Proceedings

1999 IEEE INTERNATIONAL CONFERENCE ON
ROBOTICS
AND AUTOMATION

May 10-15, 1999
Marriott Hotel, Renaissance Center
Detroit, Michigan

Sponsored by
IEEE Robotics and Automation Society

Volume 2

Pages 831-1664

Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

Keeping the Analog Genie in the Bottle: A Case for Digital Robots

Ian D. Walker

Electrical and Computer Engineering
Clemson University
Clemson, SC 29634
ianw@ces.clemson.edu

Joseph R. Cavallaro
Martin L. Leuschen
Electrical and Computer Engineering
Rice University
Houston, TX 77005
{cavallar, martinl}@rice.edu

Abstract

In this paper, we consider the case for adopting a truly 'digital' type of robot, which would evolve between a discrete and finite set of states. We adopt the point of view that the advantages of traditional robotic evolution (over the full range of a continuous domain) are often negated by complexities associated with the continuous world. The types of discrete robots discussed in this paper would keep this unwanted continuous-time 'genie' in a 'box' of discrete 'steps' during its operation. One distinct advantage of this philosophy is that a formal logical analysis can then be applied to the digital robots, since discrete-time models now correctly and completely model the robot behavior. We argue that there are significant benefits to this strategy in numerous cases, especially with respect to fault detection and fault tolerance. However, there are also disadvantages - in order to guarantee digital behavior, constraints on the robot's operations are imposed. Essentially, we gain formality of digital analysis at the expense of precision of continuous movement. Using an analogy to digital electronics, we discuss ways in which the development of digital robots could revolutionize certain aspects of robotics.

1 Introduction

Robotics is a fascinating and important example of a truly interdisciplinary field. Today, input from specialists from many different backgrounds is required to produce successful robots. The differing backgrounds of traditional robot scientists has led to an apparently natural dispersion of techniques among the different sub-areas in the field.

Mechanical and control engineers for example, traditionally design and analyze robots based on a good understanding of the (analog) hardware and (continuous) equations of motion of robot components and systems. However, the resulting designs often owe much to the experience and tastes of the particular design team, are extremely complex to model well, and it is not easy to analyze the effects of even small design changes. This leads to many 'one-of-a-kind' robot types, and makes it difficult to extrapolate experience gained in one robot application to others. This in turn hinders the transition of new results.

Computer scientists, on the other hand, would often like to see completely discrete models for robots, and often develop their algorithms as if robots truly were digital in nature [13]. However, the continuous-time nature of actual robots often 'foils' these algorithms (unexpected continuous 'behavior' between the logical steps undermines the assumptions made), which leave an important part of the physical situation unaddressed. This often frustrates the attempt to effectively apply high-level algorithms in robotics.

Although there are areas, notably robot path and task planning, where logical and continuous algorithms each contribute significantly, most areas in robotics are dominated by analysis using one or the other form of analysis. While this appears natural in most cases, a key difficulty in robot systems integration is the synergy between the different types of algorithms.

There have been numerous efforts in the use of purely analog techniques to develop high-level analysis and simulation tools for robotic and mechatronic systems (see for example [21]). However, it is difficult to incorporate 'events' (such as changes in system goals due to completion of subtasks, etc.) which happen at discrete times into an inherently analog framework. Additionally, since virtually all robots are controlled by digital computers which communicate with the analog 'world' via A/D and D/A conversion, there is a fundamental issue as to how truly 'analog' is a real robot system anyway? Care has to be taken with purely analog analysis (for example with stability proofs of controller designs based on continuous mathematics) to ensure that the effects of sampling, etc. are taken

0-7803-5180-0-5/99 $10.00 © 1999 IEEE

into account.

However, the traditional purely 'analog' approach is still the most widely used in robotics. This is in no small part due to practitioners' familiarity and comfort with, and the necessity of considering (for traditional robots), the inherently continuous nature of the kinematic and dynamic models of robots, and the analog nature of most conventional actuator and sensor systems. This is probably appropriate for many traditional cases, where the behavior of the system over a time period of interest can be described by a continuous-time system with fixed structure.

However, there are a great many situations of current interest where this description is insufficient. For robot tasks such as walking, grasping and regrasping, contact and impact with the environment, etc., the system equations change structurally at discrete (possibly unpredictable) instants of time. This is also often the case when faults occur in the robot system, or when the overall plan changes or moves on to a different phase. In all these cases, a different continuous-time model is needed to describe the modified system following each of the discrete events. The purely continuous time analysis, while valuable and critical, and the focus of almost all traditional applications, can be seen as a subset of the overall (logical) progression of the robot system, in which events at *discrete* times are seen to logically describe the evolution of the system.

The above limitations of purely continuous analysis have led to much interest recently in the notion of 'Hybrid' systems [7]. In such systems, both discrete (logical) and continuous (analog system) analyses are included, connected by an appropriate 'interface' layer. This seems to be an appropriate approach, since it explicitly considers the hybrid nature of the real system, and has been adopted in various forms by a large number of investigators. However, even in this case, if the nature of the underlying robot system is dynamically tightly coupled, this typically causes inherent difficulty in the interplay between the discrete and continuous levels [2]. In this paper we argue that, just as there are many cases where a purely analog analysis can be appropriate, there are strong motivations for a purely discrete approach in many interesting and significant examples.

We propose in this paper the adoption of 'digital robots' to address these issues. These digital robots would make the jump from the analog to digital world by operating between a specified set of discrete states (i.e. the joint or configuration space would be restricted to a finite set of states). Existing examples of such 'digital' devices in robotics include pneumatic drives (typically operated in 'bang-bang' fashion) [23]. Other such inherently digital devices are actuators based on levitation in magnetic fields, and others such as those proposed for actuation and assembly in micro- and nano- robot technology [3, 19, 26].

Notice however that in the above definition we are not restricted to 'bang-bang' types of actuators for these digital robots. The joints themselves could still be analog in nature. A 'digital' robot can be practically realized by driving each joint by stepper motors [26], where the operational space of the robot in the end effector locations corresponds to all combinations of 'steps' of the motors at the joints via the forward kinematics. Alternatively, a continuous actuator can be turned into a precise two-state actuator by replacing position feedback by a mechanical constraint and overpowering it, to reach two states under load [6].

The idea of restricting the motions of robots to a finite number of states is not new [18, 20], and there is existing theory that could be utilized by future developers of such devices. For example, there are established results for analyzing how forward [14] and inverse [6] kinematics can be computed for digital, or binary manipulators. There is also existing work in workspace analysis of such robots [4, 22]. The work in [6, 14] was motivated by specific questions resulting from analysis of hyperredundant Variable Geometry Truss manipulators. However, we feel that the benefits of such a strategy can be more fundamental and far-reaching. In the following sections, we discuss the motivation for, and possible advantages of, these 'digital' robots.

Of course, we are not suggesting that all robots should suddenly be restricted to operate over a finite set of states. There are many applications (high precision cutting, for example) where a very precise continuous path needs to be followed by the robot, and a restriction to (even a very high number of) discrete states for the end effector would be unacceptable. However, we argue that there are significant cases where such a conversion to inherently digital robots makes a lot of sense. An analogy, which we will develop further in the following sections, is that a large number of electronic systems are, and should remain, inherently analog. However, a conversion in the electronics industry of many areas from analog to digital electronics 30 years ago has brought about a major change in the way that we live our lives today. In the following sections, we pose the question of what analogous benefits might be gained by the robotics community from a similar philosophy.

2 Motivation for Digital Robots

If the concept of digital robots has been around for a long time, why are we writing this paper at this time? A primary motivation for the authors was the belief that, although there is a significant amount of scientific and technological innovation taking place in robotics, there remain significant fundamental barriers to progress.

Essentially, robotics is still in its infancy, relative to where we believe the industry can, and should, go. Perhaps a good analogy is to the automobile industry before the Model T Ford. There are a great many 'one of a kind' or 'several of a kind' robots, and in general, the art of the designer rules. Each new robot system is essentially built from scratch, with experience of the design team key. There is a general lack of a set of 'design rules' for robot design.

This situation is not without advantages - the model allows for individual innovation at all levels, but hampers large scale commercialization since there is no formal way for ideas and standards to transition across the industry. New innovations are usually built 'from the ground up', rather than being transitioned into an existing framework. Users usually need to be experts to extract the intended performance from a robot system. The situation is somewhat analogous to desktop computing before the Apple Macintosh defined a new generation of machines, which while not as versatile as contemporary PC's, were more user-friendly (perhaps only to the non-expert!) and bulletproof, and effectively changed the face of the genre.

As further perspective, consider the scenario of the evolution of the field of Computer Science. As noted in [13], in the early stages of the field, the technological components of computers were considered the prerequisites for typical courses, followed by programming, and then, eventually, abstract algorithms. However, it is now widely recognized that algorithms form the core of Computer Science [13], and the emphasis has moved away from the technology towards the issue of how best to utilize it. This has parallels to the state of robotics today, where the focus (both in the classroom and in the research arena) often tends to be more on the technology itself than how to best apply it.

Sometimes, however, the best way to apply a technology may not be directly obvious. Consider the example of the electronics industry, which initially was a purely analog domain (the fundamental equations are governed by (continuous-time) physics, as is the case for robotics), but has evolved into a domain (digital electronics) which is beyond any reasonable expectation of most practitioners 50 years ago. In the process, the face of most other technologies have been changed significantly. In the following, we further pursue this analogy between the robotics and electronics areas, in order to discuss in more detail the issues underlying digital robots.

2.1 Analogy to Digital Circuit Design and Application

Recent advances in VLSI design have allowed for large levels of integration of simple devices, the most basic of which is the two transistor CMOS inverter. The basic inverter takes a single input and produces the logical complement of the signal. A logic symbol and circuit diagram are shown in Figure 1 and the resulting ideal inverting logic behavior is shown in Figure 2. The two transistor CMOS inverter is actually a rather complex circuit and many parameters, including relative transistor sizes, input waveform, and output load, may affect its behavior. Traditional analog circuit design and analog VLSI design [9] are concerned with the operation of these circuits under a rather unstructured variety of conditions. This design methodology, although quite general, does not quickly lead to high levels of circuit integration.

Digital VLSI design is based on a combination of circuit abstractions and input/output restrictions. The switch level abstraction [10, 15, 16, 24, 27]. allows for the design of more complex digital elements, such as NAND and NOR gates, and also for the interconnection of multiple levels of logic to build structures, such as shift registers and adders. This switch level model treats each transistor as either an 'on' or 'off' switch and therefore abstracts away more complex circuit behavior. Computer-aided design tools can then handle large simulations of millions of transistors effectively.

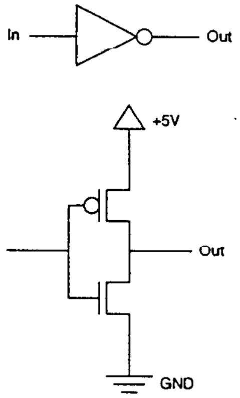

Figure 1: Basic Digital Inverter Circuit Diagram

In order to restrict VLSI circuits to operate in a digital manner, some care must be taken in ensuring the proper input signal levels and control clocking schemes. These noise margins [8, 29] put bounds on subsystem interfaces by ensuring that logic '0' values be close to '0' volts and logic '1' values be close to the power supply voltage, for example 5 volts. Also, system clock signals must be operated at a proper frequency so that all digital signals have sufficient time

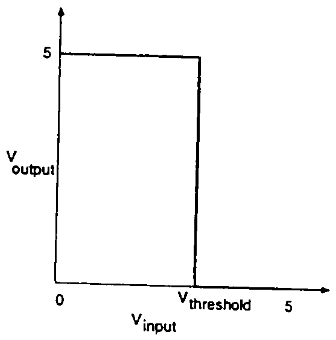

Figure 2: Basic Digital Inverter Input Output Relationship

to rise or fall. In the following sections, we will relate the digital behavior of the inverter and shift register to the movement of a stepper motor.

3 Application of Digital Robots to Today's Problems with Existing Technology

3.1 Existing Digital Robots: Stepper Motor Robots

One already existing type of robot which is inherently digital, in the sense of the above discussion, is one whose joints are driven by stepper motors [26]. Stepper motors differ from conventional motors in that, through the use and excitation of multiple coils in the motor, the rotor is attracted to exactly one of a finite number of positions, depending on the input. They are thus only able to assume discrete stationary angular positions. Figure 3 shows the operation of one step of the motor [25] as the coil 'B' is energized. An analogy can be made between the threshold voltage that causes a VLSI inverter to switch in Figure 2 and the appropriate threshold current I_B in Figure 4 that exceeds static friction and causes the motor to turn one step. Although somewhat different, an electrical and mechanical analogy applied to robot control [1] has been presented.

Stepper motors are used extensively in many applications, notably in the computer peripheral area (disk drives, etc.). Examples of robots driven by stepper motors include the Merlin robot by American Robot Corporation [11].

Stepper motors have a number of advantages over conventional motors, including simplicity of construction and control [11]. In addition, stepper motors have advantages with respect to reliability - a common failure

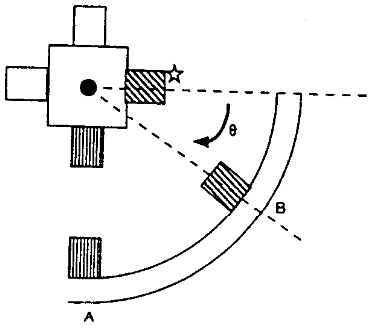

Figure 3: Mechanical Stepper Motor Sector

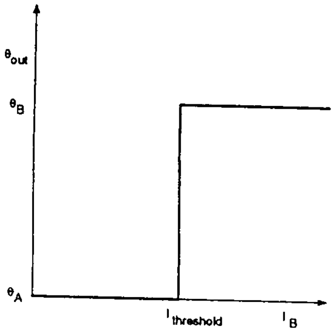

Figure 4: Mechanical Stepper Single Step Response

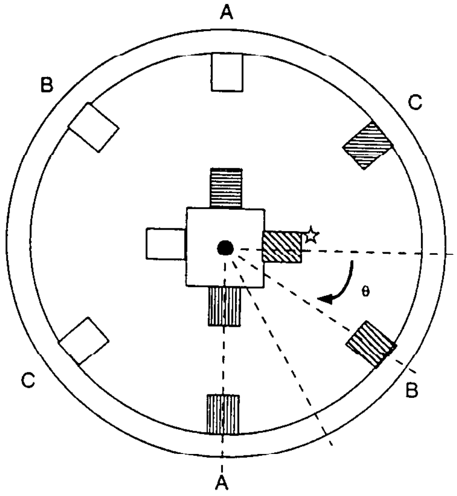

Figure 5: Complete Mechanical Stepper Motor Configuration

mode for a motor drive power amplifier is for an output transistor to short from collector to emitter, applying full voltage to the motor armature, resulting in a runaway condition in a conventional motor. For a stepper motor however, this will cause the motor to move one step and hold [11]. Thus the digital nature of the stepper motor provides certain inherent advantages.

Disadvantages of stepper motors include restrictions on the torque and speed available (in order to keep the motors from being backdriven or from 'overstepping'). However, the problem of overstepping can be alleviated by the addition of a position sensor.

Practically, this means that to guarantee that stepper motors remain in the digital mode, a user would have to specify operating conditions for each joint (maximum speed, torque, etc. to keep the stepper in its nominal mode and not slipping or missing steps) and this would probably be a non-trivial calculation for each given organization of joints, i.e. robot type.

This is an inevitable consequence of restriction of physical devices to a digital mode. However, this is pushing the difficulty from on-line to off-line (i.e. provided the system is operated within the operating conditions, it should predictably be in a specified (one of a finite number of) configuration). Deviations could only be due to a fault somewhere. This is analogous to saying that provided the operating conditions of a chip or array of electronic modules are proper (voltages at appropriate levels, signals of a low enough frequency for complete logic propagation), the system should be at one of a (possibly large) set of states which are encoded with a series of 0's and 1's.

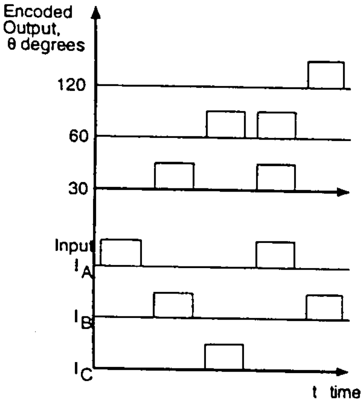

Figure 6: Stepper Motor Rotation Scenario

The analogy between digital circuits and stepper motor operation can be further extended by considering a simulation of stepper motor operation and comparing it with the behavior of a digital logic shift register. Figure 6 shows several steps in the operation of the motor and plots the input current waveform and an encoding of the stepper motor output position, θ, as a function of time. The motor begins at 0 degrees, and then moves to 30, 60, 90 degrees, and so on.

A similar digital circuit would be the shift register, drawn in a ring with buffer circuits and pass transistors in Figure 7. Upon the assertion of the *Reset* signal, a logic '1' is loaded and appears at output zero, O_0 This logic bit, which compares to the motor shaft position, can then be moved around the 'circle' by successive control signals, V_A, V_B, V_C and so on. The control signal will need to be presented at the appropriate frequency and the bit can rotate in a similar manner to the stepper motor, as seen in Figure 8.

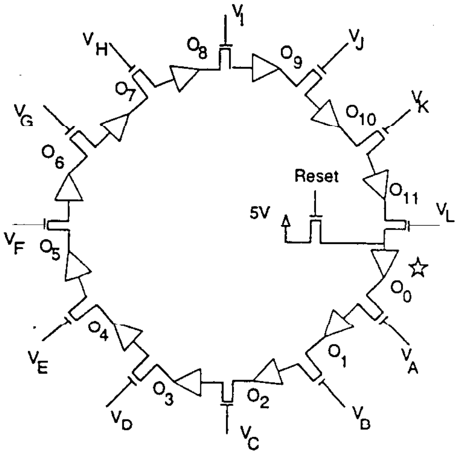

Figure 7: Digital Buffer Shift Register Analogy

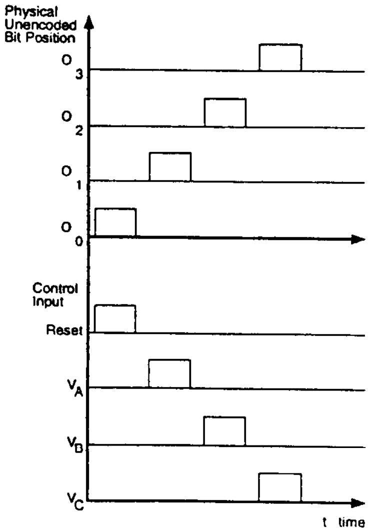

Figure 8: Shift Register Data Movement Scenario

3.2 Some Benefits of Digital Robots: Fault Detection

Some advantages of using digital robots are immediate. First, the nature of control of the devices changes completely, from being a complex issue involving significant analysis, into what becomes essentially a counting operation between discrete steps. Notice that for digital robots of the type discussed here, this is truly a correct model, not an approximation. Basically, the difficulties of the continuous-time dynamics are removed from the user and transformed into the 'operating conditions' for the device.

This has much wider benefits than simply making control simpler. For conventional robots, the analog nature of their operation means that a whole host of other issues are directly affected by unexpected variations such as unmodeled dynamics, etc.. For example, fault detection for conventional manipulators is a non-trivial task. In order to generate effective fault detection tests, thresholds taking into account unmodelable variations in the continuous-time world must be derived [28]. These thresholds are difficult to find, and it has not yet proved possible to formally prove the effectiveness of existing robot fault detection strategies. Fundamentally the problem is that errors from the continuous-time dynamic world 'leak' into the logic of fault detection, preventing a purely logical analysis.

However, for the digital robot, these problems disappear. For example, for a stepper motor (operated in digital mode, i.e. under speeds and loading such that steps are made and not missed), the model of joint evolution is (precisely) given as:

$$x(k+1) = x(k) + hv(k) \tag{1}$$

$$y(k) = x(k) \tag{2}$$

where $x(k)$ is the state of the joint (in number of steps) at time k, h is the (time) step size between k and $k+1$, and $v(k)$ is the joint velocity at time k (in number of steps per unit time h). The joint is assumed to have a position sensor (measuring the number of steps) with output $y(k)$.

For this discrete time system, the classical approach of Analytical Redundancy [5] can be applied to formally find a fault detection test (residual), which is given as:

$$0 = P(k) = [(y(k+1) - y(k))/h] - v(k+1) \tag{3}$$

This is hardly surprising, (the residual simply states that the difference in the number of motor steps between successive time steps should be the commanded number), but consider the parallel for the traditional case. In that situation, due to the inherent fluctuation in the continuous signals involved, the corresponding residual $P(k)$ is never truly zero, and a threshold needs to be devised to account for 'normal' deviations from zero. Thus, when a fault occurs, its effect on $P(k)$ must be separated from the noise. This is complex and non-trivial [28].

In the discrete-time case, $P(k)$ is an integer operation, the equation holds exactly in normal operation, non-zero $P(k)$ implies a fault condition, and no thresholds are required. Thus the analysis is reduced from a very complex one to an almost trivial case. In addition, the transfer to the digital world has resulted in linear dynamics for the joint. This allows us to use the wide range of linear methods (such as Analytical Redundancy) which constitute the vast majority of formal fault detection approaches in the literature (in previous attempts to 'prove' fault detection for robotics, linearized models have been used). Thus the switch to the digital domain has resulted in a linear model that *correctly* describes the evolution of the system, and thus allows rigorous analysis.

We thus in this case have achieved the ability to formally give a logical 'proof' for fault detection (trivial in this case, simple checking for non-zero $P(k)$), due to the reformulation of the robot motion problem from the continuous to discrete case. Essentially, due to the digital operation, the inherent modeling errors are removed to (and kept within) a lower level, allowing us to reconstruct the important information at the digital level. The situation is somewhat analogous to that of the switch in music technology from analog LP's to digital CD's. If an LP was scratched, this imperfection necessarily affected the sound as a fault, since the whole range of the continuous domain was being used for sound reconstruction information. However, for the CD, a surface scratch is less critical, since provided the information decoder can still distinguish 'zeros' from 'ones', the originally intended sound can be exactly reconstructed.

In a similar fashion a digital robot would be encoding its 'key information' digitally, and thus providing a natural barrier to 'analog noise'. This is clearly highly beneficial for fault detection, but also for a whole host of other tasks at a 'logical level', where currently formal analysis is prevented by uncertainty due to analog level uncertainty.

4 Discussion and Conclusions

While there has been much progress in robotics in the last few years, the field is still essentially in its infancy. Although current research is adding important core scientific and engineering results and understanding to the area, the field still largely consists of small groups of experts working with 'one of a kind' or 'several of a kind' devices. There is as yet no significant 'consumer robotics' or truly large scale commercial robotics products. The robot equivalent of the Model T Ford or the Apple Macintosh Computer does not yet exist.

While it is clearly important to have a core group of experts at the cutting edge of research, in a mature industry it should not be necessary to be an expert to develop or use the basic technology, as tends to be the case in robotics at present. Ways need to be found to expand robotics into a major industry, with 'many-of-a-kind' devices based on consistent industry-wide specifications. Digital robots, as suggested in this paper, offer a way to achieve this, for many potential as yet unexplored applications. Potential applications are wide-ranging, with notable exceptions being where exceptionally high precision, and probably speed, is required. For example, the toy industry is an area with a massive market, which would love to have a profitable series of robotic toys. For toys, high speed and high precision are much less important than in traditional robot applications, and thus digital robots could find a very profitable niche in this arena. Other existing industries, such as service robotics, could also benefit significantly from adoption of digital robots.

The digital robots proposed in this paper would be simpler to model, more modular and interchangeable, and more predictable and amenable to diagnosis than current 'analog' robots. The price paid for this would be the imposition of appropriate constraints on the operation of the robot in order to guarantee digital operation, in a similar fashion to digital electronics. In this way, the paradigm would be transferring the complexity from the practitioner (the user) to the developer (the specialist) - note that in traditional robotics these two are usually the same -, who would have to specify the constraints under which the robot would operate in its predictable (digital) range.

There are strong potential 'big wins' for both practitioner and developer. The practitioner would benefit from being able to exploit the relative simplicity and modularity of the resulting devices. It would be possible to compare and contrast robots developed at different locations, and initially for quite different purposes. Through the existence of a consistent set of 'digital specifications', a formal 'robot store' would come into existence, from which practitioners could choose robots and/or components. Practitioners would not need to know lower-level control or dynamics details, merely the needs of their application and the operational specifications of each candidate robot/subsystem.

For the developer, thanks to the ability to formally categorize the 'digital' subsystems (joints, etc.), it would now become feasible to formally discuss the design, construction, and analysis of robots via the chaining together of modular digital subsystems. Standard 'benchmarks' for robot operation could be set up, leading to a formal 'robot toolkit'. It would now be possible to determine the feasibility of matching systems developed at different places, initially for quite different purposes. A common set of specifications for digital operation (essentially input/output specifications) of robot modules, which would be inherent in the digital robotics technology, would also provide developers with a consistent basis with which to contrast and compare their products. Thus it would be easier to market new robots across the field, with obvious benefits for both developer and practitioner.

There are also benefits for researchers. For many robotics researchers whose primary focus is not on analog issues, the digital robots would provide effective research platforms without the need for writing or modifying low-level planning and control algorithms. There would also of course still be a significant need for 'traditional analog' robotics expertise and research. New research results at the analog (physics-based) level which could improve the performance of robot modules (improve the 'digital specifications') would be instantly applicable. A successful market for digital robotics would provide a significantly increased ability for many researchers to transfer their algorithms and devices to a much wider audience than at present.

From a fundamental research point of view, a series of fascinating and difficult challenges emerge here. Clearly, the task of defining and calculating the operating constraints on the 'modules' is both interesting and challenging. The question of how such modules could and should be connected together (possibly to generate quite new types of robots) is also of prime interest. There has already been some discussion of building evolving modular robots via 'molecules' [12, 17]. The ideas discussed in this paper touch on all these issues, which appear difficult yet solvable at this time with current technology, given sufficient motivation and effort. The potential rewards could be very great indeed.

Acknowledgments

Primary funding for this work was provided by the National Science Foundation under grant IRI-9526363, and by the U.S. Department of Energy (through Foster-Miller Inc.) under contract #DE-FG07-97ER14830, subcontracts

#805-05001 and #805-05002. The first author is also supported by the National Science Foundation under grant CMS-9796328.

References

[1] R. J. Anderson. SMART: A Modular Architecture for Robotics and Teleoperation. In *Proceedings 1993 IEEE International Conference on Robotics and Automation*, volume 2, pages 416–421, Atlanta, GA, 1993.

[2] M. Antoniotti and B. Mishra. Discrete Event Models + Temporal Logic = Supervisory Controller: Automatic Synthesis of Locomotion Controllers. In *Proceedings 1995 IEEE International Conference on Robotics and Automation*, pages 1441–1446, Nagoya, Japan, 1995.

[3] K.F. Bohringer, J.W. Suh, B.R. Donald, and G.T.A. Kovacs. Vector Fields for Task-level Distributed Manipulation: Experiments with Organic Micro Actuator Arrays. In *Proceedings 1997 IEEE International Conference on Robotics and Automation*, pages 1779–1786, Albuquerque, NM, 1997.

[4] G.S. Chirikjian and I. Ebert-Uphoff. Discretely Actuated Manipulator Workspace Generation Using Numerical Convolution on the Euclidean Group. In *Proceedings 1998 IEEE International Conference on Robotics and Automation*, pages 742–749, Lueven, Belgium, 1998.

[5] E. Y. Chow and A. S. Willsky. Analytical Redundancy and the Design of Robust Failure Detection Systems. *IEEE Transactions on Automatic Control*, AC-29(7):603–614, July 1984.

[6] I. Ebert-Uphoff and G.S. Chirikjian. Inverse Kinematics of Discretely Actuated Hyper-Redundant Manipulators Using Workspace Densities. In *Proceedings 1996 IEEE International Conference on Robotics and Automation*, pages 139–145, Minneapolis, MN, 1996.

[7] A. Göllü and P. Varaiya. Hybrid Dynamical Systems. In *IEEE Conference on Decision and Control*, pages 2708–2712, 1989.

[8] F. J. Hill and G. R. Peterson. *Computer Aided Logical Design with Emphasis on VLSI, 4th Edition*. John Wiley, New York, NY, 1993.

[9] M. Ismail and T. Fiez. *Analog VLSI: Signal and Information Processing*. McGraw-Hill, New York, NY, 1994.

[10] S. Kang and Y. Leblebici. *CMOS Digital Integrated Circuits: Analysis and Design*. McGraw-Hill, 1996.

[11] R.D. Klafter, T.A. Chmielewski, and M. Negin. *Robotic Engineering: An Integrated Approach*. Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[12] K. Kotay, D. Rus, M. Vona, and C. McGray. The Self-reconfiguring Robotic Molecule. In *Proceedings 1998 IEEE International Conference on Robotics and Automation*, pages 424-431, Lueven, Belgium, 1998.

[13] J.C. Latombe. Robot Algorithms, Robotics Research 6, MIT Press.

[14] D.S. Lees and G.S. Chirikjian. An Efficient Method for Computing the Forward Kinematics of Binary Manipulators. In *Proceedings 1996 IEEE International Conference on Robotics and Automation*, pages 1012–1017, Minneapolis, MN, 1996.

[15] M.R. Lightner. Modeling and Simulation of VLSI Digital Systems. *Proceedings of the IEEE*, 75(6):786–796, 1987.

[16] C. Mead and L. Conway. *Introduction to VLSI Systems*. Addison-Wesley, 1980.

[17] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji. A 3-D Self-Reconfigurable Structure. In *Proceedings 1998 IEEE International Conference on Robotics and Automation*, pages 432–439, Lueven, Belgium, 1998.

[18] D.L. Pieper. The Kinematics of Manipulators Under Computer Control. Technical report, Ph.D. Dissertation, Stanford University, 1968.

[19] A.A. Requicha, C. Baur, A. Bugacov, B.C. Gazen, B. Koel, A. Madhukar, T.R. Ramachandran, R. Resch, and P.Will. Nanorobotic Assembly of Two-Dimensional Structures. In *Proceedings 1998 IEEE International Conference on Robotics and Automation*, pages 3368–3374, Lueven, Belgium, 1998.

[20] B. Roth and J. Rastegar andV. Scheinman. On the Design of Computer Controlled Mechanisms. In *Proceedings First CISM-IFTMM Symposium on Theory and Practice of Robots and Manipulators*, pages 93–113, 1973.

[21] J. Scholliers and T. Yli-Pietila. Simulation of Mechatronic Systems Using Analog Circuit Simulation Tools. In *Proceedings 1995 IEEE International Conference on Robotics and Automation*, pages 2847–2852, Nagoya, Japan, 1995.

[22] D. Sen and T.S. Mruthyunjaya. A Discrete State Perspective of Manipulator Workspaces. *Mechanism and Machine Theory*, 29(4):591–605, 1994.

[23] M. Shahinpoor. *A Robot Engineering Textbook*. Harper and Row, New York, 1987.

[24] C. E. Shannon. A Symbolic Analysis of Relay and Switching Circuits. *Transactions AIEE*, 57:713–723, 1938.

[25] W. Stadler. *Analytical Robotics and Mechatronics*. McGraw-Hill, New York, NY, 1995.

[26] K. Suzumori, K. Hori, and T. Miyagawa. A Direct-Drive Pneumatic Stepping Motor for Robots: Designs for Pipe-Inspection Microrobots and for Human-Care Robots. In *Proceedings 1998 IEEE International Conference on Robotics and Automation*, pages 3047–3052, Lueven, Belgium, 1998.

[27] J.P. Uyemura. *Physical Design of CMOS Integrated Circuits Using L-EDIT*. PWS Publishing Company, 1995.

[28] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker. A Dynamic Fault Tolerance Framework for Remote Robots. *IEEE Transactions on Robotics and Automation*, 11(4), 1995. 477-490.

[29] W. Wolf. *Modern VLSI Design: Systems on Silicon, 2nd Edition*. Prentice-Hall, Upper Saddle River, NJ, 1998.