



Contributions à l'apprentissage automatique de réseau de contraintes et à la constitution automatique de comportements sensorimoteurs en robotique.

Mathias PAULIN

LIRMM (CNRS, Univ. Montpellier II)
161 rue Ada 34392 Montpellier Cedex 5

Le 3 juillet 2008

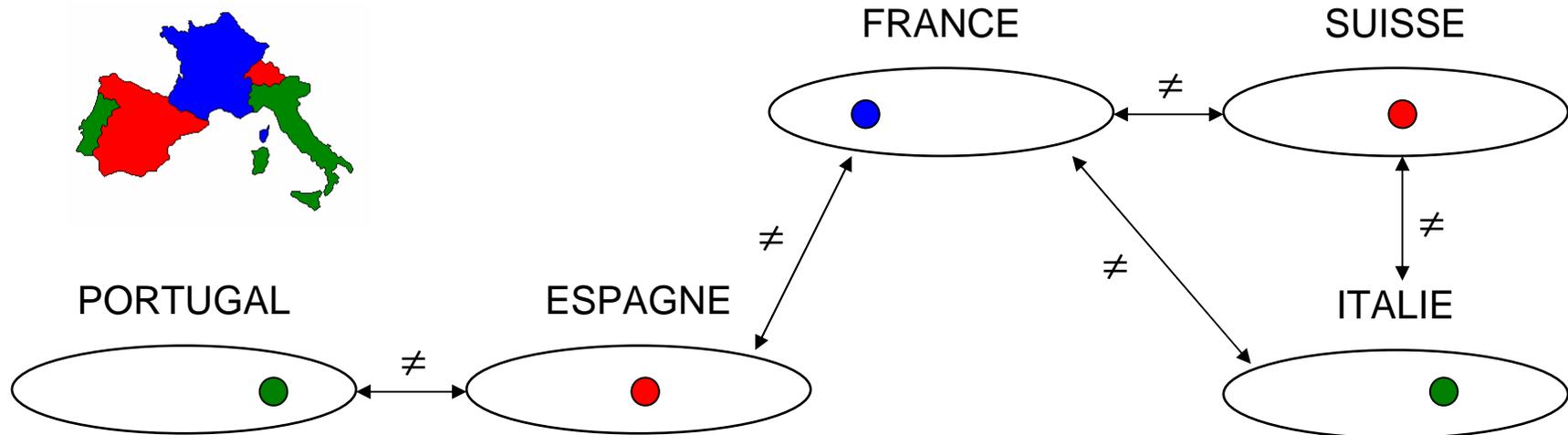
Plan de l'exposé

- Contexte
- Une approche PPC pour la constitution automatique de comportements sensorimoteurs
- Expérimentations
- Conclusion & perspectives
- Travaux connexes

Contexte

□ Un réseau de contraintes :

- ↪ Les variables X
- ↪ Les domaines D
- ↪ Les contraintes C



□ Solution = Toutes les contraintes sont satisfaites

La Programmation Par Contraintes (PPC)

□ Un paradigme déclaratif :

- Le modèle,
- Les techniques de résolution.

□ Un succès :

- Configuration de produits,
- Allocation de ressources,
- Ordonnancement,
- ...

□ Une diffusion freinée :

- Manque d'experts pour modéliser des problèmes réels,
- Modéliser efficacement un problème.

La plate-forme CONACQ

⇒ Acquisition automatique de réseau de contraintes

Utilisateur / Novice



fournit

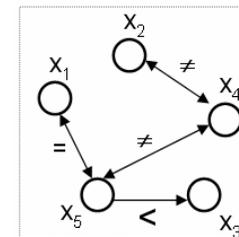
(X, D)
Biais d'apprentissage

$E = E^+ \cup E^-$
Solutions
Non solutions

CONACQ

Agent apprenant

induit



Réseau de contraintes



Apprentissage de Contraintes

Rémi Coletta, Thèse de doctorat de l'Université Montpellier II, Juillet 2006.



Une approche PPC pour la constitution automatique de comportements sensorimoteurs

Le défi de l'autonomie en robotique...

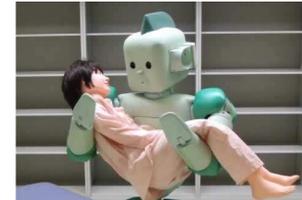
1960 - 2000



Aujourd'hui



Demain...



⇒ Une complexité mécanique croissante,

⇒ Des missions de plus en plus complexes,

⇒ Des robots de plus en plus **autonomes**

⇒ machines polyvalentes « intelligentes » capables de réaliser des missions complexes dans des environnements très variés.

Architectures logicielles de contrôle

□ Architectures délibératives

Nilson (1980)

- ↳ Centrées sur le système décisionnel (planification haut niveau)
- ↳ Manquent de réactivité, s'adaptent mal aux modifications de contexte.

□ Architectures comportementales

Brooks (1986)

- ↳ Centrées sur la réactivité du système,
- ↳ Permettent difficilement la réalisation de missions complexes.

□ Architectures hybrides

LAAS (1998), Claraty (2001), Harpic (2003)

- ↳ Cherchent à combiner les aspects décisionnels et réactifs,
- ↳ Un panel de comportements (modules temps-réel),
- ↳ Un mécanisme de choix du ou des comportements.

Comportements sensorimoteurs

Actions élémentaires

□ Comportements sensorimoteurs :

- ↳ Encapsulent des boucles de contrôle sur les actionneurs,
- ↳ Sont considérées comme les **capacités de base** du robot,
- ↳ Ils correspondent à la combinaison d'**actions élémentaires**.

□ Actions élémentaires :

- ↳ Ensemble minimal de commandes aux actionneurs,
- ↳ Commandes câblées, exécutables automatiquement,
- ↳ Modélisation par systèmes d'équations des lois physiques.

⇒ Difficultés : 1. **Combiner** et **enchaîner** plusieurs actions élémentaires.
2. Définir des comportements **robustes** et **adaptables**.

Approche proposée...

1. **Abstraire** les actions élémentaires sous la forme de **réseaux de contraintes**,
2. **Combiner** les réseaux à l'aide d'outils de **planification**,
3. **Contrôler** l'exécution d'un comportement sensorimoteur, en corrigeant les écarts existants entre les prévisions et la réalité.

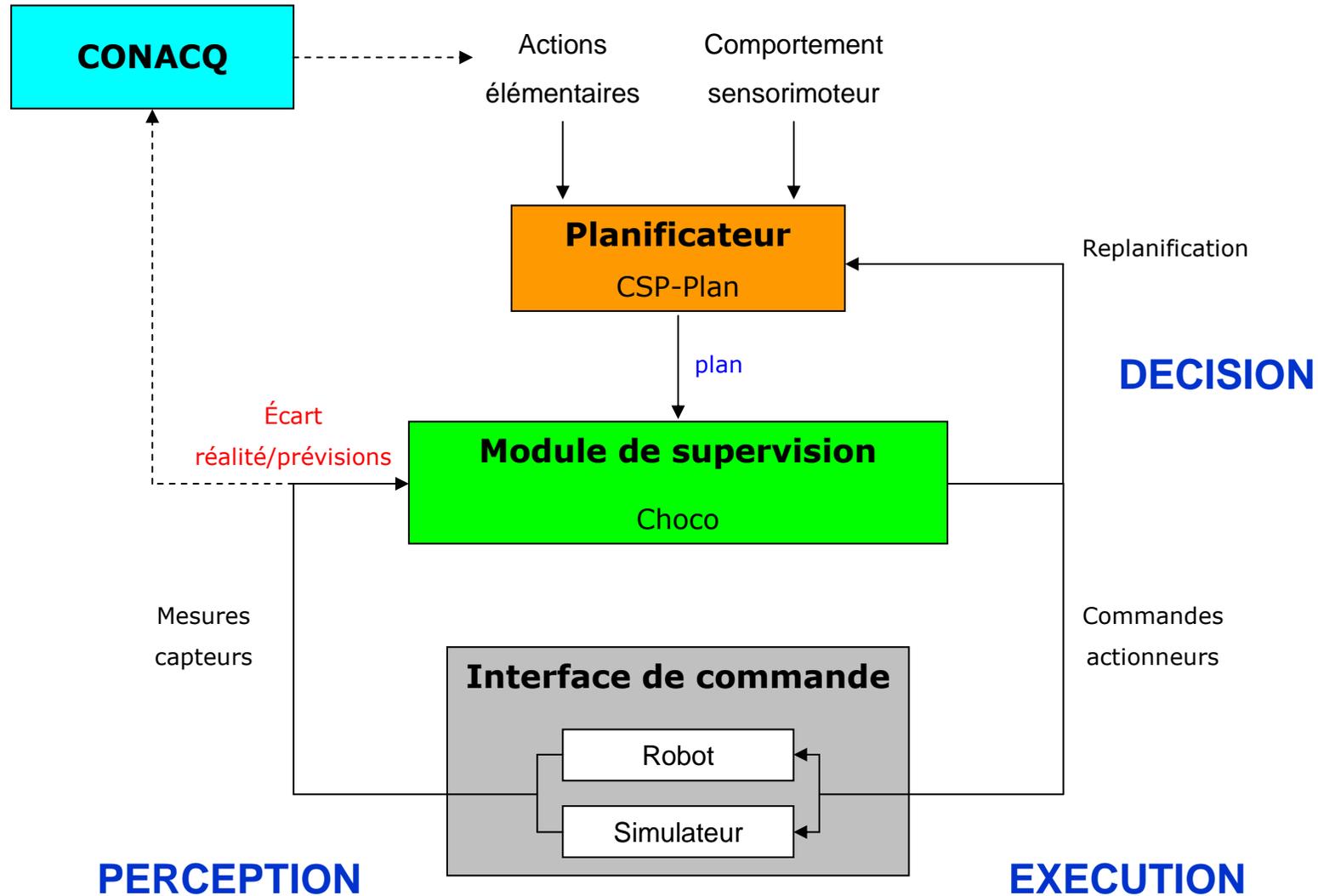
↳ Modélisation d'un **espace de solutions** ⇒ Planification plus aisée,

↳ Vocabulaire restreint, flexible et robuste ⇒ **Réactivité & adaptabilité**,

Modèle de **raisonnement** sur les aptitudes (état futur, correction, échec)

⇒ Un premier pas vers une **autonomie** comportementale non réflexe.

Architecture de supervision (Paulin 2008)



Une approche PPC pour la constitution de comportements sensorimoteurs...

1. **Abstraire** les actions élémentaires sous la forme de **réseaux de contraintes** → [Conacq](#)
2. **Combiner** les réseaux à l'aide d'outils de **planification** → [CSP-Plan](#)
3. **Contrôler** l'exécution d'un comportement sensorimoteur → [Propagation](#)

⇒ **Constitution automatique de comportements sensorimoteurs par modélisation automatique des actions élémentaires.**

Modélisation CSP des actions élémentaires

- Chaque action élémentaire a_i est modélisée par un réseau de contraintes :

{
Pre(a_i) : les conditions d'exécution de a_i ,
Actuators(a_i) : le comportement des actionneurs,
Post(a_i) : les effets de a_i .

↪ STRIPS + composante Actuators

↪ Actions élémentaires = Briques de base

Conacq : Acquisition automatique des A.E.

□ Données d'entraînement :

- ↳ Tuples étiquetés : $(\overbrace{\delta_1^I, \dots, \delta_p^I}^{\text{Descripteurs état initial}}, \overbrace{\alpha_1, \dots, \alpha_q}^{\text{Actionneurs}}, \overbrace{\delta_1^F, \dots, \delta_p^F}^{\text{Descripteurs état final}})$
- ↳ Exécutions réelles ou simulations
- ↳ Pré-traitement : Discrétisation des variables

□ Objectif :

- ⇒ $\text{Pre}(a_i) = \text{contraintes sur } \{\delta_1^I, \dots, \delta_p^I\}$
- ⇒ $\text{Post}(a_i) = \text{contraintes sur } \{\delta_1^I, \dots, \delta_p^I, \delta_1^F, \dots, \delta_p^F\}$
- ⇒ $\text{Actuators}(a_i) = \text{contraintes sur } \{\delta_1^I, \dots, \delta_p^I, \alpha_1, \dots, \alpha_q, \delta_1^F, \dots, \delta_p^F\}$

□ Biais d'apprentissage : Librairie arithmétique

Planification et supervision d'un comportement sensorimoteur

□ Planification :

- ↳ CSP-Plan (STRIPS) \Rightarrow Pré & post-conditions uniquement,
- ↳ Séquence respectant les conditions d'enchaînement des A.E.

□ Module de supervision :

- ↳ Instanciation d'un comportement sensorimoteur,
- ↳ Ajout de la composante Actuators.

□ Cycle de supervision :

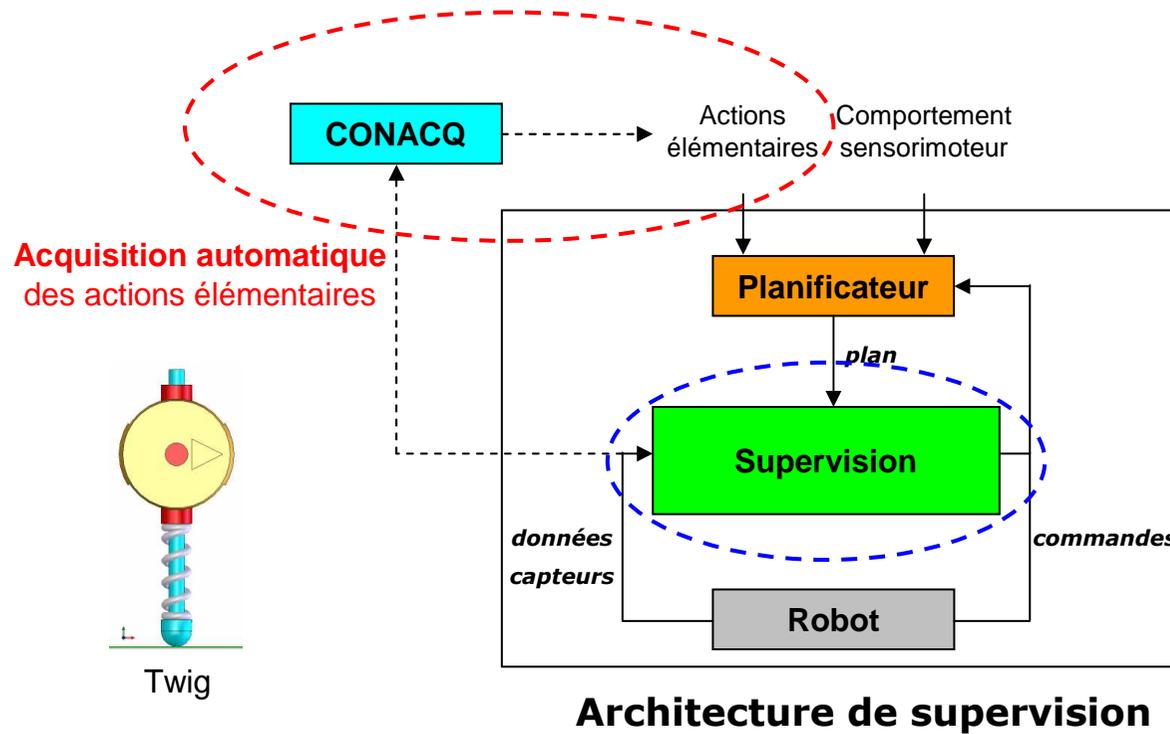
- ↳ Exécution identique aux prévisions,
- ↳ Écart mineur \Rightarrow **Corrections** mineures par **propagation**,
- ↳ Écart non « rattrapable » \Rightarrow **Replanification**.



Expérimentations

Expérimentations préliminaires

⇒ Lever deux « verrous » techniques



Les performances du résolveur de contraintes Choco

PIRI

Loi de commande bas niveau

Fréquence = 100 Hz

L'expérimentation Tribot (1)

⇒ Superviser la réalisation d'un comportement sensorimoteur.

□ Le robot Tribot :

- Robot mobile  Mindstorms NXT,
- 3 servomoteurs,
- 4 capteurs (ultrasons, pression, son, lumière),
- 1 micro-contrôleur 32 bits.



□ Objectif : Saisie d'un objet situé à proximité du Tribot.

□ 5 actions élémentaires codées en langage URBI

↳ Trouver une cible, avancer, avancer avec précision, ouvrir la pince, fermer la pince.

L'expérimentation Tribot (2)

⇒ **Confrontation à la réalité opérationnelle d'un robot.**

→ Plate-forme mécanique très simple } ⇒ **Comportements sensorimoteurs**
→ Gamme réduite de mouvements }

→ Servomoteurs et capteurs de qualité moyenne } ⇒ **Test de robustesse** pour notre architecture

⇒ Modélisation des actions élémentaires par CONACQ,

⇒ Capacité d'adaptation aux perturbations mineures,

⇒ Évaluation des performances de la replanification de comportement.

Acquisition automatique des A.E. (1)

□ Données d'entraînement :

- ↳ Tuples étiquetés : $(\text{dist}^I, \text{bump}^I, \text{pince}^I, \alpha_R, \alpha_L, \alpha_C, \text{dist}^F, \text{bump}^F, \text{pince}^F)$
- ↳ Descripteurs à valeurs entières (pas de discrétisation)
- ↳ 15 instances en moyenne

□ Modélisation de l'action Avancer avec précision :

d^I	b^I	c^I	α_R	α_L	α_C	d^F	b^F	c^F	
80	0	1	240	240	0	70	0	1	+
70	0	0	120	120	0	65	0	1	-
...									
80	0	1	24	24	24	0	79	0	-

$$\Rightarrow \left\{ \begin{array}{l} \text{Pre}(a_1) = \{(\text{dist}^I < 100), (\text{bump}^I = 0), (\text{pince}^I = 1)\} \\ \text{Actuators}(a_1) = \{(\alpha_L = \alpha_R = 24 \times (\text{dist}^I - \text{dist}^F)), (\alpha_C = 0)\} \\ \text{Post}(a_1) = \{(\text{dist}^I - 10 \leq \text{dist}^F \leq \text{dist}^I - 1), \\ (\text{dist}^F = 7 \Rightarrow \text{bump}^F = 1), (\text{pince}^F = \text{pince}^I)\} \end{array} \right.$$

Acquisition automatique des A.E. (2)

a_2 : Standard move

$$\left\{ \begin{array}{l} Pre(a_2) = \{(10 < d^I < 100), (b^I = 0)\} \\ \\ Actuators(a_2) = \{(\alpha_R = \lceil (d^I - d^F)/0.9 \rceil), \\ \quad (\alpha_C = \alpha_R), (\alpha_C = 0)\} \\ \\ Post(a_2) = \{(d^I - 36 \leq d^F \leq d^I - 27), (d^F > 10), \\ \quad (b^F = 0), (c^F = c^I)\} \end{array} \right.$$

a_4 : Open claw

$$\left\{ \begin{array}{l} Pre(a_4) = \{(d^I > 10), (b^I = 0), (c^I = 0)\} \\ \\ Actuators(a_4) = \{(\alpha_R = \alpha_L = 0), (\alpha_C = 10)\} \\ \\ Post(a_4) = \{(d^F = d^I), (b^F = b^I), (c^F = 1)\} \end{array} \right.$$

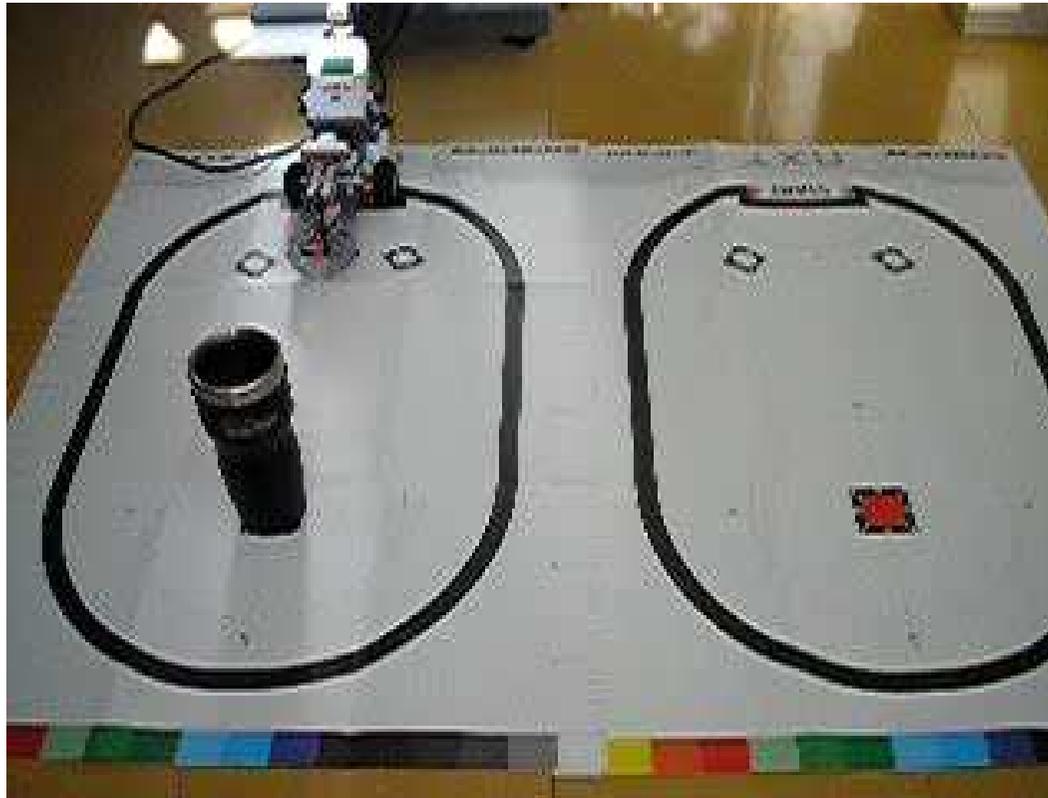
a_3 : Find target

$$\left\{ \begin{array}{l} Pre(a_3) = \{(d^I > 100), (b^I = 0)\} \\ \\ Actuators(a_3) = \{(\alpha_R = 5), (\alpha_L = -5), (\alpha_C = 0)\} \\ \\ Post(a_3) = \{(d^F \leq 100), (c^F = c^I)\} \end{array} \right.$$

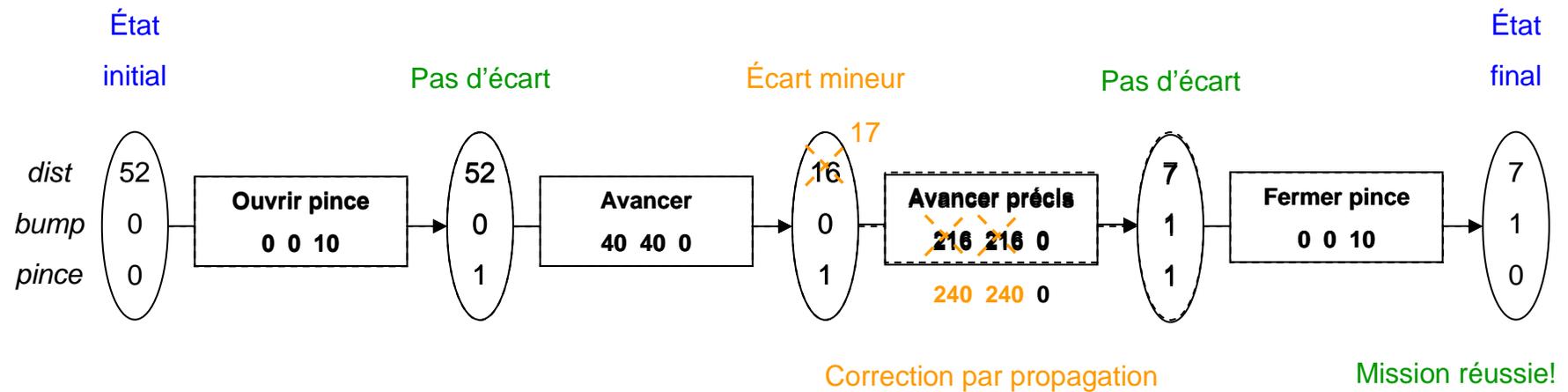
a_5 : Close claw

$$\left\{ \begin{array}{l} Pre(a_5) = \{(d^I < 100), (c^I = 1)\} \\ \\ Actuators(a_5) = \{(\alpha_R = \alpha_L = 0), (\alpha_C = -50)\} \\ \\ Post(a_5) = \{(d^F = d^I), (d^F = 7 \Rightarrow b^F = 1), (c^F = 0)\} \end{array} \right.$$

Tribot : Saisie d'un objet (1)

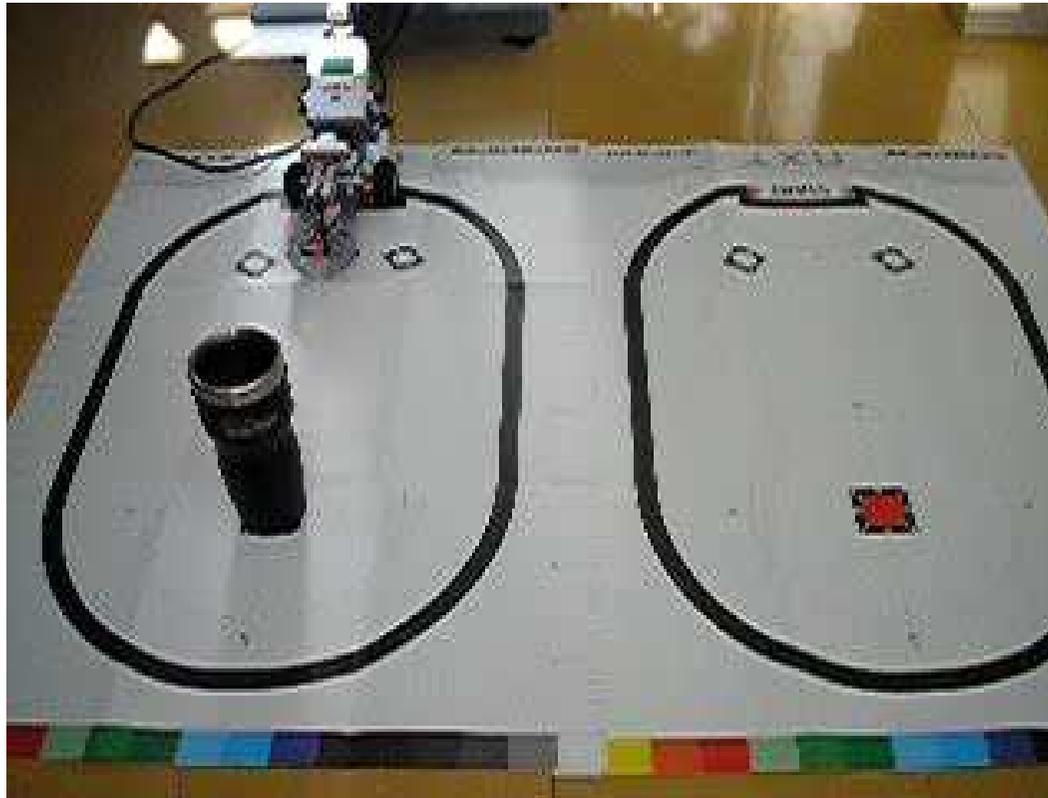


Tribot : Saisie d'un objet (1)

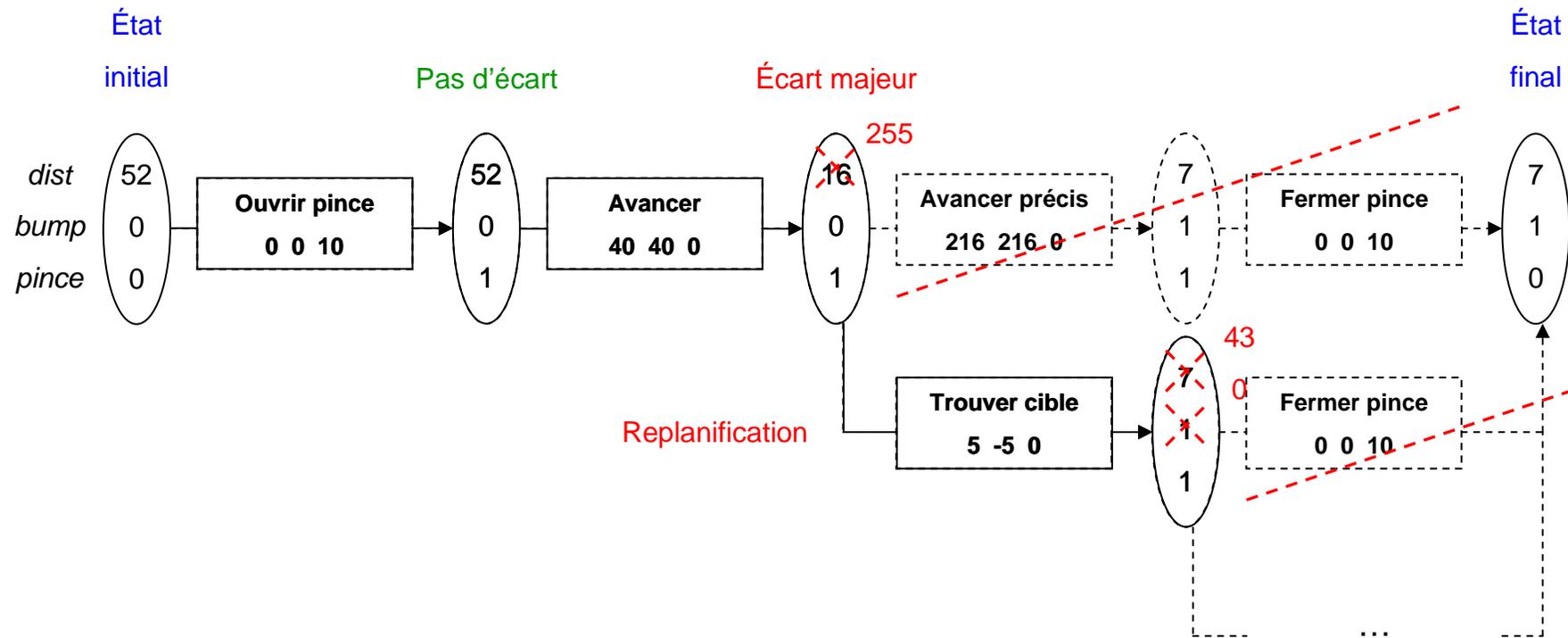


⇒ Correction *on-line* des dysfonctionnements mineurs...

Tribot : Saisie d'un objet (2)



Tribot : Saisie d'un objet (2)

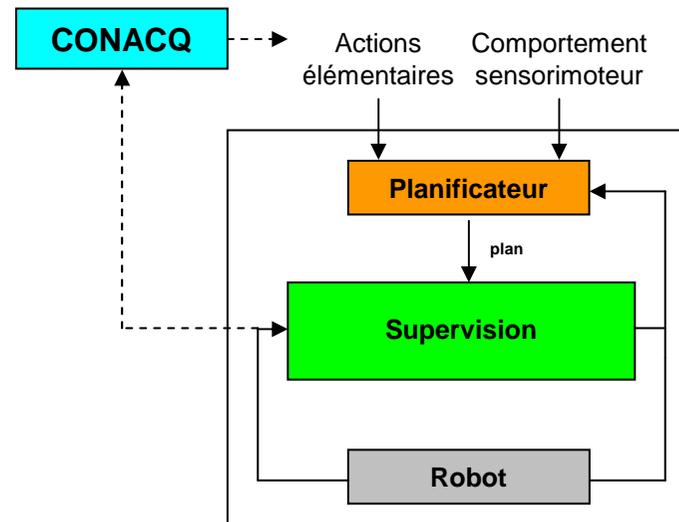


⇒ Replanification en cas de modifications majeures...



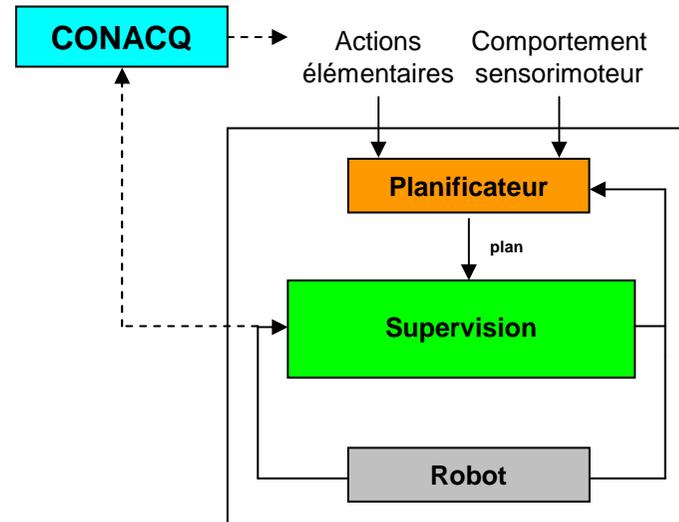
Conclusion & perspectives

Conclusion



- ↪ Une architecture logicielle utilisant les réseaux de contraintes pour **planifier** et **superviser** l'exécution de **comportements sensorimoteurs**,
- ↪ Une réponse aux impératifs d'**expressivité** et d'**efficacité** liés à la robotique,
- ↪ Conacq permet d'automatiser le processus de modélisation des A.E.

Perspectives



- Supervision de robots d'une complexité mécanique supérieure.
- Gestion des actions parallèles et/ou asynchrones.
- CSP **hybrides** pour une plus grande expressivité & efficacité
 - ↳ Performances des résolveurs hybrides actuels
 - ↳ Plate-forme d'acquisition automatique de CSP hybrides

Travaux connexes...

En collaboration avec C. Bessière, R. Coletta & Barry O'Sullivan

- Apprentissage interactif de réseau de contraintes
 - ↳ Développement d'un Conacq interactif,
 - ↳ Évaluation théorique & empirique de 3 stratégies de questionnement.
- ⇒ Amélioration sensible des performances de Conacq



Apprentissage interactif de réseau de contraintes

Mathias Paulin, Actes de la conférence RJCIA'05, Juin 2005.



Query-driven Constraint Acquisition

C. Bessiere, R. Coletta, B. O'Sullivan, M. Paulin, Actes de la conférence IJCAI'07, Janvier 2007.