HARPIC, an Hybrid Architecture Based on Representations, Perception, and Intelligent Control: A Way to Provide Autonomy to Robots

Dominique Luzeaux and André Dalgarrondo

DGA/Centre Technique d’Arcueil,
16bis av. Prieur de la Côte d’Or, 94114 Arcueil Cedex, France,
luzeaux@etca.fr, dalga@etca.fr,
WWW home page: <http://www.etca.fr/CTA/gip/Publis/Luzeaux/>

Abstract. In this paper we discuss an hybrid architecture, including reactive and deliberative behaviors, which we have developed to confer autonomy to unmanned robotics systems. Two main features characterize our work: on the one hand the ability for the robot to control its own autonomy, and on the other hand the capacity to evolve and to learn.

1 Introduction

As was mentioned recently in a call for participation to a special issue on intelligent systems design, complex intelligent systems are getting to the point where it almost feels as if “someone” is there behind the interface. This impression comes across most strongly in the field of robotics because these agents are physically embodied, much as humans are. There are several primary components to this phenomenon. First, the system must be capable of action in some reasonably complicated domain: a non-trivial environment within which the system has to evolve, and a rather elaborate task which the system should fulfill. Second, the system must be capable of communicating with other systems and even humans using specified possibly language-like modalities, i.e. not a mere succession of binary data, but some degree of symbolic representations. Third, the system should be able to reason about its actions, with the aim of ultimately adapting them. Finally, the system should be able to learn and adapt to changing conditions to some extent, either on the basis of external feedback or relying on its own reasoning capacities.

These remarks have guided our research and in order to bring up some answers and to build an unmanned ground robot that could perform well in ill-known environments, we have focused on the robot control architecture, which is the core of the intelligence, as it binds together and manages all the components. In the next sections, we will first discuss shortly what we understand under autonomy for robots and control architectures. Then we will describe the architecture we propose and give some implementation details and results. The final sections are then dedicated to the possibility for the system first to control its own autonomy depending on external input and second to learn and adapt.

1.1 Autonomy in Robotic Systems

In order to tackle autonomous robots, one has first to delimit the scope of expected results; this calls for a tentative definition of autonomy. A first necessary condition for a system to be called autonomous is to be able to fire reactions when faced with external disturbances: this yields a concept obviously parametrized by the nature and diversity of disturbances one can act against. However mere reaction to disturbances cannot be truly accepted as autonomy as it does not encompass longer-term decision abilities. A more demanding definition includes the ability to change the interaction modes with the environment: this captures the idea that an autonomous organization is not static in its functioning ways and can “adapt”. Looking closely at the implications, one sees that such an organization necessarily has to internalize external constraints, which means the ability to *integrate knowledge of its own dynamics and representation of the exterior*. To sum up, an interaction without explicit representation of both an internal world corresponding to the system and an external world relative to the environment cannot be called autonomous (consider a painting industrial robot with simple contact feedback as a counterexample). Notice that this does not mean the representations have to be entirely different (on the contrary, efficient sensorimotor closed loops require an integration of the various representations!). Concluding these paragraphs on autonomy, we see that although there are epistemological necessary conditions for autonomy, there is no absolute autonomy: a system can reasonably only be said “more autonomous” than another.

In our approach to autonomous systems, we have proceeded in a bottom-up fashion, handling first the control and perception issues, and looking for adequate representations which could integrate both these issues, leading to sensorimotor i/o behavior [4,10,11]. Since we are not interested in robot wandering aimlessly within corridors, possibly avoiding scientists strolling through the lab, but have to deal with changing situations, even rapidly changing, ranging from slight modification of the environment to its thorough transformation (or at least a transformation of the stored model: e.g. discrepancies between the cartographic memory of the world and the current location, due to stale information or possible destruction of infrastructures), we have incorporated deliberative and decision capacities in the system, that do not run necessarily at the same temporal rate than the i/o behavior execution.

In order to scale up to system level, we must then turn to control architectures, i.e. we have to detail how the various functions related to perception and action have to be organized in order for the whole system to fulfill a given objective. To achieve this, we work with *hybrid control architectures*, integrating a lower level focusing on intelligent control and active perception, and a higher level provided through the mission planning.

1.2 Robot Control Architectures

As a complex system collocating sensors, actuators, electronic and mechanical devices, computing resources, a robot has to be provided ways to organize these

various heterogeneous components in order to fulfill its prescribed mission, which furthermore may evolve in time. This is all the more important when additional constraints, such as real-time and cost issues – nowadays a major issue for operational systems – are involved. The control architecture deals with these problems and brings answers to the following questions:

- how is the system built from its basic components?
- how do the parts build up a whole?
- how should components be (re)organized to fulfill missions changing in time?

By basic components, one has to understand mechanical, electronical and software aspects, sensors, actuators, but also the ways to relate these elements and the interfaces between the various subsystems.

For a general overview of existing control architectures, see [1]. The first architectures historically introduced in mobile robots derive from the sense-plan-act paradigm taken from hard artificial intelligence, and follow a top-down approach relying on a recursive functional decomposition of the problem into subproblems down to a grain level where an explicit solution to the problem is given. Such architectures have been shown to suffer from the symbol-grounding [5], the frame and the brittleness problems [6]. In other words, they manipulate symbols which cannot be related in a constructive way to features of the environment and they have to rely on a model of the environment which has to be complete and redefined hierarchically in order to cope with the top-down functional decomposition. While this works for static environments, any unplanned situation will have dramatic impact on the robot! [3]

In reaction to that approach, bottom-up approaches have been proposed, inspired by biology and ethology. They do not rely on explicit models of the environment but on input-output reactive behaviors, which may be aggregated together to solve a more complex task. One of the most famous bottom-up architectures is Brook's subsumption architecture.

However it is generally admitted that both approaches have failed, mainly because of their radical positions: top-down approaches lead to awkward robots unable to cope with any unforeseen change of the environment or the mission, while bottom-up approaches have led to promising animal-like robots which unfortunately could not solve complex problems or missions. Building on their respective advantages (and hopefully not cumulating their drawbacks!), hybrid architectures have been investigated in recent years. They try to have on the one hand a reactive component and on the other hand a decision or planning module; the difficulty is of course the interface between these layers and here lies the diversity of the current approaches.

2 HARPIC

2.1 General Description

We propose an hybrid architecture (cf. figure 1) which consists in four blocks organized around a fifth: *perception processes*, an *attention manager*, a *behavior*

selector and *action processes*. The core of the architecture relies on *representations*.

Sensors yield data to perception processes which create representations of the environment. These perception processes are activated or inhibited by the attention manager and receive also an information on the current executed behavior. This information is used to foresee and check the consistency of the representation. The attention manager updates representations (on a periodical or on an exceptional basis), supervises the environment (detection of new events) and the algorithms (prediction/feedback control) and guarantees an efficient use of the computing resources. The action selection module chooses the robot's behavior depending on the predefined goal(s), the current action, the representations and their estimated reliability. Finally, the behaviors control the robot's actuators in closed loop with the associated perception processes.

Fig. 1. Functional diagram of the HARPIC architecture.

The key ideas of that architecture are:

- *The use of sensorimotor behaviors linking internally and externally perceptions and low-level actions:* the internal coupling allows to compare a prediction of the next perception, estimated from the previous perception and the current control, with the perception obtained after application of the control, in order to decide whether the current behavior runs normally or should be changed.
- *Use of perception processes* with the aim of creating local situated representations of the environment. No global model of the environment is used; however less local and higher level representations can be built from the instantaneous local representations.
- *Quantitative assessment of every representation* [13]: every algorithm has associated evaluation metrics which assign to every constructed representation a

numerical value that expresses the confidence which can be given to it. This is important, because any processing algorithm has a domain of validity and its internal parameters are best suited for some situations: there is no perfect algorithm that always yields “good” results.

- *Use of an attention manager*: it supervises the executions of the perception processing algorithms independently from the current actions. It takes into account the processing time needed for each perception process, as well as the cost in terms of needed computational resources. It looks also for new events due to the dynamics of the environment, which may signify a new danger or opportunities leading to a change of behavior. It may also fire processes in order to check whether sensors function nominally and can receive error signals coming from current perception processes. In practice, for instance with a vision sensor, the attention will focus on the illumination conditions, on the consistency between the movement of the robot and the temporal consistency of the representations, and on error signals sent by perception processes. With this information it is then possible to invalidate representations due to malfunctioning sensors or misused processes.
- *The behavior selection module* chooses the sensorimotor behaviors to be activated or inhibited depending on the predefined goal, the available representations and the events issued from the attention manager. This module is the highest level of the architecture. It should be noted that the quantitative assessment of the representations plays a key role in the decision process of the behavior selection: on the one hand a representation might be more or less adapted to the current situation, depending for instance on the sensor used or on the conditions of the perception acquisition (e.g. a day camera used during night operating conditions will yield representations to which a lower confidence should be assigned a priori; the same holds also for instance for a detector relying on translational invariance while the robot has such a motion that this assumption is incorrect) ; on the other hand some representations might be more interesting for some behaviors or might provide an improved help to choose between several behaviors (e.g. a wall-following behavior needs information on contours more than velocity vectors, while a tracking behavior has the opposite needs). Therefore every behavior weights also each representation depending on its direct usability, and this weight is combined with the intrinsic assessment of the representation.
- *The action selection module* regroups the lower-level controllers operating on the actuators. It uses valid representations in order to compute the control laws.

This modular architecture allows to develop independently the various processes belonging to each of the four entities, before integrating them together. Its originality relies in both the lower-level loop between perception and action, necessary for instance for active vision and any situated approach of perception, and the decoupling of the perception and the action processes, which become behaviors during their execution. The latter avoids the redundancy of common components, saves computational resources when representations are common to several behaviors and limits the conflicts when accessing the hardware resources of the robot. Compared to other hybrid architectures (the so-called three-layer

approach which develops the three following levels: symbolic level, reactive behaviors, lower-level control), we have tried to focus more on the relations between these three levels in order to take into account the heterogeneous loop aspect characterizing a complex robot.

Furthermore, the proposed architecture with both loops between perception and action, one at a lower level, the other one relying on representations, seems a plausible model from a biological point of view [2]: whereas the lower-level sensorimotor closed loop is characteristic of simpler organisms, the cognitive capacities of higher organisms, like mammals, can be explained by another loop relying on a dynamical model of the environment and the organism, running in feedforward mode with a slower time scale. This higher-level loop allows also open loop planning in order to check the validity of some representations and can trigger their update (e.g. the racing pilot who concentrates on the track and the race, when rehearsing in his head before he actually starts the race).

Finally, the attention manager is a notion used in biological vision [7], and provides a very efficient supervision concept for artificial systems, firing batch processes or reacting on discrete events. The asynchronous property of the control architecture is due to that attention manager, and we think this property is a keystone in complex systems, which have to deal with unpredictable environments and limited resources.

2.2 Implementation and Experimentation

Fundamental capacities of our architecture encompass modularity, encapsulation, scalability and parallel execution. For these needs we decided to use a multi-agent formalism that fits naturally our need for encapsulation in independent, asynchronous and heterogeneous modules. The communication between agents is realized by messages. Object oriented language are therefore absolutely suited for programming agents: we chose C++. We use POSIX Threads to obtain parallelism: each agent is represented by a thread in the overall process. We do not aim at promoting multi-agent techniques, for us it is merely an interesting formalism and our architecture could be implemented without them.

All the processing is done by a PC bi-PIII 500 MHz running Linux, with a frame grabber card. We did not use any specialized hardware or real-time operating system. The robots used both in indoor and outdoor experiments are Pioneer AT from RWI equipped with a monochrome camera, with motorized lens, mounted on a pan-tilt unit (cf. figure 2). The robot links to the computer are a radio modem at 9600 bauds and a video link which transmits 768×512 pixel images at a 25 Hz rate. The robot displacement speed is about 400 mm/s.

We have developed perception agents, for lane, obstacle and target detection, and action agents for lane following, obstacle avoiding, target tracking and lens aperture adjusting. The processing time of the perception agents varies between 5 and 30 ms according to their algorithms.

Fig. 2. Outdoor experiments with the robot (structured roads, unstructured tracks).

3 Computational Intelligence and Controlled Autonomy

Through a parallel execution of a wide range of algorithms and after the choice of the “best action” depending on the evaluation of the most adequate representation given a current situation and behavior, it is easily understandable how a high autonomy is achieved. Namely there is a straightforward scheduling of contextual modes: for instance, for an outdoor robot, there is no need to tell explicitly the robot when it drives on structured roads and when it is off-road, since the right representation computed by the various vision algorithms should be selected by the navigation behavior. As an illustration, when dealing with our robot in indoor environments with competing behaviors such as wall-following, obstacle avoidance, navigation with given heading, we have analyzed the sequence of behaviors as selected by the architecture, and there was a natural transition from wall-following to navigation with given heading and back to wall-following when a door occurred.

This is all the more important in structured outdoor environments, where curbs or road marks can suddenly disappear, either because of a crossroad or more often because the reference structure is partially damaged. Truly, we observed [13] that, while driving on roads delimited by sidewalks, detectors based on contour detection had higher assessment marks than other representations based for instance on multifractal texture indices, whereas the contrary happened when arriving on unstructured lanes such as off-road tracks.

Whereas usual roboticists would then ask for more robust perception algorithms, we prefer a more intelligent design that does not require perfect perception or action to work.

Of course, a critic arises then concerning the intrinsic stability of the behavior selector: one could fear a new behavior would be chosen every time step. In order to avoid this, we have established a dichotomy between behavior selection and action selection. A behavior has thus a different time scale than the firing of an action or the data acquisition process; a consequence is a temporal

smoothing. Besides, the behavior selector uses a knowledge base which defines which behaviors might follow a given behavior. This reduces the combinatorics and introduces some determinism in the behavior scheduling. One should note that in order not to have too much determinism, we allow the attention manager to interrupt the current behavior whenever it is necessary, depending on unforeseen detected events. Our different experiments up to now have shown that this balancing mechanism works well. Such intrinsic instability within the architecture should not be judged too severely, since we introduced intentionally asynchronism and non-determinism in order to be able to cope with unforeseen and changing situations: solving the frame problem and the brittleness problem cannot be done without introducing some structural instability, but one should take care to provide some means to control that unstability as best as possible.

As a consequence of what has been discussed up to now, a most interesting property is the fact that the robot can control its own autonomy by relying on external features, such as perceptual landmarks that can trigger special events or behaviors once they have been detected by the perception processes, transformed into representations and interpreted by the attention manager.

A typical straightforward application consists in an autonomous robot looking for traffic signs along the road, that limit its freedom of decision. A similar example is an autonomous robot, to which instructions are given for instance through sign language or even speech (speech processing algorithms are then needed among the perception processes), which yield either information on the environment or provide orders to be followed, like modification of the current behavior.

Such ability to switch from autonomous to semi-autonomous modes is from our point of view a true sign of computational intelligence: while autonomy is often conceived for an artificial system as not needing external rules to choose its own behavior, being able to follow suddenly rules while you were evolving freely just relying on yourself is a proof of intelligent design!... and a prerequisite for robots to be accepted among humans.

More elaborate control modes of the robot's own autonomy obviously can be conceived which do not necessarily rely on a priori semantics to be discovered by the perception processes. Such less explicit control of autonomy is exerted in a multirobot scenario [16], where we have implemented several robots with the previous architecture after having provided it with an additional communication process which allows the exchange of individual representations among different robots upon request. The autonomy of every robot is then implicitly limited by the autonomy of its partners, as the completion of the task is evaluated at the group's level and what is actually developed is the multirobot group's autonomy.

4 Computational Intelligence and Learning

In previous work [9,10,14], we have presented methodologies that allow to learn the sensorimotor lower-level loop. They proceed in two phases: identification of a qualitative model of the dynamical system and inversion of that model to deduce

a fuzzy-like control law. Such research has shown that furthermore higher-level representations could emerge relative to the topology of the environment or the sequencing of behaviors [12,17].

It would be interesting to see where and how similar learning mechanisms could be introduced within the control architectures. The lower reactive level which binds the perception and the action selection modules have already been dealt with as was recalled previously. Of course it is always possible to improve the already available methods: for instance, learning control laws up to now does not include any explicit prediction, as there is no quantitative analysis or comparison of the actual effect of the input with respect to the predicted effect. If this was achieved, it would be possible to include corrective terms within the learned control law (as is done in predictive feedforward control) and to have reasoning mechanisms on the consistency of the identified qualitative model relatively to the observed effects (techniques on qualitative model verification from [8] could be used for that purpose).

But it would also be interesting to add learning mechanisms at the representation level, especially concerning their update: as in biological systems, forgetting or short-term and long-term memory mechanisms could be implemented).

Concerning the introduction of learning mechanisms within action selection algorithms, the literature is rather rich [15], and most techniques described in the reference could be advantageously implemented in our work.

On the contrary, introducing learning at the attention manager level is an open problem: it could be done in order to interpret either the awaited events as a function of the current behavior and the potential behaviors that could follow it, or unexpected events that can trigger emergency answers. Another direction could be to have adaptive sequencing of the behaviors.

5 Conclusion

We have discussed robot control architectures and proposed an architecture which aims at filling the gap between reactive behavioral and deliberative decision systems, while keeping a close eye on the dynamic management of all the resources available to the robot.

We have also shown how it was possible to propose various autonomy modes, since the autonomy of the robot can be easily controlled by feeding the environment the landmarks that, after interpretation, provide selective behaviors.

Finally, we have discussed the introduction of learning techniques at the various levels of the architecture, and we have argued how such learning mechanisms, if successfully implemented, could yield not only adaptive, but also evolutive control architectures. All of this would then achieve a degree of autonomy, about which we may only dream for now...

References

1. R.C. Arkin. *Behavior-based robotics*. A Bradford Book, The MIT Press, 1998.

2. A. Berthoz. *Le sens du mouvement*. Éditions Odile Jacob, 1997.
3. M.A. Boden, editor. *The philosophy of artificial life*. Oxford Readings in Philosophy, 1996.
4. A. Dalgalarondo and D. Luzeaux. Rule-based incremental control within an active vision framework. In *4th Int. Conf. on Control, Automation, Robotics and Vision*, Westin Stamford, Singapore, 1996.
5. S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
6. P.J. Hayes. The frame problem and related problems in artificial intelligence. In J. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 588–595. Morgan Kaufmann Publishers, Inc., 1990.
7. S.M. Kosslyn. *Image and Brain*. A Bradford Book, The MIT Press, 1992.
8. B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with incomplete knowledge*. The MIT Press, 1994.
9. D. Luzeaux. Let's learn to control a system! In *IEEE International Conference on Systems Man Cybernetics*, Le Touquet, France, 1993.
10. D. Luzeaux. Learning knowledge-based systems and control of complex systems. In *15th IMACS World Congress*, Berlin, Germany, 1997.
11. D. Luzeaux. Mixed filtering and intelligent control for target tracking with mobile sensors. In *29th Southeastern Symposium on System Theory*, Cookeville, TN, USA, 1997.
12. D. Luzeaux. Catastrophes as a way to build up knowledge for learning robots. In *16th IMACS World Congress*, Lausanne, Switzerland, 2000.
13. D. Luzeaux and A. Dalgalarondo. Assessment of image processing algorithms as the keystone of autonomous robot control architectures. In J. Blanc-Talon and D. Popescu, editors, *Imaging and Vision Systems: Theory, Assessment and Applications*. NOVA Science Books, New York, 2000.
14. D. Luzeaux and B. Zavidovique. Process control and machine learning: rule-based incremental control. *IEEE Transactions on Automatic Control*, 39(6), 1994.
15. P. Pirjanian. Behavior coordination mechanisms: state-of-the-art. Technical report, University of Southern California, TR IRIS-99-375, 1999.
16. P. Sellem, E. Amram, and D. Luzeaux. Open multi-agent architecture extended to distributed autonomous robotic systems. In *SPIE Aerosense'00, Conference on Unmanned Ground Vehicle Technology II*, Orlando, FL, USA, 2000.
17. O. Sigaud and D. Luzeaux. Learning hierarchical controllers for situated agents. In *Proceedings of the 14th International Congress on Cybernetics*, Bruxelles, Belgium, 1995.