CONSTRAINT-SATISFYING SYSTEMS FOR PERCEPTUAL ROBOTICS

Alan K. Mackworth

Laboratory for Computational Intelligence, Department of Computer Science
University of British Columbia, Vancouver, B.C. V6T 1Z4, Canada
E-mail: mack@cs.ubc.ca, WWW: www.cs.ubc.ca/spider/mack

Abstract

We need methods for designing and building perceptual robotic systems that are uniform, clean, powerful and practical. None of the current methodologies satisfy all these criteria. The traditional approach has encountered the intractable engineering problems of building systems based on a hybrid on-line/off-line paradigm. I argue for a new situated agent approach based on the Constraint Net model of Zhang and Mackworth, which allows the designer to specify the robot's vision, control and motor systems uniformly as on-line systems. If the perceptual and control systems are designed as constraint-satisfying devices then the total robotic system, consisting of the robot symmetrically coupled to the environment, can be proven correct. The task of soccer was proposed as a motivational application task. Several perceptual robotic systems designed and built using this approach are described.

1 Introduction

We need practical and formal design methodologies for building integrated perceptual agents. Here I argue for a formal approach to the emerging synthesis, Situated Agents. The approach is based on the view that any agent or robot is a hybrid intelligent dynamical system, consisting of a controller coupled to its plant. The robot is symmetrically coupled to a dynamic environment.

Similarly, knowledge-based image interpretation needs to be re-interpreted. The traditional Good Old-Fashioned Artificial Intelligence and Robotics (GOFAIR) approach proposes that domain-specific knowledge is used by the robot/agent at run-time to disambiguate the retinal array into a rich world representation. The argument is that the impoverishment and ambiguity of the visual stimulus array must be supplemented by additional knowledge. This approach has failed to make substantial progress for several reasons. One difficulty is the engineering problem of building robots by integrating off-line knowledge-based vision systems with on-line control-based motor systems. Especially in active vision systems [2] this integration is difficult, ugly and inefficient [8]. Because of such objections, some in the AI-robotics community have rejected the knowledge-based approach adopting instead an *ad hoc* Gibsonian situated approach to perception that exploits regularities of the particular environmental niche of the robot [6, 1, 5, 4]. In [9], I argued that, with a radical re-interpretation of 'knowledge-based', we *can* design, build and verify quick and clean knowledge-based situated robot vision systems.

2 The Design Problem

The robot design problem is formidable, regardless of whether the robot is designed or modified by a human, by nature (evolution), by another robot (bootstrapping), or by itself (learning). A robot is, typically, a hybrid intelligent system, consisting of a controller coupled to its plant. The controller and the plant each consist of discrete-time, continuous-time or event-driven components operating over discrete or continuous domains. The controller has perceptual subsystems that can (partially) observe the state of the environment and the state of the plant.

Robot design methodologies are evolving dialectically [8]. The symbolic methods of GOFAIR constitute the original thesis. The antithesis is reactive Insect AI. The emerging synthesis, Situated Agents, has promising characteristics, but needs formal rigor and practical tools. The critiques and rejection, by some, of the GOFAIR paradigm have given rise to the Situated Agent approaches of Rosenschein and Kaebling [10], Brooks [4], Ballard [2], Winograd and Flores [12], Lavignon and Shoham [7], Zhang and Mackworth [14, 8] and many others.

3 The Robot Soccer Challenge

Theory is vacuous without an appropriate application to drive designs, experiments and implementations. In 1992, I proposed robot soccer as a grand challenge problem for the field [8] since it has the task characteristics that force us to deal all these issues in a practical way for a percepetual, collaborative, real-time task with clear performance criteria. At the same time, I described the first system for playing robot soccer. Since then it has been a very productive environment both for our laboratory [3, 11, 9, 17, 19, 18] and for many other groups around the world, stimulating research toward the goal of building vision and robotic systems that are uniform, clean, powerful and practical.

4 Constraint Nets

The Constraint Net (CN) model [15] is a formal and practical model for building hybrid intelligent systems as Situated Agents. In CN, a robotic system is modelled formally as a symmetrical coupling of a robot with its environment. Even though a robotic system is, typically, a hybrid dynamic system, its CN model is unitary. Most other robot design methodologies use hybrid models of hybrid systems, awkwardly combining off-line computational models of high-level perception, reasoning and planning with on-line models of low-level sensing and control.

CN is a model for robotic systems software implemented as modules with I/O ports. A module performs a transduction from its input traces to its output traces, subject to the principle of causality: an output value at any time can depend only on the input values before, or at, that time. The model has a formal semantics based on the least fixpoint of sets of equations [15]. In applying it to a robot operating in a given environment, one separately specifies the behaviour of the robot plant, the robot control program, and the environment. The total system can then be shown to have various properties, such as safety and liveness, based on provable properties of its subsystems. This approach allows one to specify and verify models of embedded control systems. Our goal is to develop it as a practical tool for building real, complex, sensor-based robots. It can be seen as a development of Brooks' subsumption architecture [4] that enhances its modular advantages while avoiding the limitations of the augmented finite state machine approach.

Although CN can carry out traditional symbolic computation on-line, such as solving Constraint Satisfaction Problems and path planning, notice that much of the symbolic reasoning and theorem-proving may be outside the agent, in the mind of the designer. GOFAIR does not make this distinction, assuming that such symbolic reasoning occurs explicitly in, and only in, the mind of the agent.

The question "Will the robot do the right thing?" [14] is answered if we can:

1. model the coupled robotic system at a suitable level of abstraction,
2. specify the required global properties of the system's evolution, and
3. verify that the model satisfies the specification.

In CN the modelling language and the specification language are totally distinct since they have very different requirements. The modelling language is a generalized dynamical system language. Two versions of the specification language, Timed Linear Temporal Logic [17] and Timed $\forall$-automata [13], have been developed with appropriate theorem-proving and model-checking techniques for verifying systems.

5 Constraint-Satisfying Controllers

Many robots can be designed as on-line constraint-satisfying devices [13, 16, 17]. A robot in this restricted scheme can be verified more easily. Moreover, given a constraint-based specification and a model of the plant and the environment, automatic synthesis of a correct constraint-satisfying controller becomes feasible, as shown for a simple ball-chasing robot in [17].

A constraint is simply a relation on the phase space of the robotic system, which is the product of the controller, plant and environment spaces. A controller is defined to be *constraint-satisfying* if it, repeatedly, eventually drives the system into an ϵ-neighborhood of the constraint using a constraint satisfaction method such as gradient descent.

A constraint-satisying controller may be *hierarchical* with several layers of controller above the plant. In this case each layer must satisfy the constraints appropriate to the layer, defined on its state variables. The layers below each layer present to that layer as a virtual robot plant in a suitably abstract state space [17, 18].

6 Soccer-Playing Robots

The ideas presented here have been developed and tested by application to the challenge of designing, building and verifying active perception systems for robot soccer players with both off-board and on-board vision systems.

In the Dynamo (Dynamics and Mobile Robots) Project in our laboratory, we have experimented, since 1991, with multiple mobile robots under visual control. The Dynamite testbed consists of a fleet of radio-controlled vehicles that receive commands from a remote computer. Using our custom hardware and a distributed MIMD environment, vision programs are able to monitor the position and orientation of each robot at 60 Hz; planning and control programs generate and send motor commands at the same rate. This approach allows umbilical-free behaviour and very rapid, lightweight fully autonomous robots. Using this testbed we have demonstrated various robot tasks [3], including playing soccer [11].

One of the Dynamo robots, Spinoza, is a self-contained robot consisting of an RWI base with an RGB camera on a pan-tilt platform mounted on top and binocular monochrome stereo cameras in the body. As a illustration of these ideas consider the task for Spinoza of finding, tracking and chasing a soccer ball, using the pan-tilt camera. After locating the moving ball Spinoza is required to move to within striking distance of the ball and maintain that distance. The available motor commands control the orientation of the base, the forward movement of the base, and the pan and tilt angles of the camera. The parameters can be controlled in various relative/absolute position modes or rate mode. The available rate of pan substantially exceeds the rate of body rotation. A hierarchical constraint-based active-vision controller can be specified for Spinoza that will achieve and maintain the desired goal subject to safety conditions such as staying inside the soccer field, avoiding obstacles and not accelerating too quickly. If the dynamics of Spinoza and the ball are adequately modelled by the designer then this constraint-based vision system will be guaranteed to achieve its specification.

Yu Zhang and I have recently extended these ideas to build multi-layer constraint-satisfying controllers for a complete soccer team [19]. The controllers for our new softbot soccer team, UBC Dynamo98, are modeled in CN and implemented in Java, using the Java Beans architecture [18]. They control the soccer players' bodies in the Soccer Server developed by Noda Itsuki for RoboCup. This experiment provides evidence that CN is a uniform, clean, powerful and practical design framework for perceptual robots.

7 Conclusions

I have described the motivation, some results and some current directions of a long-term project intended to develop a new approach to the specification, design, implementation and evaluation of robotic systems. The practical result is a new methodology for building intelligent, perceptual systems.

8 Acknowledgments

I am grateful to Rod Barman, Cullen Jennings, Stewart Kingdon, Jim Little, Valerie McRae, Don Murray, Dinesh Pai, Michael Sahota, Vlad Tucakov, Ying Zhang and Yu Zhang for help with this. This work is supported, in part, by the Natural Sciences and Engineering Research Council of Canada and the Institute for Robotics and Intelligent Systems Network of Centres of Excellence

References

[1] R. L. Andersson. *A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control.* MIT Press, Cambridge, MA, 1988.

[2] D. H. Ballard. Reference frames for active vision. In *Proceedings IJCAI-89*, pages 1635–1641, Detroit, MI, 1989.

[3] R. A. Barman, S. J. Kingdon, J. J. Little, A. K. Mackworth, D. K. Pai, M. Sahota, H. Wilkinson, and Y. Zhang. DYNAMO: Real-time experiments with multiple mobile robots. In *Intelligent Vehicles Symposium*, pages 261–266, Tokyo, July 1993.

[4] R. A. Brooks. Intelligence without reason. In *IJCAI-91*, pages 569–595, Sydney, Australia, August 1991.

[5] D. Chapman. Vision instruction and action. Technical Report MIT AI TR-1085, MIT, Cambridge, MA, June 1990.

[6] I. D. Horswill and R. A. Brooks. Situated vision in a dynamic world: Chasing objects. In *AAAI-88*, pages 796–800, St. Paul, MN, 1988.

[7] J. Lavignon and Y. Shoham. Temporal automata. Technical Report STAN-CS-90-1325, Stanford University, Stanford, CA, 1990.

[8] A. K. Mackworth. On seeing robots. In A. Basu and X. Li, editors, *Computer Vision: Systems, Theory, and Applications*, pages 1–13. World Scientific Press, Singapore, 1993.

[9] A. K. Mackworth. Quick and clean: Constraint-based vision for situated robots. In *IEEE Int'l. Conf. on Image Processing*, pages 789–792, Lausanne, Switzerland, September 1996.

[10] S. J. Rosenschein and L. P. Kaelbling. The synthesis of machines with provable epistemic properties. In Joseph Halpern, editor, *Proc. Conf. on Theoretical Aspects of Reasoning about Knowledge*, pages 83–98. Morgan Kaufmann, Los Altos, CA, 1986.

[11] M. Sahota and A. K. Mackworth. Can situated robots play soccer? In *Proc. Artificial Intelligence 94*, pages 249 – 254, Banff, Alberta, May 1994.

[12] T. Winograd and F. Flores. *Understanding Computers and Cognition.* Addison-Wesley, Reading, MA, 1986.

[13] Y. Zhang and A. K. Mackworth. Specification and verification of constraint-based dynamic systems. In A. Borning, editor, *Principles and Practice of Constraint Programming*, number 874 in Lecture Notes in Computer Science, pages 229 – 242. Springer-Verlag, 1994.

[14] Y. Zhang and A. K. Mackworth. Will the robot do the right thing? In *Proc. Artificial Intelligence 94*, pages 255 - 262, Banff, Alberta, May 1994.

[15] Y. Zhang and A. K. Mackworth. Constraint Nets: A semantic model for hybrid dynamic systems. *Theoretical Computer Science*, 138:211 - 239, 1995.

[16] Y. Zhang and A. K. Mackworth. Constraint programming in constraint nets. In V. Saraswat and P. Van Hentenryck, editor, *Principles and Practice of Constraint Programming*, chapter 3, pages 49–68. The MIT Press, Cambridge, MA, 1995.

[17] Y. Zhang and A. K. Mackworth. Synthesis of hybrid constraint-based controllers. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems II*, Lecture Notes in Computer Science 999, pages 552 - 567. Springer Verlag, 1995.

[18] Yu Zhang and A. K. Mackworth. A constraint-based controller for soccer-playing robots. In *IROS'98*, Victoria, BC, October 1998. To appear.

[19] Yu Zhang and A. K. Mackworth. A multi-level constraint-based controller for the dynamo98 robot soccer team. In *ROBOCUP98*, Paris, France, July 1998. Springer-Verlag. To appear.