AIC-Automatisation avancée, intelligence artificielle et cognitique

9. Robots mobiles autonomes

Jean-Daniel Dessimoz

HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD
www.heig-vd.ch

Institut d'Automatisation industrielle — LaRA — Laboratoire de Robotique et Automatisation

Hes·so Haute Ecole Spécialisée de Suisse occidentale

AIC-Automatisation avancée, intelligence artificielle et cognitique

Contenu

- Introduction
- Notion de modèle ; métrique pour le traitement d'information et pour la cognitique
- Choix d'une structure de commande
- Intelligence artificielle et « machine learning »
- Commande à logique floue
- Commande neuronale, yc. « deep learning »
- Commande multimodale
- Commande à algorithme génétique
- **Robots mobiles autonomes**
- Robot humanoïde NAO
- Conclusion

Contenu des *Exposés et exercices*

Notion de modèle ; métrique pour le traitement d'information et pour la cognitique	**4p**
Choix d'une structure de commande	**2p**
Intelligence artificielle	**2p**
Commande à logique floue	**2p**
Commande neuronale	**2p**
Commande multimodale	**2p**
Commande à algorithme génétique	**2p**
Robots mobiles autonomes et humanoïdes	**4p**
Réserve et contrôle continu (TE, corr.)	**6p**

Travaux de laboratoire associés

Estimation de grandeurs cognitives (essais en simulation avec programmes d'évitement d'obstacles)	**L-AIC-1**
Test d'intelligence artificielle selon Turing et utilisation d'Eliza	**L-AIC-2**
Commande neuronale	**L-AIC-3**
Commande à logique floue	**L-AIC-4**
Commande à algorithme génétique	**L-AIC-5**
Commande multimodale	**L-AIC-6**
Robot mobile autonome	**L-AIC-7**
Robot humanoïde NAO	**L-AIC-8**
Inférences bayésiennes	**L-AIC-9**
Sur demande, l'étudiant peut échanger l'une des manipulations ci-dessus par un autre sujet (cf. manipulations LaRA)	

AIC-Automatisation avancée, intelligence artificielle et cognitique

Contenu

9 Robots mobiles autonomes

Animats - automates en situation - robots insectes

- Labyrinthe
- Fourmi

Robots intelligents - avec modélisation

- Labyrinthe bis
- Expansion

Robots mobiles autonomes robustes

- Golf – Piaget environment

Robots coopératifs

- Robocup-at-Home

9 Robots mobiles autonomes

- **Animats - automates en situation - robots insectes**
  - Labyrinthe
  - Fourmi
- **Robots intelligents - avec modélisation**
  - Labyrinthe bis
  - Expansion
- **Robots mobiles autonomes robustes**
  - Golf – Piaget environment
- **Robots coopératifs**
  - Robocup-at-Home

9 Animats - automates en situation - robots insectes (1 de 6)

- Référence: Rodney Brooks, MIT
- « Le modèle, c’est le monde » : aucune modélisation ne serait nécessaire; il suffirait d’être « en situation » , et d’observer avec des capteurs ce qui se passe.
- C’est une réaction contre l’IA qui visait/vise une représentation complexe du monde dans l’ordinateur

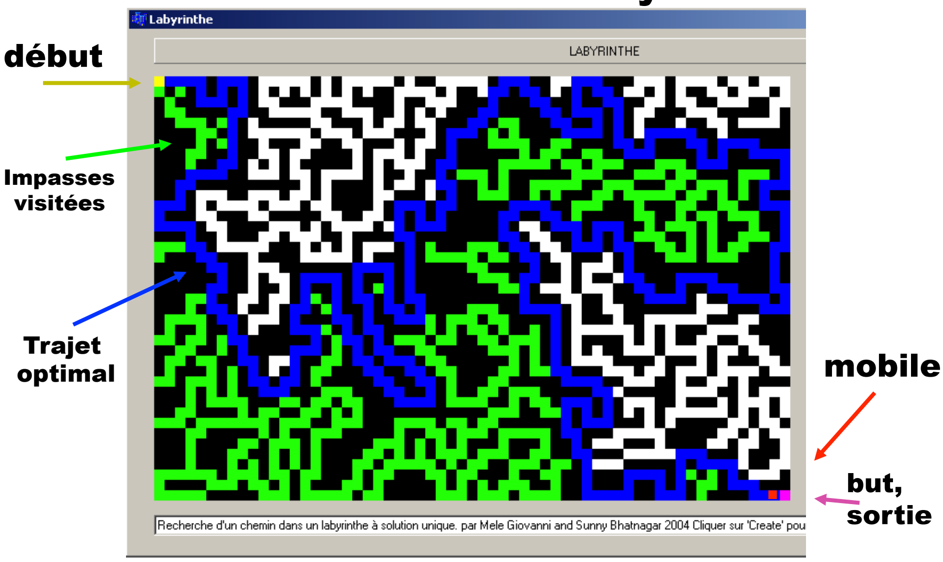
Robots mobiles autonomes

9 Animats - automates en situation - robots insectes (2 de 6)

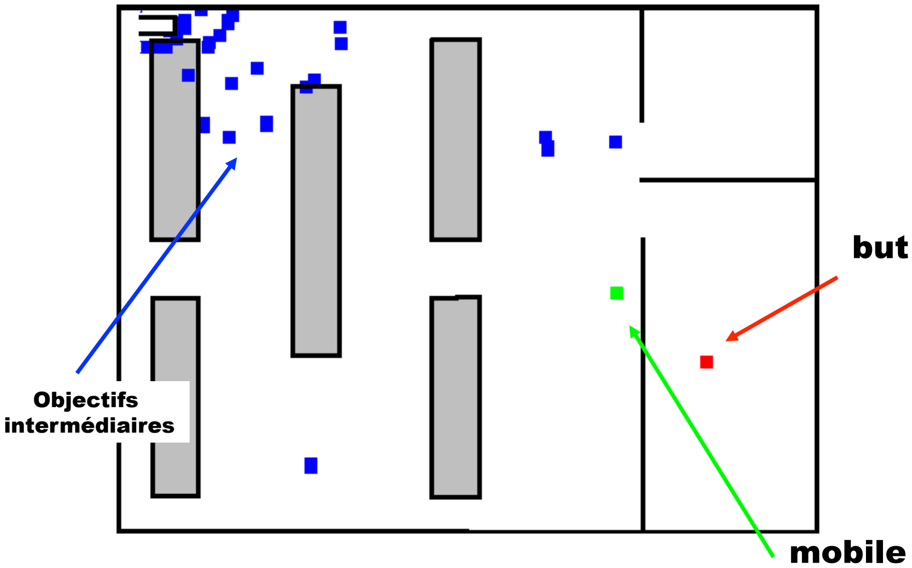
- Entrées par capteurs
- Sorties sur actionneurs
- Typiquement, entre les deux:
  - Réflexes!
- (pas de modèle, pas de mémoire)

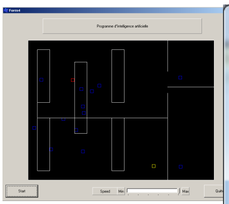
Robots mobiles autonomes

Labyrinthe

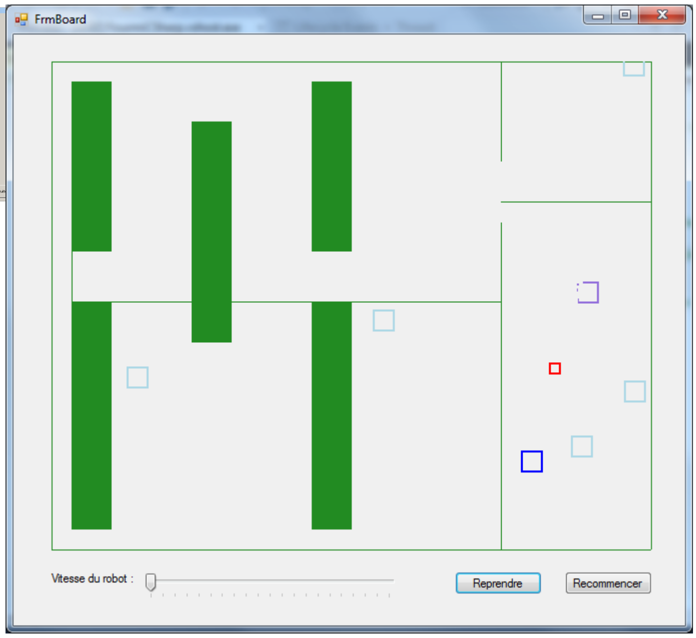


Robots mobiles autonomes « Fourmi »





Version C++, iAi-LaRA, HESSO.HEIG-VD, 2011



Version C#, Jonas Baggett, iAi-LaRA, HESSO.HEIG-VD, 2017

9 Animats - automates en situation - robots insectes (6 de 6)

Conclusion

- **Ça peut marcher! :**
  - Ex: - labyrinthe (loi simple, de type réflexe).
  - Ex: - « fourmi » (2 comportements, l'un étant réflexe et l'autre, aléatoire)
- **Limites:**
  - (pas de simulation possible, interfaces nécessaires)
  - Maintenant! (présent seulement, pas de passé, ni de futur accessible)
  - Ici! (portée spatiale toujours limitée)

9 Robots mobiles autonomes

Animats - automates en situation - robots insectes
- Labyrinthe
- Fourmi

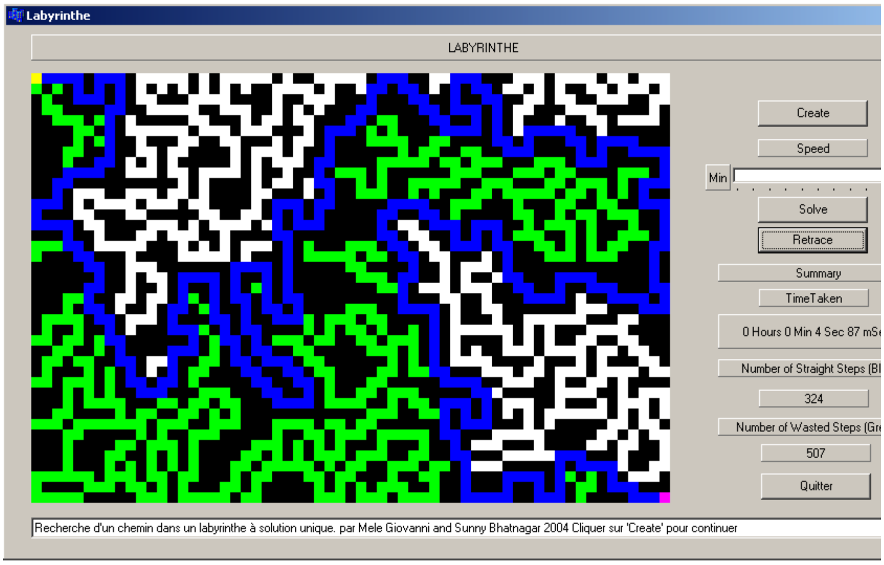
Robots intelligents - avec modélisation
- Labyrinthe bis
- Expansion

Robots mobiles autonomes robustes
- Golf – Piaget environment

Robots coopératifs
- Robocup-at-Home

Robots mobiles autonomes

Labyrinthe (2/2)



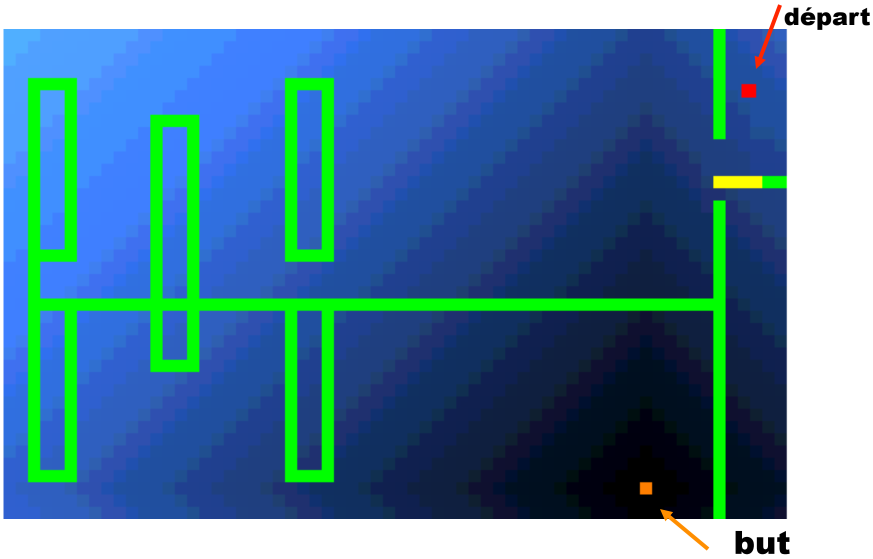
Robots mobiles autonomes

Navigation - Indice de proximité à l' objectif

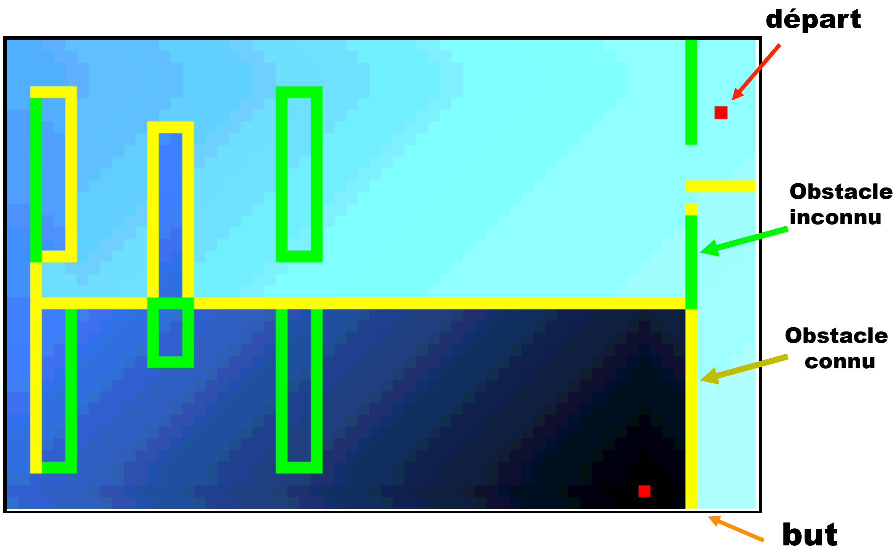
6	5	4	3	2	3	4
5	4	3	2	1	2	3
6	5		1	0	1	2
7	6		2	1	2	3
6	5	4	3	2	3	4

6	5	4	3	2	3	4
5	4	3	2	1	2	3
6	5		1	0	1	2
7	6		2	1	2	3
6	5	4	3	2	3	4

Robots mobiles autonomes « Expansion » (1/2)



Robots mobiles autonomes « Expansion » (2/2)

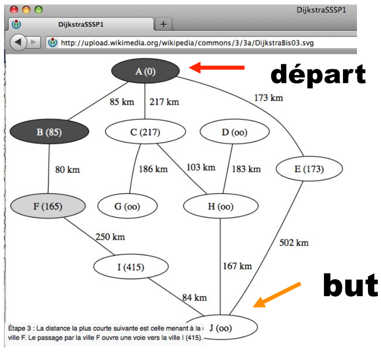


Robots mobiles autonomes

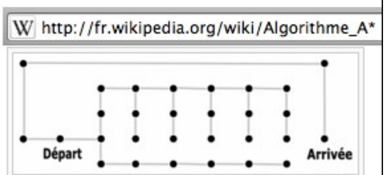
Navigation

« Classiques »

- Algorithme de Dijkstra (yc calcul de distance, ou temps, ou coûts) pour choix d'itinéraires entre villes (1959) ou serveurs internet par ex..



- Algorithme A*, Astar : sur une surface 2 D, visiter les cellules plus proches de la ligne droite entre position courante et objectif, tenant compte de façon incrémentale des obstacles.



Un labyrinthe simple, où l'algorithme A* ne sera pas très efficace

Potentiel but +, obstacles -
Mix des algorithmes proposés

Robots mobiles autonomes

9 Robots mobiles autonomes

Animats - automates en situation - robots insectes
- Labyrinthe
- Fourmi

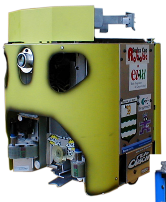
Robots intelligents - avec modélisation
- Labyrinthe bis
- Expansion

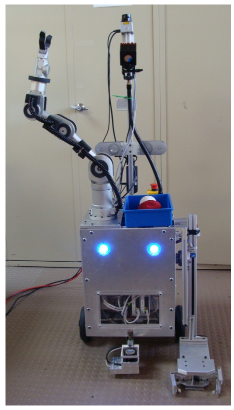
Robots mobiles autonomes robustes
- Golf – Piaget environment

Robots coopératifs
- Robocup-at-Home

Robots mobiles autonomes

ARY - Autonomous Mobile Robot with Robust Architecture and Components



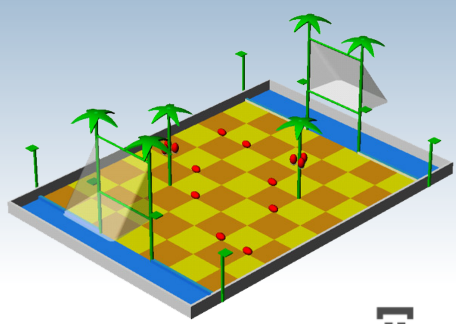


Robot-CH

Authors: Uebelhart Nicolas
Pierre-François

Robots mobiles autonomes

Lomu, an Autonomous Mobile Robot with Robust Architecture and Components



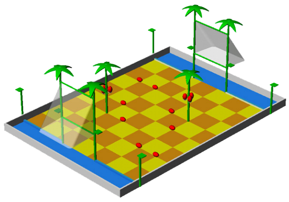
Robot-CH

Robots mobiles autonomes

The autonomous mobile robot LOMU (1 of 3)

Introduction
The A.M.R. LOMU
The architecture
Method of displacement
Control & programming
Conclusion

- LOMU was developed to take part in the contest of EUROBOT 2004



- The aim of Eurobot04 was to score a maximum of points in 90 seconds by placing small rugby balls in the adversary zone or by shooting them into the opposite goal

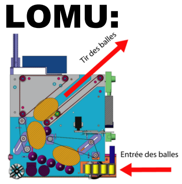
Robots mobiles autonomes

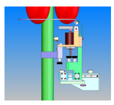
The autonomous mobile robot LOMU (2 of 3)

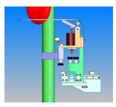
Introduction
The A.M.R. LOMU
The architecture
Method of displacement
Control & programming
Conclusion

- The mains characteristics of LOMU:

A catching balls system can store three balls in the robot. The balls are shooting the balls with a system of belts

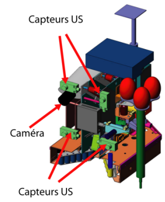






The balls are catched by a hinged arm. Equipped with an ejector finger positioned on the left side of the robot

A video camera takes an image of the play Ground at the beginning of each match to Locate the positions of the coconut trees. Four ultrasound sensors and contact sensors Allow avoid obstacles



Robots mobiles autonomes

The autonomous mobile robot LOMU (3 of 3)

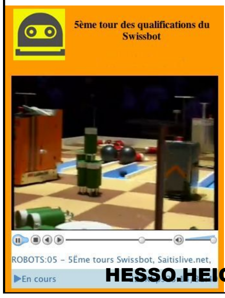
Introduction

The A.M.R. LOMU

The architecture

Method of displacement

Control & programming



- **The mains characteristics of LODUR:**

- A small robot of 10 cm x 10 cm x 15 cm approximately

- IPC - in Beck Module (PC on a chip)

- HEIG-VD FPGA for position encoder reading and PWM output for motor control:

- Improved for (Dude and) Walter

- Piaget-light programming environment

Robots mobiles autonomes

Architecture (1 of 2)

Introduction

The A.M.R. LOMU

The architecture

Method of displacement

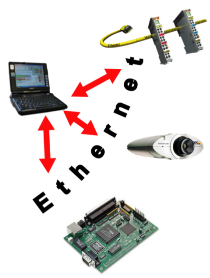
Control & programming

Conclusion

- **The internal communications are mostly based on an Ethernet network:**

The architecture of the robot is as follows:

1. A HUB. which connects physically the various components, with IP protocol
2. A compact notebook computer, which is our decision support centre
3. A specialized motion controller with trajectory interpolation
4. A video camera
5. An Inputs/Outputs Controller

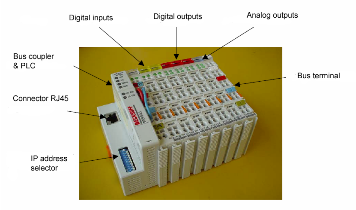


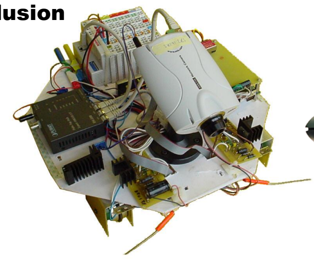
The flexibility of this architecture makes it possible, if necessary, to replace an element by another, to add or remove elements without an advanced study of their integration

Robots mobiles autonomes

Architecture (2 of 2)

- **Different components based on Ethernet network:**

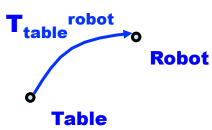


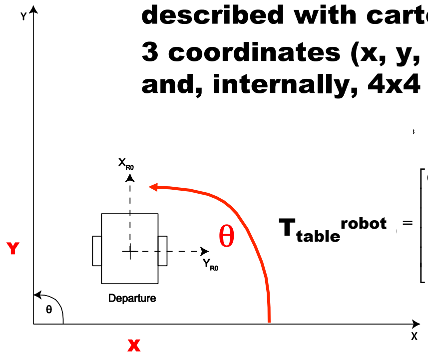


Robots mobiles autonomes

Method of displacement (1 of 5)

- **Position and orientation are described with cartesian frames, 3 coordinates (x, y, θ), graphs, and, internally, 4x4 matrices.**



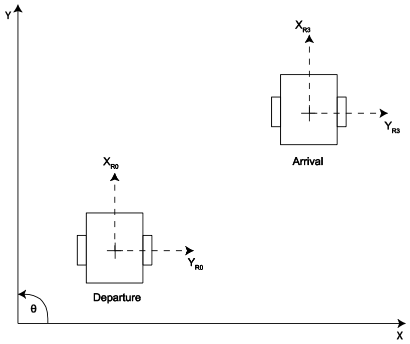


$$T_{table}^{robot} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & X \\ \sin\theta & \cos\theta & 0 & Y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Robots mobiles autonomes

Method of displacement (1b of 5)

- **The Lomu's technique of displacement is like robotics industrial arm:**



This concept allows to limit the complexity of displacement methodology using three virtual joints (θ1;d2;θ3) with the Denavit and Hartenberg's method.

Robots mobiles autonomes

Method of displacement (2 of 5)

Introduction

The A.M.R. LOMU

The architecture

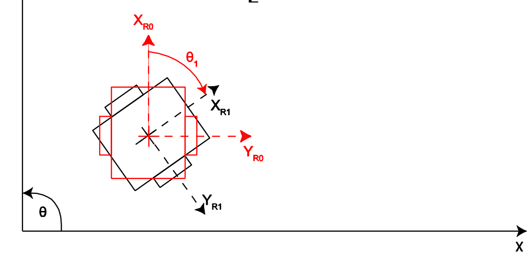
Method of displacement

Control & programming

Conclusion

- **Rotation with θ1:**

$$A_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Robots mobiles autonomes

Method of displacement (3 of 5)

Introduction

The A.M.R. LOMU

The architecture

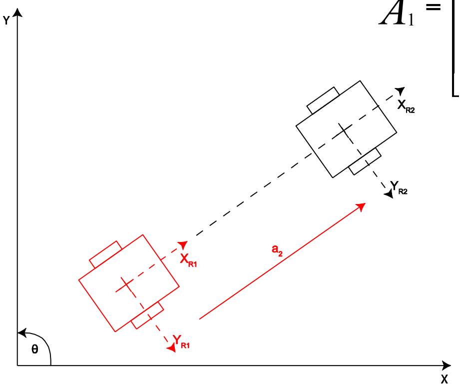
Method of displacement

Control & programming

Conclusion

- **Displacement with d2:**

$$A_1^2 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Robots mobiles autonomes

Method of displacement (4 of 5)

Introduction

The A.M.R. LOMU

The architecture

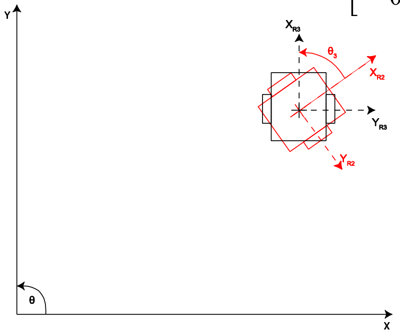
Method of displacement

Control & programming

Conclusion

- **Rotation with θ3:**

$$A_2^3 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Robots mobiles autonomes

Method of displacement (5 of 5)

Introduction

The A.M.R. LOMU

The architecture

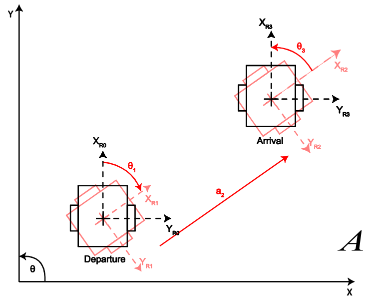
Method of displacement

Control & programming

Conclusion

- **Total displacement of the robot:**

$$A_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$A_1^2 = \begin{bmatrix} 1 & 0 & 0 & a_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^3 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 \\ -\sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The final matrix of robot's position is:

$$A_0^3 = A_0^1 \cdot A_1^2 \cdot A_2^3$$

Robots mobiles autonomes

Control & programming (1 of 5)

Introduction

The A.M.R. LOMU

The architecture

Method of displacement

Control & programming

Conclusion

- **The programming of the robot and the trajectory management are carried out in a context “Piaget” multi-agents real time implemented in C++ language.**

Robots mobiles autonomes

Control & programming (2 of 5)

- Introduction
- The A.M.R. LOMU
- The architecture
- Method of displacement
- **Control & programming**
- Conclusion

```
// PIAGET et quelques instructions de style VAL (V
     +)
location Position1 ;   // exemple : position No 1
int NSIDemarrage=1 ;   // exemple : entrée No 1
int NSOCanon=2 ;       // exemple : sortie No 2
...
// Début du match
50:WaitAGN(NSIDemarrage);//attente d'une entrée
     « vraie »
     break; case
51 : GoState(1001);     break; case
...
// Tir des balles de ping pong
500: SignalOutAGN(NSOCanon, true);//commande d'une
                              //  sortie
                             break; case
...
```


Robots mobiles autonomes

Control & programming (2b of 5)

- Introduction
- The A.M.R. LOMU
- The architecture
- Method of displacement
- **Control & programming**
- Conclusion

```
// Affichage
602: TypeStringAGN(" US Gauche: "); break; case
603: TypeIntegerAGN(DistanceCapteurUSGauche());
                             break; case
604: SleepAGN(1);  // attente d'une seconde
                             break; case
605: ;       // arrêt de la tâche
...
721: SetAGN(Position1,Trans(100,100,-90));
       // on définit une position en x,y cm,
       // avec orientation de phi degrés
                             break; case
722:  MoveAGN(Position1);  // on s'y déplace
...
1001: if(SignalIn(NSIBarriereEntree))//test d'une
     entrée
           GoState(1130);  // Palet ou...
```

Control & programming (2c of 5)

- Structure - The program is composed of various parts dedicated to different agents.


```
while (! DesiredInteraction)   {
Ticks+=1;
Task01();
Task02();                    // Move one step
Task03();                    // Read keyboard
Task04();                    // Perform point to point wheel motion
Task05();                    // Define strategy (typical user
programming context)
Task06();                    // Update Inputs/Outputs
Task07();                    // Display real and simulated status and
current configuration
Task08();                    // Compute inverse kinematics and spatial
motions
Task09();                    // Flash control LED
Task10();                    // Analyze images
Task11();                    // Manage finger reflex
Task12();                    // Manage ball operations (pick, store and
shoot)
Task13();                    // Test inputs
Task18(); }                  // Interpret "Piaget" primitives
```

The various programmed tasks are switched every 100 nanoseconds (sic) on average.

Control & programming (3 of 5)

- **Image processing:**

Image analysis includes various processes:

- Average filtering.
- Median filtering.
- Processing in specific modes.

The specific modes of processing make it possible to define the attributes which are characteristic of a colour to be isolated:

- The red mode to detect the balls.
- The green mode to locate the coconut tree.
- The magenta and blue modes to locate the luminous markers on each side of the opposite goal.

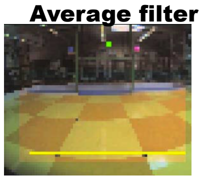
Control & programming (4 of 5)

- **Image processing:**

Example with the mode which locates markers:

Acquisition

Average filter



Detection of magenta zone

Display of the magenta target

Control & programming (4b of 5)

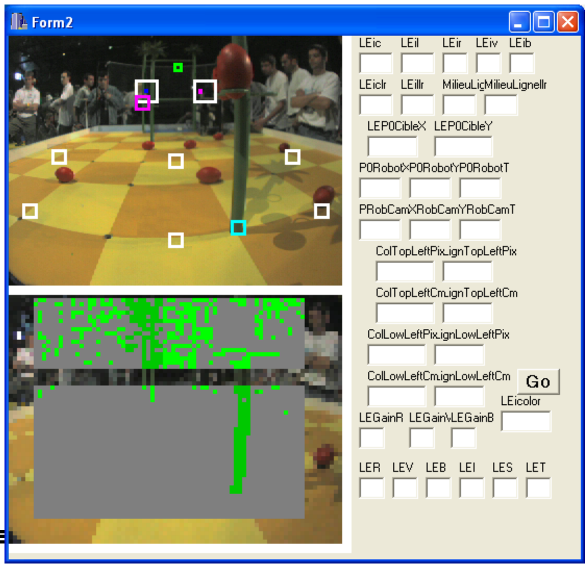


Image processing:

- extrait le vert-brun cocotier
- intègre verticalement
- extrait la zone maximale (vert foncé)
- intègre horizontalement
- filtre avec médian
- localise transition verticale

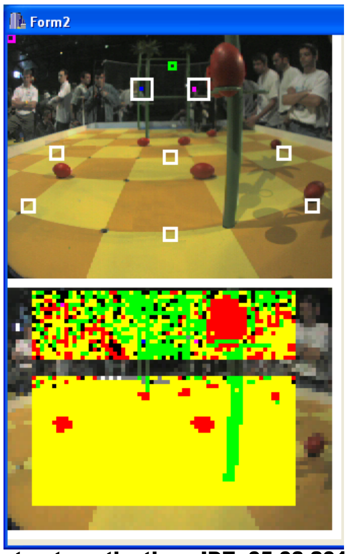
Robots mobiles autonomes

Control & programming (4b of 5)

- Introduction
- The A.M.R. LOMU
- The architecture
- Method of displacement
- **Control & programming**
- Conclusion

Image processing:

- Equilibre des blancs
  - GainsRVB
- Extrait neuf couleurs:
  - Saturation faible
    - Noir, gris ou blanc
  - Saturation forte
    - Rouge, vert, bleu
    - Jaune, cyan, magenta



Robots mobiles autonomes

Control & programming (5 of 5)

- Introduction
- The A.M.R. LOMU
- The architecture
- Method of displacement
- **Control & programming**
- Conclusion

- **Main control panel:**
  - Simulates the displacement of the robot.
  - Provides a visual monitoring of inputs/outputs.
  - allows to adjust various parameters (max speed, acceleration, etc.).
  - Check correct program operations .

- **Image processing panel:**

The control panel of the video camera allows to adjust colour video gains, to define the position of target in the picture, which allows for the correlation between distances on the playing table and relative locations of image pixels

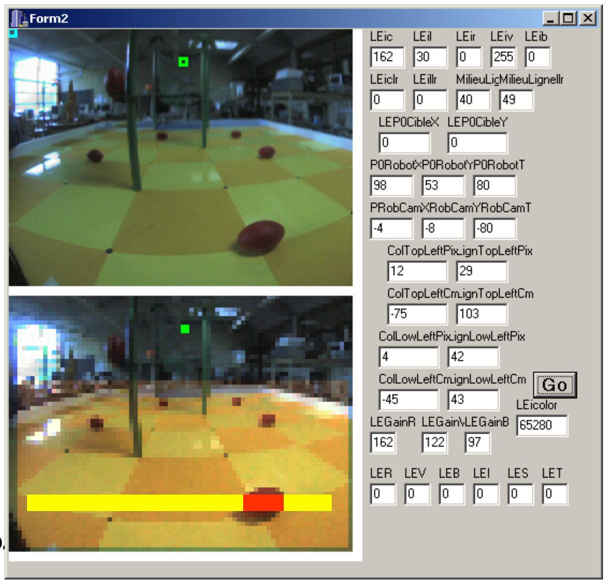
Robots mobiles autonomes

Control & programming (5b of 5)

- Introduction
- The A.M.R. LOMU
- The architecture
- Method of displacement
- **Control & programming**
- Conclusion

```
void ObserverLigneAGN(int
ALNoLigne,int
ALStartColonne, int
ALStopColonne);

void
PasserDeLImageAuTerrain(int
pColonne, int
pLigne,tPoint2DPlus
&pP0Temp);
```



Robots mobiles autonomes

Summary about ARY Robots

- Introduction
- The A.M.R. LOMU
- The architecture
- Method of displacement
- Control & programming
- **Conclusion**

- This particular architecture of the autonomous mobile robots of the Laboratory of Robotics and Automation (LaRA) of the HESSO-HEIG proved its effectiveness for the Swiss and European robotics cups (EUROBOT). The various sensors that “Lomu” uses make it possible to obtain a great safety during displacements, in terms of obstacle avoidance and localization. The structure of programming with “Piaget” makes it possible to carry out modifications quickly, and clearly facilitates programming of strategies.

Robots mobiles autonomes

9 Robots mobiles autonomes

Animats - automates en situation - robots insectes
- Labyrinthe
- Fourmi

Robots intelligents - avec modélisation
- Labyrinthe bis
- Expansion

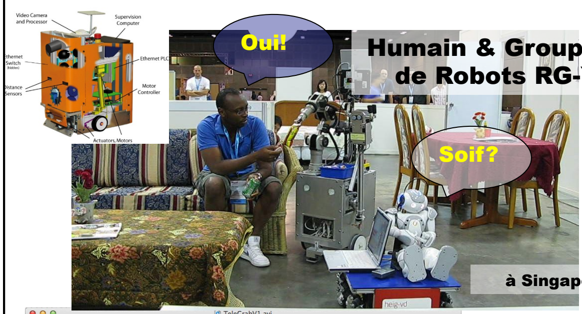
Robots mobiles autonomes robustes
- Golf – Piaget environment

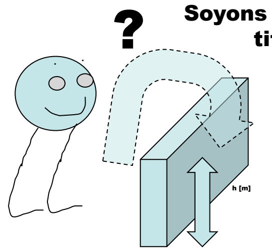
Robots coopératifs
- Robocup-at-Home

Robots mobiles autonomes

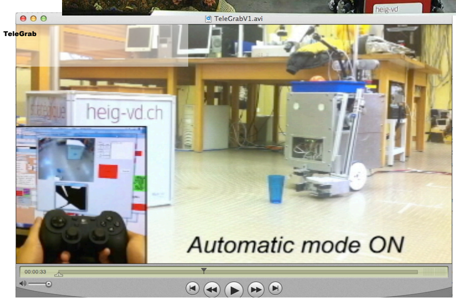
cetT

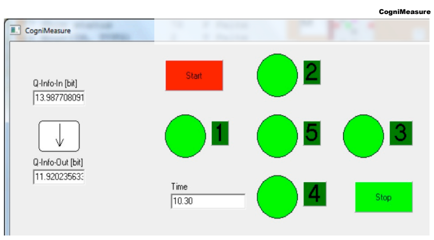
Institut d'Automatisation industrielle LaRA
Laboratoire de Robotique et Automatisation





Soyons quantitatifs!





Robots mobiles autonomes

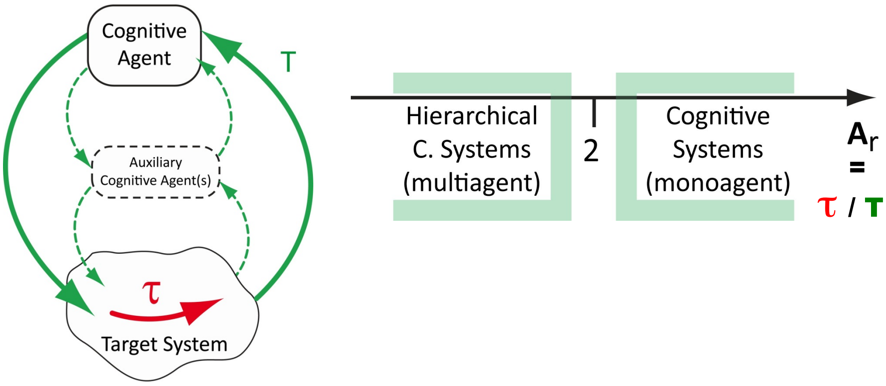
Robots coopératifs

- **Nécessité de communication avec l'homme**
  - **Vision, toucher, voix; gestion d'imprévus et d'éléments peu structurés; cf. Robocup-at-Home**
- **Robots mobiles autonomes robustes**
  - **Intégration de modules éprouvés (moteurs, capteurs, caméras, ordi, PLC, Ethernet, TCP/IP, Windows...)**
- **Environnement Piaget**
  - **Environnement temps-réel, multi-tâches, avec coordination étroite; implémentation en Pascal, C ou C++**
  - **Cf. Exemple RH-Y**

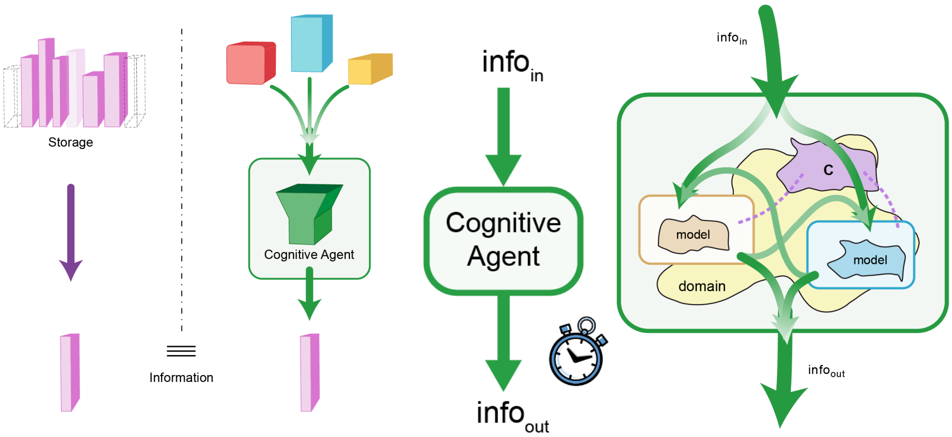
Robots mobiles autonomes

RH-Y (1 de 8)

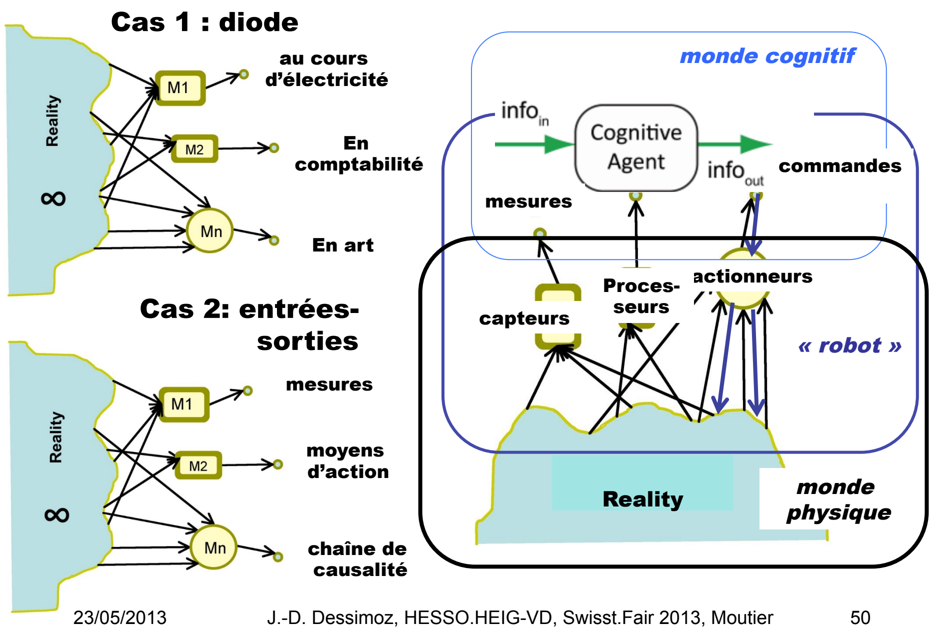
Robots coopératifs – aspects dynamiques



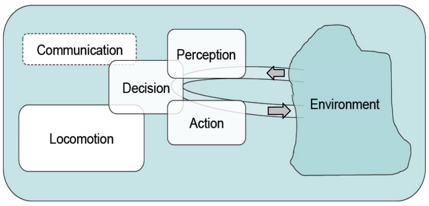
Robots coopératifs – aspects cognitifs



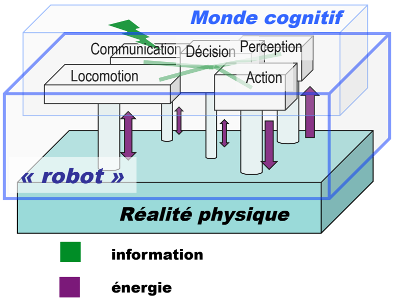
Robots coopératifs – Embodiment

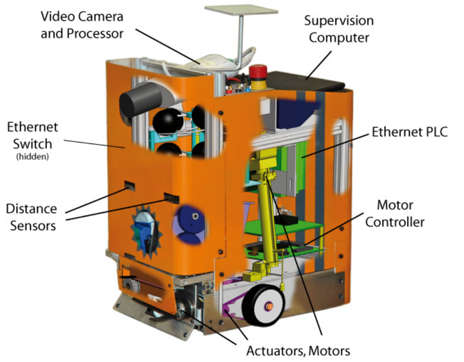


Robots coopératifs – Embodiment




```
11:  SleepAGN(0.05);                                brea
12:  if(!SignalIn(NSIStart))
       GoState(6);
       else
        GoState(20);                                brea
20:  DemarrerMatchAGN();              // start 90 s tin
                                                    brea
21:  SignalOutAGN(NSOAspirateur, true) ;  // start n
                                                    brea
22:  SignalOutAGN(NSORouleauIN, true) ; // start m
                                                    brea
23 : ApproAGN(HoleNb1, 15);                         brea
```

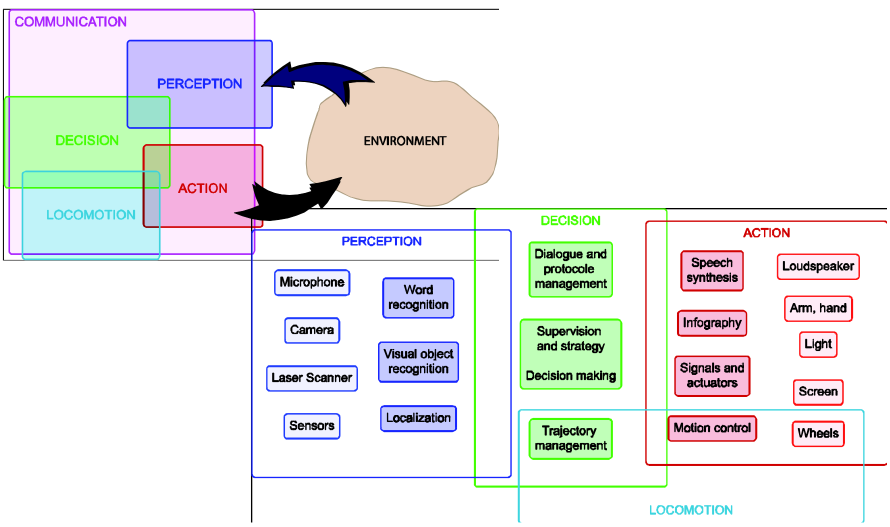


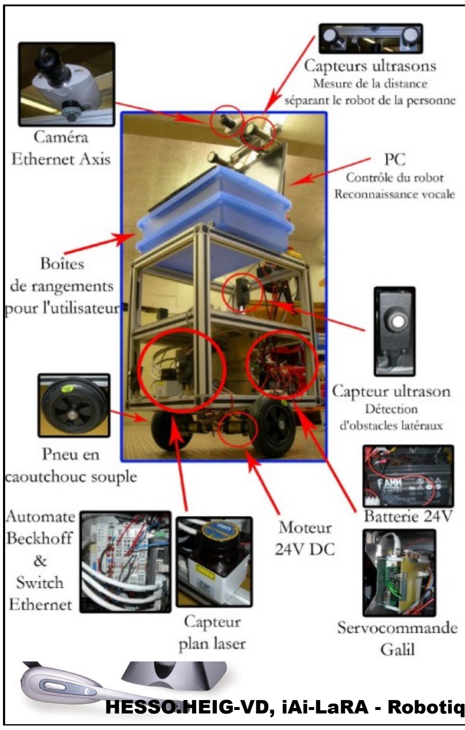


Robots mobiles autonomes

RH-Y (1b de 8)

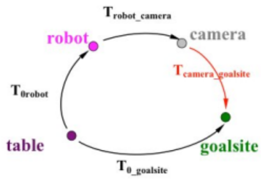
Robots coopératifs - exemple





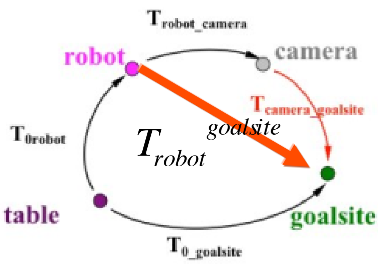
RH-Y (2 de 8)

```
while (! InteractionSouhaitee)
  {
  Ticks+=1;
  // Task01();
  Task02(); // Faire un pas
  Task03(); // Lire clavier
  Task04(); // Mouvements PTP
  Task05(); // Stratégie
  Task06(); // Entrées / Sorties
  Task07(); // Affichage
  Task08(); // Mouvements spatiaux
  Task09(); // Gestion de la diode fonctionnement
  Task10(); // Analyse d'images
  Task11(); // GestionServoCommandesUSB
//  Task13(); // Tester Entree
  Task14(); // Communication
  Task15(); // Mesures plan laser
  Task18(); // Interpréteur Piaget
  Task19(); // Voice dictation
  Task20(); // Dialogue Manager
  Task21(); // Map Manager
  }
InteractionSouhaitee=false;
return;
```



Le graphe des repères et transformation aide à trouver de nouvelles transformations

RH-Y (2b de 8)



Les transformations sont typiquement représentées:

-par des coordonnées (par ex. x,y, phi, dans le cas d'un plan),

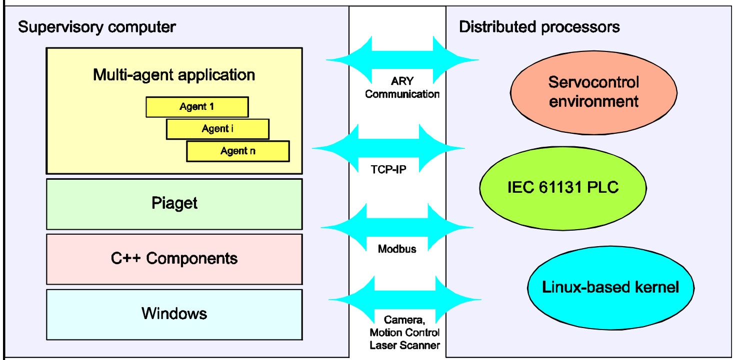
-des matrices 4x4 (par ex. ci-dessous), ou encore

-par des flèches (par ex. ci-contre)

La position par rapport au robot d'une cible observée (goalsite) peut se calculer comme suit:

$$T_{robot}^{goalsite} = T_{robot}^{camera} \cdot T_{camera}^{goalsite}$$

RH-Y (3 de 8)

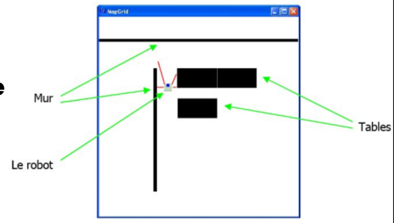


Architecture de commande de notre robot mobile coopératif RH2-Y

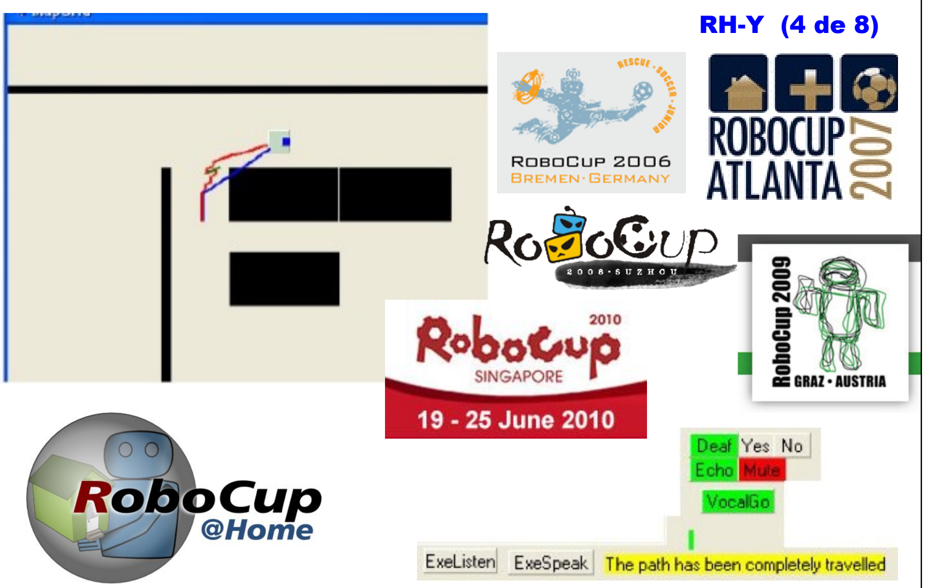
```
12: if(!SignalIn(NSIStart))
       GoState(6);
         else
       GoSTate(20);         break;case

20: ApproAGN(Table,30);   break;ca se
//Switch light on
21: SignalOutAGN(NSOLamp,true)
                          break;case
22: SleepAGN(0.05);       break:case
//Visual analysis of a row in scene
23: WatchRowAGN(R,CStart,Cstop);
```

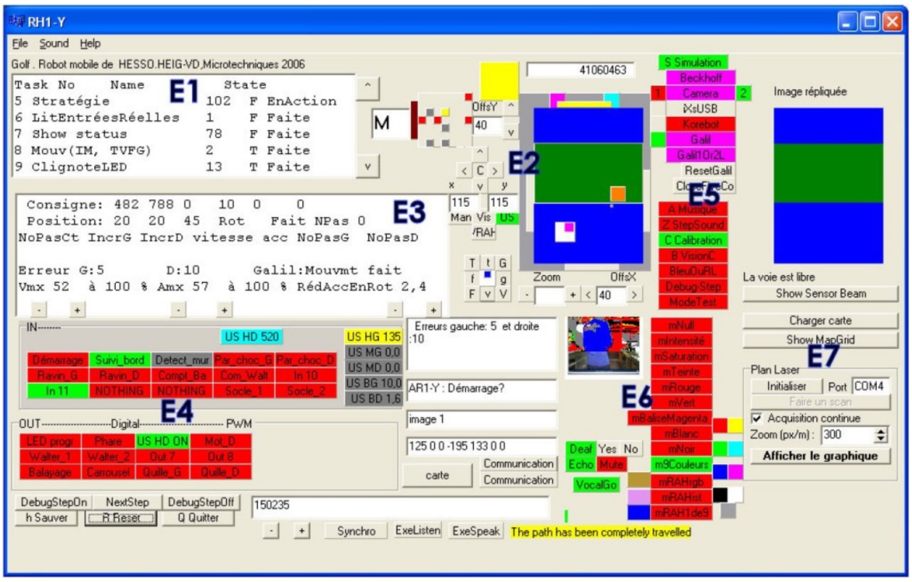
Exemple de code en Piaget



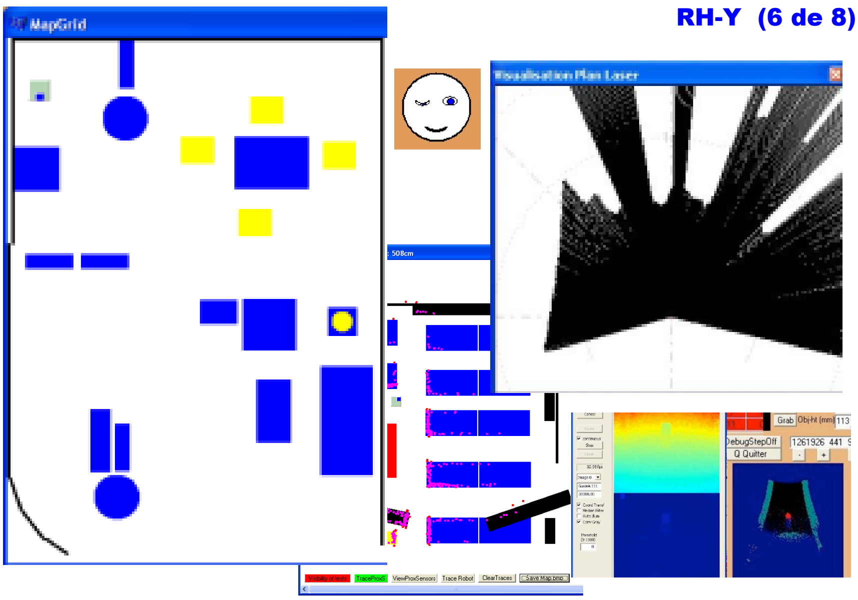
RH-Y (4 de 8)



RH-Y (5 de 8)



RH-Y (6 de 8)



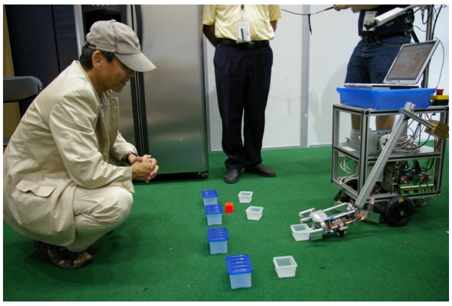
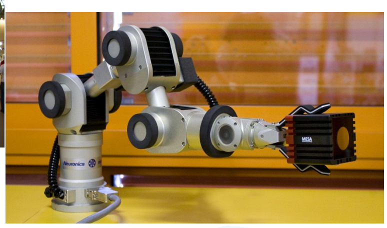


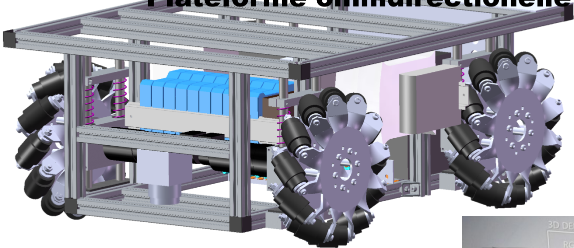
Fig. 1b Tâche « CopyCat » où le robot évolutif RH2-Y, mobile, autonome et coopératif, conçu pour Robocup-at-Home, c'est-à-dire pour l'aide aux tâches domestiques, reconnaît et reproduit les mouvements de l'homme (Atlanta 2007)

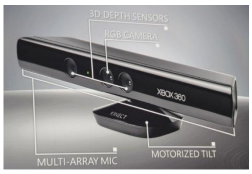
Fig. 1c Tâche « FastFollow » où le robot évolutif RH3-Y apprend les déplacements dans le logement en suivant un humain (Suzhou 2008)



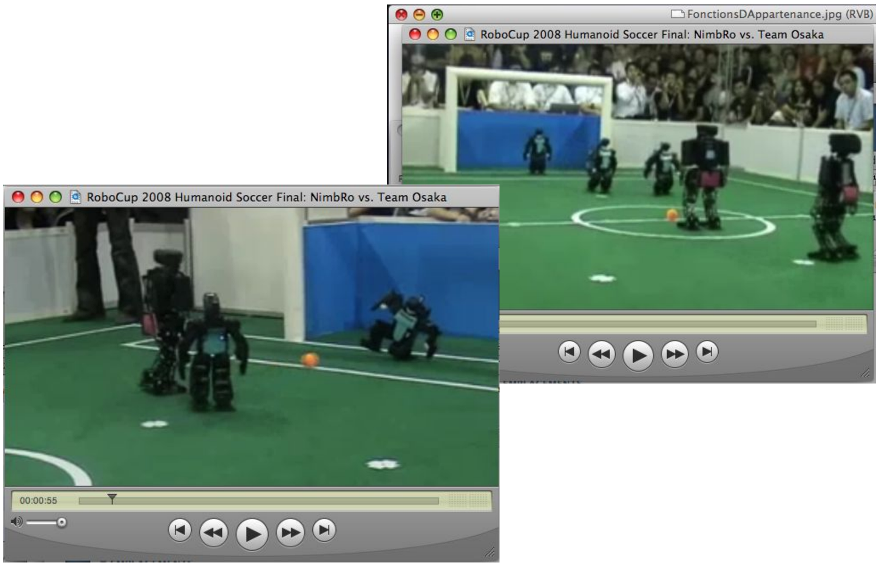
Bras léger intrinsèquement non dangereux (Katana) et ranger SR-4000

Plateforme omnidirectionelle OP-Y





RH3-Y (Suzhou 2008 en haut) **et RH4-Y** (en bas)



Equipes d'humanoïdes - taille enfant (Osaka-J-NimbRo-D)

Plateforme « standard », NAO, d'Aldebaran-F (SoftBank – J)

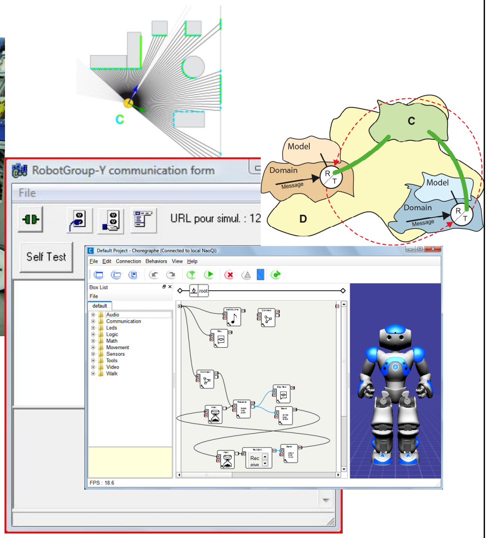
Open Challenge '10 – Safer Follow Mode

(and automated service in department stores)

- **Scientific aspects**

- The task requires :
  - Perception (incl. range data)
  - Locomotion
  - Prehension/manipulation (incl. kinematic aspects)
  - Cognition (control, decision-making, AI)
- As usual for Robocup@Home:
  - Integrated, embedded system, with real world constraints and
  - Very significant cognitic performance levels (K: ca 1MLin, E: ca 100kLin/s)

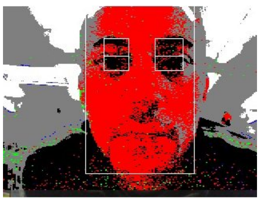
Robot Group, 3-D correlation of 2-D maps, graphic programming and Piaget communication for NAO

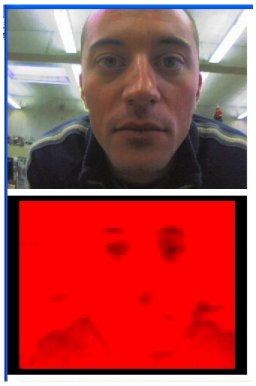


Robot Group (RG-Y) and wifi-based internal cooperation; Nao as a mediator ; Piaget integration; RH with new power drive; tablet PC in the group.

Face recognition and saturation-based, weighted hue and intensity, color correlation

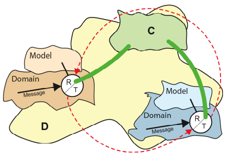
- **Persons now to be found, introduced, and later on recognized and greeted**
- **Re. Facial recognition, biometry (left), W-correlation: 2D « correlation », i.e. integral similarity estimation, based on weighted color differences (re. ISH model and weighted intensity and hue differences as a function of saturation; analog to defuzzification processes: right, for one eye)**





Open Challenge '10 – Humanoids as cultural mediators

1 of 2



- Many people **dream** of creating **artificial humans**, and other ones **fear** that other people achieve that.
- **Machines will stay machines**, and humans, humans

Complements and cyborgs; machines

Humanoids may be extremely useful as cultural mediators

- Ex. 1 : Robot group serving Daniel
- Ex.2 : Parking the car in the garage (concept)

Open Challenge '10 – a Humanoid as a Mediator

2 of 2

- **Scientific aspects**
  - The task requires :
    - Perception (incl. vocal data)
    - Gestures, Locomotion
    - Communication (incl. common culture)
    - Cognition (control, decision-making, AI)
  - As usual for Robocup-at-Home:
    - Integrated, embedded system, with real world constraints and
    - Very significant cognitic performance levels (K: ca 1MLin, E: ca 100kLin/s)
  - Specially effective in Singapore 2010:
    - Command of robot OP-Y and RH-Y by robot NAO-Y (in RIPS etc.)

Nono the humanoid, of Nao type, lower left, mediates the human and the other machines (OP-Y platform where Nono is sitting ; and RH-Y robot that brings drink and snacks)

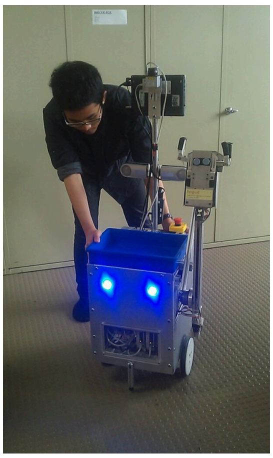
Open Challenge ‘11 – Thermal Vision
1 of 2

- **Scientific aspects**

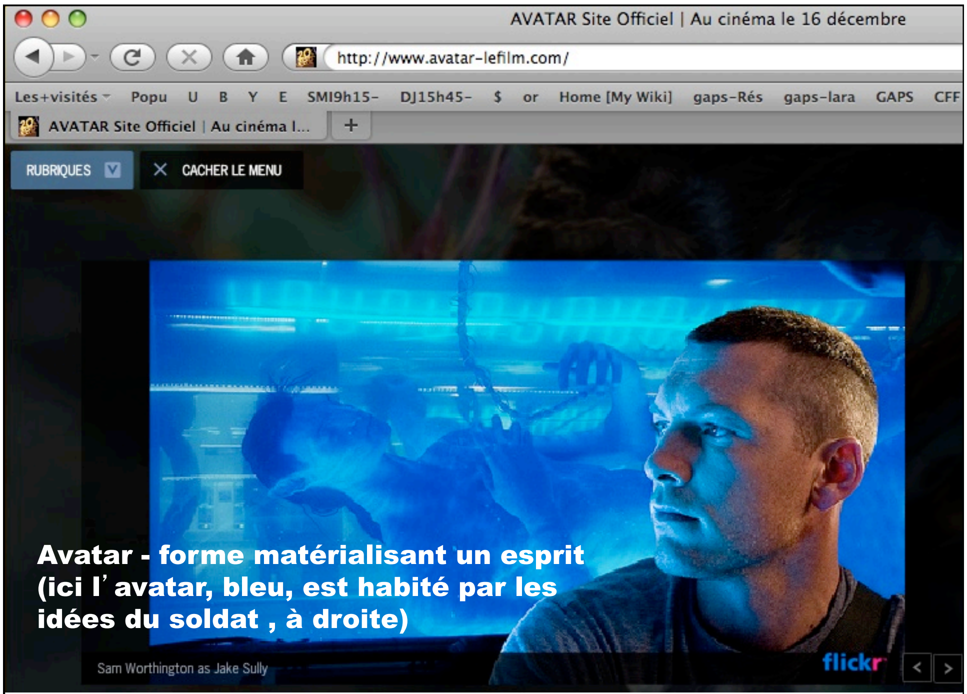
- The task requires :
  - Perception in far infrared domain
  - Extensive spatial and object calibration
- The benefits :
  - Discriminant feature for human recognition
  - Numerous other @Home potential applications
  - Very significant cognitic performance levels (K: ca 200kLin, E: ca 2MLin/s)

Open Challenge ‘11 – Compliant Motion
2 of 2

- **Scientific aspects**



- The task requires :
  - Tightly coordinatel, multi-level control
  - Decoupling of acceleration and inertia related force and torque disturbances
- As usual for Robocup-at-Home:
  - Integrated, embedded system, with real world constraints and
  - Significant cognitic performance levels (K: ca 25 Lin, E: ca 25kLin/s)



HESSO.HEIG-VD, iAi-LaRA - Robotique et automatisation, JDZ, 05.03.2017 71

Robots mobiles autonomes

Conclusion

- Les animats, automates en situation, robots insectes, peuvent faire beaucoup sans modélisation ni même mémorisation
- Pour aller au-delà du présent, c'est-à-dire raisonner avec le passé ou le futur, les robots intelligents ont un besoin absolu de modélisation
- Les robots mobiles autonomes robustes requièrent des architectures puissantes, avec des agents multiples, spécialisés par fonction et performances, avec divers points de vue, généraux ou très spécifiques
- Les robots coopératifs nécessitent en plus de gérer la communication, entre eux ou avec l' homme

HESSO.HEIG-VD, iAi-LaRA - Robotique et automatisation, JDZ, 05.03.2017 72