Dynamite: A Testbed for Multiple Mobile Robots[1]

R. A. Barman, S. J. Kingdon, A. K. Mackworth,
D. K. Pai, M. K. Sahota, H. Wilkinson and Y. Zhang[2]

Laboratory for Computational Intelligence
Department of Computer Science
University of British Columbia
Vancouver, B.C.
Canada, V6T 1W5

Abstract

We are experimenting with multiple mobile robots under visual control. The current goal of our work is to construct an architecture for robots engaged in cooperative and competitive behaviour. Multiple robots have recently been the focus of much attention; however, the little work that has been done with implemented systems has involved simple tasks and behaviours. We have initiated the Dynamo (Dynamics and Mobile Robots) Project to provide a link between theory and practice. One part of this project is the Dynamite testbed which consists of a fleet of radio controlled vehicles that receive commands from a remote computer. Robot position and orientation is determined using off-board visual sensing. We have chosen soccer playing as a domain for our experiments since it requires real-time interaction with a dynamic environment. It involves inter-robot cooperation as well as competition between teams. We outline two complementary approaches taken in our laboratory to robot control. The first, Constraint Nets, is a model for robotic systems and behaviours, which provides a theoretical foundation for systems design and analysis. The second places the emphasis in robot control on dynamic action selection; current functionality includes path planning and motion control algorithms.

1 Introduction

Dynamo (Dynamics and Mobile Robots) is an umbrella project in our laboratory for research in real-time control of mobile robots. Projects within Dynamo include systems with on-board and off-board vision.[3] In this paper, we focus on the Dynamite testbed which is based on off-board vision and off-board computation. Some possible experiments using the testbed include chasing, pushing, and soccer-playing. We have concentrated on the control of robots playing soccer. A game of soccer provides a complex and dynamic environment which challenges current approaches to robot control. Robots need to cooperate with team members as well as compete with robots on the opposing team.

We are currently attempting to have teams of robots engaged in cooperative and competitive behaviour. Recently, there has been considerable interest in multiple robots. There have been a number of experiments in this area, but they have been restricted to simple tasks and cooperative behaviour [Kub92; Nor92; Mat92]. Competition between an agent and a hostile environment

---

[1] To Appear in *Proceedings of the IJCAI-93 Workshop on Dynamically Interacting Robots*

[2] Alphabetical order

[3] Other work in the Dynamo Project can be found in [BKL+93]

has been addressed in [AC87]; however, this work involved a software agent operating in a simulation, not in the real world.

In Section 2, we describe the Dynamite testbed. It consists of a fleet of radio controlled vehicles that receive commands from a remote computer. In an integrated environment with dataflow and MIMD computers, vision programs can monitor the position and orientation of each robot at 60 Hz; planning and control programs can generate and send out motor commands at 50 Hz. This approach allows umbilical-free behaviour and very rapid, lightweight fully autonomous robots. As far as we know, it is a unique and successful approach to the trade-offs involved in mobile robot design although a related scheme was independently proposed in [Hal91].

In Section 3, two approaches taken in our laboratory to robot control are outlined. The Constraint Net (CN) approach developed by Zhang and Mackworth is a formal model for robotic systems and behaviours. CN provides a denotational semantics for real-time programming languages and for the overall behaviour of robotic systems. The desired behavior can be verified using a real-time temporal logic. The approach taken by Sahota employs behaviour based control with an emphasis on dynamic action selection. The mechanism for determining action is based on inter-behaviour and inter-robot bidding.

This paper is a description of work in progress. We hope to have teams of robots playing soccer intelligently. However, in the current state, individual robots are able to plan and track paths; robot functionality also includes shooting the ball at the opposing player's net and preventing balls from going in one's own net. Two robots can compete in a one-on-one game of soccer.

2 Dynamite: a Testbed for Experiments with Soccer-Playing Robots

In this section, we will describe the Dynamite testbed that we have developed and its usefulness in multiple robot experiments. The soccer domain allows us to explore how competition and cooperation can be accommodated and supported by robot architectures.

The mobile robot bases are commercially available radio controlled vehicles. We have six 1/24 scale racing-cars, each 22 cm long, 8 cm wide, and 4 cm high excluding the antenna. We have driven these cars under control at speeds of 140 cm/s.[4] The soccer field (244 cm by 122 cm in size) with the six cars and a ball is shown in Figure 1. The ball is the small object in the middle of the image; the two cars on the right have each been fitted with two circular colour markers allowing the vision system to identify their position and orientation. The robots are neither as flexible nor as competent as human soccer players. As a result, we have modified the environment in two ways. First, there is a wall around the soccer field which prevents the ball (and the players!) from going out of bounds. Second, there are barriers to prevent the ball from getting trapped in the corners. Since these are Canadian robots, it is not unreasonable for the soccer field to be shaped like an ice hockey rink.

The hardware used in this system is shown in Figure 2. There is a single colour camera mounted in a fixed position above the soccer field. The video output of the camera is transmitted

---

[4] This, while equivalent to a scale speed of 120 km/h, is far below the maximum speed of the vehicle which is approximately 6 m/s (or scale 500 km/h).

Figure 1. Robot Players Inhabiting the Soccer Field

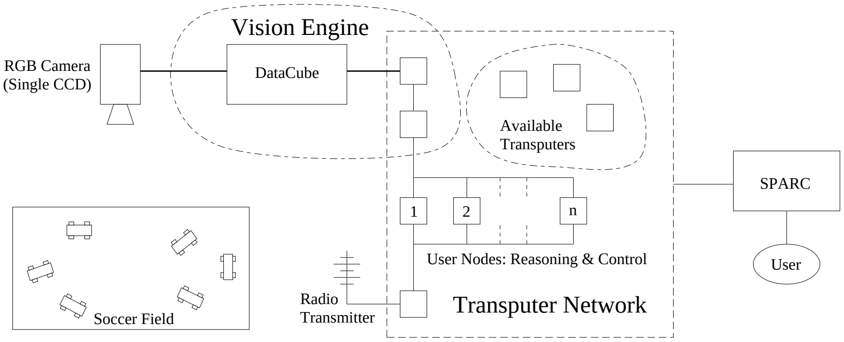

Figure 2. Hardware Setup

to special-purpose video processing hardware[5] named the DataCube in Figure 2. The DataCube is a dataflow computer which has been programmed to classify image pixels into different colour classes at video rate. This information is transmitted to a network of transputers[6] which form a MIMD computer. Additional vision processing is performed on the transputers to find the position, in screen coordinates, of the centroid of each coloured blob and to transform these positions from screen to world coordinates. The entire vision subsystem is called the Vision Engine [LBKL91]. The Vision Engine produces the absolute position and of all the objects on the soccer field. The orientation is also produced for the cars, but not the ball. This is done at 60 Hz with an accuracy in position near 1 mm.

The reasoning and control components of a vehicle can be implemented on any number of transputers out of the available pool. Currently, each vehicle is controlled by a user program running on its own transputer node[7]. An arbitrary number of nodes, labeled 1 to n in Figure 2, can be used in parallel to control n independent vehicles. The movement of all vehicles is controlled through a radio transmitter attached to a single transputer node. Commands are transmitted to the vehicles at a rate of 50 Hz.[8] System users are able to install their own planning and control routines and simply link in to the vision and transmitter systems. The operation of each user program does not effect the operation of the others. User programs can compete and cooperate with one another as well as communicate through message passing.

One of the advantages of the environment we have built is that we are able to provide a clean user interface. The user program which is to perform reasoning, planning and control is shielded from the some of the complexities in the world. The input to the user program is a sequence of vectors describing the locations of all the objects. The output from the user program is a sequence of control signals (“throttle” and “steering angle”) sent to the vehicle. We believe that our interface is of significant benefit to the user. It allows one to focus on making the robot “do the right thing” instead of worrying about implementation details. It is possible to create

---

[5] There are two boards. The Digicolour board converts analog video to a digital signal, while the MV–200 is a software reconfigurable video processing board.

[6] The network consists of a MAXTRAN transputer for communicating with the DataCube and 16 T800’s whose interconnection topology is configured in software.

[7] One plan for system development involves using two transputers: one for on-line control, and the other for off-line deliberation

[8] This is the rate at which commercial R/C equipment works. It does not match the video rate. However, this does not have a significant impact on the controllability of the vehicles since it only causes a slight increase in the latency of the servos on the vehicle.

Figure 3. Two Dynamo Vehicles: Zeno (front) and Heraclitus (rear)

this interface since we have a robust vision system and a reconfigurable network of off-board computers. Another benefit is that we are able to connect a user program to either a simulator or the real system. One of the simulators was developed by collecting statistics on the performance of a vehicle under various control signals. It has proved to be an invaluable tool in developing user programs for control.

We have worked with 1/10 scale radio controlled trucks, called Clod Busters, as well as 1/24 scale racing cars. In Figure 3 a racing car, Zeno[9], is in the foreground and a Clod Buster, Heraclitus, is in the background. A 12 inch (30 cm) ruler is also visible. Clod Busters have also been used successfully for soccer-playing[10] and for other Dynamo projects. They are large enough to support on-board video and computers; this opens a new range of experiments such as indoor and outdoor navigation.

3 Approaches

In this section we describe two complementary approaches we are taking to robot control: Constraint Nets (CN) and Dynamic Action Selection.[11] We also describe the progress we have made using the two approaches. CN provides a theoretical foundation for systems design and analysis. The dynamics and control for each robot, as well as the interaction between robots, can be modeled using CN. In the subsection on Dynamic Action Selection, the emphasis is on the control of individual robots and not on cooperation schemes. We feel that the range of possible interactions between robots depends on the underlying architecture and robot functionality. This motivates the discussion of our approaches to building individual robots.

3.1 Constraint Nets

CN is a formal model for robotic systems and behaviours. CN is composed of modules with I/O ports [ZM92b]. A module defines a transduction which is a mapping 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 language has a formal semantics based on the least fixpoint of sets of equations [ZM92a]. 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 integrated system consisting of plant, control, and environment can then be shown to have various properties, such as safety and liveness. This approach allows one to specify formally, 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 [Bro88] that enhances its modular advantages while avoiding the limitations of the augmented finite state machine approach.

---

[9] The names are inspired by Monty Python's soccer-playing philosophers sketch. Zeno and Heraclitus were particularly concerned with dynamic worlds.

[10] In fact, the Clod Buster was the only vehicle used for soccer-playing until recently. The impetus for moving to smaller vehicles was to operate more vehicles in the same workspace.

[11] These approaches do not represent the views of all the authors.

A robotic system is modeled as three machines: the robot plant, the robot control and the environment. Each is modeled separately as a dynamic system by specifying a CN with identified input and output ports. The robot is modeled as a CN consisting of a coupling of its plant CN and its control CN by identifying corresponding input and output ports. Similarly, the robot CN is coupled to the environment CN to form a closed robot-environment CN.

CN provides a denotational semantics for real-time programming languages and for the overall behaviour of robotic systems. Moreover, a real-time temporal logic has been developed for specifying the desired properties of the situated robot. The relationship between a constraint net model of a robotic system and a temporal logic specification of a desired behavior can be verified. So far, we have been able to specify, design, verify and implement systems for a robot that can track other robots [ZM92b], a robot that can escape from mazes, a two-handed robot that assembles objects [ZM92c], and an elevator system [ZM93]. For the Dynamite project we have designed a simulation of the robot plant in CN and implemented a controller for the robot in CN that can plan and execute paths. 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. Good Old Fashioned Artificial Intelligence and Robotics (GOFAIR) does not make this distinction, assuming that such symbolic reasoning occurs explicitly in, and only in, the mind of the agent [Mac93].

3.2 Dynamic Action Selection

An agent cooperating and competing with other agents in a changing world must behave appropriately. We agree with Brooks [Bro91]: “Intelligence is determined by the dynamics of interaction with the world.” It is not sufficient for an agent to select an action (or activity) intelligently. An agent must also ensure that any actions it is performing are intelligent given *the current situation.*

It is not sufficient for a disembodied off-line planner to send plans to a low-level controller. Rather, the deliberative component (traditionally a planner) must react to changes in the world as they happen; dynamic action selection is needed. Maes stated in [Mae90] the need for such a system, and argued that there are trade-offs involved in action selection. We disagree with the notion of trade-offs. It is desirable to have an agent evaluate its goals and the expected utility of its actions as often as possible. The only way to do the right thing in an unpredictable world is to look at it and decide — not to pursue one goal singlemindedly.

We propose a two layer architecture for a robot controller. The lower layer is composed of a set of task-following modules, each of which can interact with the environment to perform a well defined task or activity. This follows from similar work by [Fir92]. The deliberative layer of our controller is composed of *behaviour* producing modules. The behaviour based approach is common to much of the work on the situated agents. However, we feel that the current mechanisms for arbitrating among behaviours are inadequate. Our method for dynamic action selection is based on inter-behaviour bidding, an approach more general than other approaches [Bro88; KR90].

The main idea behind this theory of action selection is that each behaviour is best able to identify how applicable it is in a given situation. Each behaviour independently evaluates the world and reports a utility estimate or bid to the other behaviours. The behaviour with the highest bid assumes control since it has the greatest utility. This follows from the theory of *drives* in psychology [Tyr93]. The allowable bid ranges are set by the designer, just as the hierarchy of behaviours is set in the Subsumption Architecture. One advantage of a this scheme is that the bid ranges can be set at run-time or even changed on the fly in a system which learns. To clarify, there is little relation between our approach and other systems like Contract Nets [Smi80; Nor92] where there is negotiation between behaviours.

This theory of action selection can be extended to multiple robots. In this case, each robot would broadcast its intended actions and a bid which estimates the appropriateness of that action. Robots whose actions are in conflict will reevaluate the situation including the bid information of other robots. Robots with lower bids than other robots will lower their internal bid for that action which will result in the selection of some other action.

The effectiveness of the lower layer of our architecture has been demonstrated using the Dynamite testbed. Controller functionality includes motion planning, ball shooting and playing goal. Two robots can (and frequently do) compete in a one-on-one game of soccer. The controller for each robot alternates between deliberation (action selection and planning) and execution (pursuing a specific action). The functionality of the executive layer and the usefulness of the Dynamite testbed has been established. We have a video which shows the testbed and documents the two robots playing soccer. The full advantages gained from this approach will be demonstrated (very soon!) when controller performs simultaneous deliberation and execution. We expect to have two teams with three robots on each side playing soccer in the near future.

4 Conclusion

One goal of the Dynamo Project is to explore cooperation and competition among multiple robots. We have developed a flexible environment for experiments with multiple radio controlled vehicles. This is a unique and successful approach for multiple robot experiments. Much of our work is focused on the soccer world, although the Dynamite testbed can be used for other applications. Currently, only path planning, motion tracking, and some simple activities have been implemented. We have performed experiments with competition between two robots and soon hope to have teams of competing robots.

5 Acknowledgments

We are grateful to Keiji Kanazawa, Jim Little, and Dan Razzell for help with this. This work is supported, in part, by the Canadian Institute for Advanced Research, the Natural Sciences and Engineering Research Council of Canada and the Institute for Robotics and Intelligent Systems Network of Centres of Excellence.

References

[AC87] Philip Agre and David Chapman. Pengi: An implementation of a theory of activity. In *AAAI-87*, pages 268–272, 1987.

[BKL$^+$93] R. Barman, S. Kingdon, J. Little, A. K. Mackworth, D.K. Pai, M. Sahota, H. Wilkinson, and Y. Zhang. Dynamo: real-time experiments with multiple mobile robots. In *Proceedings of Intelligent Vehicles Symposium*, pages 261–266, 1993.

[Bro88] R. A. Brooks. *A robot that walks: emergent behaviors from a carefully evolved network*. Massachusetts Institute of Technology, Cambridge, MA, 1988.

[Bro91] R. A. Brooks. Intelligence without reason. In *IJCAI-91*, pages 569–595, Sydney, Australia, August 1991.

[Fir92] R. James Firby. Building symbolic primitives with continuous control routines. In *First International Conference on Artificial Intelligence Planning Systems*, pages 62–69, 1992.

[Hal91] John Hallam. Playing with toy cars: An experiment in real-time control. Technical Report DAI Research Paper No. 527, Edinburgh University, 1991.

[KR90] L. P. Kaelbling and S. J. Rosenschein. Action and planning in embedded agents. In P. Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 35–48. MIT Press, Cambridge, MA, 1990.

[Kub92] Claus Ronald Kube. Collective robotic intelligence: A control theory for robot populations. Master's thesis, University of Alberta, 1992.

[LBKL91] J. Little, R. Barman, S. Kingdon, and J. Lu. Computational architectures for responsive vision: the vision engine. In *Proceedings of Computer Architectures for Machine Perception*, pages 233–240, 1991. Paris.

[Mac93] 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.

[Mae90] Pattie Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, pages 49–70. M.I.T. Press, 1990.

[Mat92] Maja Mataric. Minimizing complexity in controlling a mobile robot population. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 1992.

[Nor92] Fabrice Noreils. An architecture for cooperative and autonomous mobile robots. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, 1992.

[Smi80] G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computing*, 29(12), 1980.

[Tyr93] Toby Tyrrell. *Computational Mechanisms for Action Selection*. PhD thesis, Edinburgh University, 1993.

[ZM92a] Y. Zhang and A. K. Mackworth. Constraint nets: A semantic model for real-time embedded systems. Technical Report TR 92-10, UBC, Vancouver, B.C., May 1992.

[ZM92b] Y. Zhang and A. K. Mackworth. Modeling behavioral dynamics in discrete robotic systems with logical concurrent objects. In S. G. Tzafestas and J. C. Gentina, editors, *Robotics and Flexible Manufacturing Systems*, pages 187–196. Elsevier Science Publishers B.V., 1992.

[ZM92c] Y. Zhang and A. K. Mackworth. Will the robot do the right thing? Technical Report TR 92-31, UBC, Vancouver, B.C., November 1992.

[ZM93] Y. Zhang and A. K. Mackworth. Design and analysis of embedded real-time systems: An elevator case study. Technical Report TR 93-4, UBC, Vancouver, B.C., February 1993.