

# On the Relationship between Robotics and Artificial Intelligence

Philippe Morignot

*Abstract— Robotics and Artificial Intelligence stand as two disciplines related to computer science. Robotics aims at building robots which autonomously perceive, (eventually) reason and act (e.g., communicate) in real physical environments. A.I. aims at representing knowledge and reasoning on it, in a way as close as possible to human reasoning. Without raising arbitrary barriers, these two disciplines are different: general conferences in the A.I. field include IJCAI, ECAI, AAAI, RFIA and many specialized conferences ; general conferences in the Robotics field include ICRA, IROS and many other specialized conferences, e.g., on control theory. However, many Robotics-claimed work include A.I. algorithms: The A\* algorithm and its variants are used for path planning, i.e., finding a continuous path from location A to location B on a map while avoiding obstacles recorded on the map; Evolutionary computing is used to optimize online merging of maps represented as occupancy grids, or in micro-robotics; many examples can be drawn along similar lines.*

*In this paper, we sketch the main goals of A.I. and of Robotics, record their differences and highlight the convergence points, aiming at a better communication between the two communities.*

## I. INTRODUCTION

ARTIFICIAL Intelligence and Robotics, as two main domains related to Computer Science, both have a long history since their introduction, which can be traced back over the centuries for their initial ideas. However it still is difficult to define these domains, in order for each one to cover the vast variety of work performed under each banner. Let us attempt at using one definition of A.I.: is relevant to Artificial Intelligence any computer program which would be said “intelligent” if its activity would be so considered when performed by a human. Such a definition leads to the imitation game, or the Turing test, in which a tester has to determine to whom it communicates with through a computer: a computer program or a human.

On the other hand, let us attempt at defining Robotics as the design of mechanical devices known as robots, a word invented by a Czech writer in the early 20<sup>th</sup> century. Given these two definitions, for A.I. on one side and for Robotics on the other side, there seems to be a wide gap between the two domains, except if we want to talk about intelligent robots, the notion of intelligence being added to the one of robot.

But the gap between the two domains is not so wide. For example, a widely adopted textbook on A.I. [14] includes two chapters on robotics (written by Sebastian Thrun): chapter 24 “Perception” and 25 “Robotics”. The overall impression produced by this chapter organization is that robotics is a sub domain of A.I., i.e., a field on which A.I. can be applied. The result seems to be that robotics looks like an application domain of A.I. “Let’s give a body to our A.I. algorithms”.

On the other hand, an introductory book on robotics [7] considers A.I. as a software module giving intelligence to a robot. “Let’s plug a brain into that robot”.

In this paper, we argue that the point where the two communities seem to meet is the notion of intelligent robot, i.e., a robot on one side which would exhibit an intelligent behavior on the other side. More precisely, we advocate that this notion unfolds into the notion of software architecture of a robotic agent, i.e., the way the algorithms running on a robotic agent are organized, in order for the mechanical and electronic device to be able to face a real dynamic environment.

The paper is organized as follows: section II presents our understanding of how the A.I. community perceives the Robotics one, and the opposite (how the A.I. community is perceived from the Robotics one). Section III proposes a short survey of the point where the two communities meet to our opinion: the software architecture of a robotic agent. Section IV discusses the theoretical point underlying to our views the remaining gap between the two communities. Finally, the last section sums up our contribution.

## II. FROM A.I. TO ROBOTICS, AND BACK

In this section, we attempt at drawing how the A.I. community perceives the robotics one on one side, and then how the robotics community perceives the A.I. one on the other side.

### A. Robotics considered from A.I.

In the early 70s, Robotics and A.I. were not as separated as now. For example, the robot SHAKEY from the SRI had its behavior driven by the first A.I. task planner, named STRIPS [5]. This was the first time a robotic agent was able to exhibit a coherent behavior while actually computing first the sequence of actions it would take later. In other words, the robot did not know at first what to do: It computed what to do (with the STRIPS task planner) before actually doing it (robotics part). One side effect of this project was the start of the A.I. planning community, resulting 40 years later in the ICAPS conferences series.

As another example, S. Koenig et al. propose an improvement of the A\* algorithm, called lifelong planning

---

This work was supported by team IMARA, INRIA Rocquencourt.

P. Morignot ([philippe.morignot@inria.fr](mailto:philippe.morignot@inria.fr)), phone : +33 (0)1 39 63 58 27, fax : +33 (0)1 39 63 54 69) is with INRIA Rocquencourt, Team IMARA, Domaine de Voluceau, B.P. 105, 78150 Le Chesnay, France.

A\* [9], which is tested in the case of path planning robotic problem. This is another example of A.I. work resulting in robotic applications.

### B. A.I. considered from Robotics

The main way A.I. is considered from the domain of Robotics seems to be a library of algorithms. The most obvious A.I. algorithm imported into the robotics community seems to be A\*, which can be used for path planning, i.e., finding a sequence of locations which leads a start location to a target location while avoiding known obstacles. This model relies on discretizing the environment, considering some cells as successors (in the successor function of the A\* algorithm) of the current cell and identifying the `IsGoalReached?` boolean function of A\* as a successor cell being the goal location. (See the previous paragraph for improvements.)

Another algorithmic import from A.I. towards robotics is genetic algorithms. For example, Li and Nashashibi use genetic algorithms for map merging [10]: given one map of some environment, how to find a rotation/translation of a second map which maximizes the number of pixels that match among the two maps? Considering this problem as an optimization problem would lead to a combinatorial explosion, and such a stochastic method can be applied successfully to merging maps coming from two different robotic vehicles.

Other algorithmic imports from A.I. towards robotics include fuzzy logic. For example, J. Perez et al. use fuzzy logic to define a small set of fuzzy rules determining the behavior of a driverless vehicle, i.e., for the control part of a robotic vehicle [13]. Given the robotic software architecture of robotic driverless vehicle (see next section), the behavior of a driverless vehicle can thus be defined by a small set of (fuzzy) rules.

## III. AGENT ARCHITECTURES

A notion relating A.I. and Robotics seems to be the one of *robotic agent architecture*, considered as the way to organize algorithms inside a robotic agent, which has to evolve in a physical real dynamic environment, thus constituting an intelligent robot.

A first robotic agent architecture is the Sense-Plan-Act loop [11] (see Fig. 1), in which the agent sequentially perceives its environment, builds an action plan through task planning, and executes it. The main problem with this architecture is that task planning is an NP complete problem, entailing that the task planning component might eventually take a very long time to produce a plan of tasks, for the agent to then execute it and eventually exhibit a motion. As a result, the overall agent might get stuck in the environment, while the environment might change and require attention --- in the worst case, the produced action plan might be obsolete once delivered because of environmental changes.

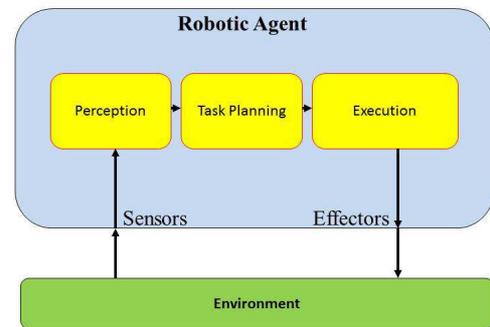


Fig. 1: The Sense-Plan-Act robotic architecture.

A radically different view is proposed by R. Brooks with the subsumption architecture [3] (see Fig. 2). In that approach, the robotic agent is composed of a finite state automaton, the parameters of which are set by an upper finite state automaton, the parameters of which ... until the slowest (and upmost) finite state automaton is reached. This architecture incorporates no deliberation at all, since no symbol is allowed [4]. The question then becomes: is such an agent intelligent?

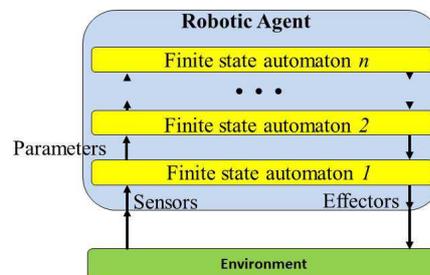


Fig. 2: The Subsumption architecture.

As opposed to the subsumption architecture, and as an improvement of the Sense-Plan-Act architecture, Hayes-Roth et al. propose a 2-layer architecture of a robotic agent [8] (see Fig. 3). In that architecture, there are 2 levels: one (lowest) for sensori-motor control loops (encoding "behaviors" in a "physical" layer), activating actual motion of the agent, and one (upmost) for recognizing a situation given perception, task planning, and plan monitoring (executing each action of the plan in sequence), in a symbolic "cognitive" layer. A major point is that these two levels run in parallel, therefore the agent can adopt a specific behavior even if A.I. task planning has not produced a task plan yet. Therefore, such a robotic agent is not stuck

deliberating, as in the Sense-Plan-Act architecture, and still incorporated A.I. task planning, as opposed to the subsumption architecture.

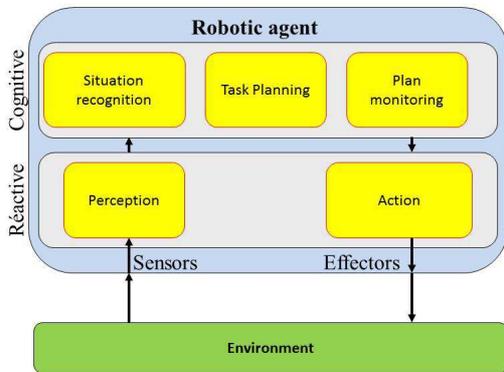


Fig. 3: The two-level robotic architecture.

Another robotic agent architecture proposes the concept of deadline as first class notion, in a 3-layer architecture [6] (see Fig. 4). In that approach, there again is a layer for deliberation (called deliberator) and a layer for physical behaviors (called controller). But in between lays another layer (called Sequencor) which activates components of the 2 previous layers, while allotting them a deadline to finish. As a result, the long computing of the deliberator can be avoided in a crude way.

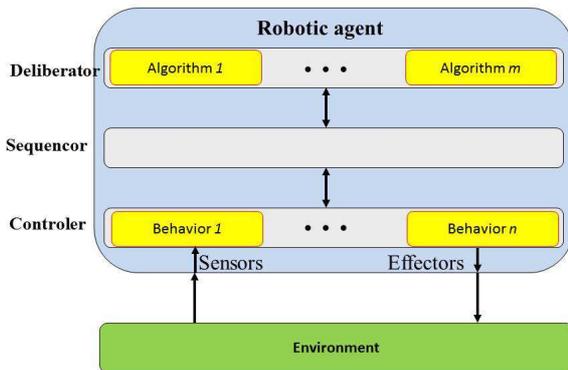


Fig. 4: The 3-level robotic architecture.

In line with the previous 3-level architecture and retaining concepts of the previous 2-layer architecture is the LAAS architecture [1] (see Fig. 5). This approach is composed of a cognitive layer (called Deliberator), which includes an A.I. task planner (IxTeT) and a procedural executive to activated produced plans (PRS), and a lowest layer (called Executive), which includes the available sensori-motor control loops of the robotic agent. In between lays a functional level, which

chooses a sensori motor control loop given the specification of the action to undertake, produced by the deliberative layer.

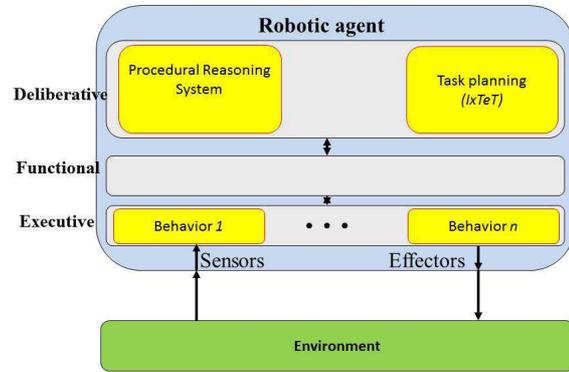


Fig. 5: The LAAS robotic architecture.

An improvement of the previous 2-level architecture, is the 2++ level architecture [2] (see Fig. 6). This architecture is close in spirit to the 2-level architecture of Hayes-Roth et al. [8], but includes an additional link between the Perception reactive component and the Action reactive component, for transmitting contingent plans in case of emergency. In case of emergency, a predefined contingent plan ("panic plan") is adopted by the agent, while the cognitive level looks for a rational action plan to take the emergency event into account. Interesting behaviors of groups have been obtained for simulated aircraft in an adverse environment [2].

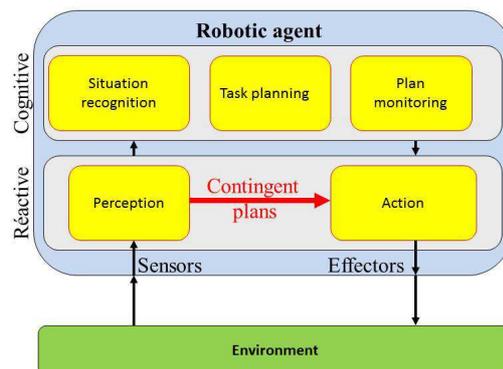


Fig. 6: The 2++-level robotic architecture [2].

A final robotic agent architecture has been proposed for intelligent transportation systems, such as automated cars in daily traffic environments [12] (see Fig. 7). This approach essentially is a Sense-Plan-Act architecture where task

planning has been replaced by path planning: the vehicle knows since the beginning its plan of action, but dynamically computes its future trajectory on a potentially crowded road, for later computing the detailed voltage to give to electrical effectors (e.g., engine). This architecture is successfully used for driverless automated vehicles such as CyberCars (see Fig. 8).

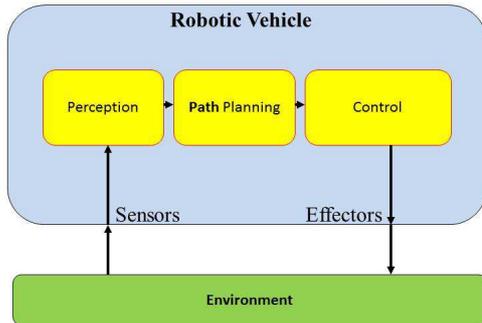


Fig. 7: The driverless robotic vehicle architecture.



Fig. 8: A CyBus, as an intelligent transportation system.

#### IV. DISCUSSION

A main characteristics of robotic models is that they must be capable of representing errors. For example, SLAM (Simultaneous Localization And Mapping) aims at both knowing where the robotic agent is on its map (localization) and building its map (mapping). Given its sensors, which mainly return a distance to potential obstacles (e.g., a laser), the problem is that if the agent knows where it is on its map, it can infer where the perceived obstacle is (by adding a vector to its own position); On the opposite, if the robotic agent knows where the obstacle is on its map, it can infer where it is on its own map (by subtracting a vector to the obstacle's position). The problem of SLAM is that none of

the two positions are known, therefore a SLAM algorithm accumulates measures in order to build an estimate of the two positions. Such a reasoning based on error shrink uses probabilities.

On the other hand, many A.I. techniques use discreteness: search algorithms, constraint programming, ontologies, to cite a few.

We believe that the main difference between Robotics and A.I. lay in the opposition between continuity (probabilities) and discreteness (integers). For example, the previous section shows examples of discretely encapsulating continuous algorithms: a continuous sensori-motor control loop (i.e., a behaviour of the physical layer of several of the previous architectures) is encapsulated into a discrete model (the architecture itself).

Similar opposition between continuous and discrete models may be found in other domains of computer science. For example, in Operational Research, the simplex algorithm (i.e., what is the value of real variables which minimize a linear cost function, given constraints on these variables represented as linear inequalities?) is based on continuous (real) values, i.e., the domain of a variable is the set of real numbers  $\mathbb{R}$ . However when we want to obtain discrete values of variables for solving the same problem, variables then belong to the set of integers  $\mathbb{N}$ , and no longer to the set of real numbers  $\mathbb{R}$ . As a result, this same problem, but with the variables domain changed, becomes NP-complete, and one possible approach is the branch & bound algorithm, since the simplex algorithm alone does not guarantee that the values of the variables will be integers (discrete). The overall behaviour of the branch & bound algorithm, encapsulating calls to the (continuous) simplex algorithm at each node of the developed tree (relaxed solution), is closed in spirit to the notion of software architectures of a robotic agent (see previous section): both notions are discrete reasoning on continuous algorithms (sensori-motor loops controlled by task planning, for software architectures; simplex algorithm controlled by a tree search, for a branch & bound algorithm).

Now the next theoretical step lays in the opposition between the set of integers  $\mathbb{N}$  (discreteness) and the set of real numbers  $\mathbb{R}$  (continuity).  $\mathbb{N}$  is included in  $\mathbb{R}$ , and there is no bijection mapping  $\mathbb{N}$  to  $\mathbb{R}$  ( $\mathbb{R}$  cannot be enumerated). But is that the end of the story?

#### V. CONCLUSION

In this paper, we discuss the relationship between Robotics and Artificial Intelligence. Although more coupled 40 years ago, the two disciplines seem to have followed different paths, leading to two different communities with specific academic forums for each. A.I. seems to consider robotics as an application domain, and Robotics seems to import A.I. algorithms when needed, as in a library. We advocate that the goal of building an intelligent robot seems to unify the two domains. Towards this we present a brief survey of the notion of software architectures of robotic agents. Furthermore, we present arguments suggesting that the difference between the two domains might lay in the

difference between discreteness and continuity, which fall on hard theoretical problems.

#### ACKNOWLEDGMENT

The author thanks Fawzi Nashashibi and the members of the RITS ex-IMARA team (INRIA Rocquencourt).

#### REFERENCES

- [1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, F. Ingrand. *An Architecture for Autonomy*. In *International Journal of Robotics Research* (Special Issue on "Integrated Architectures for Robot Control and Programming"), Vol 17, N° 4, April 1998. LAAS Report N°97352.
- [2] J. Baltié, E. Bensana, P. Fabiani, J. – L. Farges, S. Millet, P. Morignot, B. Patin, G. Petitjean, G. Pitois, J. – C. Poncet. *Multi-Vehicle Missions: Architecture and Algorithms for Distributed On Line Planning*. In Dimitri Vrakas and Ioannis Vlahavas (eds.), *Artificial Intelligence for Advanced Problem Solving Techniques*, Information Science Reference. December 2007.
- [3] Brooks, R. A. *A Robust Layered Control System for a Mobile Robot*. In *IEEE Journal of Robotics and Automation*, Vol. 2, No. 1, March 1986, pp. 14–23.
- [4] R. Brooks. *Intelligence without reason*. Proceedings of 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91), Sydney, Australia, August 1991, pp. 569–595.
- [5] R. Fikes, N. Nilsson. STRIPS, a new Approach to the Application of Theorem Proving. *Artificial Intelligence*, vol. 2(1971), pp. 189-208.
- [6] Gat, E. *Three-layer architectures*. In D. Kortenkamp et al. Eds. *A.I. and mobile robots*. AAAI Press, 1998.
- [7] R. Gélin. *Le robot: ami ou ennemi ?* Le Pommier, Paris. 2006. (in French)
- [8] Hayes-Roth, B.; Pflieger, K.; Morignot, P.; Lalanda, P. *Plans and Behavior in Intelligent Agents*. Knowledge Systems Laboratory, KSL-95-35, Stanford Univ., CA, March, 1995.
- [9] S. Koenig, M. Likhachev, D. Furcy. Lifelong Planning A\*. *Artificial Intelligence Journal (AIJ)*, 155(1-2), pages 93-146, 2004.
- [10] H. Li, F. Nashashibi. A New Method for Occupancy Grid Maps Merging: Application to Multi-vehicle Cooperative Local Mapping and Moving Object Detection in Outdoor Environment.. 12<sup>th</sup> International Conference on Control, Automation, Robotics and Vision (ICARCV'12), Guangzhou, China, 2012.
- [11] Nils J. Nilsson. *Principles of Artificial Intelligence*. Palo Alto: Tioga. 1980.
- [12] M. Parent. Advanced Urban Transport: Automation is on the way. *Intelligent Systems, IEEE*, vol. 22, no. 2, pp. 9-11, March-April. 2007.
- [13] J. Pérez, V. Milanés, E. Onieva, J. Godoy, J. y Alonso. *Longitudinal fuzzy control for autonomous overtaking*. IEEE International Conference on Mechatronics (ICM'11), 2011, pages 15-21.
- [14] S. Russell, P. Norvig. *Artificial Intelligence: A Modern Approach*. Second Edition, Prentice Hall, Pearson Education, NJ, 2003, 1081 pages.