2 Introduction

How are we to get future programmable controllers of automobiles, factories, robots, space vehicles and equipment, home appliances, and office systems to do new and changing tasks? How are we to get future robots to stack objects from a factory pallet, tidy and vacuum our house, or properly use complex sensors and effectors for any task not preprogrammed? How are we to get future computer controlled domestic appliances to do tasks such as adjusting the air-conditioning system to cope with different varieties of sick, elderly and active people? The problem is that non-programmers, and sometimes skilled programmers, have difficulty in setting up new tasks. If we expect robots and other programmable computer controlled systems to expand in the market place, then the set up cost for new tasks must be reduced, making the systems effective for non-programmers, small batch runs, varying task conditions, and so on. There will always be tasks that designers do not or cannot preprogram into a computer controlled system. The ability to be taught such new tasks is essential to a system's teachability and adaptability. This paper emphasizes the importance of the teaching; the teacher-robot interaction. Rather than the robot's *potential capabilities,* the capabilities *realizable through the interaction* are important.

Existing computer controlled systems lack teachability and adaptability [16,17]. Although complex tasks *can* be programmed into a robot, it is difficult – especially for the end user – to program new tasks.

Two popular methods of teaching fixed sequences of movements for later repetition are leading, where the robot is physically led through the sequence, and guiding, where the robot is remotely guided through the sequence [1,5,6,11,15,16,18,30]. Thus a user can specify that a fixed sequence be automated, without needing programming expertise. For example, robots have been taught spray-painting tasks using the leading method for many years [12,13].

However, once more complex task structures are needed, we must supply conditionals, iterations, hierarchical task structures, and complex data structure definitions, and we must be prepared to edit the robot's complex program to fix mistakes and make updates. Unfortunately, there are severe difficulties programming robots in this explicit manner.

Firstly, it is difficult for non-experts to specify a new procedure to a computer controlled system because existing systems need a description *in a programming language.* You can't just give a robot a rough English description of the task [6,15,16,18]. People do not find it easy to translate their natural knowledge of a task into an algorithm, nor to communicate the algorithm in a programming language. Their communication skills are interpersonal. A spray-painter knows how to spray-paint, and how to describe it to a human, but not how to write a spray-painting program. Programmers need considerable training and experience.

Secondly, explicit programming languages are all powerful for specifying procedures, but they are unsuitable for casual users because programmers need to learn tricky details of computer syntax, semantics, and internal technical characteristics [35]. More importantly programmers need to articulate procedures *explicitly,* a difficult task.

Thirdly, the programming may be difficult. Declarations about the real three dimensional world may explode even in high level languages [11]. N dimensional simultaneous programming may be unmanageable for humans [1].

Fourthly, robots lack the ability to use sensory information [24] so it is difficult to teach them sensory tasks. Existing robots are not adaptable [21] and can carry out manual tasks only in a limited range of situations [5]. Advanced robot systems are limited to specific tasks and their associated environments, and may be difficult to extend [32].

Finally, a robot programmer must be expert at both algorithm design and programming, *as well as expert at the task to be taught* [15,16,18].

The remainder of this section discusses the leading method and outlines verbal correcting. Section 3 gives the details of the verbal correcting method. Section 4 describes experimental work with verbal correcting. Section 5 compares this work to other methods for specifying procedures to computer controlled robots.

2.1 Leading a Robot by the Hand

One way to circumvent the difficulty of programming is to have the user give examples; no explicit programming is required. So one might lead a robot through several examples of a complex task, expecting it to learn to perform the task in other situations as well. The robot generalizes the taught examples to a general description of the task. The existing leading method is a trivial form of this, since there is only one example to be given and no generalization to be made. Leading teaches a fixed sequence. General techniques for programming by example have been studied by others [2,3,4,10,20,22,23,26,29,35]. For example one system [3,4] taught a 2-D mobile robot to move around arbitrary convex 2-D obstacles [3,4]. In the more distant future we might expect robots to learn in other ways; from a natural language description of a task, by imitating us, and so on. We are some way from being able to do this. Even if we could, we would have difficulty describing some tasks, for example riding a bicycle, hitting a ball, chopping wood, and so on. We *can* give examples of such tasks; for example by pushing a robot or human along on a bicycle, and by holding someone's arm to show them how to hit a tennis ball. So teaching by giving examples will be important in future controller teaching as well as present.

This paper discusses how non-programmers may edit a taught sequence, and leads to methods for the specification of procedures more complex than a fixed sequence. It proposes and demonstrates a scheme by which a teacher may verbally correct a robot as it executes a sequence; the teacher uses his or her own natural ability at verbally correcting humans. Corrections may be used to compensate for external disturbing forces applied to the arm, or to correct incorrectly led movements. The corrections invoked by verbal commands are themselves taught using the leading method. The addition of a production system of corrections would enable teaching of conditional corrections on line, by verbal correcting. The robot is then able to select its own action corrections, given the conditions sensed. My goal-seeking system would enable a teacher to teach individual actions and set goals. The robot then selects its own path of actions to achieve a goal, given the sensed conditions. These further improvements are discussed in detail elsewhere [16,17]. Briefly, verbal correcting enables only one final sequence to be taught, but the small addition of the production system of correction enables the user to give a sequence of *conditional action corrections* without having to explicitly program the conditionals in. The teacher corrects the robot in the situations requiring different corrections, and the robot can then choose its own corrections by sensing the conditions. When a goal-seeking system is also added, the robot can be shown motor-sensory goals, learn a repertoire of simple movements, and seek the goals itself along paths of movements in a network of movement actions, conditions and goals. The network may be retained from task to task, enabling the robot to build up a large repertoire of movements.

Editing led sequences In this section I give some justification for *needing* to edit sequences that have been led; why led sequences *cannot* always be correct. When a movement, say M is led, it causes a robot movement command, say C to be stored for later execution. Now C is the command that will cause M, *so long as no uncompensated forces disturb the arm.* Such forces would cause a different movement to be produced. In that case the command C must be corrected to a command C^*, which does produce M. A robot control system can compensate only for certain disturbing forces. Regardless of its complexity one could always think of a force interaction for which the control system would not compensate; for example, a simple controller might not compensate for the slower movement caused by a heavy load, while a complex controller might not compensate for force interactions while riding a bicycle, driving a car, swinging an axe, and so on. It is important to remember that when leading a robot arm, *the teacher is taking the weight, compensating for the disturbing forces himself,* and so leading movements cannot itself show the robot how to exert forces.

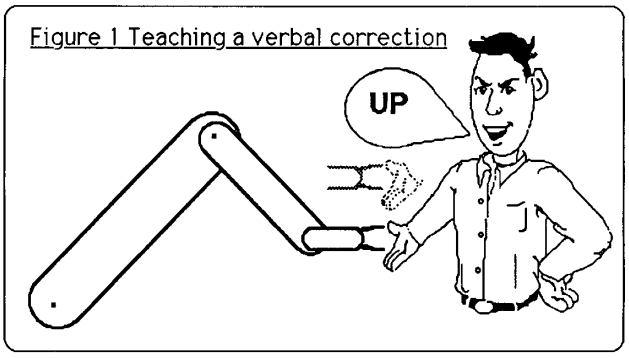
This paper presents a method by which the teacher can verbally instruct the robot to *give itself a push*; when the teacher says "UP", the effect is to get the controller to push upward somewhat more, applying more upward force. In this way previously uncompensated disturbances may be counteracted.

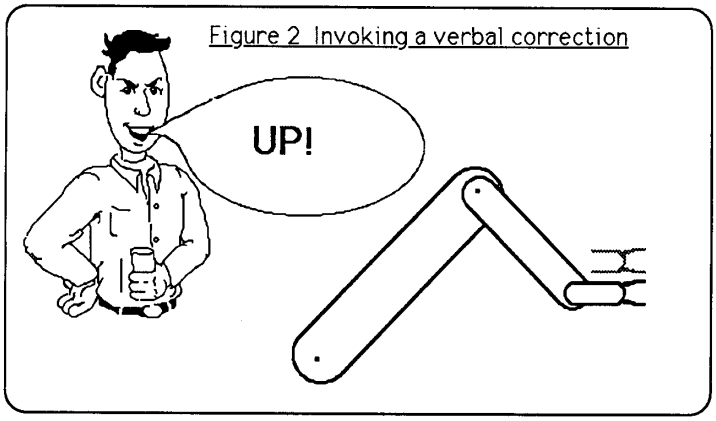
3 Verbal Correcting

3.1 The Scheme

The verbal correcting scheme makes two changes to the normal led robot. First, a list of verbal correction pairs is stored in the robot controller, and may be invoked by the teacher to cause corrections to movements. For example, Figure 1 would cause the correction pair (UP, $+1^z$) to be recorded if the led arm movement were one unit in the upward, z dimension.

Second, the ability to invoke verbal corrections is added to the teacher-robot interaction by enabling the teacher to (a) teach the verbal correction pairs, before and while he verbally corrects the robot, and (b) invoke the previously taught corrections by speaking the previously taught words of a pair. So if the teacher says "UP", then the next command sent to the robot will have $+1$ added to its z component. The new value will immediately be executed, and in addition will be stored in the recorded sequence replacing the previous command. Figure 2 depicts this process.





Sachs and Leifer [25] have used a full set of verbal corrections for controlling a robot. In addition, two special words are preprogrammed into the robot to enable a teacher to *scale* his taught corrections. Saying "LARGER" before a correction causes that set of corrections to be scaled up, say doubled, so that "LARGER UP" causes the correction of the pairs for the word "UP", perhaps three different sized corrections of "CREEP UP", "UP", and "ZOOM UP", to be doubled in size. "SMALLER" might cause them to be halved.

Corrections would be taught by the teacher pushing a button to put the robot into a special correction training mode, and then repeatedly (a) saying a word or a few words, for example "CREEP UP", and (b) leading a movement, for example an upward acceleration of 1 cm/s/s. A two-tuple,

$$< Speech, Correction >,$$

for example

$$< \text{"CREEP UP"}, (1, 0, 0, 0, 0, 0) >$$

is recorded and remembered in the verbal correction list. The correction is in Cartesian coordinates and Euler angles.

3.2 Using Verbal Correcting

When a verbal correction is made this sequence of events occurs:

1. An incorrect motor command is executed; say $(+4, 0, 0, 0, 0, 0)$
2. The teacher notices the onset of an error in the robot's movement
3. The teacher says words that have been taught as a correction; "UP"
4. The robot recognizes the words
5. The robot makes the correction, say $(+2, 0, 0, 0, 0, 0)$, to the next motor command executed, say $(+1, 0, 0, 0, 0, 0)$, forming $(+3, 0, 0, 0, 0, 0)$. (or perhaps to the motor command being executed)

I expect verbal correcting to be useful for correcting led robots' sequences for eight reasons:

1. Natural Ability The teacher may use his or her own natural ability at verbally correcting other humans, so long as the robot does not exceed human speed.

2. Error Anticipation A human teacher may anticipate errors in rapid movements [19] enabling him to correct the errors in spite of the delay between the onset of error on step 1 and the teacher's correction on step 5 above. The delay should be roughly one second.[1]

3. New Corrections Anytime If he wants to the teacher can stop the robot, put it into correction training mode and teach new corrections. So the teacher can easily teach new corrections appropriate to correcting a particular part of a particular task.

4. Current Speech Recognition Techniques Current techniques enable short phrases from a limited vocabulary to be recognised [9,21,28,25].

[1]Most of the delay should be the teacher's reaction time, which should be very roughly one second, given that human reaction times are about 200 *ms* per bit of choice [14] and there are three responses for three different sized corrections in each of six dimensions, and and modifying each one; a total of 108. See also [7,19,27,33] for further discussion of reaction times.

5. Theoretical Convergence and Stability I summarize arguments for the theoretical stability and convergence of successive verbal correcting in section 3.4 below.

6. Simple Correcting Experiment A simple trajectory correcting experiment indicates that huge errors may be corrected in less than 15 successive corrections, with the strategy given in section 3.4 [16]. Smaller corrections can be made in fewer steps.

7. Verbal Correcting Experiment As explained below, I have demonstrated a simple experimental verbal correction system with a real robot and speech recognizer. In addition verbal correcting has been implemented on a large industrial robot, also explained below.

8. Correcting led movements As well as correcting for uncompensated disturbing forces, verbal correcting can also be used to correct errors made by the teacher in the original leading.

3.3 Discussion of verbal correcting

Corrections that cannot be made Verbal correcting enables any action that can be led to be used for correcting a task. However, some actions cannot be led. For example, if a teacher could not lift a robot's arm, no upward movements could be taught. In general a correction can be made if there exist a finite number of leadable actions to produce that correction. This will always be true if there is at least one action leadable in the positive and negative directions in six independent dimensions. Since movements are finite, a finite number of leadable actions will produce any desired correction. This assumes that the teacher can lead the smallest possible action in a dimension, if he can lead any action at all in that dimension. So any correction could be produced by combining the smallest actions in each dimension.

Note that scaling corrections with "LARGER" adds no corrections not possible without it. "SMALLER" also adds no corrections.

Insertion and deletion Verbal correcting does not practically enable actions to he inserted or deleted from a sequence, since all the sequence following an insertion or deletion would need changing. Nevertheless, insertion and deletion can be achieved verbally, if one is able to stop the robot, say "INSERT" and a correction, or say "DELETE". Insertion and deletion were similarly implemented in the experimental system described in section 4.1 below. The robot cannot be stopped in this way if the task is at all dynamic; for example in a spray painting task, the act of stopping and starting the arm would completely change the painting achieved. The experimental system was concerned with the robot arm position, not its movement in time.

Successive correcting The teacher may need to observe the robot as it executes the task several times, verbally correcting it repeatedly. The stability and convergence of this successive verbal correcting are outlined below, where I argue that verbal correcting is stable and convergent under reasonable conditions.

3.4 Convergence and Stability

The teacher's correcting actions may not immediately produce the correcting movements required. The additional accelerating forces will be subject to the forces opposing the robot's movements. However, we

would hope that the corrections would cause the error to be reduced, since they cause forces to be exerted in opposition to the movement-opposing forces.

A teacher's successive corrections will naturally satisfy Dvoretzky's conditions [8,16,34] for the convergence and stability of a stochastic approximation scheme, so long as the interaction between the robot and its environment is not so complex and time-varying as to fool our teacher. If another human interacted with the robot, then a teacher might find it impossible to correct the robot's movements. That human might seek to thwart the teacher's corrections.

Dvoretzky's conditions and the way a teacher's verbal correcting satisfies them are discussed in detail in [16]. There is not room to give all the details here so only the important points are given below. Briefly, the teacher's successive correcting may converge because he may naturally adjust his corrections to ensure that

(a) correction "overshoot" tends to decrease,

(b) any increases in the movement error tend to decrease as successive corrections are applied, and

(c) correcting does not stop until the movement error is reduced to an acceptably small size.

These conditions should be met by this strategy for users:

(i) if a correction causes the movement error to be overcorrected in any dimension then make your next correction smaller, thus satisfying condition (a);

(ii) don't give up correcting just because the error is increasing but keep correcting until the error has been removed, thus satisfying (c); and

(iii) the best way to correct a movement is to try and overcorrect in each dimension, and then correct back the other way employing strategy (i).

Condition (b), requiring errors to decrease when there is no overshoot, should be satisfied by the assumption above that the teacher naturally be able to correct in the right direction.

Now, the error in a movement not the first in the sequence may be partly caused by the errors in previous movements, since an early incorrect movement may cause increased errors in later movements. This influence of previous movement errors on later movement errors can be treated as random fluctuations in those later movements. So long as these random fluctuations eventually vanish they do not prevent convergence and stability. These fluctuations can be made to vanish, from the first movement through to the end of the sequence. Once the first movement is correct, it doesn't cause an error, a fluctuation, in the second movement, so the second movement may converge, and so on. This is not to say that corrections to the *k*th movement of a sequence will be useless until all movements through to the *k-1* have converged. Rather the *k*th movement may not *finally* converge until the earlier movements have. In order to guarantee that the first movement is not affected by the last movements in a repetitive task, a teacher may provide a special "lead-in" to a task, during correcting.

4 Experimental Work

This section describes two real implementations of verbal correcting; the first on a simple experimental robot and more recent work on a large industrial robot. Briefly, the experimental system established verbal correcting to be implementable, but the sluggish robot did not enable the dynamic, online, rapid correction to be tested. The industrial robot implementation has recently been completed and is to be tested shortly. Figure 3 illustrates the form of both systems.

recognizer is only just satisfactory, since it recognized only words spoken by one person, and since at times several recognition errors were made in one correcting session. This would be frustrating in a commercial system, but was satisfactory for the demonstration. A more powerful recognizer would be required in a commercial implementation.

4.2 Industrial robot system

Verbal correcting has been implemented on a PUMA 700 industrial robot at the Alberta Research Council in Calgary. A specially written supervisory system controls the robot and speech recognizer on a 68020 development system, and also sends special dynamic path changes – which are specified by verbal corrections – via a dedicated "alter" communication line to the robot. Online verbal corrections of slow movements have been performed. The robot speed will not be increased until the system has been fully tested.

5 Other Methods

NODDY The NODDY system [3,4] acquires robot procedures, complete with control information which is not explicitly present in the examples. NODDY operates in a simple simulated 2-D world. It handles real numbers representing angles and distances. NODDY employs a preprogrammed hierarchy of actions and several preprogrammed functions both of which it uses to generalize example task sequences into one procedure.[3] One example task for NODDY is to get from a start position to an end point; a robot navigation task. It must navigate around obstacles by retreating perpendicularly to the collision surface, moving a short way parallel to the surface, then continuing toward the goal.

XPROBE XPROBE [30] combines high-level programming and leading so that movements and positions can be shown using leading, and control structures, variables and reference frames can be programmed. The user may specify different frames of reference for movements and positions shown by leading. So for example movements shown in the frame of a pallet are still correct if the pallet is moved.

Showing transformations [31] also describes a system that acquires tasks by combining leading and high-level programming. First a task is programmed, leaving some coordinate transformations undefined in the program. The transformations are taught by the teacher leading the arm to the appropriate position, specifying each undefined transformation needed.

Task level description [15] describes a language designed to allow a user to specify a task in a natural way, from the user's point of view. An example statement is:

```
PLACE BEARING1 SO
      (SHAFT FITS BEARING1.HOLE)
   AND
      (BEARING1.BOTTOM AGAINST SHAFT.LIP)
```

(p.837)

[3] See also [35,36] for more detailed explanation of Noddy.

6 Conclusion

Online verbal correction of robot sequences is feasible, and provides a method for non-programmers to edit robot sequences. Thus more sophisticated methods that use verbal correcting – production systems of corrections for example – are promising. Verbal correcting retains the advantages for the non-programmer that the popular leading method has. If robots can be verbally corrected "on the fly" then we can expect other programmable computer controllers to be similarly verbally commanded, thus taking advantage of human users' natural ability to dynamically respond in spoken language.

7 Acknowledgements

I am grateful to John Andreae for advice on the early work on verbal correcting principles. The General Manager of the New Zealand Electricity Division gave me leave without pay for the early work. The University Grants Committee, the University of Canterbury and the Defence Scientific Establishment in New Zealand supported the early work. The later, experimental work was supported by the Alberta Research Council, the University of Calgary and the Natural Sciences and Engineering Council of Canada. Tuan Vu did the experimental robot implementation and Darcy Grant the industrial robot system.

References

[1] Ambler, A.P., Popplestone, R.J. and Kempf, K.G. (1982) "An Experiment with the Offline Programming of Robots" *Proc. 12th Int. Symp. on Industrial Robots; 6th Int. Conf. on Industrial Robot technology.* Paris. France. June. pp. 491-504.

[2] Andreae, J.H. (1984) "Numbers in the head" Man-Machine Studies Progress Report UC-DSE/24, Dept. Electrical and Electronic Engineering, University of Canterbury, Christchurch, New Zealand. pp. 5-28.

[3] Andreae, P.M. (1984) "Justified Generalization: acquiring procedures from examples" PhD thesis, Dept. Electrical Engineering and Computer Science, MIT.

[4] Andreae, P.M. (1984) "Constraint limited generalization: acquiring procedures from examples" *Proc American Association on Artificial Intelligence,* Austin, TX, August.

[5] Benati, M., Gaglio, S., Morasso, P., Tagliasco, V., and Zaccaria, R., (1980) "Anthropomorphic Robotics, Parts I and II" *Biol.Cybern., vol. 38,* pp. 125-150.

[6] Bonner, S. and Shin, K.G., (1982) "A Comparative Study of Robot Languages" *Computer, vol. 15,* no. 12, pp. 82-96.

[7] Crossman E.R.F.W. (1953) "Entropy and Choice Time: The Effect of Frequency Unbalance on Choice – Response" *The Quarterly J. of Experimental Psychology, V* no. 2, pp. 41-51.

[8] Dvoretzky, A. (1956) "On Stochastic Approximation" *Proc. 3rd Berkeley Symp. on Mathematical Statistics and Probability 1* pp. 39-55.

[9] Flanagan, J.L. (1982) "Talking with Computers: synthesis and recognition of speech by machines" *IEEE Trans. Biomedical Engineering, vol. BME-29,* no. 4, pp. 223-32.

[10] Halbert, D.C. (1984) "Programming by example" Technical Report, Xerox Office Products Division, Palo Alto, California, December.

[11] Hasegawa, T., (1982) "An Interactive System for Modeling and Monitoring a Manipulation Environment" *IEEE Trans.SMC, vol. SMC-12,* no. 3, pp. 250-8.

[12] Haugan, K.M. (1974) "Spray Painting Robots: Advanced Paint Shop Automation" *The Industrial Robot,* pp. 270-2, December.

[13] Haugan, K.M. and Jarvis, D.E. (1974) "Electronically Controlled Manipulator for Spray Gun Applications" *The Industrial Robot,* pp. 119-21, March.

[14] Hyman, R. (1953) "Stimulus Information as a Determinant of Reaction Time" *J. of Experimental Psychology, vol. 45,* no. 3, pp. 188-96

[15] Lozano-Perez, T., (1983) "Robot Programming" *IEEE Proc, vol. 71,* no. 7, pp. 821-41. Invited paper.

[16] MacDonald, B.A. (1984) "Designing teachable robots" PhD thesis. University of Canterbury, Christchurch, New Zealand.

[17] MacDonald, B.A. (1987) "Improved Robot Design" *Trans.IPENZ Elec/ Mech/Chem section, vol. 14,* no. 1/EMCh, pp. 33-48

[18] McLaughlin, J.R., (1982) "TRIG: An Interactive Robotic Teach System" Working Paper 234, MIT AI Lab. June. 53p

[19] Marteniuk, R.G. (1976) "Information Processing in Motor Skills" Holt, Rhinehart and Winston.

[20] Mitchell, T.M. (1982) "Generalization as search" *Artificial Intelligence, vol. 18,* pp. 203-26.

[21] Nitzan, D., (1979) "Flexible Automation Program at SRI" *Proc JACC* pp. 754-9, Denver. June

[22] Nix, R. (1983) "Editing by example" PhD thesis, Computer Science Dept, Yale University, New Haven, CT.

[23] Nix, R. (1984) "Editing by example" *Proc 11th ACM Symposium on Principles of Programming Languages,* pp. 186-195, Salt Lake City, Utah.

[24] Rosen, C.A., (1979) "Machine Vision and Robotics: Industrial Require- ments" in Computer Vision and Sensory-Based Robots., ed. G.G.Dodd and L.Rossol, pp. 3-20, Plenum, N.Y. Symp. held at GM Labs.

[25] Sachs,J. and Leifer, L. (1979) "Voice Command of a Six-Degree-of-Freedom Manipulator" *Proc. J. Auto. Contr. Conf.* Denver. June. pp. 783-89

[26] Sammut, C. and Banerji, R. (1986) "Learning concepts by asking questions" in Machine Learning Volume 2, edited by R.S.Michalski, J.G.Carbonell, and T.M.Mitchell, pp. 167-191. Morgan Kaufmann Inc, Los Altos, CA.

[27] Sheridan, T.B. and Ferrel, W.R. (1981) "Man-Machine Systems. Information, Control, and Decision Models of Human Performance" 2nd edition. MIT Press.

[28] Simons, G.L. (1980) "Robots in industry" National Computing Centre, Manchester, England.

[29] Smith, D.C. (1975) "Pygmalion: A computer program to model and stimulate creative thought" PhD thesis, Dept of Computer Science, Stanford University.

[30] Summers, P.D. and Grossman, D.D., (1984) "XPROBE: an experimental system for programming robots by example" *Int J Robotics Research, vol. 3,* no. 1, pp. 25-39. Spring

[31] Takase, K., Paul, R.P., and Berg, E.J., (1981) "A Structured Approach to Robot Programming and Teaching" *IEEE, vol. SMC-11,* no. 4, pp. 274-89.

[32] Waltz, D., (1982) "Artificial Intelligence" *Scientific American,* October, pp. 78-83.

[33] Welford, A.T. (1980) "Choice Reactions Time: Basic Concepts" Chapter 3 in Reaction Times. edited by A.T.Welford. Academic Press.

[34] Wilde, D.J. (1964) "Optimum Seeking Methods" N.J., Prentice-Hall.

[35] Witten, I.H. and MacDonald, B.A. (1987) "Specifying Procedures to Office Systems" *Proc. Conf. on Automating Systems Development,* April, Leicester, England.

[36] Witten, I.H. and MacDonald, B.A. (1987) "Concept learning: A practical tool for knowledge acquisition?" *Proc. 7th Intl. Workshop on Expert Systems and Their Applications,* May, Avignon, France. pp. 1535-55.