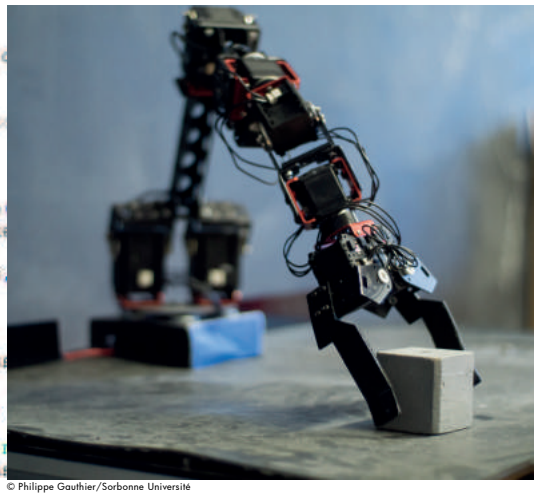
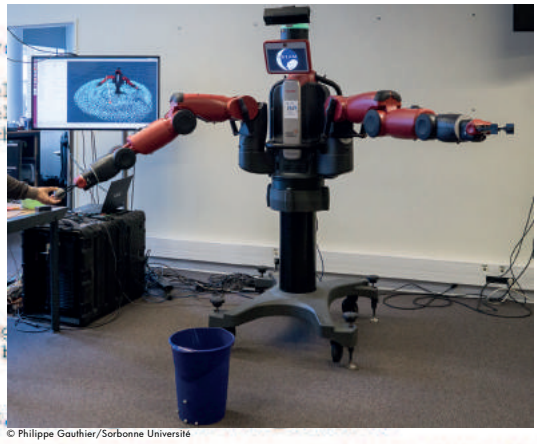
```
#include "image_processing/MotionDetection.h"
#include <pcl/registration/icp.h>
#include <pcl/filters/statistical_outlier_removal.h>
#include <pcl/filters/radius_outlier_removal.h>

using namespace image_processing;

bool MotionDetection::detect(cv::Mat& d
{
    if (_frames.size() != 2) {
        std::cerr << "detect : need ex
        return false;
    }

    cv::Mat current = _frames[1].clone
    cv::Mat previous = _frames[0].clon

    diff = cv::Mat::zeros(current.rows,

    //conversion color to grayscale
    if(previous.channels() == 3)
        cv::cvtColor(previous, previou
    if(current.channels() == 3)
        cv::cvtColor(current, current,

    //gaussian blur to eliminate some
    cv::GaussianBlur(previous, previou
    cv::GaussianBlur(current, current,


    //compute the difference between t
    cv::subtract(current, previous, di


    //binarisation of the difference
    cv::adaptiveThreshold(diff, diff,
    cv::THRESH_BINARY_INV, 1, 3);


    _resultsRects = motion_to_ROIs(di

    //clustering of the bounding boxe

//    rect_clustering(_resultsRects);
    extractResults(_resultsRects);
    return !_resultsRects.empty();
}

void MotionDetection::detect_simple(cv
{
    std::vector<cv::Mat> cf_HSV_channel
    std::vector<cv::Mat> bg_HSV_channel
    std::vector<cv::Mat> motion_mask_c

    cv::Mat current_frame_HSV;
    cv::Mat background_HSV;

    cv::Mat motion_mask;
    cv::Mat motion_mask_S_and_V;

    if (_background_BGR.empty()) {
        current_frame_BGR.copyTo(_backg
        cv::cvtColor(_background_BGR, b
    }
    else {
        cv::cvtColor(current_frame_BGR,

        motion_mask.create(current_frame_HSV.rows, current_frame_HSV.cols, CV_8UC3);

        cv::split(background_HSV, bg_HSV_channels);
        cv::split(current_frame_HSV, cf_HSV_channels);
        cv::split(motion_mask, motion_mask_channels);
```



© Philippe Gauthier/Sorbonne Université

Des robots qui apprennent



© Philippe Gauthier/Sorbonne Université

#Des robots qui apprennent

Aujourd'hui, il existe des robots dotés de capacités surpassant celles des humains pour certaines activités. Ces super-spécialistes ne peuvent toutefois faire face à une situation imprévue. Comment donner aux robots la capacité de s'adapter à des environnements variés et inconnus de leurs concepteurs ?

C'est l'enjeu des recherches de la robotique dite développementale, une branche de l'intelligence artificielle initiée à la fin des années 1990.

Face à un environnement inconnu, un robot pourrait comprendre tout seul ce qu'il peut faire, comment le faire, et accomplir sur demande des tâches domestiques. © ISIR

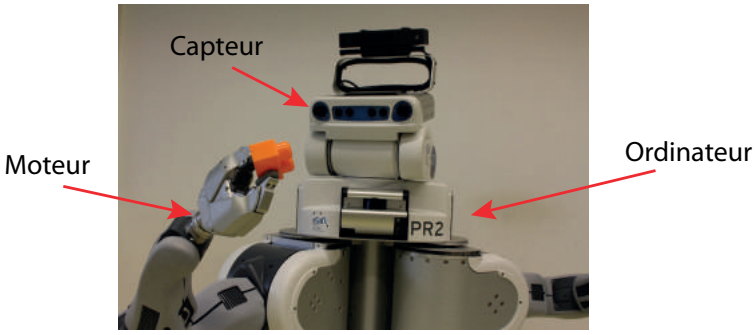
Tels de jeunes enfants qui apprennent à partir de leurs explorations, ces nouveaux robots doivent apprendre à interagir de façon appropriée avec les objets ou les humains qui les entourent, et de même que ces derniers, structurer et ancrer leurs connaissances en rêvant !

DREAM
ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE
SORBONNE UNIVERSITÉ CRÉATEURS DE FUTURS DEPUIS 1257
HORIZON 2020
cnrs
UN CHERCHEUR, UNE MANIP

Des robots et des apprentissages

Un robot est une machine programmée pour agir au profit des humains. Il est composé d'un ordinateur connecté à des capteurs et des moteurs.

Selon son programme, le robot peut répéter indéfiniment la même action, ou bien explorer son environnement dans le but de trouver de lui-même les actions à réaliser pour effectuer une tâche.



Les capteurs sondent l'environnement du robot, les moteurs permettent de le modifier. Un robot sans capteurs ni moteurs n'est pas connecté au monde réel. © Stéphane Doncieux /ISIR

#Apprentissage supervisé

{

Le concepteur du robot attend un comportement précis du robot : attraper un objet, reconnaître la voix d'une personne etc. Il fournit au robot un grand nombre d'exemples ou le renseigne sur ses erreurs (différences entre le comportement attendu et le comportement que le robot a réalisé).

(Inconvénient : le robot ne peut apprendre seul, sans l'aide d'un superviseur). }

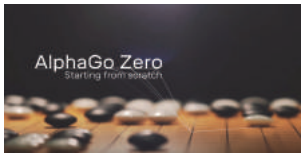
#Apprentissage par renforcement

{

Le concepteur du robot n'attend pas un comportement précis mais identifie un but, associé à une valeur numérique fixe : la récompense. En cherchant à maximiser la récompense que lui procurent ses actions, le robot finit par trouver celles qui sont les plus efficaces.

(Inconvénient : les actions possibles étant connues à priori et peu nombreuses, ce type d'apprentissage est insuffisant dans des environnements variés (environnements ouverts)). }

Des « robots » qui n'en sont pas : les machines sans capteurs ni moteurs.



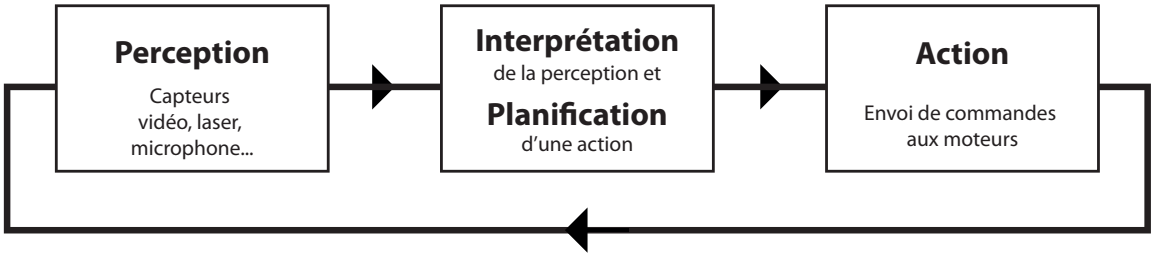
© DeepMind

Les machines qui battent les joueurs d'échecs ou de go sont dotées d'un d'algorithme d'intelligence artificielle qui leur permet d'apprendre à partir de nombreux exemples ou en affrontant des joueurs artificiels. Elles ne jouent qu'aux échecs ou au Go et doivent être reprogrammées pour jouer à d'autres jeux.

Un robot, ça se programme

Un robot doit réagir pour éviter des obstacles, planifier des actions pour atteindre un but, apprendre... tout cela simultanément. Chacune de ces fonctions « cognitives » est prise en charge par un programme particulier qui s'assemble aux autres dans une « architecture cognitive ».

#L'architecture cognitive « sense, plan, act »

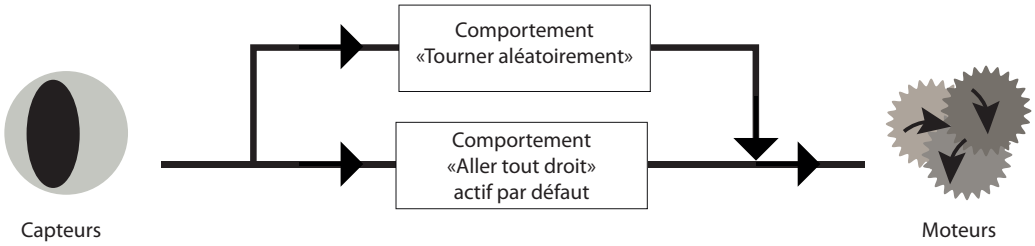


{

En sondant l'environnement, les capteurs permettent au programme de planifier des actions exécutées par les moteurs. Cette boucle se répète tant que le robot fonctionne. Elle inclut une représentation figée de l'environnement qui limite les capacités d'apprentissage du robot.

}

#L'architecture cognitive « orientée comportement »



Si (pas d'obstacle) alors {aller tout droit;}
sinon {tourner d'un angle aléatoire;}

{

Architecture novatrice des années 1990, elle utilise un assemblage de comportements simples (deux au minimum) activés par des signaux issus des capteurs. Cette architecture ne convient pas à des comportements complexes qui nécessitent une planification.

}

Un robot aspirateur utilise plusieurs comportements. © Roomba

La représentation du monde chez un robot

Comment programmer un robot pour qu'il avance en suivant une ligne noire sur fond blanc ?

#Le concepteur du robot commence par fixer une représentation de l'environnement adaptée à la tâche : le « monde » du robot est en noir et blanc.

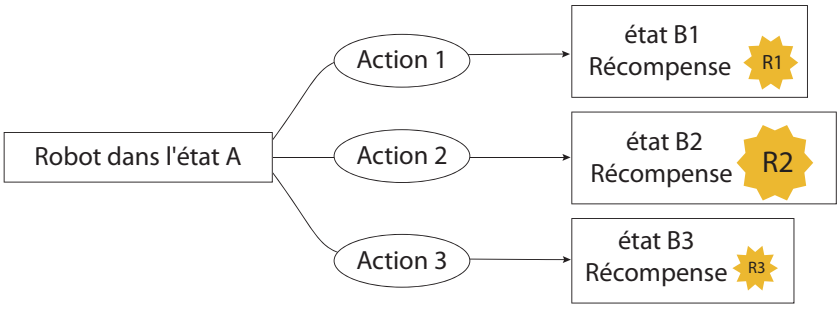
#Régulièrement, les capteurs adaptés à la détection de contraste lumineux renseignent le programme sur la position des zones noires. Le programme peut alors commander les moteurs de façon à garder le robot sur la ligne noire.

Avec une représentation du monde figée (ici en noir et blanc), le robot ne peut pas suivre une ligne rouge ou servir un café à un humain !

L'apprentissage par renforcement

Doté d'un algorithme d'apprentissage par renforcement, un robot explore son environnement et apprend comment agir pour obtenir une récompense maximale. Les récompenses sont associées à un état ou à une action.

#Une récompense après chaque action

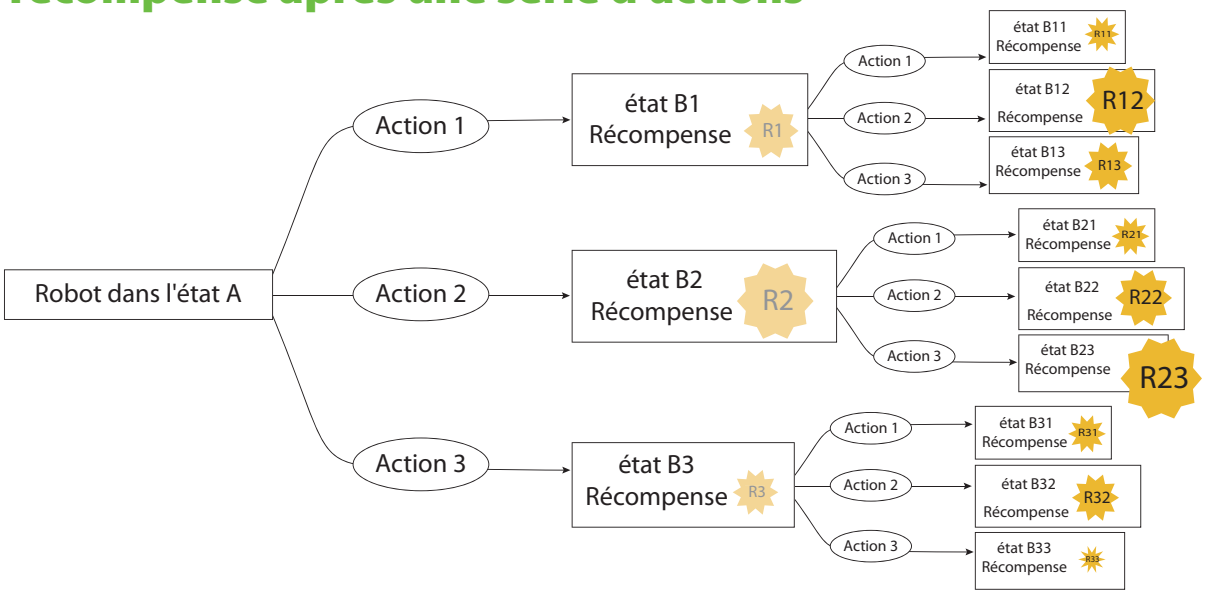


{

Le robot passe d'un état à un autre en effectuant une action.
Au cours de son exploration, il apprend à connaître la relation entre l'état courant, une action, l'état suivant et la récompense associée. Le concepteur du robot fixe les récompenses dans un tableau ou par l'intermédiaire d'un programme de calcul.

}

#Une récompense après une série d'actions



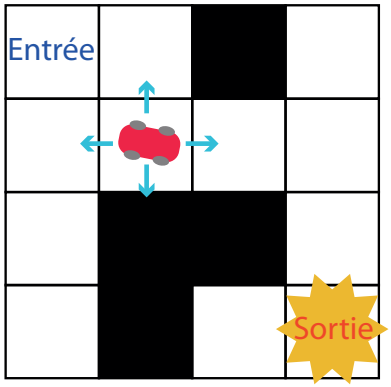
{

Le robot génère un comportement complet composé d'une suite d'actions sur l'environnement et observe la récompense finale qu'il obtient (R23 pour la suite d'actions 2 et 3 par exemple).

}

Les concepteurs de robots recourent également à des principes issus de la sélection naturelle darwinienne.
Des comportements complets, déterminant quelles actions choisir selon les états, sont générés aléatoirement puis sélectionnés selon la récompense qu'ils procurent. Ils sont ensuite mutés et croisés entre eux afin d'en produire d'autres, évalués à leur tour.

Un robot peut-il apprendre à sortir d'un labyrinthe ?



Le robot a les moyens de savoir sur quelle case il se trouve.

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	1

Valeurs de récompense affectées à chaque case, fixées initialement par le concepteur du robot.

0	0,4	0	0,5
0,4	0,5	0,6	0,8
0,3	0	0	0,9
0,2	0	0,5	1

Valeurs de récompense après la phase d'exploration du labyrinthe par le robot.

Les valeurs de récompense croissantes repèrent le chemin à prendre.

#Les valeurs de récompense initiales affectées à chaque case sont fournies par le concepteur du robot.
Au cours de son exploration du labyrinthe, le robot les réajuste selon un principe de rétropropagation de la récompense : si au hasard de son exploration, les actions du robot (avancer tout droit, tourner à droite ou à gauche, reculer) le mènent à la case de sortie, le programme du robot affectera à la case précédente une valeur de récompense légèrement inférieure à 1, et ainsi de suite jusqu'à la case d'entrée. Les cases noires, inaccessibles, gardent une récompense nulle.

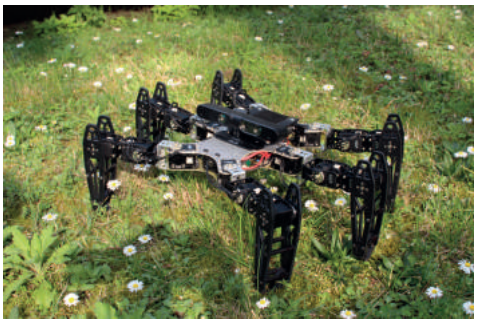
#A la fin de cette phase exploratoire, l'apprentissage du robot est terminé. Il est capable de sortir rapidement du labyrinthe à partir de n'importe quelle case.

L'approche évolutionniste

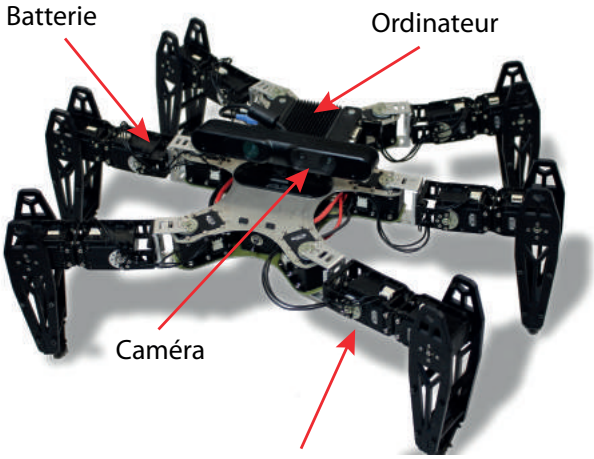
L'approche évolutionniste s'est développée en optimisation à partir des années 1960. Appliquée à la robotique depuis 1990, elle bénéficie aujourd'hui de la grande puissance de calcul des ordinateurs. Les algorithmes d'intelligence artificielle évolutionnistes croisent et sélectionnent un grand nombre de comportements adaptés. Ils permettent ainsi aux robots de faire face à une large variété de situations, y compris des pannes.

#L'hexapode et ses six pattes

L'hexapode est un robot capable d'apprendre à marcher tout seul. Lorsqu'une ou plusieurs de ses pattes sont en panne, ses apprentissages antérieurs lui permettent de trouver une autre façon de marcher en quelques essais.



© Antoine Cully/Jean-Baptiste Mouret Sorbonne Université



{

En explorant son environnement réel et simulé, l'hexapode trouve différentes façons d'avancer, sur une, deux, trois.... ou six pattes. Chaque stratégie de marche est évaluée en fonction de la récompense dispensée, d'autant plus grande que la marche s'effectue rapidement et en ligne droite. Des dizaines de milliers de stratégies sont mémorisées dans un « répertoire de comportements ».

}

#L'hexapode s'adapte à une panne

Sans savoir quelle patte ne fonctionne plus, le robot trouve une autre façon de marcher en une quarantaine de secondes !

{

En cas de panne, le robot puise des comportements dans son répertoire et sélectionne à l'aide d'une méthode d'optimisation 5 à 10 comportements de marche à tester. Le robot ne retient que celui qui procure la récompense la plus grande pour aller se faire réparer à sa base.

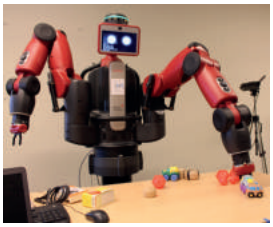
}

L'approche développementale

Les approches évolutionnistes permettent aux robots d'apprendre dans des environnements variés mais elles sont lentes. Des approches plus rapides limitent les actions du robot et nécessitent une connaissance étendue de son environnement. L'approche « développementale » qui procède par étapes comme le font les enfants, peut s'appuyer sur une approche évolutionniste pour déterminer des actions appropriées utilisées ensuite par des algorithmes rapides.

#1- Babillages

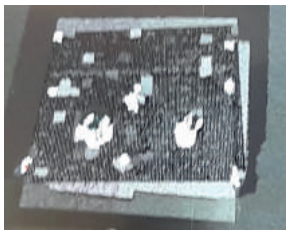
Le robot apprend lentement, avec peu de connaissances initiales sur son environnement.
Peu à peu, il construit des cartes d'informations.



© ISIR

Exploration

Le robot interagit avec son environnement afin de récolter des informations sur la mobilité des objets.



© Marielle Vergès/EPPDCSI

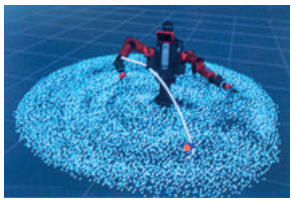
#1-1 Carte de saillance

construite par le robot : en blanc, les objets mobiles (probabilité de mobilité = 100%);
en noir les zones indéplaçables (probabilité de mobilité = 0%).

© Philippe Gauthier/Sorbonne Université

Simulation

La simulation préserve le robot.
En une vingtaine d'heures seulement, le robot effectue 400 000 simulations qui, pour des lancers de balle par exemple, prendraient 60 fois plus de temps dans la réalité et solliciteraient à outrance la mécanique du robot.



© Philippe Gauthier/Sorbonne Université

#1-2 Répertoire de comportements

La balle (en rouge) suit une trajectoire simulée (en blanc).
Les points d'impacts des différents lancés sont représentés en bleu clair.

#2- Exploitation

Le robot utilise l'expérience acquise dans l'étape de babillage afin de produire des connaissances qu'il exploite ensuite pour apprendre plus efficacement.

Le robot essaye les lancers qu'il a appris dans l'étape du babillage.
Si la balle tombe à côté, il recalcule une trajectoire plus adaptée.
Il relance ainsi la balle jusqu'à ce qu'elle tombe dans le panier.

© Philippe Gauthier/Sorbonne Université

#3- Consolidation et adaptation

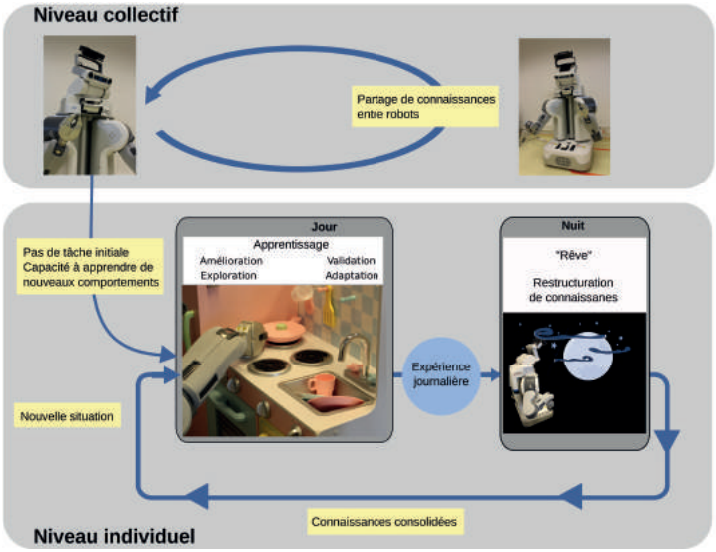
Le robot consolide ses connaissances et les applique dans d'autres situations. Face à une situation imprévue, il ne réapprend pas de « zéro ». Ici, le robot arriverait en quelques essais à lancer la balle dans un panier placé en hauteur.

Le projet DREAM

#DREAM (rêve) est un projet de recherche européen en robotique développementale.
C'est un projet interdisciplinaire entre robotique et neurosciences.

#Il vise à concevoir des algorithmes permettant à des robots de faire face à des environnements ouverts. Ces algorithmes alternent des phases d'interaction avec l'environnement et des phases d'analyse ou de restructuration semblables à celles de notre sommeil.

#Le projet s'intéresse également à l'étude des processus cognitifs qui restent encore incompris chez les êtres vivants. Ainsi, les algorithmes développés en robotique dans le cadre de ce projet pourraient améliorer les connaissances en neurosciences et inversement les études sur le vivant pourraient contribuer à améliorer les algorithmes de robotique.



#Les partenaires : Sorbonne Université (France), Armines (France),
Universidade da Coruna (Espagne), VU/VUmc (Pays-Bas) et University of Edinburgh (Royaume-Uni).

L'ISIR

Institut des Systèmes Intelligents et de Robotique

// Domaines de recherche

{

#Modélisation, conception, commande de robots et systèmes robotiques (mouvement, manipulation)

#Capteurs, perception, interfaces

#Cognition et apprentissage

#Interaction humain-machine physique et cognitive

}

// Domaines d'application

{

#Santé, chirurgie, assistance, réhabilitation

#Robotique de service

#Robotique d'intervention

#Micro-technologies

}

Le robot PR2 apprend à manipuler des objets.
© Philippe Gauthier/Sorbonne Université

L'Institut des Systèmes Intelligents et de Robotique (ISIR) est une unité mixte de recherche (UMR 7222) rattachée à la faculté d'Ingénierie de Sorbonne Université (UFR 919) et à l'Institut des Sciences de l'Information et de leurs Interactions (INS2I) du Centre National de la Recherche Scientifique (CNRS). L'Institut National de la Santé et de la Recherche Médicale (INSERM) est également tutelle de l'une de ses équipes, l'équipe de recherche labellisée U1150.

DREAM

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

SORBONNE UNIVERSITÉ CRÉATEURS DE FUTURS DEPUIS 1257

HORIZON 2020

cnrs

UN CHERCHEUR, UNE MANIP