

TPN
A 549

一般不引用

8863612



**A Robot Ping-Pong Player:
Experiment in
Real-Time Intelligent Control**



E8963612

Russell L. Andersson
AT&T Bell Laboratories



The MIT Press
Cambridge, Massachusetts
London, England

PUBLISHER'S NOTE

This format is intended to reduce the cost of publishing certain works in book form and to shorten the gap between editorial preparation and final publication. Detailed editing and composition have been avoided by photographing the text of this book directly from the author's prepared copy.

© 1988 AT&T Bell Laboratories

Published by The MIT Press, Cambridge, Massachusetts.

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

This book was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Andersson, Russell L.

A robot ping-pong player.

(The MIT Press series in artificial intelligence)

Bibliography: p.

Includes index.

1. Robotics. 2. Artificial intelligence. 3. Table tennis. 4. Real-time data processing.
I. Title. II. Series.

TJ211.A53 1987 629.8'92 87-32526

ISBN 0-262-01101-8

**A Robot Ping-Pong Player:
Experiment in
Real-Time Intelligent Control**



The MIT Press Series in Artificial Intelligence

Edited by Patrick Henry Winston and Michael Brady

Artificial Intelligence: An MIT Perspective, Volume I: Expert Problem Solving, Natural Language Understanding, Intelligent Computer Coaches, Representation and Learning edited by Patrick Henry Winston and Richard Henry Brown, 1979

Artificial Intelligence: An MIT Perspective, Volume II: Understanding Vision, Manipulation, Computer Design, Symbol Manipulation edited by Patrick Henry Winston and Richard Henry Brown, 1979

NETL: A System for Representing and Using Real-World Knowledge by Scott Fahlman, 1979

The Interpretation of Visual Motion by Shimon Ullman, 1979

A Theory of Syntactic Recognition for Natural Language by Mitchell P. Marcus, 1980

Turtle Geometry: The Computer as a Medium for Exploring Mathematics by Harold Abelson and Andrea diSessa, 1981

From Images to Surfaces: A Computational Study of the Human Early Visual System by William Eric Leifur Grimson, 1981

Robot Manipulators: Mathematics, Programming, and Control by Richard P. Paul, 1981

Computational Models of Discourse edited by Michael Brady and Robert C. Berwick, 1982

Robot Motion: Planning and Control edited by Michael Brady, John M. Hollerbach, Timothy Johnson, Tomás Lozano-Pérez, and Matthew T. Mason, 1982

In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension by Michael G. Dyer, 1983

Robotics Research: The First International Symposium edited by Michael Brady and Richard Paul, 1984

Robotics Research: The Second International Symposium edited by Hideo Hanafusa and Hirochika Inoue, 1985

Robot Hands and the Mechanics of Manipulation by Matthew T. Mason and J. Kenneth Salisbury, Jr., 1985

The Acquisition of Syntactic Knowledge by Robert C. Berwick, 1985

The Connection Machine by W. Daniel Hillis, 1985

Legged Robots that Balance by Marc H. Raibert, 1986

Robotics Research: The Third International Symposium edited by O. D. Faugeras and Georges Giralt, 1986

Machine Interpretation of Line Drawings by Kokichi Sugihara, 1986

ACTORS: A Model of Concurrent Computation in Distributed Systems by Gul A. Agha, 1986

Knowledge-Based Tutoring: The GUIDON Program by William Clancey, 1987

AI in the 1980s and Beyond: An MIT Survey edited by W. Eric L. Grimson and Ramesh S. Patil, 1987

Visual Reconstruction by Andrew Blake and Andrew Zisserman, 1987

Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence by Yoav Shoham, 1988

Model-Based Control of a Robot Manipulator by Chae H. An, Christopher G. Atkeson, and John M. Hollerbach, 1988

A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control by Russell L. Andersson, 1988

To Amy With Love

SERIES FOREWORD

Artificial intelligence is the study of intelligence using the ideas and methods of computation. Unfortunately, a definition of intelligence seems impossible at the moment because intelligence appears to be an amalgam of so many information-processing and information-representation abilities.

Of course psychology, philosophy, linguistics, and related disciplines offer various perspectives and methodologies for studying intelligence. For the most part, however, the theories proposed in these fields are too incomplete and too vaguely stated to be realized in computational terms. Something more is needed, even though valuable ideas, relationships, and constraints can be gleaned from traditional studies of what are, after all, impressive existence proofs that intelligence is in fact possible.

Artificial intelligence offers a new perspective and a new methodology. Its central goal is to make computers intelligent, both to make them more useful and to understand the principles that make intelligence possible. That intelligent computers will be extremely useful is obvious. The more profound point is that artificial intelligence aims to understand intelligence using the ideas and methods of computation, thus offering a radically new and different basis for theory formation. Most of the people doing artificial intelligence believe that these theories will apply to any intelligent information processor, whether biological or solid state.

There are side effects that deserve attention, too. Any program that will successfully model even a small part of intelligence will be inherently massive and complex. Consequently, artificial intelligence continually confronts the limits of computer science technology. The problems encountered have been hard enough and interesting enough to seduce artificial intelligence people into working on them with enthusiasm. It is natural, then, that there has been a steady flow of ideas from artificial intelligence to computer science, and the flow shows no sign of abating.

The purpose of this MIT Press Series in Artificial Intelligence is to provide people in many areas, both professionals and students, with timely, detailed information about what is happening on the frontiers in research centers all over the world.

Patrick Henry Winston
Michael Brady

Preface

The work described in this book is the result of a sequence of experiments that began with the idea of building an honest real-time (60 Hz) vision system. The sight of a slick, high-tech robot plodding along at a snail's pace has always rubbed me the wrong way. John Jarvis originally pointed out that moments were a simple enough, and well enough understood, computer vision operation that we really ought to be able to do them faster. They seemed well suited to making a vision system that was fast enough to serve as a sensor for a fast robot, even if the scenes had to be quite simple. I set off to make a vision system that would process simple scenes quickly, rather than process more complex scenes slowly, as did the majority of the vision and robotics community. I could then investigate tightly coupled robot/vision systems.

Two years later, I had a working vision system, and in the meantime had created a system to control a PUMA 260 robot in C. Suddenly, the need for a demonstration of the potent capabilities of the work was upon me. For reasons no longer clear, but doubtless related to the simplicity of the resulting image, I threw together a system to catch a ping-pong ball rolling in 2-D across a table. The ping-pong ball had the advantage of being very white, and the ball would break rather than allow the robot to come to harm.

After a subsequent explosion in the complexity of the code, the robot caught most of a certain class of trajectories with high reliability. It would even catch balls bounced off a piece of foam shortly before they were to be caught, amply demonstrating the true real-time nature of the system. It was clear, however, that catching the remainder of the balls would require more subtle strategies not easily captured in my program — it was likely that even trying would cause the whole thing to collapse under its own weight. Nonetheless, the demonstration was a vast success, and the volume of demonstrations prevented me from doing much additional work on the system, affording me the time for thought.

Since I could catch in two dimensions, it only stood to reason that I should be able to do it in three. About this time, I heard of the challenge/contest announced by Professor John Billingsley of Portsmouth Polytechnic (U.K.) for robot ping-pong, and since I had already thought along these lines, Professor Billingsley's proposal provided an excuse for further ruminations. (Many thanks to Professor Billingsley for his well-thought-out rules.)

Some time later, I heard a rumor that Dan Kodeschek of Yale had thought of catching objects. I put off all consideration of ping-pong, and rushed to create a 3-D catcher, since I knew it should be straightforward. The resulting system proved able to catch balls in a styrofoam coffee cup, exhibiting some spectacular behavior due to primitive error recovery code.

By this time, however, it was clear that I was going about this entirely the wrong way. Alarmingly, it appeared that if I attempted to extrapolate the system to full ping-pong, it was never going to work. There was not going to be a magic algorithm that could compute an ideal ping-pong player's response, as had been the case (at least in practice) in the previous demonstrations. Trying to address the different cases was going to result in a nonfunctional morass; the line between when to apply one strategy and when to apply another was thin, bordering on nonexistent.

What I had to do was to capture the subjective skills of a trained human. A person's skilled techniques appeared then, and now, equally applicable to a person playing ping-pong, or to a person working on an assembly line. In both cases, people make many clever motions, without seeming to expend much thought, and one can improve their performance by giving small hints such as "keep your elbow up." By contrast, a robot would need a whole new algorithm.

This book describes an experiment to try to capture this type of skilled behavior, resulting in a robot ping-pong player which can play and beat human players. Along the way to a working system, I discovered and had to solve many interesting problems. The vision system, the expert robot controller, the low-level robot controller, and the overall system design were significantly affected by the need to make the system respond accurately in a dynamic environment. By working only with static problems, the vision and robotics communities have not yet had to understand dynamic environments.

Accordingly, this book begins to pave the way. At least one chapter is devoted to each of the four major subsystems above; each chapter describes what the subsystem must accomplish, how it does so and why, and quantitatively documents its performance. Although the book has the technical content of a work oriented at robotics professionals; computer scientists, physicists, hobbyists, and the technically curious should be able to read the book and gain insight into how the system works and is put together — and an appreciation of how good the human machine really is.

The management of AT&T Bell Laboratories, and especially John Jarvis, deserve immense credit for seeing the technical merit behind a project of little instantaneous gain. I think this characteristic reflects extremely well on the quality of the institution.

I am indebted to many people for their aid in the successful completion of this work. Many thanks to Professor Richard Paul (University of Pennsylvania), for his support, encouragement, immense practical experience, and especially, wise judgement. To Professor Ruzena Bajcsy (Penn) I owe the emphasis of carrying through ideas to working results, thus eliminating the need and possibility of trying to justify ideas by hand-waving. Professor Rodney Brooks and Professor Tomás Lozano-Pérez (both of MIT) have my thanks for their suggestions on how to improve the book's readability.

Thanks to George Whyte and Richard Seide, the able mechanical and electronic construction crew. Thanks to Ray "Plot" Soneira and his indomitable `ray_plot`, which supported the computer graphics in this book. I am indebted to the creators of the MEGLOS multiprocessor system and the MULGA VLSI design system: Bob Gaglianello, Howard Katseff, Krish Ramakrishnan, Beth Robinson, Brian Ackland, Jay O'Neill, and Neil Weste. Thanks are also due my other colleagues at the Labs who were always ready with ideas of strategies and effects the system ought to deal with, and put up with the sound of ball against table, paddle, and wall, and the screams from man and machine as the latter attempted autorobocide at spectacular speeds. Thanks for hanging in there, King Pong.

Finally, many, many thanks to my wife, Amy, who provided marathon-level encouragement, and mental, moral, physical, and logistical support.

Holmdel, New Jersey
November, 1987

R.L.A.

**A Robot Ping-Pong Player:
Experiment in
Real-Time Intelligent Control**

Contents

| | |
|--|-----------|
| Series Foreword | x |
| Preface | xi |
| Chapter 1 Introduction | 1 |
| 1.1 Introduction to Robots and Their Limitations | 1 |
| 1.2 Introduction to Robot Ping-Pong | 5 |
| 1.3 Why Ping-Pong is a Good Problem | 8 |
| 1.4 A Preview of the Work | 9 |
| 1.5 Implications and Application Areas | 11 |
| 1.6 Organization of This Book | 12 |
| Chapter 2 Robot Ping-Pong | 13 |
| 2.1 Robot Ping-Pong Rules | 13 |
| 2.2 Comparison to Human Ping-Pong | 15 |
| 2.3 Ping-Pong Physics | 17 |
| Chapter 3 System Design | 27 |
| 3.1 Task Partitioning | 27 |
| 3.2 Processors and Network | 29 |
| 3.3 Achieving Accurate Timing | 30 |
| 3.4 Data Logging System | 32 |
| 3.5 Real-Time Debugger | 36 |
| 3.6 Summary | 38 |
| Chapter 4 Real-Time Vision System | 39 |
| 4.1 Current Vision Techniques | 39 |
| 4.2 Basic Approach to the Vision System | 41 |
| 4.3 Moment Generator | 42 |
| 4.4 Stereo and Calibration Techniques | 48 |
| 4.5 Imaging Dynamic Scenes | 59 |
| 4.6 Three-Dimensional Trajectory Analysis | 65 |
| 4.7 Vision System Accuracy | 68 |
| 4.8 Summary | 73 |

| | |
|--|------------|
| Chapter 5 Robot Controller | 75 |
| 5.1 Robot Mechanism | 75 |
| 5.2 Controller Electronics Architecture | 77 |
| 5.3 Robot Software Architecture | 80 |
| 5.4 Basic Elements | 81 |
| 5.5 Trajectory Generation | 87 |
| 5.6 Trajectory Following: Prediction and Control | 104 |
| 5.7 Robot Controller Performance | 107 |
| 5.8 Summary | 109 |
| Chapter 6 Expert Controller Preliminaries | 111 |
| 6.1 What is an Expert Controller? | 111 |
| 6.2 Rule-Based Systems | 112 |
| 6.3 Actual Architecture | 116 |
| 6.4 Model Data Structure | 120 |
| 6.5 Summary | 129 |
| Chapter 7 Expert Controller | 131 |
| 7.1 Functionality Required | 131 |
| 7.2 Initial Planning | 132 |
| 7.3 Temporal Updating | 139 |
| 7.4 Exception Recovery | 147 |
| 7.5 Learning? | 154 |
| 7.6 Summary | 159 |
| Chapter 8 Robot Ping-Pong Application | 161 |
| 8.1 Program Overview | 161 |
| 8.2 Examination of Program Components | 163 |
| 8.3 Execution Case Studies | 189 |
| 8.4 Implementation Limitations | 197 |
| 8.5 Summary | 203 |
| Chapter 9 Conclusion | 205 |
| 9.1 Synopsis | 205 |
| 9.2 Future Research Topics | 206 |
| 9.3 Man versus Machine | 207 |
| Appendix 1: Robot — The Official Rules | 209 |
| Appendix 2: Vision Performance Plots | 213 |
| Appendix 3: Robot Performance Plots | 225 |
| Appendix 4: Ping-Pong Top Level | 237 |

| | |
|--|------------|
| Appendix 5: Initial Planning and Updating | 241 |
| Appendix 6: Flight Path Correction | 251 |
| Appendix 7: Acceleration Limit Correction | 253 |
| References | 261 |
| Index | 269 |

Chapter 1

Introduction

"A Robot Ping-Pong Player: Experiment in Real-Time Intelligent Control" analyzes the construction of the first robot to play ping-pong* against humans and machines (Figure 1). What are the limitations of present robots? What makes ping-pong hard? Why investigate ping-pong? What are the applications of the technology we developed? The following introductory chapter will address these issues, then provide an overview of the book's organization.

1.1 Introduction to Robots and Their Limitations

The general public thinks of R2D2 and C3PO when someone mentions robots, but today's robots are far from that stage, mechanically or intellectually. When confronted with a real robot, the mechanism makes the biggest impression. However, in our (admittedly biased) view, it is not the mechanism that is of greatest interest, but the computer system and its programming that controls the mechanism. Figure 2 divides the robot control system into three levels. Each level maintains a specific duration of foresight, with a corresponding constraint on the time available for planning.

The robot mechanism occupies the hierarchy's lowest level: motors, encoders, and their projections into the electronic domain. Specialized joint servo processors interact with the interfaces of each joint, implementing control algorithms that cause the robot to attain specified positions, velocities, or torques. The actuator bandwidths restrict the available processing time to a millisecond per iteration. The joint servos

* Ping-Pong is a trademark of Parker Brothers.

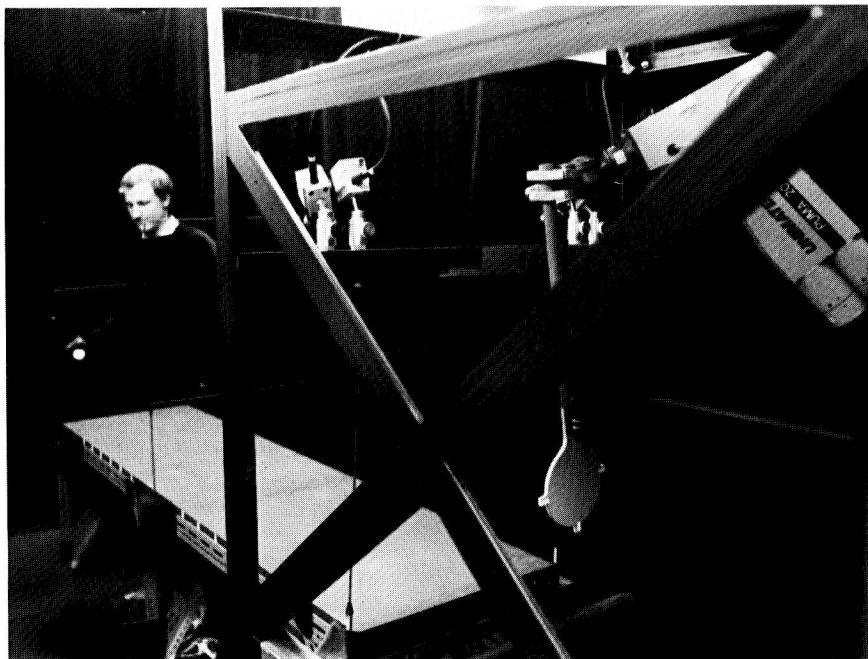


Figure 1. Robot and Environment. The robot ping-pong player takes on a human opponent, the author, at robot ping-pong. Two of four video cameras and the robot arm lurk in the supporting framework.

present a clean electronic interface to the middle control level, which can specify desired robot trajectories.

At the hierarchy's summit, a high-level planner generates objectives for the robot and monitors its performance. The planner manipulates an abstract view of the robot system, even though it may supply details such as the sizes of parts from a CAD/CAM database. The planner is slower than and desynchronized from robot operation, but must be able to monitor and modify the robot system to improve its performance. Either a program or a human may implement the high-level planner.

The intermediating controller translates the planner's demands to the servo controller's realities. The controller must plan a sequence of trajectories for the servos which satisfy both the abstract requirements of

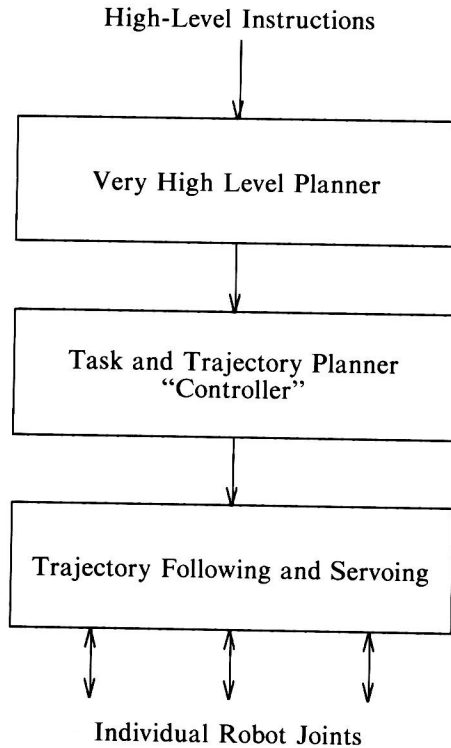


Figure 2. High-Level Robot Diagram. The planner is an artificial or natural intelligence system. We will be concerned with the controller, which must integrate the task and robot constraints.

the task and the complex and concrete limitations of the physical apparatus. In the face of ignorance, the controller has no choice but to adopt very conservative estimates of the manipulator's abilities. Note that the generic term "robot controller" refers to both the task and trajectory planner, and the lower-level servoing functions.

Consider a conventional robot feeding a punch. The high-level planner might produce this plan: