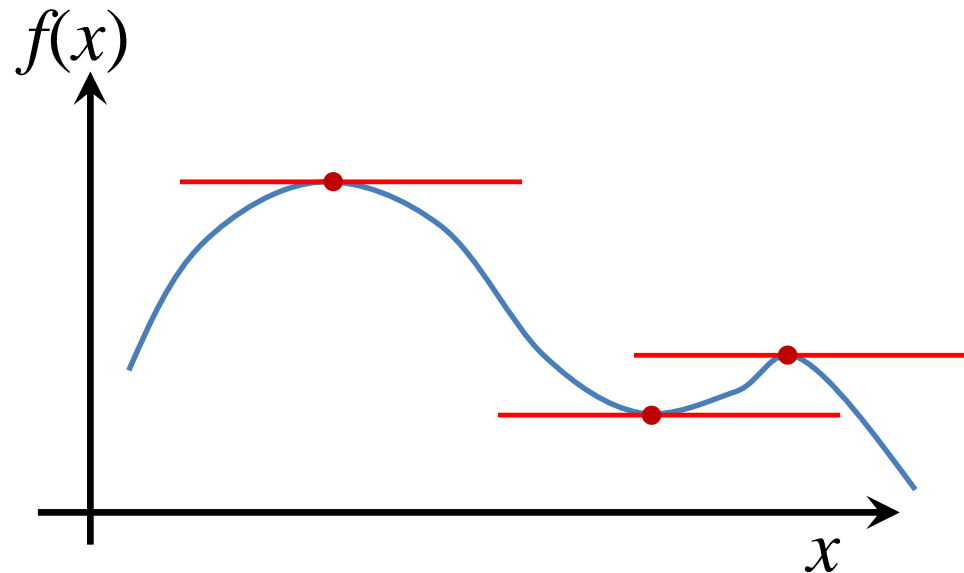
– e.g. $\frac{\partial}{\partial x_2} [3x_1 + 4x_2 + 7x_3] = 4$

notez le d arrondi ↗

(Jacobiennes du filtre de Kalman Étendu)

Maximum/minimum d'une fonction f

- À un maximum/minimum local ou global, la dérivée de la fonction f est nulle



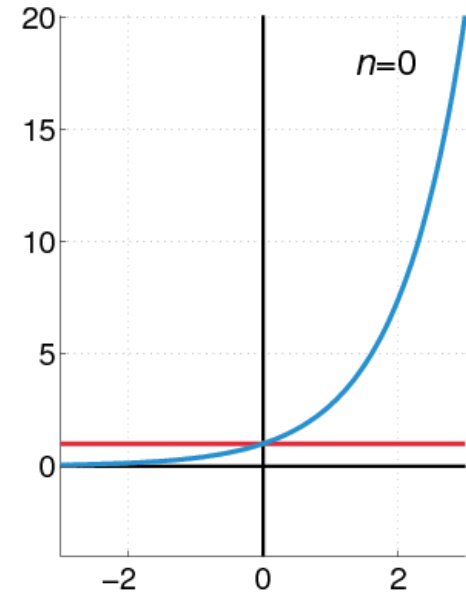
Séries de Taylor

Série de Taylor

- Représentation d'une fonction f par une série infinie de dérivées de f à un point a

$$f(x)|_a = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots$$

(factoriel $1!=1$ $2!=1 \times 2=2$ $3!=1 \times 2 \times 3=6$ $4!=1 \times 2 \times 3 \times 4=24$)

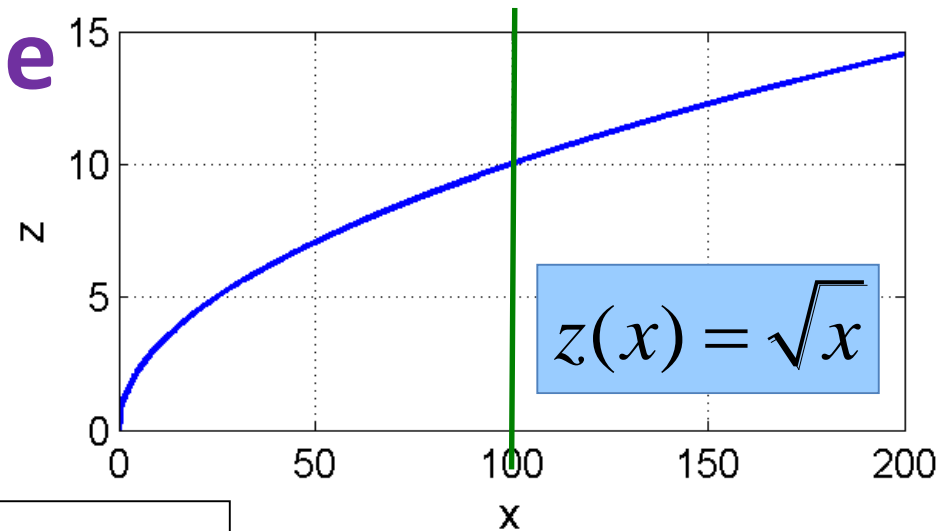


- Va nous servir à **approximer** des fonctions non-linéaires autour d'un endroit a particulier

(un peu partout dans le cours)

Série de Taylor : exemple

- Approximer $z(x)$
autour de $x = a = 100$



$$z(x) = z(a) + \frac{z'(a)}{1!} (x-a) + \frac{z''(a)}{2!} (x-a)^2 + \dots$$

$$z' = \frac{d}{dx} \sqrt{x} = \frac{1}{2\sqrt{x}}$$

$$z'' = \frac{d^2}{dx^2} \sqrt{x} = -\frac{1}{4x^{3/2}}$$

$$z_1(x) = 10 + \frac{1}{2\sqrt{100}} (x-100)$$

approximation avec 1 dérivée

$$z_2(x) = 10 + \frac{1}{2\sqrt{100}} (x-100) - \frac{1}{2 \cdot 4(100)^{3/2}} (x-100)^2$$

approx. avec 2 dérivées

Série de Taylor : exemple

- Approximer $z(x)$ autour de $a=100$

$$z_1(x) = 10 + 0.05(x - 100)$$

$$z_2(x) = 10 + 0.05(x - 100) - 0.000125(x - 100)^2$$

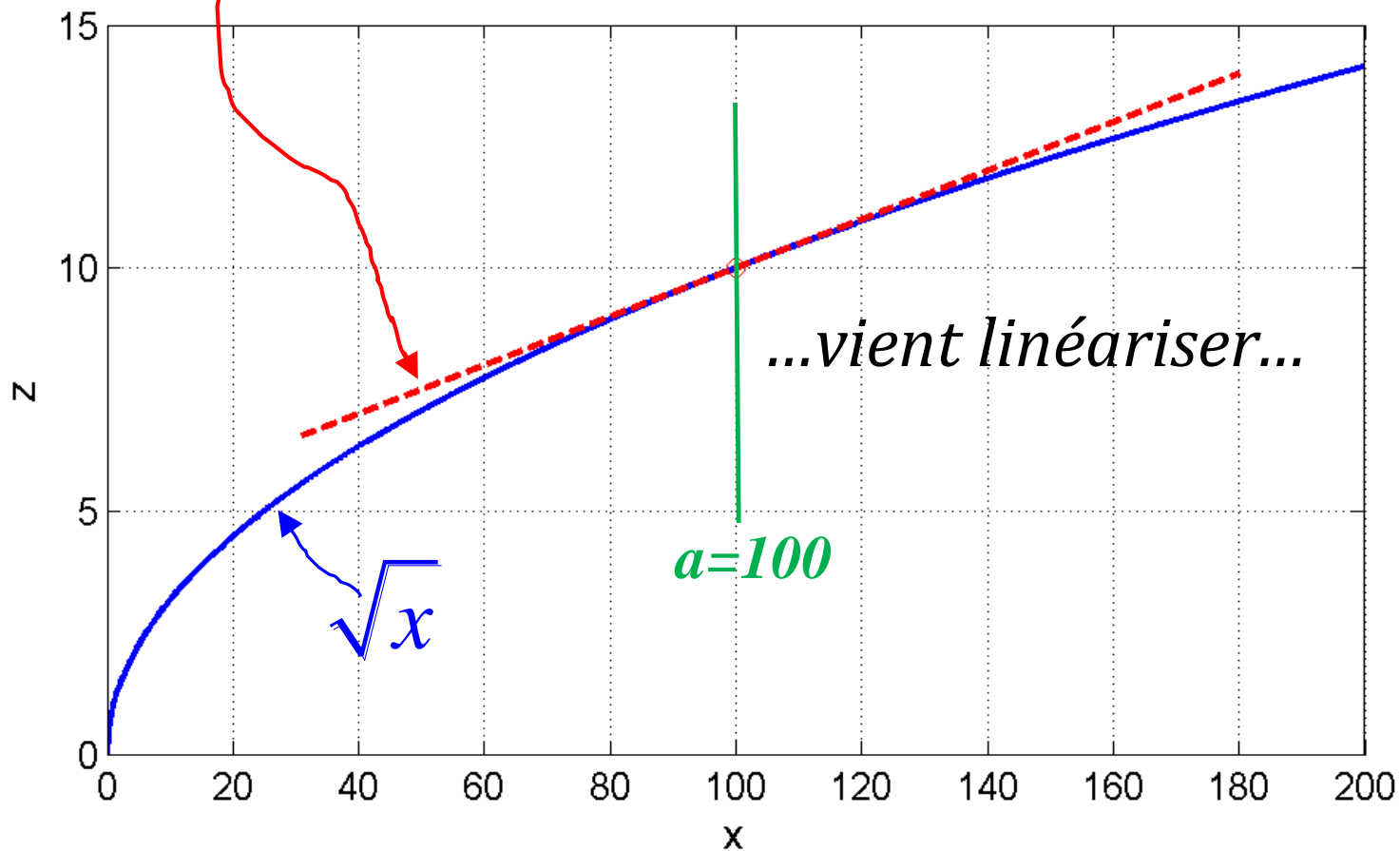
approximations

x	$z(x)$ exacte	$z_1(x)$	$z_2(x)$
100	10	10	10
100.5	10.0250	10.0250	10.0250
105	10.2470	10.2500	10.2469
110	10.4881	10.5000 (0.114 %)	10.4875 (0.0056%)

% d'erreur

Série de Taylor : linéarisation

$$z_1(x) = 10 + \frac{1}{2\sqrt{100}}(x - 100) = 10 + 0.05(x - 100)$$



Approximation linéaire de sin, cos

- Rendre les fonctions trigonométriques courantes *linéaires*, pour des petits angles

$$\theta \ll 1 \text{ rad}$$

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots$$

$$a = 0$$

Fonction	Série de Taylor	Approximation linéaire
$\sin(\theta)$	$\sin(0) + \cos(0)\theta - 0.5\sin(0)\theta^2 + \dots$	θ
$\cos(\theta)$	$\cos(0) - \sin(0)\theta - 0.5\cos(0)\theta^2 + \dots$	1

Note : $\frac{d}{dx}\sin(x) = \cos(x)$, $\frac{d}{dx}\cos(x) = -\sin(x)$

Série de Taylor : résumé

- Nous permettre d'approximer n'importe quelle fonction
- Choisis un « point d'opération » a
- Choisis le nombre de terme
 - pour rendre linéaire, seulement la 1^{ère} dérivée!

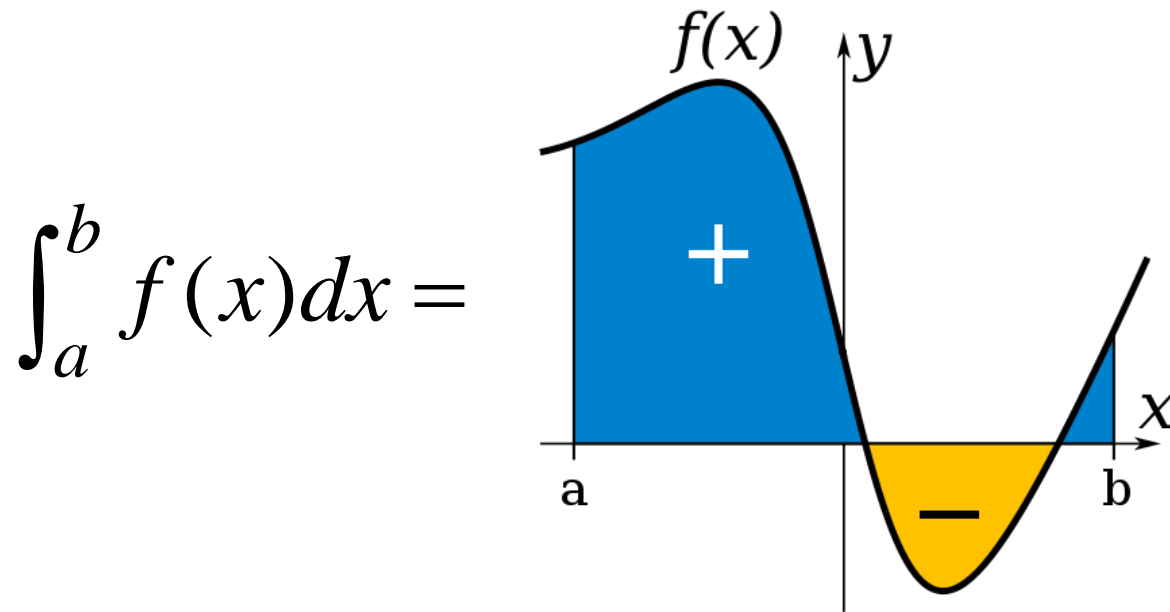
$$f(x) \approx f(a) + f'(a)(x - a)$$

- fonction linéaire == facile à utiliser

Intégration

Définition

- Aire sous la courbe

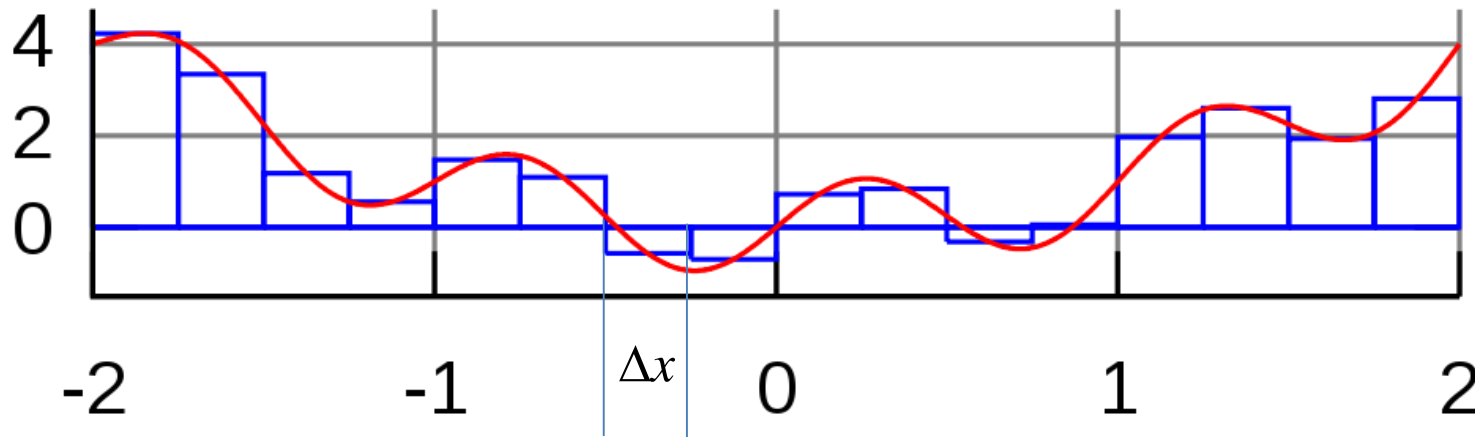


- Opération inverse de la dérivée

Comment la calculer?

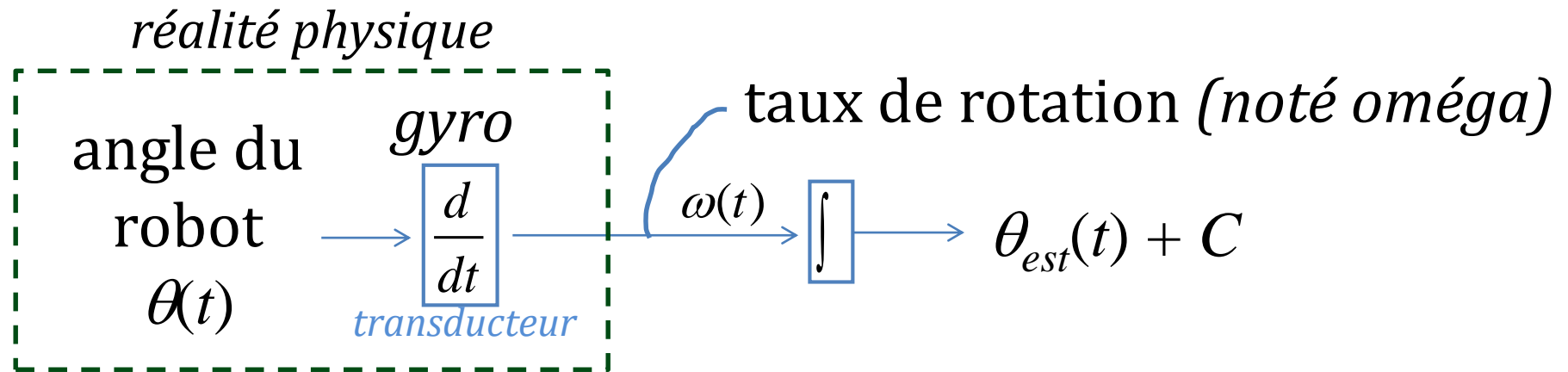
- Dans le cours, on fera essentiellement de **l'intégration numérique**

$$\int_a^b f(x)dx \Rightarrow \sum_{i=a}^b f[i]\Delta x$$



Exemple avec gyroscope

- Gyroscope : donne un **taux** de rotation ($^{\circ}/s$)

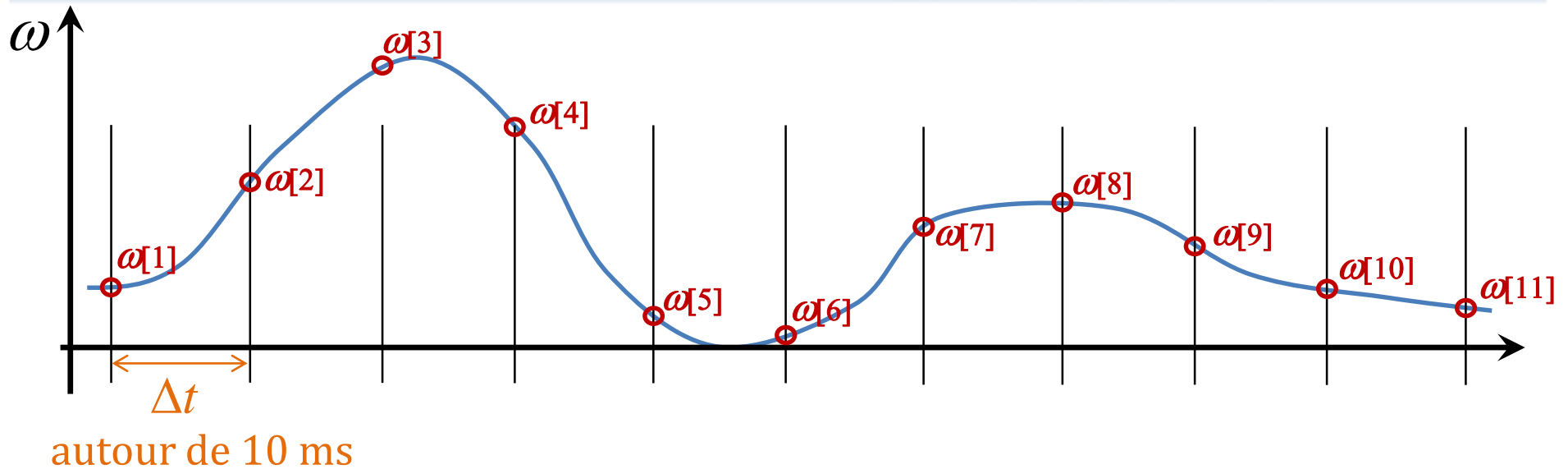


- Signal $\omega(t)$ échantillonné à chaque Δt : $\omega[i]$
- On retrouve l'angle par :

$$\theta[i+1] = \omega[i+1]\Delta t + \theta[i]$$

où i est un index (entier)

Exemple avec gyroscope



```
dt = 0.01 % Periode d'échantillonnage
```

```
w = [0.2, 0.3, 0.1, -0.05, -0.07, -0.02];
```

```
theta(1) = w(1)*dt + 0.04; le « C » des intégrales non-définies  
est l'angle initial (si connu)
```

```
for index = 2:size(w,2)
```

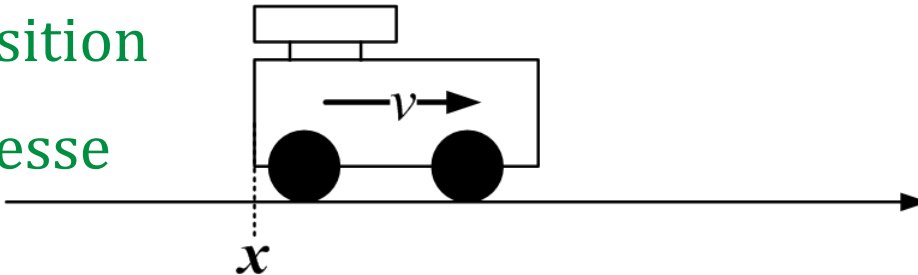
```
    theta(index) = w(index)*dt + theta(index-1);
```

```
end
```

(voyez aussi la fonction `cumsum`)

Robotique : combine matrice + intégration

- Système *linéaire* (parfois *approximatif*) représentant l'évolution de l'état d'un robot dans le temps

- Vecteur d'état $X = \begin{bmatrix} x \\ v \end{bmatrix}$
 ← position
 ← vitesse


équations du système $\begin{cases} x_{t+1} = x_t + v_t \Delta t \\ v_{t+1} = v_t \end{cases}$ *intégration numérique simplifiée*

$$X_{t+1} = \begin{bmatrix} x_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ v_t \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} X_t$$

(utilisé dans la section estimation d'état)