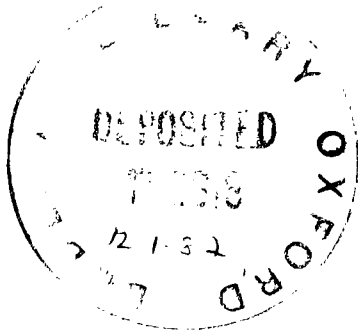


Applications of Mathematical Modelling
in Demand Analgesia



by

Peter Lammer
St. John's College

thesis submitted for
the Degree of Doctor of Philosophy
University of Oxford
Trinity Term 1986

Applications of Mathematical Modelling
in Demand Analgesia

Peter Lammer
St. John's College

thesis submitted for
the Degree of Doctor of Philosophy
University of Oxford
Trinity Term 1986

ABSTRACT

This thesis describes applications of mathematical modelling to systems of demand analgesia for the relief of acute postoperative pain. It builds upon work described in the D.Phil. thesis of M.P.Reasbeck.

Following major surgery, patients are given a hand-held button which they press when in need of pain relief. The relief is afforded by automatic intravenous infusion of opiates. New clinical demand analgesia hardware, PRODAC, has been developed and data have been collected with it in two major trials involving a total of 80 patients. Patients' drug requirements have been found not to be correlated with body weight, contrary to conventional teaching. The type of operation was also found to have no significant influence upon drug requirements. The performance of transcutaneous nerve stimulation (TNS) as a method of analgesia for acute postoperative pain has been studied and found to be poor. Reasbeck's mathematical model of patients in pain has been corrected and extended. The representation of pharmacokinetics has been enhanced by modelling the transfer of drug between blood plasma and analgesic receptor sites as a first-order process. The time constant of this process has been calculated for morphine using a novel method and found to be 12 minutes. On line estimation of 2nd order pharmacokinetic time constants has been found in simulation not to be feasible. New software has been used to tune the revised model to the clinical data collected with PRODAC. Model behaviour is now demonstrably life-like, which was not previously the case. Blood samples taken during demand analgesia have permitted a comparison between measured and estimated drug concentrations, with good results.

Acknowledgements

My gratitude and appreciation are due first and foremost to Dr. Oliver Jacobs, whose outstandingly conscientious, able and persistent supervision has been invaluable.

I am also extremely grateful to Dr. Roy Bullingham and Dr. Henry McQuay, who, as my medical supervisors, have helped me and given much advice throughout my work.

Dr. Michael Reasbeck's D.Phil. work laid the foundations for my own research and I am absolutely indebted to him, in more ways than can be listed here.

Thanks are due too to many other people who gave invaluable help in a variety of ways, including Dr. Wassim Badran, Mr. Jeremy Bowles, Dr. Jolyon Cox, Dr. Lyn Davies, Mr. Steve Evans, Dr. Chris Glynn, Dr. Keith Godfrey, Dr. Jan Hruska, Dr. Duncan McCallum, Dr. Andrew Moore, Dr. Anne Phillips, Miss Patsy Poppleton, Dr. Jeff Uppington, the nursing staff at the Nuffield Orthopaedic Centre and at the Radcliffe Infirmary, and last but by no means least, the eighty or so patients who consented to take part in clinical trials reported here.

A number of companies provided help in the form of loans and donations of various equipment, including EPSON UK Ltd, Executive Computers Ltd, Graseby Dynamics Ltd, Grafox Ltd,

Robert Moss PLC and Sophos. Their generosity is much appreciated.

Finally, sincere thanks are due to SERC, who supported me for three of my years at Oxford, and to the Szeben-Peto foundation of Canada, which funded the first two sets of clinical equipment.

CONTENTS

CHAPTER 1 INTRODUCTION

1.1	CLINICAL PAIN RELIEF	1
1.1.1	Problems with Conventional Methods	2
1.1.2	Demand Analgesia	3
1.2	APPLICATION OF CONTROL ENGINEERING METHODOLOGY	4
1.2.1	Criteria of Performance	4
1.2.2	Demand Analgesia as a Control Problem	4
1.2.3	The Work of Reasbeck	7
1.2.3.1	Scope	7
1.2.3.2	The Model	7
1.2.3.3	Realisation of Model	10
1.2.3.4	Limitations	12
1.3	OTHER RELATED WORK	12
1.4	OBJECTIVES AND OUTLINE OF THIS THESIS	14

CHAPTER 2 NEW DEMAND ANALGESIA EQUIPMENT

2.1	INTRODUCTION	17
2.2	SPECIFICATIONS AND REQUIREMENTS	18
2.2.1	Reasbeck's Hardware	18
2.2.2	The Need for New Hardware	18
2.2.3	Requirements for the new Hardware	20
2.2.3.1	Choice of Single Board Computer	22
2.3	PRODAC - DESCRIPTION OF HARDWARE	24
2.3.1	General construction	24
2.3.2	Button Interface	27
2.3.3	Syringe Driver Interface	29
2.3.4	Display	34
2.3.5	Keypad	35
2.3.6	Data Storage	37
2.3.7	RS232 Interface	37
2.3.8	Warning Devices	38
2.3.9	Printer	38
2.3.10	Power Supply	39
2.4	PRODAC - DESCRIPTION OF SOFTWARE	40
2.4.1	Choice of programming language	40
2.4.2	Software development	42
2.4.3	Structure of the Software	43
2.4.4	Initialisation	45
2.4.5	Background Software (not interrupt driven)	45
2.4.6	Foreground Routines (interrupt driven)	46
2.4.7	Data Storage	48
2.5	USE OF PRODAC	51
2.5.1	Normal operation	51
2.5.2	Housekeeping functions	51
2.5.3	Data capture	52
2.6	SUMMARY	53

CHAPTER 3	INITIAL SIMULATION STUDIES	
3.1	CHANGES TO OXPANG AND REASBECK'S MODEL	54
3.1.1	Patient Demand Non-linearity	54
3.1.2	Randomisation Procedure	58
3.2	SIMULATION RESULTS	59
3.2.1	Batch Processing	59
3.2.2	Effects of Real Syringe Driver Behaviour	59
3.2.3	Comparison of Controllers and the Automatica Paper	61
CHAPTER 4	FIRST PRODAC TRIAL	
4.1	MOTIVATION	63
4.2	DESIGN OF TRIAL	64
4.3	RESULTS	66
4.3.1	Intramuscular Phase	67
4.3.2	Demand Analgesia Phase	67
4.3.3	Assay Results	71
4.3.4	Clinical Experience with PRODAC	71
4.4	DISCUSSION	72
4.4.1	Summary of Results	72
4.4.2	Potency Ratios	73
4.4.3	Influence of Weight	76
4.4.4	Influence of Operation	79
4.4.5	Demand Rate and Press Factor	81
4.5	CONCLUSIONS	82
CHAPTER 5	PHARMACOKINETIC MODEL REVISIONS	
5.1	MOTIVATION FOR CHANGES	85
5.2	DETERMINING PLASMA-TO-RECEPTOR TRANSFER KINETICS .	89
5.2.1	Established Methods	89
5.2.2	Modified Method using Time to Peak Effect	90
5.3	DIRECT 2ND ORDER MODELLING OF TISSUE LEVEL	97
5.3.1	Aim and Motivation	97
5.3.2	Finding 2nd Order Approximation	99
5.3.3	Modified Estimation Scheme	100
5.3.3.1	Changes to State-Space Model	100
5.3.3.2	Changes to Covariance Matrix Handling	101
5.3.4	Performance	101
5.4	CONCLUSIONS	103
CHAPTER 6	NEW OFF-LINE SOFTWARE	
6.1	MOTIVATION AND REQUIREMENTS	104
6.2	SIM	106
6.2.1	Functionality	106
6.2.2	SIM Details	107
6.2.2.1	Randomisation procedure	108
6.2.2.2	Variable Duration of I.M. Dose	110
6.2.2.3	Plasma/Tissue Distinction	110
6.2.2.4	Accurate Simulation of PRODAC	110

6.2.2.5	Shared Code and Variables with PRODAC	111
6.2.2.6	Running Multiple Simulations	111
6.2.2.7	Extendable Modular Menus	112
6.3	UTILITIES	113
6.3.1	PRODUMP	113
6.3.2	PATCONV	113
6.3.3	PATSTAT	113

CHAPTER 7 MODEL TUNING AND VALIDATION

7.1	INTRODUCTION	115
7.2	CRITERIA OF MODEL BEHAVIOUR	115
7.3	TUNING THE MODEL TO THE MEPTAZINOL TRIAL RESULTS .	116
7.4	VALIDATION	119
7.5	CONCLUSION	123

CHAPTER 8 SECOND PRODAC TRIAL

8.1	INTRODUCTION	124
8.2	METHODS	125
8.3	RESULTS	126
8.4	DISCUSSION	128
8.4.1	Comparison of Active and Placebo TNS	128
8.4.2	Measured vs. Estimated Plasma Levels	130
8.5	CONCLUSIONS	132

CHAPTER 9 CONCLUSIONS

9.1	SUMMARY OF ACHIEVEMENTS	135
9.2	SUGGESTIONS FOR FURTHER WORK	140
9.2.1	Model-Based Control	140
9.2.2	Psychology	142
9.3	FINAL PERSPECTIVE	144

APPENDIX A AUTOMATICA PAPER

APPENDIX B PRODAC CIRCUIT DIAGRAMS

APPENDIX C PRODAC SOFTWARE

APPENDIX D SOFTWARE FOR 2ND ORDER FIT

APPENDIX E BIBLIOGRAPHY

CHAPTER 1

INTRODUCTION

1.1 CLINICAL PAIN RELIEF

Clinical relief of acute postoperative pain is generally achieved with analgesic drugs such as morphine. These act at receptor sites in the spine and brain. A patient emerging from a general anaesthetic after surgery is typically given intramuscular (i.m.) injections every four hours or so, in sufficient quantity to reduce pain to an acceptable level, yet not so much that undesirable side-effects such as ventilatory depression pose a threat to his safety. The major factor determining the relief of pain is the concentration of drug at the receptor sites. This is closely dependent on the concentration in the bloodstream, or 'blood plasma level'. The propagation of drug through the body is described by 'pharmacokinetics', whereas the effect caused by the concentration of drug at the receptor sites is described by 'pharmacodynamics'.

1.1.1 Problems with Conventional Methods

After an i.m. injection, the plasma level of the drug rises for about half an hour to reach a peak value, and then slowly decays. This means that both for the first half-hour after an injection and towards the end of the three- or four-hour period between injections, the blood plasma level is liable to be insufficient for pain relief. To cope with this by giving larger doses of drug, so that the plasma level never drops to an insufficient value, would lead to unacceptably high peak plasma levels posing a potential danger to the patient. A slightly better solution is often to give a constant intravenous (i.v.) or i.m. infusion of the drug, so that a stable plasma level is maintained. Intravenous infusions have the advantage that the drug is carried more rapidly to the receptor sites without having first to diffuse through muscle and fat into the bloodstream.

There is however a fundamental difficulty which prevents any such fixed regime from achieving consistently good performance. Pain is an extremely subjective phenomenon which can fluctuate strongly over the course of hours or even minutes. It is unpredictable and, even under apparently identical conditions, varies widely from one patient to another. Furthermore the effect of analgesic drugs is similarly variable from one patient to another; some patients need only very little drug, possibly experiencing a substantial placebo effect as well, whereas others need large quantities of drug or, in extreme cases, fail to get any

relief at all. For these reasons analgesia, like other applications of drug delivery, is a natural candidate for the use of feedback.

1.1.2 Demand Analgesia

During the past decade, various devices have been developed to allow the patient to play a part in determining how much drug should be given, and when. The most common approach is to give the patient a hand-held button, which is to be pressed when pain relief is required. On the basis of this button-pressing, a control device infuses drug into the patient's bloodstream. Intravenous infusion is generally used, to permit a fast analgesic response. Strategies of this nature, in which the patient is allowed to close a feedback loop, are loosely termed 'demand analgesia'. They offer an inherent advantage over conventional therapy by making it possible to match drug delivery to the individual patient's requirements. Commercially available devices commonly operate in a mode known as 'proportional control', in which a small infusion bolus of set size is given (subject to safety limits) in response to each pressing of the button. In some cases this is combined with a background infusion which may or may not be self-adjusting.

1.2 APPLICATION OF CONTROL ENGINEERING METHODOLOGY

1.2.1 Criteria of Performance

A good demand analgesia system should infuse drug such that the patient's demands are not suppressed completely but are reduced to an acceptable average rate such as one or two demands per hour. This acceptable rate should be achieved as uniformly as possible for all patients, but with the amount of drug administered varying from patient to patient as required. The drug should be administered in such a way as to track the patient's analgesic requirement, avoiding excessively high plasma levels.

Pure proportional control devices are inherently unable to fulfil the joint requirements of maintaining a uniform demand rate and varying the amount of drug delivered, as their control laws include no element of learning or adaptation to the needs of individual patients. The first work to attack the problem of optimal control of postoperative pain from the point of view of feedback control theory is that described by M.P. Reasbeck in his D.Phil. thesis [Reasbeck, 1982].

1.2.2 Demand Analgesia as a Control Problem

Fig. 1.1 shows how the demand analgesia patient can be regarded as part of a feedback control loop. A number of problems such as non-linearity and uncertainty, which can make good control of a simple loop such as this difficult, are manifestly present.

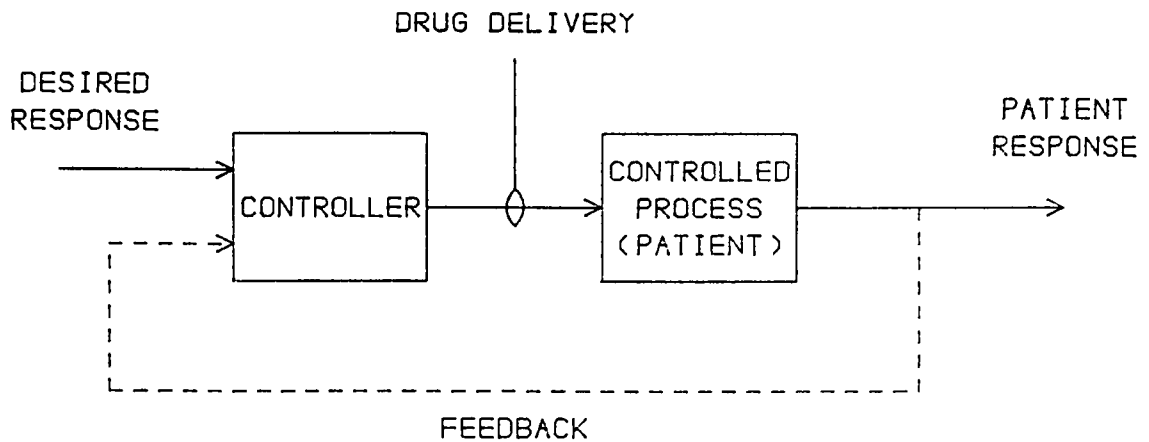


Fig. 1.1 Demand Analgesia as a Feedback Control Loop

The controlled process in this case, the patient, is an immensely complex system about much of which little is known, and which must be modelled simplistically and approximately. The only really certain knowledge of the 'plant parameters' is that they are likely to vary widely from one patient to another. This applies to almost every aspect of the problem, from the pharmacokinetics of the drug to the patient's reaction to pain. Unpredictable disturbances are present in the form of the surgical wound as well as fluctuations in the patient's overall mood and well-being.

The single greatest problem is that there is no direct way of measuring the controlled variable, pain, as this is an inherently subjective phenomenon which is dependent not only

upon physiological but also psychological factors. Traditional measurements of pain involve asking the patient to choose a word from a list like 'mild', 'severe', 'none', or to point on an analogue scale to the position which represents the pain he has felt. Methods like this are both taxing for the patient and obviously unsuitable for on-line automatic control, so that the button-pressing of demand analgesia seems to be the best feedback signal available with which to quantify the patient's perception of his pain. The patient is told simply

"if it hurts, press the button".

This feedback signal is subject to strong non-linearity in that the patient will press the button when pain relief is insufficient, but not when it is adequate or excessive. Likewise, the controller action is constrained to be non-negative; it is possible to inject drug or not to inject it, but it is not possible to remove it from the patient.

There is also a time-delay aspect as it is not until several minutes after infusion of a bolus of drug that peak analgesic effect occurs.

1.2.3 The Work of Reasbeck

1.2.3.1 Scope - Reasbeck set up a mathematical model of the patient in pain, including simple representations of pain generation, drug action and pain perception. This model was tuned until simulations based upon it showed results apparently similar to available records of clinical demand analgesia. A scheme was developed, based on a novel non-linear Bayes' Rule estimator and an extended Kalman filter (EKF), to perform suboptimal estimation of the states of the model in the face of the strong non-linearities involved. Control laws based on these estimates were tested in simulation and promised to afford pain relief considerably better than that available with proportional control. Hardware was developed to implement on-line estimation and control, and was used to provide demand analgesia for a number of patients recovering from major orthopaedic surgery. These clinical trials gave encouraging preliminary validation of the improvement in pain relief offered by sophisticated control schemes.

1.2.3.2 The Model - Fig. 1.2 is a block diagram of Reasbeck's mathematical model of patient and pain. It shows that analgesic drug infused intravenously affects the blood plasma level and thus the receptor sites to give a pain-killing effect referred to as 'comfort' c and measured in units (p) defined below.

Reasbeck's work assumed that drug concentration at the receptor sites in brain and spinal tissue was identical to

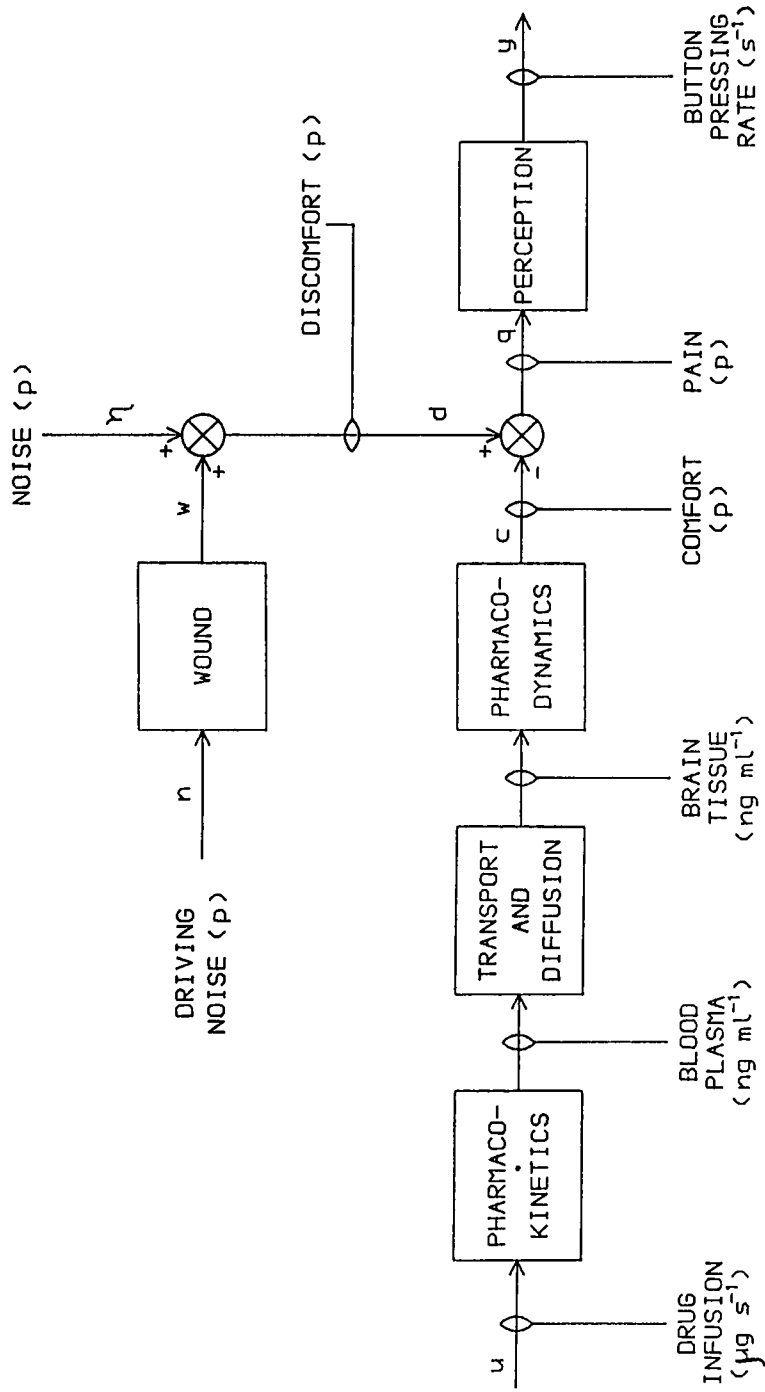


Fig. 1.2 Structure of the Mathematical Model

plasma level, i.e. that the block labelled 'transport and diffusion' in Fig. 1.2 could be ignored. The pharmacodynamic relationship between plasma level and comfort is modelled as a linear, time-invariant function, the patient-dependent slope of which is called 'relief'. The model accounts not only for comfort but also for the 'discomfort' d which is to be relieved. This is modelled as the output w of a Wiener process driven by white noise n , to which is added a further random disturbance η . In the absence of any published information on the subject, it is assumed that pain is perceived via a variable q which is the arithmetic difference between comfort and discomfort:

$$q = d - c \quad (1.1)$$

The variables c , d , q are measured in units called 'pangs' (p), defined as 'the amount of pain which, when perceived by button-pressing, as here, produces presses at the rate of 1 s^{-1} '. The output from the model is the patient's response y , measured in button-pressing units (s^{-1}), to his perceived pain q . The mechanism relating y to q includes a demand nonlinearity by which y is zero for all negative values of q .

1.2.3.3 Realisation of Model - The model is of a continuous-time system. Reasbeck's work used discrete-time approximations throughout, since they provide a more suitable framework for real-time work with digital computers - an approach which has been continued in this thesis. One of the crucial points about demand analgesia is that the system should use the fast-acting i.v. route of drug administration, to permit as rapid a response as possible to demands when the patient is in pain, which in turn dictates an upper limit of the order of one minute for the sample interval of a control algorithm. Reasbeck used a sample interval of one minute, which fulfils this requirement but also allows sufficiently complex calculations to be carried out in real time with microprocessor-based equipment.

For the purposes of simulation, estimation and control the model was formulated in terms of a five-dimensional state space, the state vector x being composed as follows:

1. x_1 - blood plasma level of drug concentration.
2. x_2 and x_3 - auxiliary states to implement 3rd order pharmacokinetics.
3. x_4 - the wound w .
4. x_5 - 'relief'.

In terms of these variables comfort c and discomfort d are given by

$$\begin{aligned}c &= x_1 x_5 \\ d &= x_4 + \eta\end{aligned}\tag{1.2}$$

The corresponding dynamical equations can be written in discrete-time vector-matrix form

$$x(i+1) = Ax(i) + bu(i) + \xi(i)\tag{1.3}$$

where the state transition matrix A is linear, b is a column vector relating drug input u to the states x_1 , x_2 and x_3 , and ξ is a column vector containing the noise n which drives x_4 . The output equation for net pain

$$q = x_4 - x_1 x_5 + \eta\tag{1.4}$$

is nonlinear, due to the term $x_1 x_5$ which is a product of two states. This nonlinearity led to the use of an extended Kalman filter (EKF) rather than a normal Kalman filter. The separate Bayes' Rule estimator was used to handle the demand nonlinearity relating q and y , providing a pseudo-measurement for q (on the basis of real measurements of y) to the EKF. Assumed modelling errors were handled explicitly, with the diagonal terms of the covariance matrix being corrected every sample interval. Details of the non-linear state estimation scheme are summarised in [Jacobs et al., 1985], which is

included as Appendix A of this thesis.

1.2.3.4 Limitations - Reasbeck was unable to collect sufficient clinical data against which to tune and compare the behaviour of the mathematical model. This was primarily because the hardware with which the trials were carried out was unsuitable for prolonged clinical use; it was cumbersome, complicated, expensive and unreliable. For the same reason it was not possible to provide a true clinical validation of the superior performance of Reasbeck's controllers. Furthermore there were errors and inadequacies in the mathematical model, particularly concerning the representation of pharmacokinetics, pain perception and inter-patient variations, which were found in retrospect to undermine the usability both of simulation results and of the controller designs.

1.3 OTHER RELATED WORK

Clinical investigations of demand analgesia have been carried out by researchers in England [Evans et al., 1976; Slattery et al., 1983; Hull and Sibbald, 1981], Germany [Lehmann, 1984; Lehmann, 1985a; Lehmann, 1985b], Sweden [Tamsen et al., 1979; Tamsen et al., 1982; Dahlstrom et al., 1982] and the United States [Keeri-Szanto and Heaman, 1972; Keeri-Szanto, 1976; Graves et al., 1983; Bennett, 1985a]. Published reports from these sources have provided useful details of demand rates, plasma levels, pain scores and many other aspects of clinical experience. These have been used both to guide the work

described in this thesis and as yardsticks against which to compare the results obtained. One of the most comprehensive studies to date is that of K.A. Lehmann, who has compared the performance in demand analgesia of 11 different drugs, with 40 patients per drug [Lehmann, 1984]. Most work has been carried out in postoperative pain, though demand analgesia has also been used successfully in care of chronic pain [Bennett, 1985b] and in obstetrics [Barrier, 1985].

Work by researchers in pharmacokinetics and pharmacodynamics has been important to this thesis both for the background to these topics and for specific results, eg. [Godfrey, 1983; Garrett et al., 1967; Holford and Sheiner, 1981; Hull, 1979; Mather et al., 1975; Paalzow, 1982; Stanski et al., 1978; Wagner, 1968; Wagner, 1971].

Various papers on psychophysics were referred to when investigating the demand nonlinearity. These include [Adair et al., 1968; Cross et al., 1975; Sternbach and Tursky, 1974; Stevens, 1957; Stevens and Greenbaum, 1966].

No work other than Reasbeck's and that described in this thesis appears to have been undertaken in explicit application of control engineering methodology to the control of pain. There is however a body of published work on feedback control of other physiological variables such as blood pressure and muscle relaxation, eg. [Arnsperger et al., 1983; Auer and Rodler, 1981; Berger and Brown, 1984; Collins and Arzbaecher,

1979; Linkens et al., 1982; Rametti, 1985]. Many of these applications have in common with demand analgesia the constraint of non-negative controller action, as well as non-linearities and uncertainties of varying severity, but all of them differ from demand analgesia in controlling a variable which, unlike pain, can be measured in a relatively well-defined way.

1.4 OBJECTIVES AND OUTLINE OF THIS THESIS

The ultimate aim of the research is to provide better understanding and relief of pain, adapting to the needs of individual patients, by using the contributions of control engineering to design optimal demand analgesia systems. The objectives for this thesis stated at the start of the Author's work were

1. to develop new clinical hardware for implementation of model-based control and for collection of real patient data
2. to improve and tune the mathematical model on the basis of clinical work and off-line simulation studies
3. to develop new model-based controllers and investigate their performance

Objectives 1 and 2 have been achieved successfully. Foundations have been laid for the third objective, but this is of sufficient scope for a separate thesis and is commended

to future researchers.

Chapter 2 of the thesis describes the hardware and software design of new demand analgesia equipment, 'PRODAC', which has been fully successful in clinical practice. Chapter 3 describes initial simulation studies which investigated the performance of different control laws and led to changes in the modelling of the demand non-linearity. The first clinical trial to be carried out using PRODAC involved 60 patients suffering from postoperative pain. Its results, which yielded new information about the influence of upon drug consumption of body weight and operation type, are described in Chapter 4. The representation of pharmacokinetics was enhanced by modelling the transfer of drug between plasma and tissue, and using a novel method to derive the time constant of this transfer for morphine. A scheme for on-line estimation of pharmacokinetic time-constants was also investigated, and these results are reported in Chapter 5. New off-line software, needed for various reasons, is described in Chapter 6, while Chapter 7 concerns the task of using it to tune the revised model to the clinical results of Chapter 4. The model now exhibits life-like behaviour, which was not the case previously. A second clinical trial was carried out and is reported in Chapter 8. It provided large numbers of blood samples, using which estimated plasma levels during demand analgesia could be compared with measured values. Finally, Chapter 9 summarises the achievements of the thesis and makes suggestions for further work.

A number of publications have arisen directly from the work described in this thesis, including [Jacobs et al., 1983; Jacobs et al., 1985; Lammer et al., 1985; McQuay et al., 1985; Jacobs and Lammer, 1986]. Aspects of the work have been presented at the First International Workshop on Patient Controlled Analgesia held at Leeds Castle in 1984, at a subsequent international symposium in London and at an Institute of Measurement and Control conference held at Sheffield University in 1985.

CHAPTER 2

NEW DEMAND ANALGESIA EQUIPMENT

2.1 INTRODUCTION

New clinical hardware was needed for the research to make progress. The Author designed and built 'PRODAC', a microprocessor-based demand analgesia device. The prototype was fully successful, and a further three units have been built to date. This chapter describes

- The reasons for needing new hardware and the requirements of it.
- Details of the design and considerations which influenced it.
- PRODAC's operating software.
- Features of PRODAC's use.

2.2 SPECIFICATIONS AND REQUIREMENTS

2.2.1 Reasbeck's Hardware

The hardware used for Reasbeck's clinical work comprised six separate items: an RML 380 Z microcomputer, a double 8" floppy disk drive, a video monitor, a keyboard, an 8085-based pump controller in a 19" rack, and a Vickers Treonic syringe pump.

The Vickers pump was mounted on the patient's bed, and the other components, connected by cables, were mounted on a mobile trolley. The equipment was powered from the hospital mains supply. Control programs were written in Control Basic, with some machine-code subroutines to drive the pump controller. The operating system (CP/M), the Control Basic interpreter and the programs were all loaded into the computer from floppy disk at the start of a session, and the data generated during a session were stored likewise on floppy disk.

2.2.2 The Need for New Hardware

The old hardware, as described above, was convenient and suitable for initial studies; it was made largely of off-the-shelf components and used a convenient general-purpose interpretive language. Data-storage was made easy by the use of a general-purpose computer with disks, and modifications to the control algorithm could be made in situ. However it became clear during the course of the approximately 30

clinical trials carried out by Reasbeck with this apparatus, that it was unsuitable for extended clinical use, for a number of reasons:

- Reliability: On several occasions clinical trials had to be ended prematurely, either because of a hardware fault or because the control program crashed. This was unacceptable in a clinical environment and impeded useful research. The unreliability can probably be attributed to inelegant and superfluous complexity in the hardware, and to dependence on the notoriously noisy hospital mains power supply.
- Ease of Use: The apparatus was not easy to use, especially for anyone not well acquainted with running Control Basic programs on the 380Z. This was particularly true when problems arose, as it was easy to crash the program by pressing the wrong key inadvertently. When the program did crash, it left the clinician uncertain as to what had happened, how much drug had been given, and how to proceed. It was out of the question for normal nursing staff to be left in charge of the system.
- Physical Portability: Two main problems of portability arose with the apparatus. It could only be moved on a trolley and could not be carried without being dismantled completely. This meant that it would have been almost impossible to use the

apparatus in different hospitals when this was desired. The second major problem was that the patient could not be moved from one ward to another without unplugging the apparatus from the mains and thus breaking off the trial (which could not then be continued).

- Cost: The apparatus was too expensive (approx. £5000 in 1979) for more than one set to be used, thus greatly limiting the extent of the clinical trials and research.

For these reasons it was decided that new hardware needed to be designed and built, before further research in demand analgesia could proceed.

2.2.3 Requirements for the new Hardware

It was agreed by the Author and his supervisors that new hardware should satisfy the following general requirements:

- Control function: A similar overall control function to that of the old hardware should be provided. Inputs to the device should include operator instructions from a keypad and patient response in the form of button pushing. Outputs from the device should include messages to an alphanumeric

display, visual and audible warning devices, hard copy to a small printer, a serial interface for data capture, and the control output in the form of driving-signals for a Graseby Dynamics MS 16 Syringe Driver.

- Size: All components of the new device, excepting the syringe driver, should be contained in one housing, and the device should be easily portable.
- Power supply: The power supply should be designed to allow the device to be disconnected from the mains for up to 24 hours without the running being affected. This implies that CMOS circuitry should be used to minimise power consumption and thus the size of rechargeable batteries.
- Ease of Use: The hardware and software should be designed to be intrinsically fool-proof and easy to use. It should be impossible for the device to fail unsafely, or for the operator to crash the software.
- Data retention: The device should be able to store data (i.e. the results of clinical trials) in battery-backed static RAM, and be able to transfer such data to another computer days or weeks after the trials.

- Programmability: It should be easy to program the device in a high-level language on an off-line computer, with the program then being installed in the form of EPROM chips.
- Cost: The components for each device should cost less than £1000.
- Standards: The hardware must satisfy the IEC standard for medical electrical equipment, as detailed in IEC Publication 601-1. This is equivalent to BS 5724.

2.2.3.1 Choice of Single Board Computer - It was decided that the hardware should be based around a commercially available single-board computer, which should be able to fulfil as many of the requirements as possible, thus minimising the need for additional hardware. The following criteria were specified:

- All integrated circuits should be CMOS, to allow sufficiently low power consumption
- The CPU should preferably be software-compatible with the Z-80, to allow maximum flexibility in the choice of programming language and software development tools
- The board should accommodate up to 64 kByte of mixed RAM and EPROM for program and data storage. This size should be ample for future requirements.

- A real-time clock (time-of-day/date) should be provided
- The board should provide a UART for serial communications and at least 16 lines of parallel I/O for sundry functions
- It should be possible to battery-back the RAM and real-time clock when the other components on the board are powered down, in order to keep time and preserve data in RAM

Over forty single-board computers on the British market were considered as candidates, but none fulfilled more than three of the above requirements. However, the Author visited the Electronica '82 fair in Munich and found a German board, the XMOS 801 made by ELSA GmbH, which satisfies all the above criteria. It also provides low-voltage interrupt generation, two 16-bit counter-timers, a barcode-reader interface and an on-board backup battery, as well as a number of other features which allow great flexibility in tailoring the board to the user's requirements. It is in standard 'eurocard' format, 160 x 100 mm. This board was duly bought, and forms the heart of the new hardware.

2.3 PRODAC - DESCRIPTION OF HARDWARE

Circuit diagrams for PRODAC are to be found in Appendix B.

2.3.1 General construction

PRODAC is enclosed in a commercially available steel case with front and rear panels of aluminium. The display and keypad are mounted on the front panel, and are covered with an additional Perspex window which has a cutout allowing access to the keypad. This arrangement protects the front panel from splashed liquids - a precaution which has proved to be worth taking. The back panel has a mains connector, fuses and switch, and an ammeter to show the charging current of the battery. Also on the back panel are sockets for the syringe driver, the patient button and the RS232 connector, as well as a key for switching PRODAC on. The printer paper emerges through a slot. Figs. 2.1 and 2.2 show the front and back panels respectively.

A rectifier and regulator for charging the lead-acid battery are mounted on a separate printed circuit board. The printer is at the rear of the case mounted on the inside of the back panel. The XMOS 801 single board computer is mounted horizontally on an aluminium chassis-plate, and above it is a second eurocard with most of the remaining circuitry. Fig. 2.3 shows the internal layout.

This second, specially built circuit board is connected to the



Fig. 2.1 PRODAC Front Panel



Fig. 2.2 PRODAC Back Panel

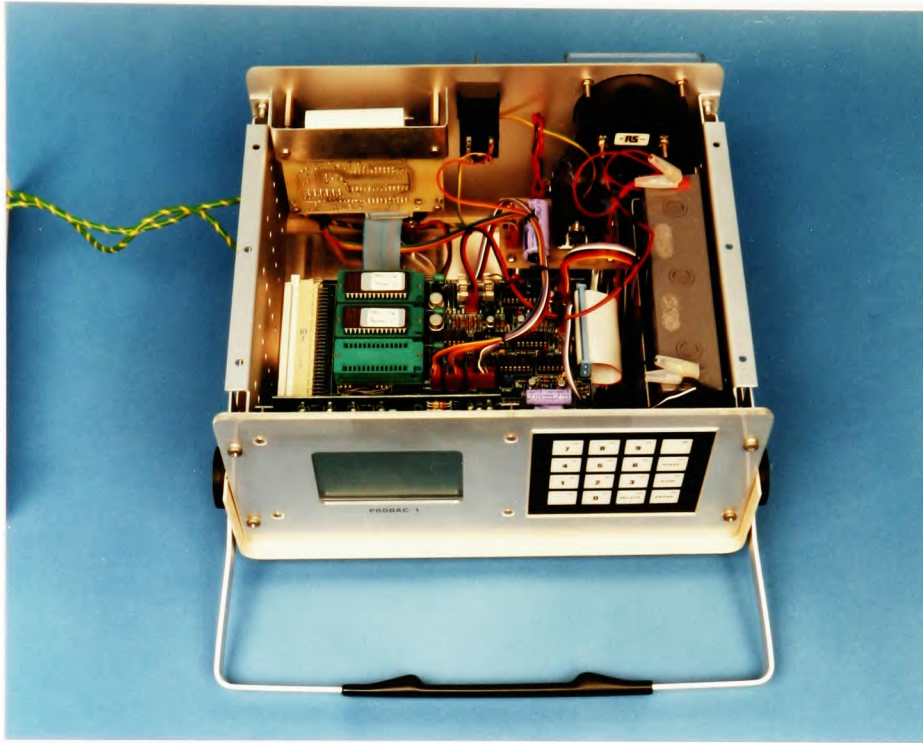


Fig. 2.3 PRODAC Internal Layout

XMOS 801's main bus using 64-way cable, and to its parallel and serial I/O using 40-way cable. All other components such as the display, the printer, the syringe driver etc. connect to the second card. This card also has three zero-insertion-force (ZIF) sockets mounted on it, which are mapped onto the address/data bus of the XMOS 801 below: this allows easy changing of the EPROMs containing the control algorithm, without dismantling the whole apparatus or damaging the integrated circuits.

For the prototype PRODAC the second board was constructed in a 'breadboard' manner, to allow the many changes and modifications necessary as the design evolved. When the prototype had seen extensive clinical use and the design was sufficiently stable, a proper double-sided printed circuit board was designed. This is now used in all existing PRODACs.

2.3.2 Button Interface

The feedback signal in most demand analgesia systems comes from a button pressed by the patient when he is in pain. In the hardware used by Reasbeck there were numerous problems concerned with obtaining accurate readings of the number of times the patient consciously pushed the button. On various occasions the numbers of button-presses measured by the hardware were orders of magnitude higher than the true value, leading to potentially dangerous situations in which the amount of drug delivered was greater than the correct dosage. This was attributed to inadequate debouncing of the button, and the circuitry was modified by Reasbeck and others in a number of ways in attempts to prevent spurious button-presses being recorded. Standard debouncing methods such as the use of low-pass RC networks proved insufficient, and the modifications included the introduction of a minimum number of seconds which had to elapse after the button was pressed, before a second pressing would be registered. While not being completely successful, this measure certainly succeeded in reducing the recorded number of button-presses to a more realistic level. Unfortunately it thus also invalidated part

of the mathematical model of the patient, as used by the control algorithm's estimation scheme.

The crux of the problem in fact lay elsewhere: in the type of button used. This was namely of the single-pole-single-throw type, with values of logic 0 and logic 1 being generated from the single contact by using a pull-up resistor. Problems arise with this sort of switch when the button is not pressed cleanly and firmly but in a slow, wavering manner, and in particular when the switch is held by partial pressure very close to the make-or-break point. This can easily occur if, for example, a patient falls asleep with the button half pressed. In this situation the slightest movement or vibration can repeatedly cause very large numbers of recorded presses, and even the use of a dead time does not solve the problem.

The solution used in PRODAC replaces the single-pole switch with a single-pole-double-throw switch using the circuit shown on page 6 of Appendix B. This circuit is not vulnerable in the same way to marginal contact, because a change in the output can only be effected by the switch moving all the way from one contact to the other - a mechanical hysteresis of about 3 mm. Not only is the problem thus solved, but the circuitry required is also much simpler and more economical. The circuit is taken from a design found in [Horowitz and Hill, 1980]. The debouncing method can only be implemented as neatly as this by using CMOS, due to a) the symmetry of a CMOS

output's ability to source or sink current and b) the output resistance of a CMOS gate (of order 200 ohms), which saves the gate from self-destruction when it is short-circuited momentarily. The button presses are counted by connecting the debounced output of the switch to an interrupt line.

2.3.3 Syringe Driver Interface

In the old hardware the syringe containing analgesic drug was driven as follows: the main computer (380 Z) sent commands to an 8085 microprocessor via a serial line. These commands were read and interpreted by a slave program running on the 8085, which in turn sent infusion-rate data to the Vickers Treonic syringe driver, using a parallel I/O port. One possible hazard with this scheme was that if the 380 Z software crashed for any reason (as happened relatively often), the syringe might continue to be driven at the rate set by the last received command until it emptied or the circuitry was switched off by an operator. This was overcome by making the 8085 switch itself off automatically if no new command was received within 90 seconds of the previous command. If the syringe emptied or jammed, the Vickers syringe driver sent a signal to the 8085 board, which in turn sent a message to the 380 Z. Quite apart from the expense and physical bulk of this system, the complexity of having three autonomous control devices daisy-chained together led to inherent unreliability and loss of direct control. No feedback was possible to tell the 380 Z how much drug had actually been infused in any particular minute.

With the new hardware a new syringe driver was chosen - a Graseby Dynamics MS 16, shown in Fig. 2.4, which is much smaller, cheaper and less power-consuming than the Vickers pump.

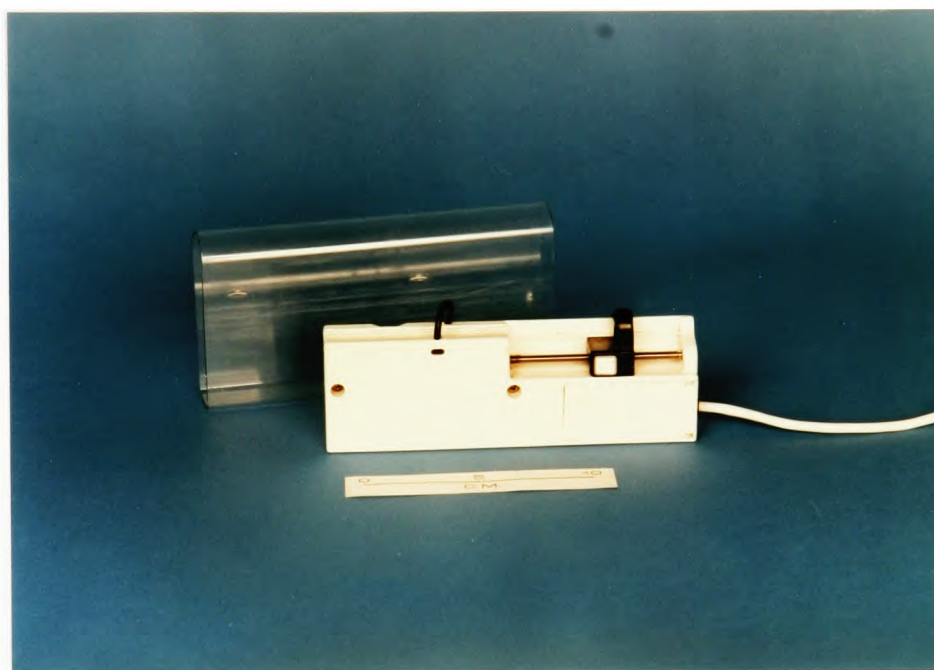


Fig. 2.4 Graseby MS 16 Syringe Driver

The MS 16 has a CMOS control circuit which drives the syringe at a rate set by two screws on the outside of the housing. The range of infusion rates offered is from 0 to 99 mm hr⁻¹ plunger travel. The MS 16 uses a standard small dc motor to drive a lead screw, which moves the plunger of the syringe. A cam on this lead screw opens a microswitch three times per revolution to feed back information on how far the plunger has moved. This means that the MS 16 moves the plunger in

discrete increments corresponding to approximately 1/3 revolution of the lead screw. The lead screw thread is such that one revolution corresponds to approximately 0.75 mm plunger travel.

The initial proposal was to make use of the MS 16's own control circuitry, with a two-digit BCD (binary-coded-decimal) interface to determine the rate of infusion. This would use eight lines of the XMOs 801's parallel I/O, with an additional line being used for feeding back the status of the MS 16 (enabling detection of an empty syringe). This proposal was rejected for two reasons.

The main one is as follows: when a low infusion rate is set, the mean time between incremental plunger movements is usually much greater than one minute, and in general any specified infusion rate can only be realised with any degree of accuracy by leaving it set for a period of several minutes. However, as explained in Section 1.2.3.3, there is an upper limit of the order of one minute for the sample interval of a control algorithm. A suitable algorithm would therefore not only be unable to specify meaningfully a particular infusion rate for any particular minute, but in addition there would not even be any feedback as to how much drug actually had been infused in that minute.

The second, less damning reason is that the system would use an excessive number of parallel I/O lines needed for other

tasks, and would necessitate the physical complication of using 12-way cable between the syringe driver and the main equipment housing.

For these reasons a different approach was adopted, offering direct control of the syringe driver by the main computer, and using an inherently safe and simple scheme whereby the syringe driver is switched on by software and switched off by hardware. The controlling software runs as a concurrent process invisible to the main control algorithm. The details of this scheme are as follows:

The circuit schematics of the system are shown on page 3 of Appendix B. The motor of the syringe driver is switched by the transistor T1, which in turn is controlled by the flip-flop made of two NOR gates. To set this flip-flop and start the motor running, the NSC-800 writes a low-high-low pulse to port pin PB1 of the NSC-810; this triggers the monostable, which in turn sets the flip-flop. As the motor runs, the output shaft of its gearbox turns the main lead screw and with it the cam, which operates the microswitch S1. This switching action is debounced by the RC network and Schmitt trigger, and triggers the second monostable to reset the flip-flop and thus switch off the motor. The lines to port pins PB2 and PB3 allow the computer both to detect whether the pump is running, and to disable it if required.

The software to drive the system works in the following

manner. Once every minute, the main algorithm goes through one iteration and calculates the control output for the next minute, in terms of the (integer) number of incremental plunger movements to be made in that minute. This number is made available to an interrupt routine which is entered approximately once every 0.5 seconds. The routine starts the motor by pulsing port pin PB1 and sets a software flag. On subsequent entries, the interrupt routine checks to see whether the motor has switched itself off by examining pin PB2. If this is the case, the number of plunger movements still to be made is decremented by one and the number of movements made successfully is incremented. If on the other hand the motor has still not switched itself off after five seconds, the interrupt routine disables it using pin PB3, and sends a message to the display asking for the syringe to be refilled (the motor jams when the syringe plunger reaches the end of its travel). By this method it is possible to feed back to the main algorithm how many plunger movements were made, and thus how much drug was delivered in the previous minute, with maximum accuracy. This ability is vital to any estimation/prediction scheme which bases its estimates in part on the past control output, as well as for the implementation of cumulative safety levels for drug dosage.

This system is also robust from a safety aspect; if the software crashes, the syringe driver is automatically switched off by the hardware action of the cam, whereas if the switch-off hardware should fail for any reason, the driver is

switched off from software exactly as if the syringe had come to the end of its travel. It is therefore not possible for a single failure to endanger the patient, and the probability of the system failing in both ways simultaneously is extremely low. Protection is offered against hardware failure by using two transistors in series, rather than one, to switch current to the syringe driver motor. A further safety measure can be imposed by making sure that the total amount of drug in the syringe at any moment is less than the lethal dose.

2.3.4 Display

The task of the display in PRODAC is twofold; it must support dialogue with the operator, during which data can be entered, and it must display up-to-date information while the control function is being performed. These requirements are such that it must be possible to display the full range of normal ASCII characters. Since the up-to-date information during a run should ideally be displayed all the time (rather than just on demand by the clinician), the display should consume very little power. This effectively rules out the use of plasma displays, light-emitting-diode arrays or cathode ray tubes, leaving only one choice: a liquid crystal display. After consideration of the wide range of LCD displays on the market, an Epson EA-Series model was chosen. This can display eight lines of twenty characters each, and is equipped with its own RAM and ROM as well as two LSI (large-scale-integration) controller chips, all in CMOS. It can be mapped directly onto the bus of a microprocessor system, occupying four bytes of

memory or I/O address space. A certain amount of software complexity is required to drive the display, due to the way characters are mapped by the two LSI controllers onto the LCD panel, but all character generation is performed automatically and there is a standard ASCII interface for the characters themselves.

In PRODAC the display has been mapped in the I/O locations 00 to 03 inclusive. The contrast and viewing angle of the display can be set using the potentiometer connected to terminals VR1 and VR2.

2.3.5 Keypad

It is necessary for a clinician to be able to enter data and make commands. The most flexible way of achieving this would be the provision of a general-purpose keypad, rather than a separate key for every possible function. This was the approach used in the 380 Z system, in which all commands were typed as words on a standard alphanumeric keyboard. The Author felt that a better and more foolproof scheme would be to use a small keypad offering the digits 0 to 9, 'delete' and 'enter' keys for correcting mistakes and entering data, and 'start' and 'stop' keys to control the overall behaviour of the system. This approach has also the advantage that a small keypad takes up less physical space, and is easier to handle as an I/O device than a full alphanumeric one.

The keypad chosen is of the membrane variety, offering a

matrix of 4-by-4 unlabelled keys, a waterproof surface, and a total thickness of only 1 mm, which allows it simply to be stuck to the front panel of the hardware next to the display. It is made by W.H.Brady GmbH.

Two main approaches are possible in designing software to handle input from a keyboard. The first is to poll the keyboard repeatedly, typically at a rate of the order of 50 Hz. In such a system the keyboard is often scanned using the pins of a parallel I/O port. The second approach requires the keyboard to be scanned and encoded by an autonomous device, which can interrupt the main processor when a key is pressed. The first system is often used when there is to be a great deal of keyboard activity, or when no interrupt facility is available, but has the disadvantage that it not only uses a large number of parallel I/O lines, but also overloads the main processor with an additional task. The second system has the disadvantage of requiring extra hardware, but is otherwise a more elegant solution, especially as it does not distract the main processor unnecessarily from more important things during long spells in which no keys are pressed. A suitable single-chip CMOS encoder for 4-by-4 keypads is available, and the NSC-800 has five interrupt lines, so the second approach was adopted with no reservations. It was decided to access the encoder chip using five I/O lines from the NSC-810 rather than mapping it directly onto the system bus, as the first is the simpler and more flexible solution. It can be seen in the circuit schematics that the 'data available' line from the

encoder chip is connected to the NSC-800 interrupt line through a monostable: this is to avoid holding the interrupt line low for the full time that a key is held pressed. There is no tactile feedback of key-movement, but this is compensated by the software generation of audio feedback using the buzzer described in Section 2.3.8 below.

2.3.6 Data Storage

The data-storage facility is implemented using three 8 KByte static CMOS RAM chips, located on the XMOS 801 board. This memory, like the real-time clock, is doubly battery-backed so that data are preserved when PRODAC is switched off. Some of the memory is used by the system for stack space and variable storage, leaving approximately 22 KByte for patient records: this is more than adequate for up to 50 typical patients.

2.3.7 RS232 Interface

An RS232 interface is needed for communication of the internally stored data to another computer (q.v. Section 2.5.3 below). It is also used in the development stage to allow communication between a monitor program running on the XMOS 801 single board computer and a standard terminal. The XMOS 801 is equipped with a CDP 1854 UART, which performs most of the functions required. Additional circuitry, as can be seen in the schematics, is necessary to generate RS232 voltage levels for transmission, as well as buffering the received signal. Included in the RS232 interface is the

ability to detect a short circuit between pins 24 and 25 of the standard D-type connector on the back of the case; this allows PRODAC to detect the presence of a data capture computer and enter a suitable mode for transfer of data. The cable used to connect the two devices must have pins 24 and 25 (which are normally unused) linked together.

2.3.8 Warning Devices

Two warning devices are incorporated into PRODAC: a yellow light-emitting diode on the front panel and an audible piezo buzzer mounted inside the case. The l.e.d. and the buzzer are used to attract attention from the nursing staff when the syringe empties, or if any other condition requiring attention should occur. The buzzer also provides audio feedback for the keypad, making it much easier and more positive to use.

The l.e.d. and the buzzer are both switched on and off using transistors controlled by port pins of the NSC810.

2.3.9 Printer

A small dot-matrix impact printer, has been incorporated in PRODAC. This is to provide an additional hard backup copy of the cumulative volume of drug delivered, hour by hour. The printer used has 24 columns and can print the full range of ASCII characters as well as graphics. It is controlled by a single-chip microcomputer, mounted with the printer to form a single unit, which accepts serial or parallel input. In this

application it is driven in serial mode, using a port pin of the NSC810. Two further pins are used as well; one to detect whether the control chip is busy and one to switch the main power supply to the control chip, which is only switched on when the printer is required.

2.3.10 Power Supply

PRODAC is designed so that it can be used either in conjunction with the mains, or powered just by its internal rechargeable battery. It should be possible to switch the mains supply on and off without causing any disturbance to the operation of the device; only the charging of the battery is affected.

To avoid the very difficult problems of restartability in a system such as this, a policy of hard switch-on and soft switch-off has been adopted: the device is switched on by pressing a button on the back panel, and can only be switched off by the main processor. Using this approach makes it impossible for PRODAC to be switched off accidentally during a run, and guarantees that the internal records can be maintained correctly. The switching circuit which performs this function was developed from an original design by Dr. J. Hruska and is shown on page 8 of Appendix B. An important aspect is that a clinician can nevertheless stop and restart delivery of the drug at will during a run, using the STOP and START keys on the keypad.

The 6V lead-acid battery has a capacity of 5.7 ampere-hours, which allows PRODAC to run independently of the mains for approximately two days. A lead-acid battery was chosen in preference to nickel-cadmium cells because of the much lower self-discharge rate, as well as the robust charging behaviour. The battery is of fully-sealed construction, with no danger of leakages.

PRODAC conforms with the IEC regulations for medical electrical equipment as set out in IEC Publication 601-1.

2.4 PRODAC - DESCRIPTION OF SOFTWARE

2.4.1 Choice of programming language

The programming language used for generating the software to reside in EPROM on PRODAC has to fulfil a number of requirements; not only must it allow high-level formulation of complex mathematical algorithms in such a way that they can easily be understood and modified, but it must also permit easy access to assembly language subroutines, while at the same time being close enough to the machine to minimise the need for using these. This is because the PRODAC software has to provide operating system functions, such as driving the display and keyboard, as well as running a control algorithm. A standard language which is easily portable from one processor to another would have the advantage that any successor to PRODAC using another processor could make use of the existing software with minimum modifications. A further

advantage of portability is the possibility of testing parts of the software, such as the control algorithm, on a large computer such as the VAX 11/780, which should allow simulations to be carried out much faster than on a microcomputer. The language must support the use of floating-point arithmetic and transcendental functions, and must run fast enough on the NSC800 at 2.5 MHz to satisfy the requirement that the control algorithm should complete one iteration in less than one minute. Finally, the language must be able to produce ROMable code! (Some compilers for certain languages such as Pascal produce code which modifies itself during execution, and can therefore not run in read-only memory).

Three main candidates for the language were considered: Control-Basic, a real-time version of interpreted BASIC; FORTH, a threaded interpretive language using reverse polish notation; and 'C', a compiled structured language. The initial choice was for FORTH, but this proved to be a mistake since the difficulty of reading and writing complex mathematical expressions outweighed by far any of the advantages offered by the language. When this became clear, C was chosen instead and was found to be suited ideally to the task.

C is a medium-level, general purpose language which was initially designed for writing compilers and operating systems. It is not only very fast in execution but also

highly portable. A standard C is available for the VAX 11/780. After carrying out a survey of the various C compilers for Z 80 systems, the Aztec C II package was bought from Manx Software in the USA. This can be used in conjunction with a standard Z 80 assembler and linker - a necessary feature, since some of the low-level routines governing PRODAC's operation must be written in assembly language.

2.4.2 Software development

The tool acquired for developing the PRODAC software is a DEC Rainbow microcomputer. This is a dual-processor machine which can run Z 80 or 8086/88 software, using a special version of the CP/M operating system, and can also be used as a VT102 terminal. It has twin floppy disk drives which give a total storage capacity of approximately 800 kByte. A memory expansion board, which increases the amount of user RAM to 128 kByte, was bought to allow adequate working space for the C compiler.

The Rainbow has been used primarily as a vehicle for writing, compiling, assembling, linking and testing C programs. The resulting object code can then be transferred to an EPROM programming device and blown into suitable EPROMs, which are finally plugged into the ZIF sockets on the top circuit board inside PRODAC.

A permanent link to the Departmental VAX 11/780 has

facilitated the transfer of programs between the two computers (q.v. Section 6.2.2.5).

In the initial stages of development the Rainbow was also used as a terminal for the machine-language monitor bought with the XMOS 801 single board computer. This was for developing and debugging the software which interacts directly with the PRODAC hardware, such as the routines which handle the display, the keypad and the syringe driver. This approach is by far the best way of developing such assembly language routines when no In Circuit Emulator is available, since it allows control over the microprocessor in an interactive way.

2.4.3 Structure of the Software

A full software listing can be found in Appendix C. PRODAC's software can be divided into two basic parts:

1. System software, which is specific to PRODAC. This consists of declarations of variables, initialisation routines, hardware-dependent I/O drivers, interrupt routines, the main operating software and code managing the storage of data in RAM. All these elements of the software are designed to remain the same, independent of the control algorithm.
2. Control algorithm software, which is common to PRODAC and to the off-line simulation program described in Chapter 6. This includes declarations of variables and code for functions concerned with the control

algorithm. It is the part of the software which needs to be changed when, for example, changing the safety limits for drug infusion or replacing proportional control with a stochastic control algorithm.

Three crucial integer variables link these two parts of the PRODAC software:

1. 'demands' - the number of demands made, which is set by the system software and read by the control algorithm.
2. 'fixesdue' - the desired control output, in terms of the number of incremental movements to be made by the syringe plunger. This is stipulated by the control algorithm and acted on by the system software.
3. 'givenfix' - the actual control output, i.e. the volume of drug successfully infused, in the same units as above. This is set by the system software and read by the control algorithm.

The two latter variables can differ when, for example, the syringe empties or when the clinician requests a temporary suspension of drug delivery by pressing the STOP key.

The following descriptions are of the System software.

2.4.4 Initialisation

1. Hardware initialisation. The code which PRODAC starts executing on power-up is written in assembly language. It sets up the stack, provides interrupt vectors and context-switching code for the foreground routines, and initialises various hardware items such as the parallel I/O, the UART, the display and the real-time interrupt generation. It ends with a jump to the C function 'main()', following standard C convention [Kernighan and Ritchie, 1978].
2. Software initialisation. This, written in C, is contained in the function 'initial()' and consists mainly of initialisation of variables used by the system.

2.4.5 Background Software (not interrupt driven)

It is common practice to give the patient an i.m. 'loading dose' of analgesic drug, after surgery but before the patient re-emerges from general anaesthetic, so that pain relief will already be effective when the patient awakes. PRODAC engages the clinician in a preliminary dialogue, asking to be told the code-number of the patient and the time since administration of any loading dose of drug: this information is stored with the patient records.

The main program loop is defined in the function 'body()',

which implements a one-minute sample interval. Once per minute it calls the control algorithm, handling the three variables described in Section 2.4.3 above, and invoking the record-keeping function as appropriate. Once per hour it prints out the new cumulative total dose on the built-in printer.

2.4.6 Foreground Routines (interrupt driven)

Four hardware interrupt sources are used in PRODAC, each serviced by a separate foreground routine.

1. Real Time Clock - The MM 58174A chip on the XMOS 801 single board computer generates an interrupt once every 31/60th second, which is serviced by the C function 'RSTA()'. This function is of particular importance as it controls the operation of the syringe driver, using the method already described in Section 2.3.3. It also updates the display of current time and of the total volume of drug delivered, as well as monitoring the keypad while the main control program is running. Only interrupt-driven software such as this can achieve the concurrency needed between the real-time tasks of computing a control output and of handling the syringe driver and keypad, without wastage of processor time: this has influenced the overall design of both hardware and software so as to maximise PRODAC's ability to implement control

algorithms with heavy computational requirements.

2. Power Failure - In the unlikely event of the battery voltage falling too low for PRODAC to continue operation, the function 'RSTB()' provides for a controlled shutdown. The interrupt is provided by the voltage watchdog on the XMOS 801, which triggers when the (nominally) 5V supply falls below 4.6V. The printer and syringe driver, if active, are switched off, the internal records are updated and a message on the screen requests the clinician to recharge the device from the mains supply. When this message is responded to, PRODAC switches itself off. This has never yet happened in the approximately 80 clinical trials using PRODAC.
3. Keypad - The function 'RSTC()' is called whenever a key is pressed. It maintains a small type-ahead buffer into which it inserts the new character; characters are removed from the other end of the buffer by the main software (eg. during the various dialogues) as and when required. Audio feedback is provided when a key is pressed, except when the control function is active.
4. Patient Button - When the demand button is pressed, the foreground function 'INTR()' increments a software counter for the number of demands.

2.4.7 Data Storage

Data are stored for each patient on a minute-by-minute basis, for various purposes. One is the collection of input/output information about the patient, for subsequent off-line use in tuning the mathematical model. The data are also used off-line for estimating time-histories of individual patients' drug levels, so that these can be matched to the results of blood sample assays. The most important data to store are the number of demands made by the patient and the amount of drug infused. Certain other data are stored too, to aid reconstruction from patient records of the course of events during any particular session. PRODAC automatically time- and date-stamps the individual patients' records.

The software is designed to allow PRODAC to store records for up to 50 patients. This provides a buffering which greatly eases the logistics of running clinical trials involving staff based in different places and with different duties; the task of data-collection can be completely asynchronous to that of providing demand analgesia for patients.

A compact form of storage is used, with allocation of the available memory as required. For each patient there is header information of fixed size and run-time data of variable size. The header information is composed as follows:

item	size (bytes)	meaning
----	-----	-----
code	3	patient identification code, as entered at the keypad by the clinician
stime	5	time of start of demand analgesia
sdate	8	date of start of demand analgesia
imdtme	5	time since loading dose, as entered by the clinician
duration	2	duration of run, in minutes (stored as a binary integer)

Two further items stored with the header information but used only by PRODAC are pointers to the first and last records associated with that particular patient.

In the standard configuration space is reserved for fifty such blocks of header information.

The run-time data are composed of 5-byte records, each associated with a particular minute. For minutes in which nothing happens, no record is kept: this is the main way in

which the storage is made efficient. Data records are composed as follows:

name	size (bytes)	meaning
----	-----	-----
minute	2	the time in minutes from the start of demand analgesia
dems	1	number of demands made during that minute
given	1	number of movements of syringe plunger made during that minute
events	1	a bitwise coded set of events: bit 7: syringe emptied bit 6: STOPped on request bit 5: reSTARTed bit 4: run terminated bit 3: power failure bit 2: unassigned bit 1: unassigned bit 0: unassigned

For any minute in which demands were made, drug was given or one of the listed events occurred, a new record is allocated from the memory still available.

2.5 USE OF PRODAC

2.5.1 Normal operation

Use of PRODAC is simple. The data management functions are completely invisible to normal users such as nursing staff, who are provided with a seven-page manual.

2.5.2 Housekeeping functions

PRODAC provides five special housekeeping utilities, which are invisible and inaccessible to the clinician. To access them, PRODAC must be switched on with a special cable, as described in Section 2.3.7, connected to the RS232 port. The presence of such a cable is detected by the initialisation software, which branches to the appropriate set of options. These are:

1. set the time
2. set the date
3. erase all records
4. dump all records
5. paper feed

2.5.3 Data capture

The tool used in this research for collection of data from PRODAC was an Epson HX20 portable computer. Any computer with an RS232 port and a suitable program could be used, but a portable computer has the advantage that it can easily be taken into a hospital ward. The HX 20 is light, compact, inexpensive and very versatile - it is ideally suited to this task. The data transfer programs are very short and simple, written in BASIC. They are normally stored in the HX20's battery-backed main memory, and patient records are loaded to and from the machine's built-in micro cassette drive.

The procedure for data collection is simple. First the RS232 cable is connected to PRODAC and to the HX20, and both machines are switched on. Option 4, to dump records, is selected on PRODAC and the data transfer program is run on the HX20. When this beeps to show that it is ready to receive data, the user presses PRODAC's 'START' key and transmission begins. Data transfer is at 1200 baud, using a software handshaking protocol. All transmitted data are in the form of printable ASCII characters - if desired these can be printed out on the HX20 screen or printer during transmission. When transmission is complete, the HX20 rewinds its tape and PRODAC asks whether the record should be retained or erased. It then returns to the service menu.

In the course of this research the data were transferred from the HX20 to the Rainbow and thence to the VAX for subsequent

use. This deployment of 'go-betweens' arose because the VAX used for off-line work was physically situated miles away from the hospital in which the trials took place. In other cases it would be simpler to connect PRODAC directly to the destination computer for data-collection.

2.6 SUMMARY

This chapter has described the task of designing PRODAC. The device has subsequently proved fully successful in clinical use, in its simultaneous roles as a reliable demand analgesia device suitable for use by ordinary nursing staff in the wards and as a research tool for data collection. Over 1400 patient-hours' clinical experience have been gained to date, and a total of four PRODACs have been built. Clinical experience is reported in Chapters 4 and 8. PRODAC has been presented on two occasions at international conferences, and is described in the first book to have been published on demand analgesia [Lammer et al., 1985].

CHAPTER 3

INITIAL SIMULATION STUDIES

In the period before PRODAC began to yield new clinical results extensive simulations of demand analgesia were carried out using adaptations of Reasbeck's Pascal program OXPANG [Reasbeck, 1982], running on VAX 11/780. A number of corrections were made to the mathematical model and to the program, and certain particular questions were investigated. All simulations were of patients with hip joint replacement operations, using fentanyl as the postoperative analgesic, and with a total simulated time for each run of 24 hours (3 hours asleep followed by 21 hours awake).

3.1 CHANGES TO OXPANG AND REASBECK'S MODEL

3.1.1 Patient Demand Non-linearity

The patient output, in the form of button pressing, is a non-linear function of the perceived pain, defined as the difference between comfort and discomfort. When the perceived pain is zero or negative, the patient makes no response, whereas a perceived pain greater than zero will cause the patient to press the button. In general - both intuitively

and from practical observation - the greater the pain, the higher the frequency with which the patient will press the button, but the precise form of this relationship is not known.

Reasbeck proposed in his thesis that for values of perceived pain greater than zero the nonlinearity took the form

$$Y = 2^{(q/a)} - 1 \quad (3.1)$$

where Y is frequency of button pressing
 q is perceived pain
 a = $1 / [\log_2 (n+1)]$
 n is the sample interval

This could be rewritten as

$$Y = (n+1)^q - 1 \quad (3.2)$$

and as such is evidently completely erroneous, since it makes the patient response an arbitrary function of the observation period n. It is not completely clear where this function came from, but its use in the model seemed to lead to good control in the clinical trials, and on this ground it was accepted. In mass simulation, however, it led to some ridiculous freaks including a fictitious patient who supposedly pressed the

button more than 10^{38} times in one particular minute.

Scrutiny of papers concerning psychophysical experiments on the perception of pain [Adair et al., 1968; Cross et al., 1975; Sternbach and Tursky, 1974; Stevens, 1957; Stevens and Greenbaum, 1966] suggested that the output function might have the form

$$Y = (q - q_0)^p \quad (3.3)$$

where q_0 is a threshold and p is a constant. However there is neither any conclusive evidence that p should take any value other than unity (values of p in various experiments, all of which concerned artificially induced cutaneous pain, varied between 0.8 and 1.5), nor any indication as to the value of q_0 in this case. It was therefore decided to use a simple function of the form

$$Y = a q \quad (3.4)$$

scaled by the constant a so that the values of pain q generated in simulation would correspond approximately with the demand rates y observed in clinical trials. It was found that this could be achieved by preserving the definition of the pang (i.e. 'the amount of pain causing button pressing at the rate of one press per second'), and scaling down numerical

values of pain and relief by a factor of 5.

This modest step was of some significance; for the first time in this research it was possible to assign non-arbitrary numerical values of pain to the wound and to the quantity 'relief'.

It is illuminating to examine why the false output nonlinearity, used by Reasbeck and described above, led to occasionally freakish simulations and yet inherently safe clinical trials. The reason is that in simulation the output nonlinearity is used both for generating levels of button pressing and for estimating from them the value of the perceived pain, whereas in clinical trials the button pressing was generated by the true output function (the patient himself), and the false nonlinearity was used only as a basis for estimation. In the latter context, the relationship is effectively reflected to give values of pain as a function of button pressing and as such, this particular output function acts as a limiting filter on the number of demands made by the patient; even extremely high demand rates are interpreted by the estimation algorithm to mean only moderate pain, without loss of sensitivity to slight pain, and the problem which arises in simulation (i.e. that conversely, moderate pain would cause extremely high demand rates) does not occur. It is therefore most unlikely that excessive patient response could lead to dangerously high drug delivery rates, and the success in clinical trials using this nonlinearity can

certainly be attributed in part to the way in which robustness and sensitivity were accidentally combined.

The corrections were incorporated both into the model and into the estimation scheme; not only does the simulation of patient behaviour use the demand nonlinearity, but the Bayes' estimation algorithm is also designed around it.

3.1.2 Randomisation Procedure

The role of the randomisation procedure in simulation of patients is to produce a realistic range of different patients. This is achieved by randomising certain parameters away from their nominal values, including the constants in the model of drug kinetics, the initial value of the wound, the wound time constant and the value of relief. The manner in which these are randomised must be subjected to constraints of physiological realism, for otherwise the resulting 'patient' is liable to exhibit unrealistic behaviour. This becomes particularly apparent in mass simulations, when one freak result can skew the overall statistics. Preliminary modifications were therefore made to the randomisation procedure to keep simulated parameter values within realistic limits. Later, when the new simulation program was written, this procedure was further modified as described in Chapter 6.

3.2 SIMULATION RESULTS

3.2.1 Batch Processing

The whole simulation program was restructured so that instead of each simulation being carried out interactively from a graphics terminal, a batch process could be submitted in which large numbers of simulations were made to take place, with the results being stored in a file.

3.2.2 Effects of Real Syringe Driver Behaviour

An important question arising from the limitations on the range of drug infusion rates, as described in Section 2.3.3, is that of the optimal drug concentration and syringe size to be used. If very concentrated drug is used, the time between syringe changes is long and therefore good, but the control is coarse and hence liable to be poor. If on the other hand the drug is made very dilute, then fine control can be achieved but at the expense of a very short syringe life, which is unacceptable in clinical practice. The physical dimensions of the syringe used also play a role here; a 5 ml syringe has a smaller cross-sectional area than a 20 ml syringe, and so a finer gradation of volumetric dosage is possible. However the syringe capacity determines the mean time between syringe changes for any given concentration, and this must also be taken into account. This question was investigated in simulation, to determine how concentrated the drug should be made without causing unacceptably bad performance, and how to choose the syringe size. The criteria of performance used

were the demand rate, the mean dosimetric infusion rate and the maximum blood plasma level of fentanyl, all of which should be low. The simulations were performed for Stochastic and Hybrid control. For each of these controllers, and for each patient, five simulations were carried out, each using a different dosage of fentanyl to correspond to each incremental movement of the plunger. These simulations suggested that the quality of hybrid or stochastic control might remain acceptably good, provided fentanyl delivery was not more than approximately 1 μg per plunger-movement.

The best approach was thus judged to be choosing the longest syringe which can be accepted by the driver, regardless of its diameter, and then adjusting the drug concentration such that one incremental movement of the plunger delivers 1 μg fentanyl; this should thus maximise the mean time between syringe changes while still permitting good control.

It should be noted that realistic modelling of the behaviour of the drug infusion device is crucial to the usefulness of simulations of demand analgesia. Similarly, it is dubious to attempt discrete-time control in demand analgesia if the discrepancy between the desired and actual amounts of drug infused in any particular sample interval is both unknown and ignored. The design considerations of Section 2.3.3 and the simulation studies described here therefore represent an important contribution to the quality of modelling and control

achievable by the techniques used in this research.

3.2.3 Comparison of Controllers and the Automatica Paper

To provide a statistical basis upon which to compare Reasbeck's clinical results with simulation results, and to clarify the theoretical differences in performance between various control regimes, mass simulations of 500 patients were carried out. The results were reported in Automatica [Jacobs et al., 1985] and are to be found in Table 2 of that paper, in Appendix A of this thesis. Each simulated patient was subjected to 5 different control regimes; the proportional, stochastic and hybrid controllers described in Reasbeck's thesis were used, as was an additional regime without feedback action which gave constant infusion of fentanyl at $40 \mu\text{g hr}^{-1}$. This was to give a better assessment of the performance improvements possible using feedback. The total computing requirement for such a simulation was approximately 20 hours' VAX CPU time (elapsed time approximately 3 days, depending on the CPU demand from other jobs).

Three results were recorded for each patient and each control regime: the average number of demands per hour, the average hourly infusion rate of the drug, and the peak blood plasma level (incorrectly tabulated in the Automatica paper as 'Max. tissue level') of fentanyl.

This indicates that both stochastic and hybrid controllers can offer a marked improvement in performance over proportional

control or simple infusion; the demand rates are significantly lower, corresponding to less pain, for only slightly higher mean drug infusion rates. It can also be seen that the stochastic and hybrid controllers give rise to lower variances in the demand rate, and higher variances in the drug infusion rate than do proportional control or constant infusion: this is synonymous with better control on the one hand and better adaptation to the patients' needs on the other, in accordance with the criteria listed in Section 1.2.1. The figures for maximum plasma level, which are not significantly higher for stochastic or hybrid control than they are for proportional control, suggest that the increased delivery of analgesic under model-based control can be achieved without increased risk of poisoning the patient.

However the Automatica paper was written early in 1983, and the results published there were obtained with a mathematical model which had never been tuned properly to clinical data, because such data were not available. These results must therefore be regarded as tentative, although they provide an indication of the potential improvements to be gained from using model-based control. The work described in the ensuing chapters lays a firmer foundation upon which to carry out simulation studies of this sort, but at the same time the revisions which have taken place have shown Reasbeck's controller designs to be conceptually wrong. The question of designing model-based controllers in the light of the revised model is discussed in Chapter 9.

CHAPTER 4

FIRST PRODAC TRIAL

4.1 MOTIVATION

Clinical trials were needed to provide real data against which to tune the mathematical model. Previous clinical data, described in [Reasbeck, 1982], were not sufficiently numerous to provide a solid statistical basis for carrying out this task. High inter-patient variability in almost every aspect of the mathematical model means that it is not enough to make a detailed study of one patient only, tuning the model so that simulation mimics accurately the results of that single study: the correct approach is to amass data for a large group of patients, then tune the model so that simulations of the same group give rise to similar mean statistics with comparable standard deviations.

The clinical work described in this chapter was motivated also by the need to study the button pressing in demand analgesia as a method of measuring pain relief. The classical method is to ask the patient questions about his pain, but recent developments in analgesic technique, including demand analgesia, have given rise to the need for an alternative

measurement method which can be used continuously over a period of hours or days. So-called 'potency ratios' between drugs are found by comparing the pain relief they give. Comparison of the potency ratios found using demand analgesia to measure pain relief with those found using classical methods should thus provide information on the relationship between button pressing and classical pain measures.

In this study meptazinol, a new drug, was compared with morphine and pethidine, both of which are well-established analgesic opiates. Proportional control was used throughout the trial.

4.2 DESIGN OF TRIAL

The trial took place at the Nuffield Orthopaedic Centre, Oxford, between December 1983 and December 1984. It was conducted using a double-blind method in 60 consenting patients undergoing elective orthopaedic surgery. Patients could be selected for the study if their age was between 20 and 75 years and their weight between 45 and 100 kg. All patients had general anaesthesia. They were given postoperative analgesia with one of three drugs - meptazinol, morphine or pethidine - which were allocated in coded ampoules so that the identity of the drug was unknown both to the patient and to the nursing staff. The patients had had one of three operations: total hip replacement, menisectomy (knee) or laminectomy (spine). The allocation of drugs was such that for each operation there were equal numbers of patients

treated with each of the three drugs.

In the recovery ward, when patients complained of moderate to severe pain, they were given an i.m. dose of 1 ml of the drug subsequently used for demand analgesia. Analgesic measurements were then performed at specified times using standard verbal rating and visual analogue scales presented by a trained nurse observer. The same nurse observer was used throughout the trial. The measurements were continued for up to 6 hours or until the intensity of pain required further analgesia, whichever was sooner.

Demand analgesia was then instituted using PRODAC and a dedicated i.v. line into the patient. Drug concentrations were chosen to be roughly equianalgesic by conventional wisdom - 15 mg ml^{-1} for morphine, 100 mg ml^{-1} for pethidine and meptazinol, which assumes the potency of both meptazinol and pethidine to be 6.7 mg per mg morphine. The bolus size given in response to each button press, approx. 0.05 ml, was chosen such that patients would be likely to press the button 5 or 10 times per hour. This combined choice of bolus size and drug concentration, which is the 'gain' of proportional control, was important; very effective analgesia could be attained if the control output of drug from the machine were such that the patient's analgesic needs were, in general, equalled or exceeded by the effect of the drug - but in this case the patient would make only few demands, and the resulting data would provide a poor basis on which to tune or validate the

model. This was one of the reasons for using proportional rather than stochastic control in this trial; as explained in [Jacobs et al., 1985], good control goes hand-in-hand with low information content in the resulting data. The size of bolus dose was therefore chosen as a compromise between obtaining good data on the one hand, and providing acceptable analgesia on the other.

Patients came out of surgery in the early afternoon and were on demand analgesia until 8.00 a.m. the following morning. Blood samples were taken at the beginning and end of demand analgesia, and were deep frozen awaiting analysis after the end of the whole trial. These samples were of limited interest as none of them were taken during demand analgesia and they could not therefore be used to relate measured and estimated plasma levels. The second PRODAC trial, described in Chapter 8, included extensive blood sampling and made such comparison possible.

4.3 RESULTS

There were no statistically significant differences (details are given in [McQuay et al., 1985]) between the drug groups for age, sex, height weight, girth, surgery time or operation type - it can therefore be assumed that these factors are not responsible for any significant differences between the results for the three groups.

Individual patient results for the i.m. phase, the demand

analgesia phase and the assay results are all shown in Table 4.1, which extends over two pages. At the time of writing, some of the assay results were still not available: the appropriate places in Table 4.1 have been left blank. Assay results which were off-scale (probably due to use of a drug-contaminated syringe for taking the blood sample) are indicated by an asterisk.

4.3.1 Intramuscular Phase

Five different classical measures of analgesia were computed for the i.m. phase of the study. Relative potencies were calculated for the three drugs by comparing the mean unit of pain relief or pain intensity difference obtained per mg drug on these analgesic measures. As the correlation between these is good and they are all essentially measures of the same quantity, only two are listed here, namely TOTPAR (Total Pain Relief) and SPID (Score of Pain Intensity Difference) [Wallenstein and Houde, 1975]. These are shown in Table 4.1, as is the duration of the i.m. dose.

4.3.2 Demand Analgesia Phase

The individual results are listed in Table 4.1. For each patient the following details are shown:

1. Duration of the demand analgesia phase (mins).

patient code	age	sex	kg	op	drug	im (mins)	TOTPAR (%)	SPID (%)	duration (mins)	dose (ml)	dems /hr	rate (ml/hr)	press factor	level1 (ng/ml)	level2 (ng/ml)
WY001	60	f	80	h	me	370	79.1	75.0	670	6	.31	.53	1.00	43	14
WY002	69	f	65	h	mo	300	56.2	29.1	586	11	.57	1.12	1.22	112	50
WY003	45	m	80	h	pe	195	6.2	5.5	721	72	3.72	5.99	1.26		
WY005	64	f	59	h	mo	195	31.2	12.5	725	27	1.39	2.23	1.04	95	67
WY006	64	f	63	h	pe	140	6.2	0	862	53	2.73	3.68	1.36	334	299
WY007	35	m	75	s	me	186	20.8	22.2	942	66	3.41	4.20	1.03	118	100
WY008	39	f	54	s	mo	307	60.4	41.6	736	34	1.75	2.77	1.26	34	69
WY009	46	f	58	s	pe	380	35.4	25.0	807	31	1.60	2.30	1.07	154	240
WY010	20	f	53	s	pe	205	33.3	25.0	846	13	.67	.92	1.18	347	116
WY011	37	f	65	s	mo	145	22.9	8.3	1093	156	7.69	8.56	2.05	193	347
WY012	54	f	80	s	me	270	27.0	22.2	679	25	1.29	2.20	1.47	92	79
WY013	65	m	83	k	me	175	8.3	-8.3	839	121	6.14	8.65	1.68	74	28
WY014	64	m	71	k	mo	190	6.2	0	1054	65	3.35	3.70	1.63	95	*
WY015	71	f	72	k	pe	160	0	0	980	32	1.65	1.95	1.14	482	273
WY016	74	m	69	k	me	235	10.4	0	792	93	4.18	7.04	2.21	271	171
WY017	67	f	81	k	pe	250	4.1	2.7	704	92	2.99	7.84	5.41	207	*
WY019	46	m	65	s	mo	200	16.6	8.3	1014	26	1.34	1.53	1.73	104	17
WY020	52	m	76	s	me	255	12.5	0	910	66	3.41	4.35	1.02	63	100
WY021	34	m	70	s	mo	105	0	0	1020	184	7.77	10.82	1.80	70	258
WY022	21	m	66	s	me	190	12.5	4.1	769	202	8.26	15.76	1.38	99	194
WY024	24	m	70	s	pe	260	37.5	20.8	797	23	1.19	1.73	1.05	208	188
WY025	66	m	85	h	mo	140	0	0	955	165	8.41	10.36	1.56	116	392
WY026	70	f	73	h	mo	270	25.0	25.0	916	121	6.24	7.92	1.14	60	202
WY027	62	m	68	h	me	235	27.0	25.0	651	139	6.60	12.81	1.81	74	369
WY028	62	f	76	h	me	90	2.0	0	1065	55	2.84	3.09	1.17	148	29
WY029	68	f	64	h	pe	265	20.8	8.3	953	59	3.02	3.71	1.64	186	395
WY030	62	f	97	h	pe	160	0	0	1032	161	7.33	9.36	2.18	*	758
WY031	47	m	100	h	pe	145	4.1	-4.1	888	111	5.19	7.50	2.52	201	41
WY032	39	f	57	h	pe	150	2.0	0	896	112	5.52	7.50	1.81	246	709
WY033	66	f	70	h	me	325	43.7	72.2	710	32	1.42	2.70	2.67	299	373
patient details	i.m. phase results							demand analgesia results							assay results

Table 4.1 Individual Patients' Results for First PRODAC Trial

patient code	age	sex	kg	op	drug	im time (mins)	TOTPAR (%)	SPID (%)	duration (mins)	demands	dose (ml)	dems /hr	rate (ml/hr)	press factor	level1 (ng/ml)	level2 (ng/ml)		
WY034	74	m	68	h	mo	355	56.2	55.5	769	55	2.84	4.29	.22	1.45	28	402		
WY035	55	f	66	h	mo	330	52.0	29.1	861	12	.62	.83	.04	1.00	53	28		
WY036	63	m	57	h	me	370	47.9	50.0	758	27	1.39	2.13	.11	1.59	42	24		
WY037	23	m	75	s	pe	145	8.3	13.8	997	151	7.48	9.08	.45	1.31	226	1077		
WY038	44	m	80	s	mo	200	18.7	16.6	955	53	2.73	3.32	.17	1.00	106	518		
WY039	58	m	68	s	mo	120	6.2	0	776	41	2.12	3.17	.16	1.41	85	20		
WY040	71	m	57	s	pe	260	39.5	4.1	741	26	1.34	2.10	.10	2.00				
WY041	22	f	57	s	me	105	2.0	2.7	318	174	7.64	32.83	1.44	1.53				
WY042	32	m	84	s	me	220	41.6	33.3	815	79	4.02	5.81	.29	1.11				
WY043	64	m	71	k	pe	150	20.8	13.8	959	139	7.07	8.69	.44	2.32				
WY044	52	f	51	k	me	135	4.1	0	977	62	3.20	3.80	.19	1.00				
WY045	69	f	60	k	me	150	4.1	0	941	251	8.51	16.00	.54	1.67				
WY046	65	f	80	k	mo	235	27.0	27.7	677	34	1.75	3.01	.15	1.13	98	126		
WY047	35	f	72	k	mo	195	25.0	4.1	767	29	1.50	2.26	.11	1.53	64	90		
WY048	67	m	71	k	pe	115	14.5	8.3	905	85	4.13	5.63	.27	1.29				
WY049	75	m	66	h	me	190	8.3	16.6	1054	85	4.31	4.83	.24	1.55				
WY050	70	f	66	h	pe	375	60.4	41.6	850	8	.41	.56	.02	1.60				
WY051	55	m	67	h	me	165	6.2	2.7	720	183	8.46	15.25	.70	1.43				
WY052	59	m	86	h	mo	165	25.0	16.6	739	13	.67	1.05	.05	1.63	133	42		
WY053	54	m	87	h	pe	105	8.3	5.5	833	29	1.50	2.08	.10	1.16				
WY054	60	m	78	h	mo	110	0	0	789	14	.72	1.06	.05	1.08	112	14		
WY055	50	m	83	h	me	255	43.7	36.1	679	86	4.10	7.59	.36	2.39	34	66		
WY056	59	m	49	h	mo	365	45.8	50.0	884	31	1.60	2.10	.10	1.19				
WY057	72	f	80	h	pe	210	12.5	0	773	11	.57	.85	.04	1.22				
WY058	61	f	63	h	pe	150	0	5.5	802	409	8.15	30.59	.60	3.44				
WY059	57	f	53	h	me	110	0	0	770	179	8.31	13.94	.64	2.03				
WY060	51	m	59	h	mo	325	43.7	37.5	787	64	3.30	4.87	.25	1.33	84	*		
WY061	35	m	79	s	pe	105	6.2	5.5	834	117	5.62	8.41	.40	3.77				
WY063	75	f	56	h	me	305	33.3	25.0	830	141	7.07	10.19	.51	1.57	76	277		
WY065	62	f	74	k	mo	110	12.5	8.3	892	132	6.11	8.87	.41	2.81	97	92		
patient details			i.m. phase results							demand analgesia results							assay results	

Table 4.1 cont. Individual Patients' Results for First PRODAC Trial

3. Total volume of drug delivered (ml).
4. Mean hourly demand rate.
5. Mean hourly volumetric dose rate.
6. Press factor.

Items 4 and 5 above are naturally derived from items 2 and 3, but in the face of variable duration provide a better basis for comparison of drug consumption and button-pressing.

The 'press factor', formulated in the course of this work, is defined as the total number of button-presses, divided by the total number of 1-minute sample intervals in which they were made. It serves as a basic quantification of the pattern in which button-pressing takes place, and in terms of the mathematical model can be interpreted as the average margin by which discomfort exceeds comfort when the button is pressed.

To illustrate the difference between press factor and demand rate, consider two hypothetical patients. Patient A presses the button on one occasion only each hour, but on that occasion makes ten demands in quick succession - perhaps through anxiety or in the face of sudden intense pain. Patient B on the other hand presses the button once every six minutes. For each patient the demand rate is the same at 10 demands/hour, yet the time histories of demands made by the two patients look radically different. This difference is quantified by the press factor, which for patient A has a

value of 10 and for patient B a value of 1. When the sequence of demands serves as the measurement signal for an estimation scheme, as in this research, the distinction between patients A and B becomes extremely important. It is therefore vital that the task of tuning the mathematical model should include a matching of the press factor arising in simulated time-histories to that found in real clinical patients, not relying solely upon matching the demand rates.

Press factor is discussed at greater length in Section 4.4.5 below and in Chapter 7.

4.3.3 Assay Results

The drug samples taken at the beginning and end of demand analgesia were assayed after the end of the study. Morphine assays were carried out using radio-immuno-assay methods by Dr. R.A. Moore at the Nuffield Department of Biochemistry, Oxford, whereas the pethidine and meptazinol assays were carried out in Nottingham by arrangement with Wyeth Research.

The assay results are listed in Table 4.1.

4.3.4 Clinical Experience with PRODAC

The study described in this chapter provided the first major practical test of PRODAC in routine clinical use. Trouble was encountered on two occasions, with patients numbers WY018 and WY023. In each of these cases the machine 'crashed', with the

result that the two trials concerned had to be replaced by numbers WY065 and WY061 respectively. The reason for these failures was found to be that the level of the +5V supply to the logic circuitry fluctuated excessively during printing, stopping the main microprocessor from working. The fault was corrected by the addition of two buffer capacitors - since then there have been no failures.

PRODAC was received very well both by patients and nursing staff. Once or twice the night nurses, who were unfamiliar with demand analgesia, were uncertain as to how they should deal with an emptied syringe. This was clearly explained in the instruction leaflet, but it became apparent that people tend not to read such documents unless forced to. The software was modified slightly to provide more comprehensive advice when the syringe emptied, which appears to have solved the problem.

4.4 DISCUSSION

4.4.1 Summary of Results

A summary of the main results, of Table 4.1 is shown in Table 4.2, with breakdowns both by drug and by operation type. Standard deviations are shown as well as mean values.

drug	op	number	i.m. time (mins)		demands/hour		dose rate (ml/hr)		press factor	
			mean	sd	mean	sd	mean	sd	mean	sd
ALL	ALL	60	210.4	82.0	6.31	6.20	0.28	0.24	1.65	0.76
ALL	H	30	228.8	92.0	6.03	6.16	0.27	0.21	1.60	0.56
ALL	S	18	203.2	74.2	6.66	7.42	0.31	0.32	1.51	0.64
ALL	K	12	175.0	45.0	6.46	3.81	0.29	0.14	1.99	1.15
ME	ALL	20	216.8	80.9	8.69	7.36	0.40	0.31	1.57	0.46
MO	ALL	20	218.1	85.0	4.20	3.18	0.21	0.15	1.45	0.42
PE	ALL	20	196.3	78.2	6.03	6.40	0.25	0.16	1.94	1.09

Table 4.2 Summary of Demand Analgesia Phase Results for First PRODAC Trial

4.4.2 Potency Ratios

The classical pain score results for the i.m. phase of the study [McQuay et al., 1985] showed the potency of meptazinol relative to morphine as 7.0 mg meptazinol per mg morphine, with a 95 percent confidence interval of 6.0 - 8.0 mg per mg morphine. This is very close to the assumed potency ratio of 6.7 mg per mg morphine used to specify the drug concentration for the trial, and is thus a reflection of the finding that there was no significant potency difference between morphine and meptazinol during the i.m. phase at the concentrations used.

For pethidine there was a much greater variation in the relative potency given by different analgesic measures: the mean value was 18.3 mg pethidine per mg morphine, but the 95 percent confidence interval of 0.0 - 37.0 mg per mg morphine included the dosing potency of 6.7 mg per mg morphine.

There were no significant differences between the drugs for mood or sedation scores, but pethidine produced significantly greater respiratory depression than meptazinol.

For the demand analgesia phase, potency ratios were calculated on the basis of the hourly drug consumption rates. The demand analgesia potency ratio for meptazinol was substantially greater than the i.m. value at 12.7 mg per mg morphine and thus well outside the i.m. confidence limits of 6.0 - 8.0 mg

per mg morphine. The potency of pethidine, on the other hand, was 7.9 mg per mg morphine, which is within the 95 percent confidence limits calculated for the i.m. phase although it is less than half the i.m. mean value of 18.3 mg per mg morphine.

Several possible reasons for this difference in potency ratios obtained for meptazinol from the two parts of the study are discussed in [McQuay et al., 1985]. One further possible reason could be the existence of a pharmacodynamic nonlinearity in the form of a threshold concentration below which meptazinol has little or no effect. In the i.m. phase this threshold would generally have been exceeded by a large margin, whereas in the demand analgesia phase, when mean drug levels tend to be lower than in the i.m. phase, patients would have to dose themselves beyond the threshold to obtain adequate analgesia. This would lead to the sort of discrepancy seen here. Little is known about nonlinearities in the pharmacodynamics of analgesic opiates - the modelling and simulation work described in this thesis assumes linear pharmacodynamics only for want of usable information. A number of researchers have published what they call 'minimum effective concentrations' for analgesic opiates [Lehmann, 1985a], but these refer to the mean drug levels generating sufficient comfort to counteract the discomfort of a postoperative wound, rather than to pharmacodynamic dead space. A study of pharmacodynamic non-linearities would be a valuable topic for further research in this field.

4.4.3 Influence of Weight

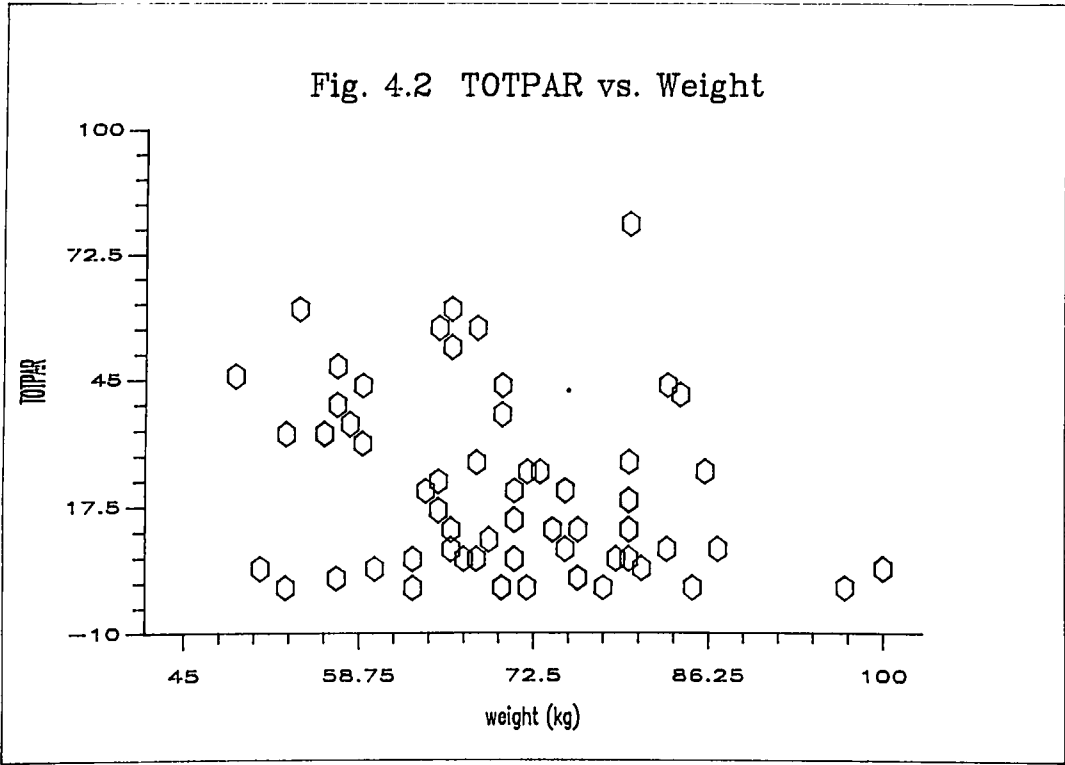
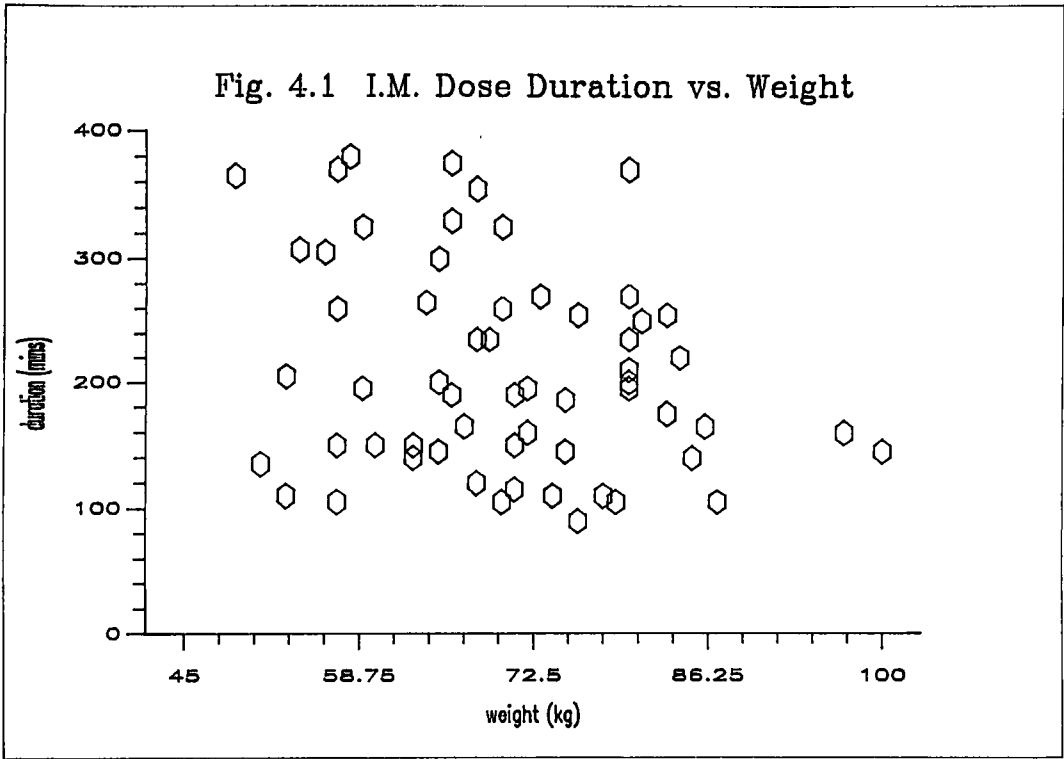
It is part of conventional clinical 'wisdom' and standard clinical practice that the amount of analgesic drug required by a patient is dependent upon the patient's body weight. Postoperative doses of analgesic opiates are generally specified in mg per kg of body weight; it makes intuitive sense that a heavier patient will require more drug than a lighter one, since the drug is distributed over a larger volume and will thus be more dilute when it reaches the receptor sites at which it acts.

These conventional ideas are contradicted convincingly by the results of the study.

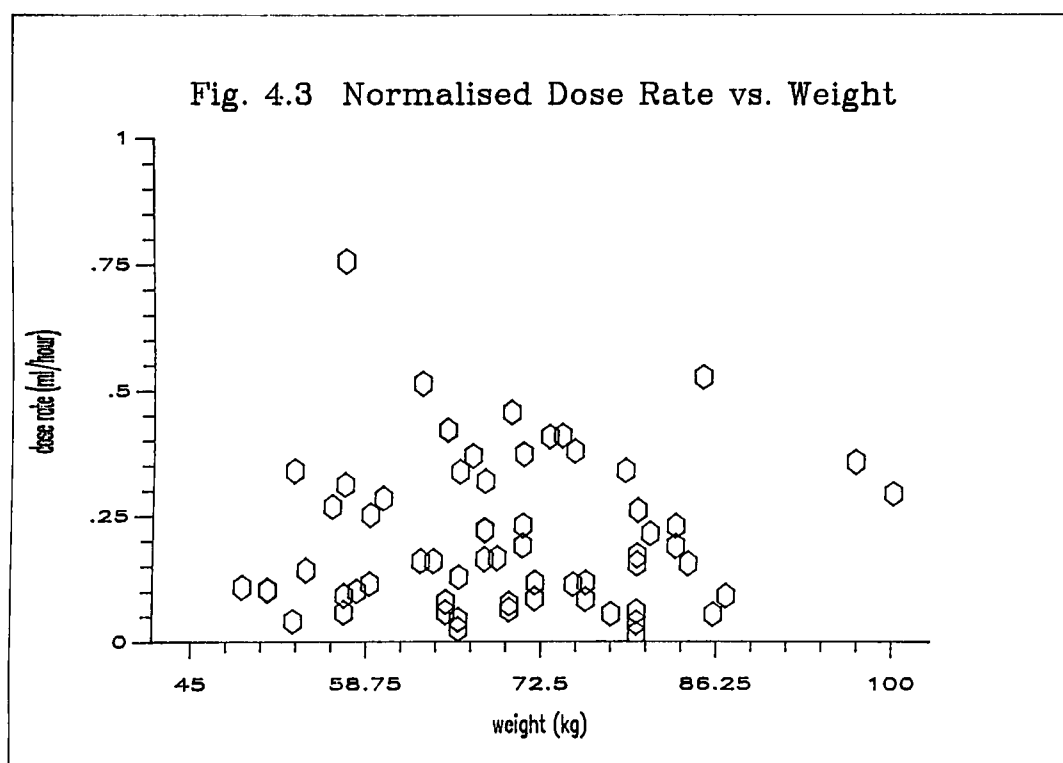
Fig. 4.1 shows the duration of the i.m. dose, which was the same quantity of drug for all patients, versus the patients' body weight. While a line drawn through these points might have slightly negative gradient, showing that the i.m. dose was effective for a shorter time in heavier patients, the correlation is so poor as to be statistically meaningless.

Fig. 4.2 shows the TOTPAR score of pain relief during the i.m. phase versus body weight. Here there is no apparent correlation.

Finally, Fig. 4.3 shows the mean rate at which patients dosed themselves during demand analgesia, again versus body weight. The dose rates are shown normalised relative to morphine to



eliminate the differences in potency between the three drugs. This is the most convincing demonstration of the lack of correlation between body weight and drug requirement. If conventional ideas were correct, then patients free to give themselves drug according to their needs should exhibit a positive relationship between body weight and drug requirement: this is most obviously not the case.



This finding is supported by those of other researchers, though it does not seem to be discussed in detail in their published work. Tamsen mentioned at the PAA Workshop in London, June 1984, that he had found no correlation between

drug requirement and body weight. Lehmann likewise states in a paper that he has found no correlation [Lehmann, 1984].

The implication for this thesis is that contrary to previous assumptions, patient body weight can be neglected in the mathematical model. For the clinical world the results imply that when demand analgesia is not available a fixed dosing regime will generally lead to a better coverage of patient requirements than one proportional to body weight. It would be important in that context to consider the correlation between undesirable side-effects and body weight for a fixed dose.

4.4.4 Influence of Operation

As can be seen from the summary of results in Table 4.2, the mean values of TOTPAR and SPID achieved for different operations ascend in the order knee, spine, hip, which would imply that knee operations are more painful than spines, which in turn are more painful than hips. This same tendency is shown by the mean duration of the i.m. dose. However the sample sizes involved also rise accordingly - there were 12 knee patients, 18 spines and 30 hips - and examination of the spread of results for the three operations reveals that there is not in fact a statistically significant difference. This is confirmed by an analysis of variance.

Furthermore, there was no significant difference between the three types of operation in terms of duration of the i.m.

dose, mean demand rate, mean hourly dose rate or press factor.

The mean press factor for knee operations was somewhat higher than for hips or spines. This is largely due to a skewing caused by patient WY017, who had an unusually high press factor of 5.41 (this patient is likewise largely responsible for the higher mean press factor calculated for pethidine than for morphine or meptazinol).

The patient records were also analysed to see whether the hour-by-hour variations in demand rate showed any differences between the groups, following the line of thought that some wounds heal faster than others. This too yielded no difference between the types of operation.

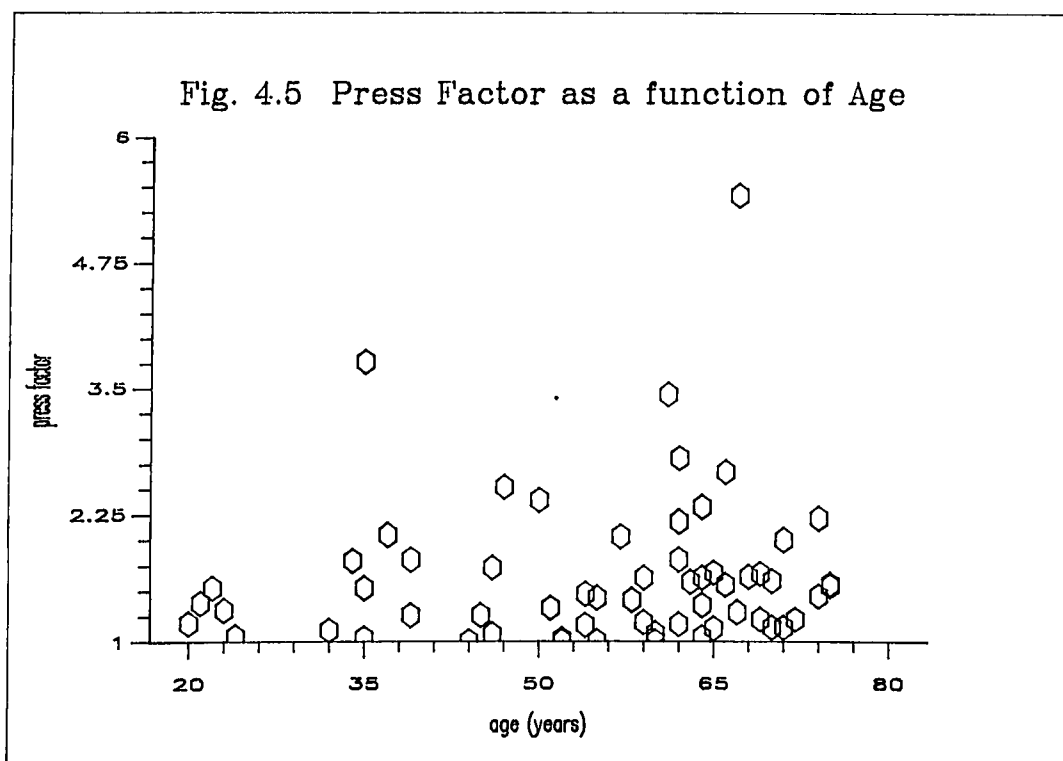
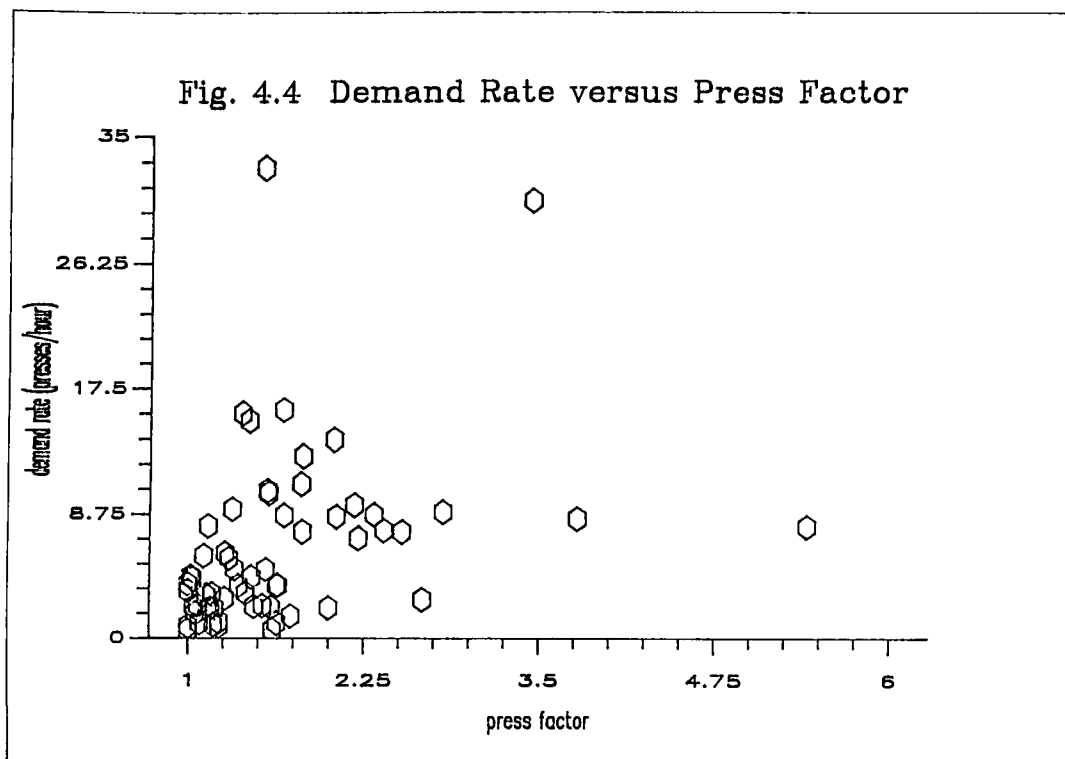
Reasbeck used different values of wound driving noise s.d. and wound time constant to characterise different types of surgery. These figures were based on everyday clinical experience of doctors since there were no suitable clinical data available upon which to draw. It may be that there are differences in the time-course and severity of discomfort resulting from different operations. In the case of the three types of orthopaedic surgery studied here, however, no differences are discernible for the demand analgesia part of the study and therefore no distinctions need to be made between them for purposes of simulation, estimation or control - this is again supported by the results of Lehmann, who found no difference in drug consumption between patients with major

abdominal surgery and those with back surgery [Lehmann, 1984]. This finding thus represents a significant and desirable simplification of the mathematical model.

4.4.5 Demand Rate and Press Factor

It is illuminating to examine the distribution of demand rate against press factor, as shown in Fig. 4.4. The spread of points, particularly the outliers, shows clearly that the two statistics are independent: some patients (low press factor, high demand rate) press the button only once or twice on average when they need drug, yet their overall demand rate is very high, perhaps because they obtain little pain relief from the drug. Other patients (high press factor, low demand rate) make many demands at once when they require drug, perhaps because they have been subject to an intense bout of new pain, yet all in all they do not consume much drug.

Both demand rate and press factor were plotted against patient age. There was no correlation between demand rate and age, but the press factor results in Fig. 4.5 show a certain tendency for the spread of press factor to increase among older patients. This must be attributed in part to the fact that more older patients than younger ones took part in the study - it would not seem to be of sufficient significance to need to be catered for in the mathematical model.



4.5 CONCLUSIONS

The study described here has provided for the first time data which are sufficiently rich and detailed to allow meaningful tuning of the mathematical model of patients in pain (q.v. Chapter 7). This could not have been achieved without the development of PRODAC, described in Chapter 2, which has proved itself well, both as a flexible research tool and as a reliable and simply operated piece of clinical equipment.

Analysis of the results obtained in the two parts of the trial shows a generally good correlation between classical verbal pain ratings and demand analgesia drug consumption, as measures of pain and thus of drug potency ratios. Of particular interest to this thesis is the analysis of the roles played by body weight and operation type for the purposes of mathematical modelling. The results have shown that contrary to traditional teaching, the amount of drug required by the patient is not dependent upon the patient's weight. Furthermore, while the type of operation appears to influence the pain relief afforded by a fixed i.m. dose of drug, no significant difference is discernible between the three operations studied here with regard to any aspect of demand analgesia.

The identification and definition in this thesis of the press factor is an important contribution to the work of modelling demand analgesia. It greatly enhances the characterisation of time histories of demands, yet is simple in concept. It will

be seen in Chapter 7 that the press factor provides a precise criterion for tuning one of the model parameters which until now had been set only by guesswork.

CHAPTER 5

PHARMACOKINETIC MODEL REVISIONS

As was stated in Chapter 1, the path from infusion of analgesic drug to the occurrence of analgesic effect is traditionally divided into two stages:

1. The propagation of the drug through the body to receptor sites (at which the drug acts), described by 'pharmacokinetics'.
2. The effect caused by the presence of drug at the receptor sites, described by 'pharmacodynamics'.

This chapter is concerned with changes which were made to Reasbeck's pharmacokinetic model in order to achieve a more conceptually correct model structure, more realistic model behaviour and thus ultimately better control.

.

5.1 MOTIVATION FOR CHANGES

Data published for analgesic drugs refer in general to the kinetics of the concentration of drug observable in the blood plasma. Linear exponential models are well known to result from fitting time constants to observed plasma concentrations

after a bolus dose [Godfrey, 1983]. A common representation of these results has transfer function

$$G(s) = \sum_{i=1}^n \frac{K_i}{1 + sT_i} = \frac{b_0 + \dots + b_{n-1}s^{n-1}}{a_0 + a_1s + \dots + a_ns^n} \quad (5.1)$$

where the value of n required to obtain a good fit is generally not more than 3. The non-zero numerator coefficients $b_1, \dots, b_{n-1}$ in equation (5.1) predict an instantaneous response to a bolus input. This amounts to assuming, as pharmacologists conventionally do, that the drug is distributed instantaneously and uniformly throughout the body, such that the average concentration in the plasma immediately following drug delivery is equal to the mass of drug delivered divided by the apparent 'volume of distribution'. Naturally this is not in fact so; it takes a certain time for the body to act as a mixing device and for drug which has been injected into one part to appear in anything resembling a uniform concentration in the blood in the rest of the body. In the period immediately following injection there will parts of the body (eg. downstream of the point of injection) in which the plasma level is considerably higher than the predicted instantaneous peak, whereas other parts of the body further away from this point or upstream of it will initially have very low or zero plasma level. Nevertheless, pharmacokinetic descriptions based on these

assumptions are widely available, mathematically simple, and (if properly understood) useful.

A major problem in trying to establish generally 'true' pharmacokinetic values for particular analgesic opiates is the enormous variability both of individual patient responses and of the concentrations measured by different assay methods. In a typical large sample of patients, all of whom have been given injections of the same quantity of morphine, the ensuing plasma concentrations measured can differ by a factor of 40 between the highest and the lowest [R.A. Moore, private communication]. Individual requirements for analgesia in the face of postoperative pain are similarly variable. Even mean values of fundamental parameters such as the volume of distribution or the clearance, published by different research workers for the same drug, can vary by factors of 3 or more [Lehmann, 1985a]. Similar variability can be observed in mean values published for pharmacokinetic time constants.

It is not known precisely where and how analgesic opiates act, but it is likely that the main analgesic effect depends upon the presence of drug at receptor sites in tissue of the CNS (central nervous system), i.e. the brain and the spinal cord. The mechanism whereby drug is brought to these sites is a combination of transport in the blood and diffusion from the blood vessels through tissue to the receptors. This leads to a delay between the injection of drug into the bloodstream and the occurrence of peak analgesic effect.

Reasbeck worked on the assumption that this delay was at most one minute and could thus be neglected. This is not so. While the onset of effect (and this generally includes a placebo element; the patient's perception of his pain changes simply because he knows something has been done about it) can sometimes be seen within one minute, the delay between i.v. injection and the occurrence of peak effect is typically of the order of several minutes. The design of a respectable feedback control system should not neglect this delay if it is large compared to the sample interval, as is the case here with a sample interval of one minute. Reasbeck's work assumed further that receptor-site levels of drug concentration would be the same as those in the blood. This too is most unlikely to be true except in the steady state, since the mechanism of transfer from blood to receptors includes diffusion through tissue: a concentration gradient must exist for this transfer to take place.

It is therefore insufficient to assume that analgesic effect depends directly upon the plasma drug concentration: some account must be taken of the transfer between plasma and receptor sites.

5.2 DETERMINING PLASMA-TO-RECEPTOR TRANSFER KINETICS

5.2.1 Established Methods

Little appears to be known about the kinetics of transfer of drug from blood to the receptor sites, though a number of papers published over the past two decades have suggested approaches for tackling the problem [eg.: Garrett et al., 1967; Wagner, 1968; Wagner, 1971; Holford and Sheiner, 1981; Paalzow, 1982]. All these authors assume that the transfer can be represented as a first-order process, as this is the simplest representation giving qualitatively correct results. Experiments to determine the kinetics of such a process directly would require not only exact knowledge of the receptor-site locations but also measurement of the drug concentrations, which would involve taking frequent samples of brain and spinal tissue: very difficult and almost certainly unacceptable. Instead, the natural approach is to measure the effect of the drug and to infer from it the tissue concentration. To do this it is necessary to possess (or assume) knowledge of the pharmacodynamics of the drug.

In his 1971 paper, Wagner describes fitting measured data of the effect of LSD following i.v. injection to predicted brain tissue concentrations of the drug, based on plasma concentration measurements and an assumed first-order transfer between plasma and tissue. Holford and Sheiner quote calculated time constants of such transfers for four drugs, all of which have a well-measurable effect and for which

specific knowledge of the non-linear pharmacodynamics was available.

A problem with determining such figures for analgesic opiates is the difficulty of measuring the effect accurately and repeatably, especially with the short interval between measurements necessary to find the response over the course of the first minutes after an injection. In addition, as was mentioned in Chapter 4, very little indeed is known about the pharmacodynamics of these drugs. The Author therefore took a slightly different approach, using the time to peak effect rather than the actual values of a series of pain relief measurements. The former seems to be more consistent from patient to patient than the latter and this approach has the further advantage that no knowledge or assumptions of the pharmacodynamics are necessary, other than that peak analgesic effect coincides with peak receptor site concentration.

5.2.2 Modified Method using Time to Peak Effect

The plasma-to-tissue transfer can be approximated as a first-order process

$$C_r(s) = C_p(s) \cdot \frac{K_r}{1 + sT_r}$$

(5.2)

where C_r and C_p are the Laplace transforms of the receptor

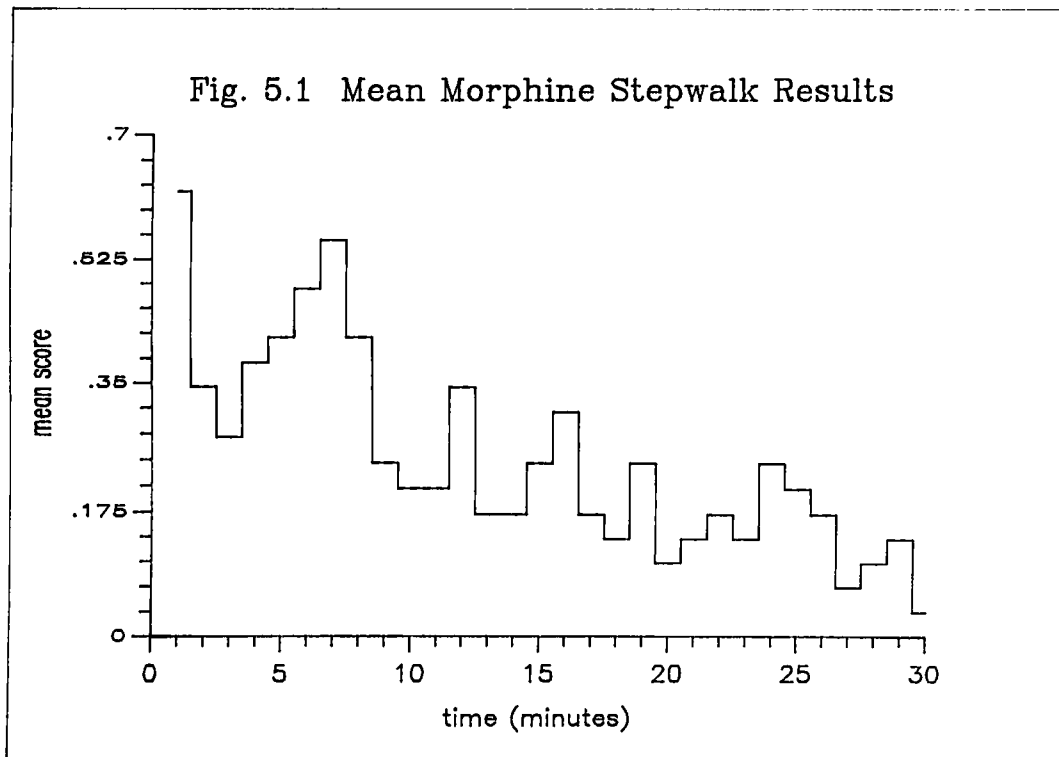
site and plasma drug concentrations respectively, K_r and T_r are constants. Assuming that the receptor site drug concentration is reflected in the analgesic effect perceived by the patient, the time-constant T_r of the first-order process can be determined from the (reasonably well-established) plasma kinetics combined with clinical observations of the time delay between injection of the drug and the occurrence of peak analgesic effect. The gain K_r is of less immediate interest (unless the receptors can be identified and the drug levels there measured as outlined above) and can be assumed arbitrarily to be unity without compromising the overall approach. This unity assumption implies that in the steady state, tissue levels of drug concentration will be equal to those in the blood - which is not unreasonable.

The first step was to determine the mean time between injection of an i.v. dose of opiate and the occurrence of peak analgesic effect. It was found that a study conducted two or three years previously by McQuay and Bullingham, the results of which had never been analysed, might provide an answer. This was the so-called 'Stepwalk' trial. A number of patients suffering from postoperative pain were given 10 mg morphine i.v.. For the next 30 minutes they were asked once per minute

"Is your pain better, worse or the same?",

meant with reference to the previous minute. These results

were now exhumed and analysed. The method used was to rate 'better' as +1, 'same' as 0 and 'worse' as -1: these figures were then averaged minute-by-minute across all patients to produce a mean 30-minute time-history, shown in Fig. 5.1.



Inspection of individual records revealed that patients were probably not answering the question accurately. An i.v. injection of drug leads to rapid onset of analgesia followed by a gradual lessening. Correct answering of the question ought to reflect $\frac{dq}{dt}$, where q is the perceived pain of equation (1.1), as the patients track successive changes in their pain. Assuming constant discomfort d , the graph of Fig. 5.1 ought

therefore to be strongly positive for the first few minutes, corresponding to the rapid onset of pain relief, but thereafter ought to be zero or slightly negative, corresponding to the gradual decline. This is patently not so; there is not even a single minute in which the summed result is negative! One or two patients answered 'better' almost every minute throughout the half-hour.

It seems almost certain that the patients' answers on the whole reflected not $\frac{dq}{dt}$, as was intended, but simply how they felt, namely q . When patients' mental faculties are clouded, as here, by morphine and pain, the task of answering difficult questions is made harder still - and it is much easier for a patient to assess the intensity of his pain than to assess its derivative with respect to time.

Looked at in this light, Fig. 5.1 makes more sense. The initial peak in minute 1 is almost certainly attributable to placebo relief from the injection. The patients relax, their anxiety is relieved, they wish to please the nurse asking the question and they feel confident because something has been done for their pain. The ensuing minutes are probably a more accurate reflection of the effect of the drug: the curve rises to a peak around minutes 6 and 7, then decays irregularly. It is likely that the placebo effect in demand analgesia is much lower than with nurse administration of drug, effectively removing the first-minute peak of Fig. 5.1.

This implies that the peak analgesic effect occurs on average about 6 - 7 minutes after i.v. injection of morphine. By itself, Fig. 5.1 would not be the firmest of evidence - however it corresponds well not only with everyday clinical experience [R.E.S. Bullingham, H.J. McQuay, C.J. Glynn, R.A. Moore; private communication], but also with the results of the first PRODAC trial described in Chapter 4. Inspection of individual PRODAC records from that study showed that patients very often made a group of demands over the course of a few minutes: while contradicting an assumption of instantaneous and strong analgesia, this would be explained well by a 6-minute average delay to peak effect. A patient in pain will presumably continue to make demands until he feels sufficient relief, typically somewhere between minutes 2 and 6. Furthermore, Paalzow draws a graph in [Paalzow, 1982] of the predicted disposition of morphine in the human brain following intravenous injection, based on measurements taken from experiments with rats. This too shows a rapid onset, reaching a peak 5 or 10 minutes after injection.

Having thus estimated the time to peak effect of i.v. morphine, the time constant T_r in (5.2) can be found. Reasbeck describes the derivation of the transfer function representation of 'established' plasma pharmacokinetics [Reasbeck, 1982], reproduced here as

$$C_p(s) = \frac{K_1}{1 + sT_1} + \frac{K_2}{1 + sT_2} + \frac{K_3}{1 + sT_3} \quad (5.3)$$

Cascading this with the plasma-to-receptor transfer function of (5.2) leads to the overall transfer function of (5.4), relating the injection of drug i.v. to the tissue level.

$$C_r(s) = \left[\frac{K_1}{1 + sT_1} + \frac{K_2}{1 + sT_2} + \frac{K_3}{1 + sT_3} \right] \frac{K_r}{1 + sT_r} \quad (5.4)$$

The value of K_r is of little interest, as already explained, but given the (established) values of T_1 , T_2 and T_3 , T_r will determine the delay between injection and peak effect. Taking the inverse Laplace transform of (5.4) yields the impulse response

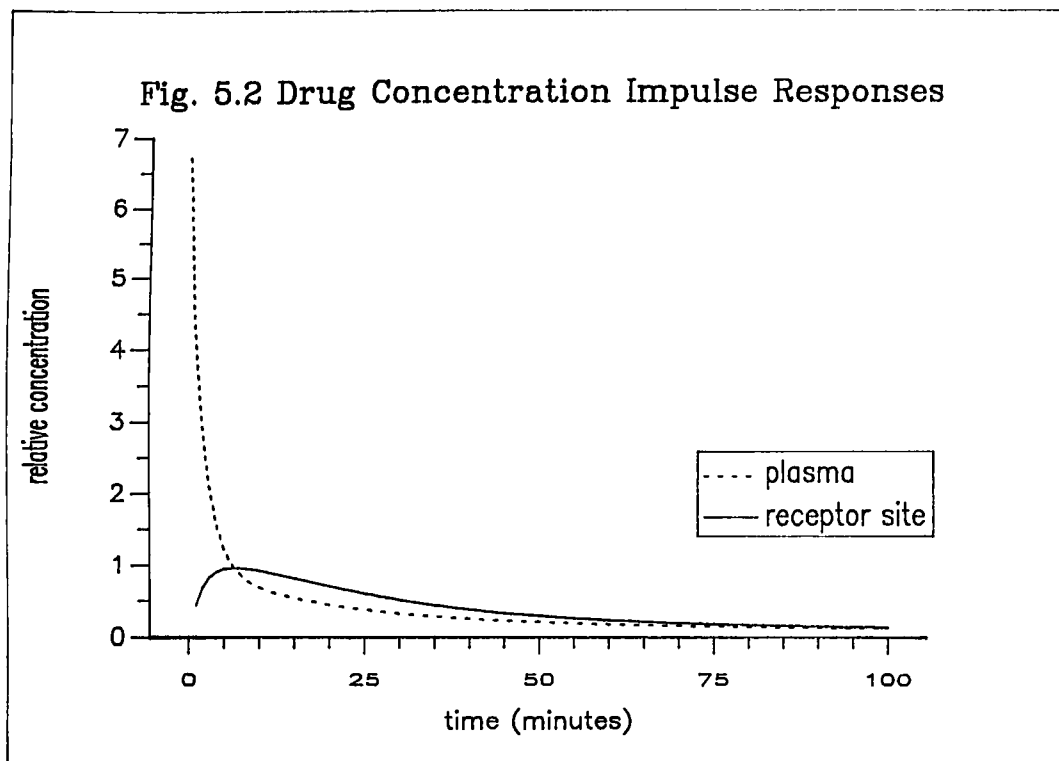
$$c_r(t) = \sum_{i=1}^3 \frac{K_i K_r}{T_i - T_r} \left[e^{-\frac{t}{T_i}} - e^{-\frac{t}{T_r}} \right]$$

(5.5)

The normal method would then be to differentiate (5.5) with respect to time and equate the resulting expression to zero to find the time at which the impulse response reaches its maximum value, finally solving this equation for T_r . The resulting equation is difficult to solve analytically and so instead (5.5) was solved numerically using the software package LOGISTIX [Grafox Ltd, 1985], with T_r being tuned manually until the peak of the impulse response occurred approximately 6.5 minutes after the impulse. The value of T_r thus found was 12 minutes. A bolus i.v. injection of drug, as studied here, is a good approximation to an impulse.

Fig. 5.2 illustrates the difference between the standard plasma impulse response and the tissue impulse response now derived.

The assumption until now that no distinction need be made between plasma and receptor site concentrations led to the unrealistic simulation of sharp 'spikes' of drug concentration in the CNS - and thus of analgesic effect - supposedly resulting instantaneously from an i.v. infusion of a bolus of drug, corresponding with the 'plasma' curve of Fig. 5.2. The new representation, shown by the 'receptor site' curve, is intuitively much more convincing; it shows the rapid but not instantaneous rise of drug concentration to a peak after a few minutes, following which it gradually decays.



5.3 DIRECT 2ND ORDER MODELLING OF TISSUE LEVEL

This section describes what proved ultimately to be a fruitless trail, but was potentially of considerable interest and is thus documented here.

5.3.1 Aim and Motivation

The great variability between patients, discussed in Section 5.1 above, applies to the time constants T_i and coefficients K_i of equation (5.4). It would be of value if an on-line estimation scheme could estimate not only drug levels, as described in Reasbeck's thesis, but also these parameters T_i

and K_1 for individual patients. This could improve the performance of estimation and control in adapting more realistically to individual patients, as well as providing clinically interesting information.

It becomes immediately apparent that such estimation of four time constants on the basis only of the patient's button pressing is an ambitious undertaking. However inspection of Fig. 5.2 led to the thought that the 4th order tissue impulse response shown there could be quite well approximated by a 2nd order function as in equation (5.6),

$$C_f(s) = \frac{K_f}{(1 + sT_a)(1 + sT_b)}$$

(5.6)

bypassing the breakdown into equations (5.2) and (5.3). In this way the entire route from drug injection to receptor site level could be characterised by two time constants and one gain. This would make estimation of time constants more feasible.

5.3.2 Finding 2nd Order Approximation

The aim is to choose K_f , T_a and T_b in equation (5.6) so that it provides as close an approximation to equation (5.4) as possible. The following method was adopted at the suggestion of Dr. O.L.R. Jacobs.

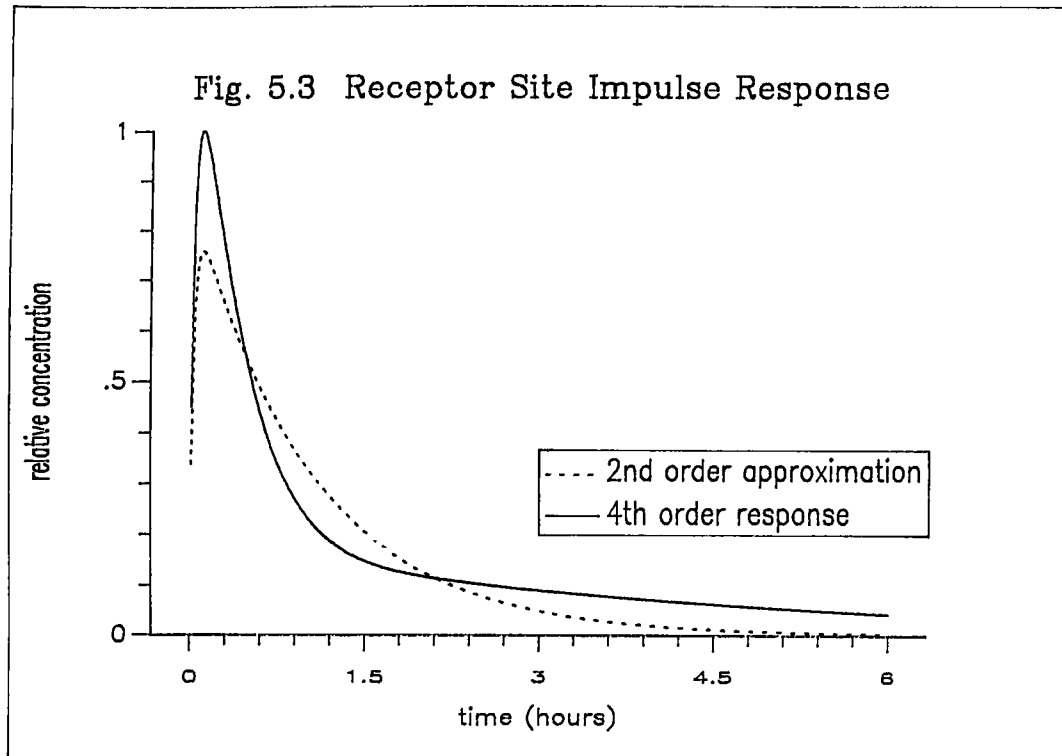
Parseval's theorem can be used to express the integral over all time of the difference between the impulse responses corresponding to C_r and C_f as

$$I = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} (C_r(s) - C_f(s))(C_r(-s) - C_f(-s)) ds \quad (5.7)$$

Choosing K_f , T_a and T_b to minimise I gives the required approximation. Since $C_r - C_f$ can be expressed as a simple ratio of polynomials in s , I can be calculated analytically by standard methods. A computer program was written to calculate I using the recursive method given in [Astrom, 1970], linking it to a NAG library routine to perform the minimisation. This program is listed in Appendix D.

Fig. 5.3 shows the resulting 2nd order approximate impulse response superimposed upon the 4th order 'receptor site' response of Fig. 5.2, demonstrating that the above method for

finding the best approximation works well.



5.3.3 Modified Estimation Scheme

Certain modifications needed to be made to the existing estimation scheme to use the 2nd order approximation:

5.3.3.1 Changes to State-Space Model - The state vector used so far included three elements, x_1 , x_2 and x_3 , formulated in such a way that x_1 represented the overall plasma level, while x_2 and x_3 represented two of its components, corresponding overall to the three terms of equation (5.3). In the light of Section 5.1 this had been augmented by a further state x_6 to

represent tissue level.

To use the 2nd order approximation, x_1 , x_2 , x_3 and x_6 were replaced with states x_1' , x_2' , x_a and x_b . The tissue level was represented by x_2' , with x_1' providing the intermediate state required for 2nd order response. The time constants T_a and T_b were represented by x_a and x_b . The constant K_f was not estimated separately as it is indistinguishable to the estimator from 'relief'. The state space matrices were reformulated as required, the system now being considerably more non-linear than before with states x_a and x_b appearing in the elements of the state transition matrix. This matrix therefore now needed to be updated once each sample interval using the new estimates of x_a and x_b .

5.3.3.2 Changes to Covariance Matrix Handling - Since the sources of pharmacokinetic modelling error, namely the time constants, were now themselves represented as states, these modelling errors were handled automatically by the covariance matrix and did not need to be treated explicitly as before.

5.3.4 Performance

The 2nd order approximation worked well in producing plausible simulation results. However the main objective - estimating time-constants on line - was thwarted by the difficulty of the problem. Even when the demand nonlinearity was replaced with a true linear measurement, simulation studies showed that performance in estimating the two time constants was too

haphazard to be useful. This difficulty is probably due to the very low content in the button-pressing signal of information about the time constants.

A further disadvantage was that the 2nd order approximation does not permit explicit estimation of plasma levels. Assays of drug concentration in blood samples taken during demand analgesia provide the only real measurement to which the performance of the estimation scheme can be related, so that estimation of plasma levels is much needed.

Finally, the approximate impulse response shown in Fig. 5.3, although it is the 'best possible' 2nd order approximation, has the undesirable property that it decays to zero much faster than is generally true. This means that the estimator would be liable to underestimate the level of drug remaining in the body three or four hours after injection, thus undermining the use of safety limits based on allowable drug concentrations.

For these reasons it was decided not to proceed with the 2nd order approximation.

5.4 CONCLUSIONS

This chapter has described the enhancement of Reasbeck's model through drawing the distinction between plasma and tissue drug concentrations. This results in a 4th order description of the tissue drug concentration response to intravenous injection. A simplified, 2nd order approximation to this response was derived but for a number of reasons was not subsequently used.

The results have been incorporated into the model by adding to the existing five states described in Section 1.2.3.3 a sixth, x_6 , representing tissue level. Comfort c is thus now the product of tissue level x_6 and relief x_5 .

The derivation of the time constant T_r in modelling receptor site (i.e. tissue) response for morphine is believed to be original, both in its method and in obtaining a result. The pitfalls which beset the methods proposed in the literature, such as the need for accurate knowledge of pharmacodynamics and absolute values of effect data, are avoided. The work described in this chapter thus represents a significant contribution not only to the task of refining the mathematical model of patient and pain, but also to the science of pharmacokinetics.

CHAPTER 6

NEW OFF-LINE SOFTWARE

6.1 MOTIVATION AND REQUIREMENTS

Off-line software is needed in this research for

1. simulation of patients, including verification of the mathematical model and testing of estimation and control schemes.
2. analysis of real patient data collected using PRODAC.

The PASCAL program OXPANG, written by Reasbeck and described in his thesis fulfilled the first of these functions, while a modified version was used to run clinical data through the estimation scheme. An adaptation of OXPANG was used for the work described in Chapter 3.

With the advent of PRODAC and the clinical trial described in Chapter 4 new requirements emerged for off-line software. These included

1. exact simulation of PRODAC's behaviour, including quantisation of control output, safety limits and the minute-by-minute relation between input and output.
2. ability to simulate large numbers of patients, with specified combinations of drug and operation.
3. simulated patient records should be produced in the same format as real patient records from PRODAC.
4. PRODAC should share with the off-line software the same source code files for estimation and control algorithms.

As it was not possible to fulfil these requirements with OXPANG or a direct adaptation thereof, it was decided to write completely new off-line software. The natural choice of language for this purpose was 'C', as it allows sharing of PRODAC code and variables, as well as being more suitable for the task than PASCAL. Simulation of individual patients, simulation of many patients and analysis of real patient data using the estimation scheme were all integrated into the one main program, entitled SIM and described below. It should be clear that although different in many respects, SIM is still based on OXPANG. The estimation scheme, apart from those modifications already outlined in Chapters 3 and 5, is that presented in Section 1.2.3.3 and described in detail in Appendix A. The scheme is used, for example, to produce the estimated levels of Figs. 8.1, 8.2 and 8.3 in Chapter 8.

6.2 SIM

6.2.1 Functionality

SIM is structured as a hierarchical system of menus. The main choices open to the user on entering the program are

1. Run simulation/estimation. This is followed by a choice of

1. single simulated patient
2. many simulated patients
3. single real patient

All such simulations and/or estimations are based upon the same format of treatment as the meptazinol trial: an i.m. loading dose may or may not be given upon leaving the operating theatre, following which demand analgesia is available for up to 24 hours.

2. Choose control algorithm. This allows a choice between proportional control, Reasbeck's hybrid control and other algorithms. These choices are of relevance only for simulated patients.
3. Choose estimation algorithm. This option allows a choice between different estimation algorithms, a facility which, it is envisaged, will be used in future work to explore the performance of alternative algorithms.

4. Set parameters for simulation/estimation. These parameters include the assumed and actual modelling errors, wound driving noise, volume of smallest syringe step, volume of i.m. dose and so on, as well as parameters for the control and estimation algorithms in use.
5. Display results. This is to produce a graphic display of time-histories (normally over 24 hours) of individual patients' results. All simulated and estimated states of the model can be displayed, as well as square roots of diagonal terms of the covariance matrix, and other items such as control output. Assay results can be displayed for real patients, any number of graphs can be overlaid on the screen and the results can be reproduced on a hard-copy device.

Each of these options in turn leads to further menus or prompts as appropriate. Use is made wherever possible of default settings for menu selections; this makes the program more convenient to use.

6.2.2 SIM Details

This section gives details of certain aspects of SIM, as well as highlighting some of the differences between OXPANG and SIM.

6.2.2.1 Randomisation procedure - To simulate different patients, certain parameters in the model are randomised away from their nominal values - this topic was referred to briefly in Chapter 3. In OXPANG the randomisations were characterised by a single positive number Q such that each simulated parameter value P was made to differ from its nominal value P_0 according to

$$P = P_0(1+\xi Q) \quad (6.1)$$

where ξ is an independent normal random variable with zero mean and unit variance.

However the nature of patient parameters P such as the nominal wound or the pharmacokinetic time constants is such that they are inherently non-negative and that uncertainties about them are proportional rather than absolute. It is therefore inappropriate to characterise inter-patient variability according to equation (6.1), particularly because large negative values of ξ are liable to produce negative values of P which are meaningless. Equation (6.1) is also more likely to generate unrealistically small values of P than unrealistically large values, an imbalance for which there is no physiological justification.

A more plausible, log-normal distribution of P is

$$\begin{aligned}
 P &= P_0(1+\xi Q) & \text{for } \xi \geq 0, \\
 P &= P_0/(1-\xi Q) & \text{for } \xi < 0
 \end{aligned}$$

(6.2)

which is equally likely to generate $P = 2 P_0$ as $P = 0.5 P_0$. SIM therefore embodies a randomisation of this form.

To preserve physiological realism in the randomisation, SIM imposes limits on the range for each variable. Commonsense bounds on these parameters of the form

$$P_{\min} < P < P_{\max}$$

(6.3)

are therefore generally such that

$$\begin{aligned}
 P_{\min} &= P_0/k \\
 P_{\max} &= P_0 k
 \end{aligned}$$

(6.4)

where k is a constant reflecting the degree of uncertainty. Limits are imposed in SIM by setting k to $1+3Q$; this is equivalent to clipping ξ in (6.2) to ± 3 . The 'simulation error', Q , can be set by the user to control the spread of randomisation.

6.2.2.2 Variable Duration of I.M. Dose - Postoperative demand analgesia is often combined, as in the meptazinol trial, with an i.m. loading dose. Demand analgesia generally commences when the loading dose wears off and the patient begins to feel pain, rather than after some fixed interval. SIM provides an option to vary the size of i.m. dose, as well as simulating its patient-dependent duration.

6.2.2.3 Plasma/Tissue Distinction - A further state x_6 was added to the model to represent tissue level, as described in Chapter 5. Comfort c of equation (1.1) is therefore now x_6x_5 rather than x_1x_5 .

6.2.2.4 Accurate Simulation of PRODAC - It is important that simulation should include accurate reproduction of the input/output behaviour of the demand analgesia device. As described in Sections 2.3.3 and 3.2.2, the range of control outputs available to PRODAC is quantised, being a function of the drug concentration, the cross-sectional area of the syringe and the smallest step of the syringe driver. This quantisation is duplicated in SIM.

A further aspect is that PRODAC works using the following minute-to-minute cycle:

1. read number of demands in previous minute.

2. calculate new control output.
3. deliver new control output.
4. wait for start of new minute and then go to 1.

As a result the control output is delivered during the minute following that in which demands are made - SIM reproduces this behaviour accurately.

6.2.2.5 Shared Code and Variables with PRODAC - An important feature of SIM is that it shares with PRODAC the source files for the control algorithm software described in Section 2.4.3. This permits testing of control algorithms in simulation before using them in PRODAC. It also helps to guarantee that a simulation of PRODAC will be accurate: the simulation work described in Chapter 7 uses the same source code for proportional control and its safety limits as did the PRODACs used in the trials described in Chapters 4 and 8. The shared software is linked to the remainder of the simulation program via the three variables 'demands', 'givenfix' and 'fixesdue', as described in Section 2.4.3.

6.2.2.6 Running Multiple Simulations - Because of the variations between patients, discussed at several points already, not only clinical trials but also simulations must be performed on large numbers of patients to obtain meaningful mean results. This function has been incorporated into SIM. The number of patients to be simulated, and the operation, drug and code-name for each patient are all specified in a

master file in a standardised and simple way. When directed to perform multiple simulations, SIM reads the details from the specified file and produces a PRODAC-style file of results for each patient. For control regimes which do not use on-line estimates, such as proportional control, SIM automatically disables estimation during multiple simulations; this reduces the computation time for each patient by a factor of about 50. The statistics for the resulting group of patients can then be calculated using the utility PATSTAT, described below.

6.2.2.7 Extendable Modular Menus - A major problem with OXPANG was the style in which it was written; this made it extremely difficult and time-consuming to modify the program. In SIM a very different approach has been adopted, which has proved to be flexible and efficient. All menus of options are organised as linked lists of structures, handled by a single recursive menu-running function. This makes it very easy to add, remove or modify options in a particular menu, or even to add or remove menus in the overall nested structure. The structures associated with each option allow further information to be 'tacked on', so that, for example, selection of a particular drug from the list of drugs gives automatic access to the time-constants, concentration, relief etc. for that drug. This makes the software elegant and easy to modify.

6.3 UTILITIES

The following utility programs are needed to fulfil certain off-line functions.

6.3.1 PRODUMP

This runs on the DEC Rainbow microcomputer and is used to transfer data from the Epson HX20 to floppy disk. The two computers are connected via their RS232 ports.

6.3.2 PATCONV

Raw patient data files from PRODAC do not contain certain items of information such as the operation performed and the drug used. PATCONV is a program which converts a group of files from the raw form to that used by SIM and by a further utility, PATSTAT.

6.3.3 PATSTAT

The patient data files consist of minute-by-minute records and are not suitable for looking at overall statistics such as press factor, mean drug consumption and so on. PATSTAT is a program which reads a complete group of patient data files and produces statistics for that group. Total drug consumption, total demands, mean drug consumption rate, mean demand rate and press factor are computed for each patient. These results are then summarised across the group, with overall mean and standard deviation calculated for each of the statistics.

This is repeated for each of the subgroups made by bracketing patients according to drug, operation or both. The figures in Table 4.2, for example, were extracted from individual patient records using PATSTAT. In particular, the program is essential for the task of tuning the model as described in Chapter 7.

.

CHAPTER 7

MODEL TUNING AND VALIDATION

7.1 INTRODUCTION

The model of pain and analgesia can only be tuned to produce lifelike behaviour on the basis of many real patients and mass simulations. Real patient data became available in sufficient quantity with the clinical trial described in Chapter 4, while the program SIM described in Chapter 6 embodies the revised structure of the model.

Until this time, model tuning had been performed by guesswork and by intuitive assessments of what 'realistic behaviour' might amount to. This chapter describes the task of tuning and validating the behaviour of the model on a more rigorous, quantitative basis than had hitherto been possible.

7.2 CRITERIA OF MODEL BEHAVIOUR

The primary criteria to be matched when tuning the model to real data are

1. The average demand rate, in button presses per hour of demand analgesia.
2. The average press factor (q.v. Section 4.3.2).

The demand rate is affected both by the potency of the drug and by the severity of the wound. The press factor reflects the average margin by which discomfort exceeds comfort when the button is pressed - this depends both on the rate of fluctuation of the wound and on the size of the disturbances represented by η (q.v. Section 1.2.3.2 and equation (1.2)).

These criteria should be matched so that the statistics for 60 simulated patients are similar to the real ones shown in Table 4.2. The differences between drugs should also be reflected in simulation, as should the spread of results.

7.3 TUNING THE MODEL TO THE MEPTAZINOL TRIAL RESULTS

Four different parameters were tuned to make the simulation results life-like. The starting point was to take the nominal initial value of the wound as 0.5 p. This is equivalent to saying that with no analgesia at all, a patient having had major orthopaedic surgery would be in such pain as to press the button once every two seconds. Inspection of individual patients' results in the meptazinol trial suggested that there are often strong fluctuations in the discomfort caused by the wound. The wound driving noise n was therefore chosen such that the standard deviation of the wound was equal to the nominal initial value, 0.5 p. A typical wound time-history

generated by these values is shown in Fig. 7.1, and corresponds plausibly with the intra-patient variations in analgesic demand rate seen in the trial and demonstrated by the typical patient in Fig. 7.2.

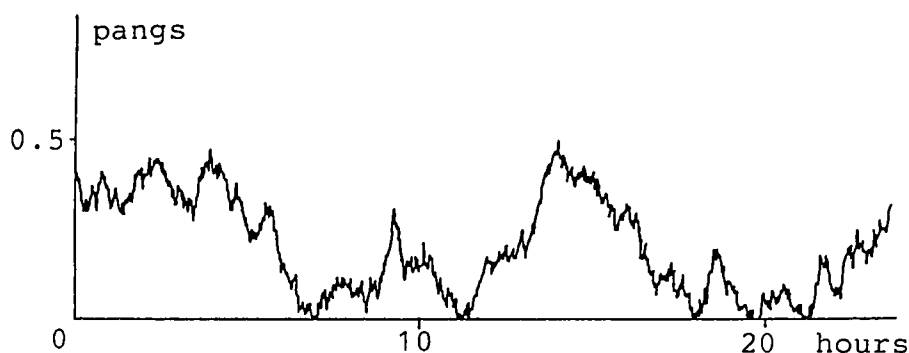


Fig. 7.1 Simulated Wound

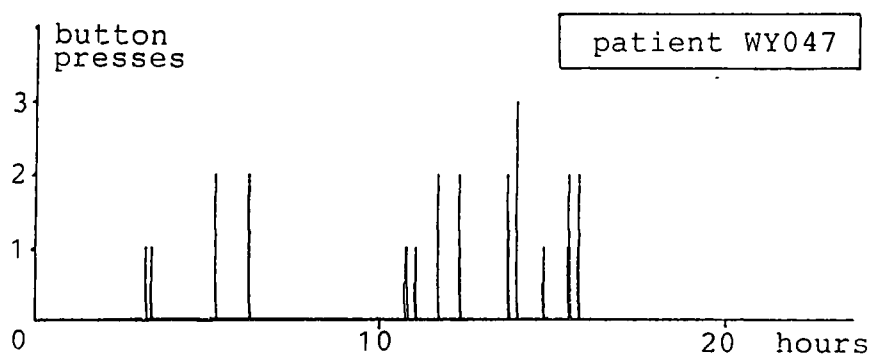


Fig. 7.2 Real Patient Demands

The next step was to tune the nominal relief of each of the three drugs, in parallel with selecting a realistic value for the standard deviation of the additive noise η . The only reasonably firm prior knowledge concerning relief was the approximate relation between the values for the three drugs, as had been used to select the nominally equianalgesic concentrations mentioned in Chapter 4.

A first guess at relief was obtained by considering the plasma levels of drug typically found in patients following an i.m. injection of 100 mg pethidine [Mather et al, 1975], and assuming that in the first hour these would give rise to a peak value of comfort, eg. 1 p, which would compensate more than adequately for the typical discomfort of 0.5 p caused by the wound. This is a reasonable approach since the average time in the trial for a patient to feel pain again, following the initial i.m. dose, was approximately three hours, and the i.m. levels measured by Mather decayed to about half their peak value after about the same time. The values of relief for morphine and meptazinol were set correspondingly. From these initial values the relief was adjusted until the average demand rate was approximately correct. The additive noise η was set initially to have s.d. 0.02 p, but this was found to cause too high a press factor and needed to be reduced. The s.d. of η also affects the demand rate, and so a certain amount of iteration was needed in tuning relief on the one hand and the s.d. of η on the other. Finally, the relief values of the individual drugs were tuned so that the simulations reflected differences seen in the trials.

Table 7.1 shows the results for a typical set of 60 simulated patients, which can be seen to resemble closely the real results in Table 4.2. Deviations about the mean values were deliberately kept smaller in simulation than in reality, as the real results included a number of extreme outliers which the simulation did not attempt to reproduce. Final nominal

values of the tuned parameters giving rise to Table 7.1 are:

1. The initial value of the wound: 0.5 p
2. the s.d. of the wound w: 0.5 p
3. the s.d. of the noise η : 0.01 p
4. the 'relief' afforded by each drug

Morphine: 0.0077 p(ng ml^{-1})⁻¹
 Pethidine: 0.00080 p(ng ml^{-1})⁻¹
 Meptazinol: 0.00052 p(ng ml^{-1})⁻¹

The potency ratios (relative values of relief) thus found between these three drugs correspond well with those found in the trial itself. The non-tuned parameters of the model had the following values:

Simulation error Q: 0.2

Wound half-life (mins): Hip 480, Knee 960, Spine 1440

Pharmacokinetics:

	T_1	T_2	T_3	T_r	K_1	K_2	K_3
Morphine	1.92	18.9	244	12.0	0.13	0.19	0.50
Pethidine	2.00	20.0	230	12.0	0.15	0.20	0.50
Meptazinol	2.00	20.0	250	12.0	0.15	0.20	0.50
	min				min l ⁻¹		

7.4 VALIDATION

A certain degree of validation is implicit in having tuned the model to real results as described above. Validation is also provided by the good correspondence between the drug potency ratios calculated from tuned values of relief and those found by conventional methods as described in Chapter 4. Model behaviour can however be tested more rigorously, by simulating a clinical regime different from that according to which the model was tuned and then comparing these simulated results with appropriate clinical results.

drug	op	number	i.m. time (mins)		demands/hour		dose rate (ml/hr)		press factor	
			mean	sd	mean	sd	mean	sd	mean	sd
ALL	ALL	60	228.1	140.8	6.11	3.14	0.31	0.16	1.67	0.32
ALL	H	30	225.0	132.7	6.23	3.12	0.32	0.16	1.86	0.30
ALL	S	18	198.4	137.4	6.41	3.47	0.33	0.18	1.45	0.23
ALL	K	12	280.3	150.6	5.36	2.52	0.28	0.13	1.53	0.17
ME	ALL	20	145.7	85.8	8.64	2.95	0.45	0.15	1.77	0.30
MO	ALL	20	292.5	153.1	4.47	2.29	0.23	0.12	1.63	0.33
PE	ALL	20	246.1	132.0	5.21	2.41	0.27	0.12	1.62	0.31

Table 7.1 Summary of Results for Simulation of First PRODAC Trial

The TNS trial described in Chapter 8 provides such validation. The trial used the same drug concentration and control algorithm as the morphine group in the meptazinol trial of Chapter 4, but was different in a number of ways. The patients were not given any form of loading dose, and the trial was carried out in a different hospital under the supervision of different staff. Conditions for the patients were also made different by the use of TNS and the taking of blood samples, as described in Chapter 8. It is thus good, but not surprising, that the overall TNS results are similar to those obtained by simulating the TNS trial with the tuned model.

The ideal approach, however, would be to run another trial using the same protocol as the meptazinol trial but with a substantially different control algorithm. This was not possible during the course of the research described here, for such a trial requires major funding as well as one to two years to prepare and carry out.

Failing this, a suitable set of results for validation of the model might be provided by the morphine group in the trial described in [Lehmann, 1984]. No loading dose was given, and the bolus in response to each button press was 1.92 mg morphine - 2.5 times greater than the 0.77 mg used in the trials described in this thesis. This should lead to a somewhat lower demand rate and higher drug consumption. Lehmann publishes a demand rate of 1.3 demands per hour, which

is indeed lower than the rate of 4.2 seen for the morphine group in Table 4.2. However Lehmann's figures for overall drug consumption are also lower - 2.6 mg hr^{-1} as opposed to 3.2 mg hr^{-1} .

The conditions of the Lehmann trial were simulated approximately using SIM, for 20 patients. Unfortunately there are no published figures from which press factor could be calculated for the real patients, so comparison of real and simulated results is limited to mean demand rate and drug consumption. Simulation results showed a mean of 2.12 demands per hour, with drug consumption at 4.5 mg hr^{-1} - these figures are higher than those (1.3, 2.6) reported by Lehmann. This implies that levels of pain perceived by his patients were lower than those generated in simulation and arising in the meptazinol and TNS trials. There are various possible reasons. In particular the physical surroundings of the patient play a substantial role in determining pain perception [Chapman, 1985], as do differences in cultural background [Melzack, 1977], and the variations in such respects between two trials, conducted by different staff in different hospitals in different countries, are very difficult to allow for accurately. The discrepancy between the two results is thus probably attributable to experimental error of this sort.

7.5 CONCLUSION

This chapter has described how the model was tuned for the first time in a precise, quantitative way on the basis of 60 real patients' results. No previous work had attempted to do this. Guesswork and arbitrary parameter values have largely been eliminated and the revisions and improvements to the model described in Chapters 3, 5 and 6 have contributed to making simulations considerably more lifelike.

The tuned model has been validated to some extent. Simulation of the trials described in Chapters 4 and 8 reproduces the clinical results with high fidelity but further clinical work would be necessary for a more rigorous validation.

Until the press factor was defined, η had been set purely by guesswork. As can be seen from Section 7.3, matching the simulated average press factor to the clinical value imposes an extra constraint on the model. This leads to the precise balance of relief and η needed to match both press factor and demand rate, and is a further indication of the importance and contribution in this thesis of press factor as a means of characterising demand time-histories.

CHAPTER 8

SECOND PRODAC TRIAL

8.1 INTRODUCTION

This chapter describes the second main clinical trial carried out with PRODAC. Demand analgesia was used to assess the effectiveness of another method of analgesia, Transcutaneous Nerve Stimulation (TNS), which involves application of electrical signals through the skin using electrodes stuck to the body at carefully chosen points. It has been described as 'electrical acupuncture' and has enjoyed commercial success, not least because it is easy and safe for patients to use without medical supervision. TNS devices are battery-driven and easily carried in a pocket. None of the serious side-effects which beset analgesic opiates, such as nausea, ventilatory depression or dependence, are incurred by TNS - for this reason in particular it has proved useful in treatment of chronic pain. The only recognised side effect of TNS is that of possible allergy to the electrode jelly and the electrode fixing tape.

Much of the medical world has regarded TNS with scepticism, for its effect seems to be unreliable. The aim of this trial

was to confirm or deny a role for TNS in the treatment of postoperative pain. The possible benefit is that TNS may reduce or even obviate some patients' need for narcotics, resulting in fewer problems for the patients as well as a reduced demand on nursing time spent administering drugs.

The trial took place between November 1984 and November 1985 at the Radcliffe Infirmary, Oxford. It is the first placebo study of TNS, as well as the first to use demand analgesia.

8.2 METHODS

A double-blind placebo controlled method was used, in which each of twenty patients coming from the operating theatre after spinal laminectomy was fitted with one of two TNS devices, labelled 'A' or 'B'. One of these was fully functional, whereas the other had its electrodes disconnected inside but was externally identical to the first. The clinician did not know which of the two machines was the placebo. Patients were told that they might or might not be able to get pain relief from their machine.

The devices each had settings for frequency and amplitude of the output signal, which could be set by the patient. In addition to the TNS equipment, patients were also given PRODAC, with 15 mg ml^{-1} morphine. They were told to experiment with the adjustments on the TNS device so that they could find the setting which gave them the best pain relief. If this was still not enough, they were to resort to PRODAC to

obtain what further relief they required. Patients were not told explicitly that the TNS device might be a placebo, but simply that the pain relief available might not be adequate. No loading dose of analgesic was given, i.m. or otherwise.

Comparison of the drug consumption rates for the ten patients in each group should thus give an indication of the difference in effect between the functional TNS device and the placebo. Comparison with the meptazinol trial results should further give some indication of the extent of any placebo effect.

Blood samples (10 ml) were taken once at the start of the trial, then every hour for the first 6 hours, and then once again at the end of the study, which lasted approximately 24 hours.

8.3 RESULTS

Demand analgesia results and blood sample assay results for the individual patients are shown in Table 8.1. The symbol '-' indicates that a blood sample was not taken. The plasma morphine concentrations in blood samples taken at the start of the trial should all theoretically be zero, as patients were given no morphine pre- or peroperatively. Small non-zero values may be due to inaccuracy in the assay or a delay before the sample was actually taken (by which time the patient would already have morphine in the blood). Large initial concentrations are more likely to be due to use of a morphine-contaminated syringe for taking the blood sample.

patient code	age	kg	tns type	duration (mins)	demands	dose (ml)	dens /hr	rate (ml/hr)	press factor	blood sample number						'7'	
										0	1	2	3	4	5		6
TNS001	43	72	a	1471	185	8.88	7.55	0.36	3.36	0	1	1	1	22	40	46	-
TNS002	24	55	b	1369	48	2.40	2.10	0.11	2.40	0	42	14	10	8	10	7	-
TNS003	44	61	a	1453	53	2.71	2.19	0.11	1.02	0	31	47	35	62	34	34	34
TNS004	60	47	a	1446	202	10.40	8.38	0.43	1.84	96	-	223	166	175	177	256	210
TNS005	47	77	b	2604	175	8.95	4.03	0.21	1.79	39	33	33	106	106	85	-	85
TNS006	38	80	a	1438	204	9.18	8.51	0.38	2.87	0	256	268	220	163	247	237	136
TNS007	42	56	b	1452	111	5.73	4.59	0.24	1.37	0	19	37	63	63	153	66	152
TNS008	35	62	a	1502	10	0.52	0.40	0.02	1.11	7	6	8	52	40	41	28	19
TNS009	35	84	a	1430	35	1.81	1.47	0.08	1.03	0	0	13	9	10	28	23	16
TNS010	50	72	b	1430	73	3.33	3.06	0.14	1.01	12	31	40	50	42	49	37	12
TNS011	48	61	b	2447	34	1.75	0.83	0.04	1.79	0	0	0	0	1	-	-	7
TNS012	46	81	a	1494	175	8.62	7.03	0.35	1.61	-	0	18	14	28	35	33	-
TNS013	38	62	b	1323	148	7.51	6.71	0.34	1.95	4	377	452	351	282	242	26	26
TNS014	49	59	b	2402	53	2.73	1.32	0.07	1.39	*	991	157	161	62	42	38	33
TNS015	42	78	a	1416	156	7.33	6.61	0.31	2.08	0	0	52	80	51	34	53	91
TNS016	35	83	b	1433	48	2.48	2.01	0.10	1.20	114	47	85	60	66	32	54	31
TNS017	61	66	a	1528	15	0.77	0.59	0.03	1.15	0	14	23	16	48	16	12	-
TNS018	60	76	b	1379	240	11.66	10.44	0.51	2.45	4	54	44	45	78	136	156	75
TNS019	64	75	b	1545	146	7.53	5.67	0.29	1.40	40	83	126	130	190	148	231	45
TNS020	42	74	a	1409	114	5.21	4.85	0.22	2.59	2	44	30	33	31	41	44	30
patient details			demand analgesia results							assay results							

Table 8.1 Individual Patients' Results for Second PRODAC Trial

This is almost certainly the case with patient 14, for whom the measured concentration in the first sample is so high as to be out of range for the assay. In this case it seems likely that continued use was made of the contaminated syringe, as the time 1 assay concentration, though lower, is still unrealistically high. A patient who really had as high a plasma morphine concentration as this would probably not be alive, let alone awake. Initial plasma concentrations below 14 ng ml^{-1} were taken as being within acceptable bounds of experimental error. Patients 4,5,14,16 and 19, for whom this criterion was not met, have been excluded from the analysis in the discussion of assay results. A summary over all patients of the demand analgesia results is given in Table 8.2.

8.4 DISCUSSION

8.4.1 Comparison of Active and Placebo TNS

The machine labelled 'A' was the placebo, that labelled 'B' was active. The results of Table 8.2 show that there was little difference between the two in terms of demand rate, mean hourly drug consumption or press factor, although the figures are slightly in favour of active TNS. Moreover, the mean Area Under the Curve (AUC) of plasma morphine levels, calculated by Dr. R.A. Moore, was noticeably higher for the active group ($507 \text{ ng ml}^{-1} \text{ hr}$) than for the placebo ($360 \text{ ng ml}^{-1} \text{ hr}$): the reverse of what would be expected if TNS worked. The same was true of the mean peak plasma concentration, which for the active group was 142 ng ml^{-1} and

tns type	number	demands/hour		dose rate (ml/hr)		press factor	
		mean	sd	mean	sd	mean	sd
ALL	20	4.42	2.98	0.22	0.14	1.77	0.66
a (placebo)	10	4.76	3.12	0.23	0.15	1.87	0.80
b (active)	10	4.08	2.78	0.20	0.14	1.68	0.46

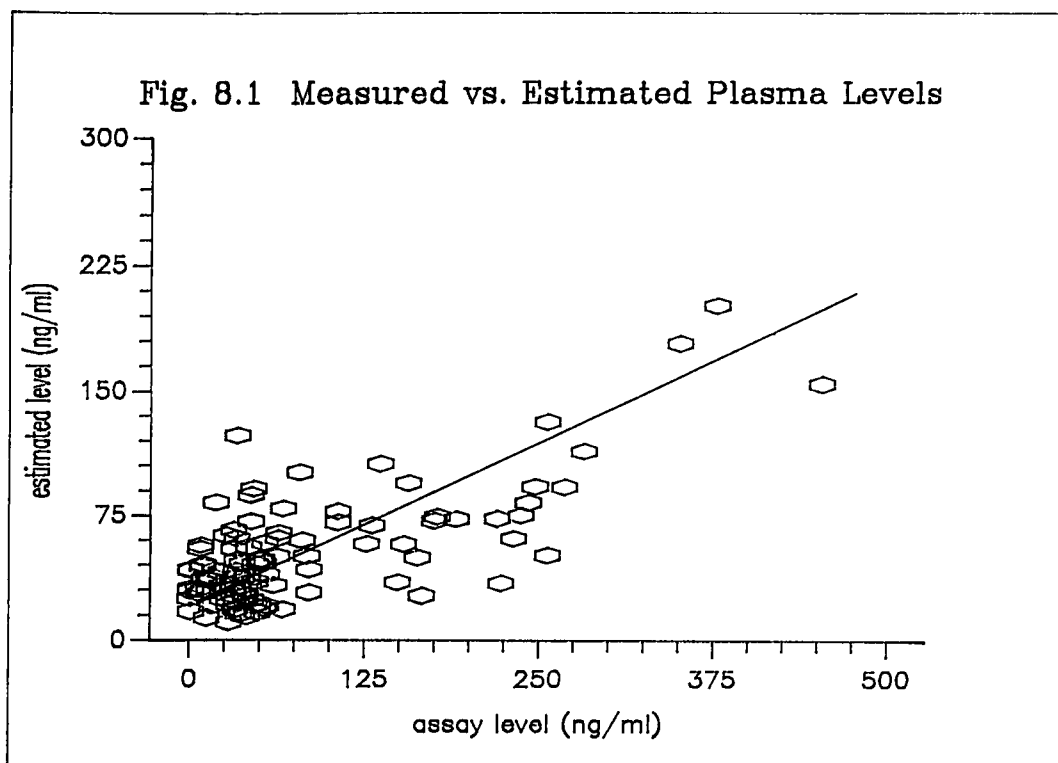
Table 8.2 Summary of Demand Analgesia Results
for Second PRODAC Trial

for the placebo group 93 ng ml^{-1} . In calculating these statistics, only results from the seven hourly-spaced blood samples at the start of the trial were used (i.e. the final sample, if taken, was ignored).

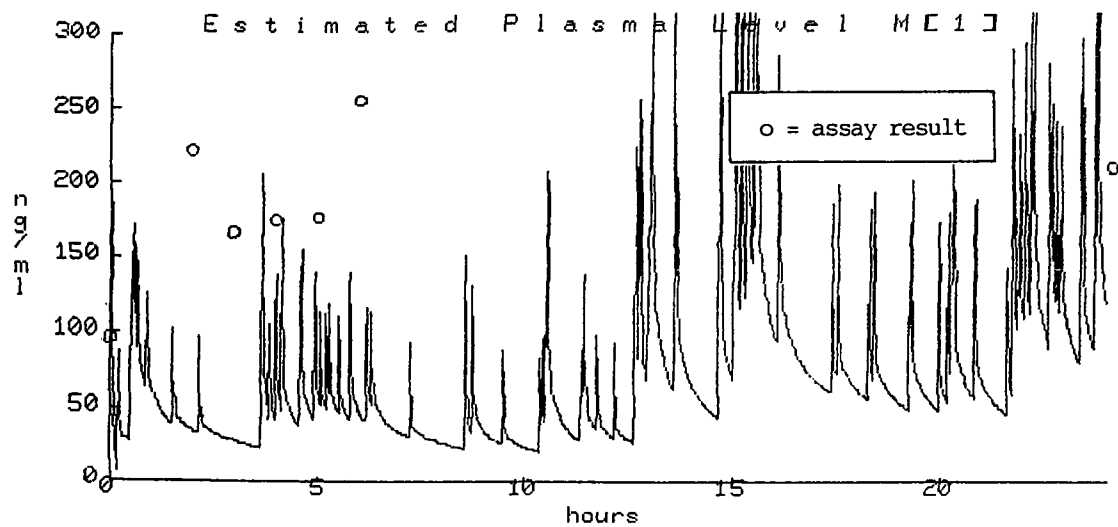
8.4.2 Measured vs. Estimated Plasma Levels

The TNS trial was of particular interest for this research because of the relatively large number of blood samples taken. Plasma drug concentration is the only state of the mathematical model which can readily be measured. Linking these measurements to plasma-level estimates produced by the estimation scheme is thus an important part of validating both the estimation scheme and the model. The assay results are subject to two sorts of error. The first is error in the concentration value, which has been discussed above. The second is error in the timing of the blood sample. The clinical staff performing the trial were asked to take blood samples at precise hourly intervals. This is often difficult in practice: it must be assumed that in reality samples were taken between 0 and 15 minutes late. The summarised correlation between measured and estimated values is shown in Fig. 8.1, which plots these quantities for all 20 patients.

To allow for uncertainty in the exact timing of blood samples, the figures used for estimated plasma level were the average values of that quantity over the 15-minute intervals in which the samples were assumed to have been taken. As can be seen, the points in the graph lie loosely grouped about the line

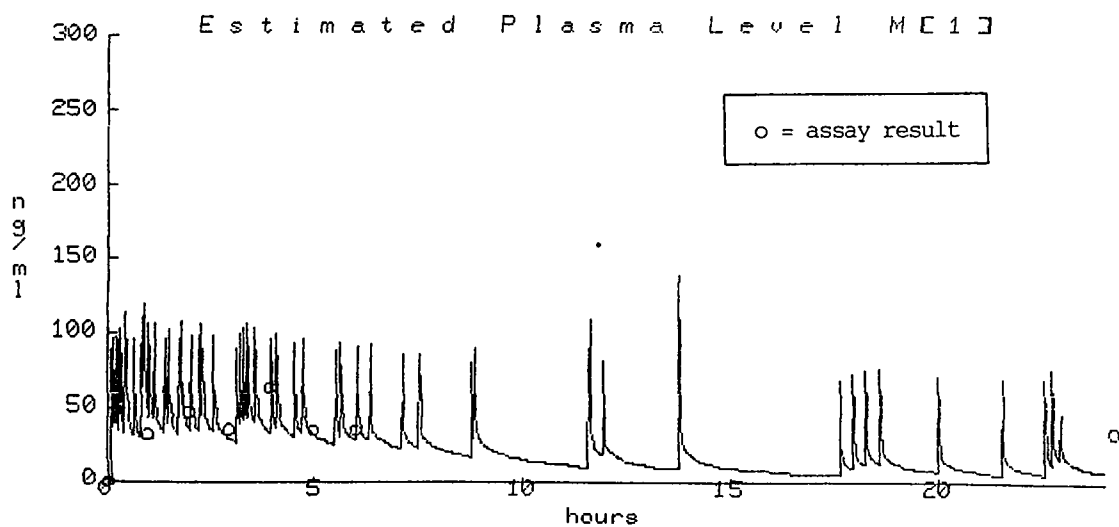


drawn. Ideally the line should have unit slope and pass through the origin, whereas in fact it shows measured values on average twice as high as estimated values. However, given that mean values for pharmacokinetic parameters (upon which the estimated values are based) vary by factors considerably larger than this from one study to another, the correlation is very good. What cannot be seen in Fig. 8.1 is that estimation of plasma levels was much better for some patients than for others. Figs. 8.2 and 8.3 show time histories for two of the patients, illustrating this.



Real patient: tns004

Fig. 8.2 Poor Correspondence between Estimation and Assays



Real patient: tns003

Fig. 8.3 Good Correspondence between Estimation and Assays

8.5 CONCLUSIONS

The TNS trial has fulfilled various functions. It has provided a degree of validation to the tuned model, as described in the previous chapter. It is the first study carried out in the course of this research to take substantial numbers of drug samples during demand analgesia, on the basis of which to validate the estimation of plasma drug concentrations. The correlation between measured and estimated values is qualitatively good and quantitatively within the experimental error of pharmacokinetic measurements.

The trial has provided PRODAC with further clinical testing: no problems were encountered and the machine was very well received by patients, nursing staff and doctors. For research purposes, the value of assay results could be increased considerably by modifying PRODAC's software to allow the timing of blood samples to be recorded accurately. It would not be difficult to assign one of the keys on the front panel for this purpose. A doctor taking a blood sample would immediately press the specified key: this would allow PRODAC to generate an internal record of the timing, accurate to the nearest minute, thus eliminating the uncertainty which degraded the usability of assay results in this trial. This would be a much better solution to the problem of timing uncertainty than would any attempt to force doctors to take blood samples at very precisely specified times.

Finally, the trial has used PRODAC to show that TNS does not

make a significant contribution to care of acute postoperative pain. While not of direct relevance to this thesis, this is a valuable result for the medical world.

.

CHAPTER 9

CONCLUSIONS

9.1 SUMMARY OF ACHIEVEMENTS

This thesis presents work on the improvement of a mathematical model of patients suffering acute postoperative pain. It has used a combination of on-line clinical work and off-line simulation studies to refine both the structure and the performance of the model.

To fulfil the need for detailed real clinical data the Author has developed PRODAC, a demand analgesia machine for research purposes. PRODAC is unique among demand analgesia machines in terms of its programmability: it is the most flexible research tool of this sort available and is the only machine on which control laws involving complex calculation can be implemented. It is also unique for its ability to keep highly detailed records of the progress of a large number of clinical trials, which can be transferred at any time to a computer for analysis; no other machine at present can perform any of these functions, all of which have been crucial to the work described in this thesis. Table 9.1, reproduced from the book 'Patient Controlled Analgesia' by kind permission of Blackwell

Scientific Publications, lists some of PRODAC's features in comparison with other machines used by leading researchers in demand analgesia. PRODAC has received extensive clinical use, with over 1400 patient-hours to date, and has been extremely well received by patients, nursing staff and doctors. Minor modifications to hardware and software were made following the clinical experience reported in Chapter 4. Four PRODACs have been made to date, one of which has been supplied on a commercial basis to the Department of Anaesthetics at Bristol Royal Infirmary.

It has been shown that correct implementation of model-based control imposes particular requirements upon the mechanical design of a syringe driver and its interface to a demand analgesia device. This guided the design of PRODAC. The further implication, that there is an optimal drug concentration for such demand analgesia use, has been investigated in simulation. It was found that an acceptable compromise is reached between syringe life and quality of control when the concentration is such that drug is delivered in minimum increments of size $1 \mu\text{g}$ fentanyl (or the equivalent for other drugs).

The output nonlinearity, which relates the frequency of button pressing to a patient's perceived pain, is now modelled as a simple function which is linear for positive values of pain and zero for negative values. This replaces the complex and erroneous function described in Reasbeck's D.Phil. thesis.

	ODAC	Cardiff Palliator	Prominject	Harvard PCA 4000	Abbott Lifecare PCA Infusor	Leicester Micropalliator	Oxford PRODAC* Palliator MS 402	
Incremental dose	Variable in ml	Variable in mg	Variable in units of mass (mg, µg, lµ etc.)	Variable in ml	Variable in ml	3 set values in ml	Variable	Variable in mg
Background infusion mode available	Yes	No	Yes	No	No	Yes	Yes	Yes
Additional adaptive or non-continuous infusion mode available	Yes	No	Follow-up infusion for 1 hour can be selected	No	No	No	Yes	No
Concentration setting	Yes results in printer output in mg	Set as dilution in µl/mg 1-400	Variable in units of mass-/l	No	No, pre-packed drugs in special syringes used	No	Variable	Variable in mg/ml 1-99
Lock-out time	Variable in minutes 1-99	Variable in minutes 1-99	Variable in minutes 5-999	Variable in minutes 3-60	Variable in minutes 5-99	Fixed 10 minutes	Variable	Variable in minutes 1-30
Infusion rate or infusion time	Background variable in ml/minute 0.01-0.99	Variable in ml/hour 1-99	1 minute for bolus dose 1 hour for optional follow-up infusion	Bolus dose 2.5 ml/minute	Fixed 4 ml/minute	Increment fixed, background fixed	Variable	Variable in minutes 3-15 plus min for fast injection
Pre-set max† dose or dose rate	Max dose per hour 1-99 ml/hour	No	Max, dose set in continuous infusion mode only	6 doses/hour causes alarm	Max, dose in 4 hours 5-30 ml by 5 ml steps	No	Variable	No
Cumulative dose display	On printer	Yes	Yes	Yes	Yes	No, of increments	Yes	Yes
Patient demand signal	2 presses in 1 second	2 presses in 1 second	2 presses in 1 second	Single press	Single press	2 presses in 1 second	Single press	2 presses in 2 seconds
Printer	Yes	No	Yes	Yes	No	No	Yes	Interface
Battery power	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
Manufacturer or status	Janssen Scientific Belgium	Graseby Medical Ltd UK	Pharmacia AB, Sweden	CR Bard Inc. USA *	Abbot Laboratories, USA	Clinical prototype	Clinical research machine	Graseby Medical Ltd, UK

* The Oxford PRODAC is programmable in a high-level language. Any combination of the tabulated parameters over a wide range of values can be achieved.

† A combination of dose setting and lock-out time setting will give a maximum dose/hour for all machines, but this is not a pre-set feature.

Table 9.1 Comparison of Demand Analgesia Machines

Simulation work exposed errors in Reasbeck's mathematical model, but also reaffirmed that potential improvements in pain relief might be gained using model-based control schemes. These include lower and more consistent demand rates from patient to patient, greater adaptation of the delivery of drug to the needs of individual patients, and the delivery of greater amounts of drug without incurring a corresponding increase in peak plasma level.

The first major trial to be performed using PRODAC compared three different drugs (meptazinol, pethidine and morphine) and three different operations (hip, knee and spine) in a total of 60 patients. It showed that contrary to previous expectations the different operations did not give rise to significant differences in the results. It also showed that, again contrary to previous expectations and conventional teaching, there was no correlation between the amount of drug required and patients' body weight. The Author introduced 'press factor' as a basic characterisation of the patterns of time-histories of button pressing. Results from the trial confirm very clearly that press factor and demand rate are independent in practice as well as in theory.

The mathematical model structure has been extended to take into account the transfer of drug between blood plasma and the analgesic receptor sites at which drug action takes place. This transfer is modelled as a first order process with unit gain and time constant T_r . The Author has used a novel method

to find T_r from experimental data. The method makes use of the time between infusion of the drug and the occurrence of peak analgesic effect, rather than relying upon the actual values of a series of effect measurements. As a result the only pharmacodynamic assumption needed is that peak concentration coincides with peak effect, and any non-linearities can be ignored. The value thus found for T_r is 12 minutes. It is believed that this is the first in-vivo measurement of this time constant for morphine in man.

The possibility of on-line estimation of pharmacokinetic time constants has been investigated and found not to be feasible, even using a simplified representation of tissue pharmacokinetics with two such time constants instead of four. The reason such estimation is not possible is primarily the low content in the button-pressing signal of information concerning these time constants, rather than inherent unobservability.

Off-line software embodying the revised model structure has been used to tune the model to the clinical data collected in the first PRODAC trial. The model now exhibits life-like behaviour, which was not previously the case. The s.d. of the noise η , which had hitherto been set only by guesswork, has now been assigned the precise value of 0.01 p by matching the mean press factor in simulated patients to that found in the first PRODAC trial. This confirms the direct relevance of press factor as a characteristic of patients' demand

behaviour.

The second clinical study to be conducted with PRODAC, involving 20 patients, has given validation of the tuned model. It has also shown that there is no significant contribution to be offered by TNS in treatment of acute postoperative pain. Assay results of blood samples taken during the study correlate well with plasma levels estimated by off-line software from PRODAC records, though clinical experience has shown room for improvement in the logistics of taking blood samples and recording the time of sampling.

Firm foundations have thus been laid for further research into the application of model-based control to demand analgesia. A sound mathematical model exhibiting life-like behaviour is the basis upon which such schemes must rest: this has been achieved. The second requirement for application of model-based control is the existence of suitable equipment and software with which to implement ideas and analyse results, and this too has been achieved.

9.2 SUGGESTIONS FOR FURTHER WORK

9.2.1 Model-Based Control

A primary objective of this research is to determine whether model-based control laws can improve the performance of demand analgesia systems. Reasbeck's stochastic and hybrid controllers were designed to offer one-step-ahead dead-beat

control; they attempt to drive the receptor site level of drug concentration at the next sample interval to a level expected to cover the predicted analgesic requirement. They do this, however, using the false assumption that plasma and tissue levels are identical, ignoring the facts presented in Chapter 5. While this does not mean that Reasbeck's controllers will necessarily perform badly, it is likely that a control law designed upon the correct conceptual basis will perform better.

New model-based controllers should be designed in the light of the model revisions. One approach might be to apply conventional linear concepts such as PID control to the estimated states so as to drive the estimated net pain q to a chosen value. Another possibility might be to use k -step-ahead control to take into account the delay between infusion of the drug and the occurrence of its peak effect. Reasbeck's controllers should be re-tested and their performance compared with new designs. Chapter 3 discussed the reason for safe behaviour of Reasbeck's controllers in clinical practice: this was attributable in part to use of the wrong demand non-linearity, which gave protection in the case of excessive button pressing. This is no longer the case, and all model-based controllers should be tested in simulation for benign behaviour in this respect.

Clinical trials involving substantial numbers of patients (eg. more than 50) should be carried out to confirm simulation

results for new controller designs.

9.2.2 Psychology

At present, although it now performs well and has been much improved, there is an underlying imbalance in the mathematical model. On the one hand, the representation of pharmacokinetics has received much attention and is described by 4th order differential equations. On the other hand, the model relies upon extremely crude and simple assumptions about the psychological elements - which are of quite fundamental importance - of pain perception and the patient's reaction in the form of button pressing. Chapter 3 dealt briefly with the psychophysical relationship between perceived pain and button-pressing, and indicated that there is little established knowledge which could help to formulate a better representation of that particular issue. However an area which has so far been ignored, and which might be tackled fruitfully, is the possibility that a patient's decision to press the button may be affected not only by his current pain, but also by his experience over the previous minutes and hours of pressing the button and obtaining pain relief.

There is a body of psychological knowledge, eg. [Schwartz, 1984; Mackintosh, 1974], concerning such interactions between previous experience and current decisions, based on a variety of studies in humans and animals. It would be of interest to integrate some, at least, of this knowledge into the

mathematical model.

At present the model has been tuned to clinical results which used proportional control with a particular gain and particular safety limits. As was mentioned in Chapter 7, the ideal way to validate the tuned model would be to use a different control law, both in simulation and in clinical practice, and to compare these results. The findings of Schwartz, Mackintosh and others imply that there is heuristic coupling between the control law used and the demand behaviour of the patient. It could thus be expected that a model which takes account of this coupling, and has been tuned to real results using one particular control law, will perform better when used to simulate patients subject to a different control law than will a model which does not represent any interactions between previous experience and current decisions. Given that future work must concentrate in part upon the assessment of different control laws, this could prove to be a significant aspect of model behaviour.

The uses for the mathematical model are twofold: as a means of understanding the mechanisms of pain and its relief, and as a conceptual basis upon which to design, test and implement good control laws. It could benefit in both respects from psychological expertise, for pain is as much a psychological as a physiological phenomenon.

9.3 FINAL PERSPECTIVE

The scope for achieving real improvements in care of acute pain is very large. Demand analgesia offers immediate benefits through its use of feedback, but the quality of its contributions must depend ultimately upon the control law it implements. In the Symposium on Patient-Controlled Analgesia in London, July 1984, there was much confusion among leading researchers about: how to choose bolus sizes; whether or not a background infusion should be used; if so, should it be self-adjusting; what measures could be best for all patients; how should they adapt to different patients; and so on. It became clearer than ever that a conceptual framework, upon which to base these questions and from which to approach the answers, is very valuable.

The work reported by Reasbeck and this Author is the only known attempt to develop such a framework, in the form of the mathematical model, and to answer the questions of optimal control by means more methodical and guided than guesswork. It is hoped that the fruits of this research will ultimately be seen in an everyday clinical usage of demand analgesia, embodying control laws which have benefited from the application of control engineering methodology. For this to happen, efforts must continue to be directed towards designing control laws and then using these in large-scale clinical trials, to demonstrate to the medical world the real results achievable.

APPENDIX A
AUTOMATICA PAPER

.

Modelling Estimation and Control in the Relief of Post-operative Pain*

O. L. R. JACOBS,† R. E. S. BULLINGHAM,‡ P. LAMMER,† H. J. McQUAY,§ G. O'SULLIVAN‡
and M. P. REASBECK†

Mathematical modelling, nonlinear estimation and on-line microcomputer control are successfully applied in a clinical feedback system to give improvements in pain relief with the conclusion that methodology of modern control engineering can contribute to practical drug-delivery systems.

Key Words—Bayes methods; biomedical; computer control; control engineering computer applications; demand analgesia; drug-delivery systems; estimation; Kalman filters; medical systems; nonlinear systems; patient-controlled analgesia.

Abstract—The paper arises from collaborative work to explore contributions which the technology of control engineering can make to pain relief. One particular demand analgesia system was studied in which Fentanyl was intravenously infused to relieve pain during the first day after total hip-replacement. A mathematical model is presented in which pain is modelled as the difference between discomfort and comfort and is measured by the frequency at which the patient presses a button. This is inherently nonlinear because there is no button-pressing when comfort exceeds discomfort. A novel nonlinear estimation scheme to handle the nonlinearity is implemented in an on-line microcomputer to provide model-based stochastic control. Experimental results from trials on 20 real patients and 500 simulated patients validate the model and show that the model-based control gives substantially better pain relief than is achieved by conventional demand analgesia, and is robust against modelling errors. Directions for further work are indicated and it is concluded that there is promise for model-based on-line estimation in demand analgesia and promise for the mathematical model as a basis for further research on mechanisms of pain and its relief.

1. INTRODUCTION

CLINICAL relief of pain can be regarded as a classical control problem. The block diagram of Fig. 1 represents a patient as a controlled process whose input u is the administration of pain-killer, whose

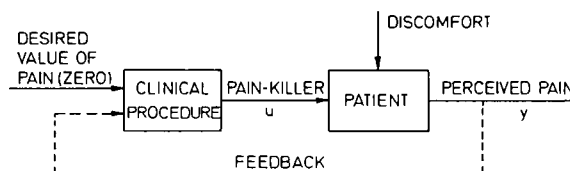


FIG. 1. Clinical relief of pain.

output y is perceived pain and which is subject to disturbances in the form of discomfort, for example resulting from surgery. The input u is controlled by a clinical procedure which aims to regulate the output y to zero. Characteristics of the controlled process and of the discomfort experienced are determined by a variety of factors, physiological, pharmacokinetic, psychological and clinical, which are known to be time-varying, patient-dependent and situation-dependent but which are not all well specified. This type of uncertainty is common to control systems of all sorts: it is classically overcome by using feedback.

In conventional clinical pain relief, the feedback is provided by nursing staff (*British Medical Journal*, 1978) who, at intervals of order 4 h, may administer analgesic as prescribed by a clinician. The nurse's decision whether or not to intervene is based on her observation and interpretation of the patient's pain. This conventional regime is, for two reasons, imperfect and likely to be far from optimal:

- (i) There can be loss of information, or excessive noise, in the feedback path between the patient's pain and the nurse's observation. Some patients may not like to complain of pain, and others may be overfearful. Most nurses will have many other tasks on hand besides monitoring an individual patient's pain.

* Received 2 March 1983; revised 19 January 1984. The original version of this paper was presented at the 6th IFAC Symposium on Identification and System Parameter Estimation which was held in Washington, D.C. during June 1982. The Published Proceedings of this IFAC Meeting may be ordered from Pergamon Press Limited, Headington Hill Hall, Oxford OX3 0BW, U.K. This paper was recommended for publication in revised form by Associate Editor Herman R. Weed under the direction of Editor H. A. Spang III.

† Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.

‡ Nuffield Department of Anaesthetics, University of Oxford, Oxford, U.K.

§ Oxford Regional Pain Relief Unit, Abingdon Hospital, Abingdon, U.K.

- (ii) The discrete-time control interval of order 4 h is very slow compared to the times over which pain may vary and compared to the pharmacokinetic time constants for many analgesics.

The regime nevertheless achieves some pain relief. Its main advantages are that it requires no special equipment and that it is known to be safe. Safety is important because current analgesics for relieving severe pain are all of opiate type and can produce dose-related side-effects which range from the unpleasant, e.g. nausea and vomiting, to the hazardous, e.g. ventilatory depression.

This paper arises from collaborative work to explore contributions which technologies of control engineering could make to pain relief. Preliminary results on the value of mathematical modelling and of on-line computer control have already been reported (Jacobs *et al.*, 1981, 1982). We now present a novel mathematical model of the patient and his pain, and use it as the basis for discussing and improving pain-control systems in which the feedback loop is closed by the patient himself to provide what is known as "demand analgesia".

In demand analgesia the patient is provided with a hand-held button and is instructed to press it whenever he feels uncomfortable. The timing of button-pressing provides a feedback signal to control an infusion pump which automatically administers intravenous analgesic: intensity of button-pressing is not monitored. Figure 2 illustrates the main features of a demand analgesia system and makes the point that the controller, whose input is button-pressing y and whose output is analgesic infusion rate u , could be implemented in an on-line microcomputer. This system eliminates the above-mentioned imperfections of conventional nursing regimes and could therefore be expected to give much better pain relief. A valuable by-product is that time-histories of button-pressing could easily be recorded to provide data of scientific value as a contribution to understanding mechanisms of pain, its perception and relief.

Demand analgesia was first described by Sechzer (1968) and has subsequently had clinical trials at hospitals in Canada, Britain and Sweden. It mostly used special equipment giving an intravenous bolus dose of preset size in acknowledgement of a button-press by the patient. Tamsen *et al.* (1982) summarize this work in which "investigators were impressed by the large variation in individual analgesic demand

and the almost unanimous enthusiasm among their patients over the analgesia afforded by demand analgesia" and comment, on the basis of their own trials, that demand analgesia "appears to be a therapeutic strategy which is very well suited for adults who are rational and not in circulatory shock. The additional costs involved in the necessary apparatus will probably restrict its use to patients in severe pain." Most clinical trials have been in the relief of post-operative pain, a common and unwanted side-effect which is often the subject of intensive care in a recovery ward for a day or so after some major surgery.

Systems of demand analgesia are essentially nonlinear. They include a diode-type nonlinearity, shown in Fig. 3, which is characteristic of demand-activated control systems. This nonlinearity arises because the patient makes demands only when he is in pain: no feedback signal is generated when the comfort due to analgesia exceeds the discomfort which is being relieved. The effect is that negative values of the feedback signal from the controlled process are clipped to zero.

Our mathematical model explicitly accounts for the demand nonlinearity. The model is derived from the point of view of control engineering; it uses structural knowledge where this is available, for example about pharmacokinetics, but otherwise consists of simple behavioural assumptions. Although crude, the resulting model provides a basis for designing and improving systems of demand analgesia. It could also be relevant to future scientific studies of pain and its mechanisms.

The mathematical model is presented in Section 2 of this paper. It leads in Section 3 to a discussion of existing demand analgesia systems and to the conclusion that these are nonoptimal because they use linear controllers which cannot be properly matched to the demand nonlinearity. Improved suboptimal performance can be obtained by model-based separated stochastic control using a novel nonlinear estimation scheme (Jacobs, 1983) in which Bayes's rule is implemented on-line to account for the demand nonlinearity and is cascaded with a conventional extended Kalman filter.

This novel control scheme is described in Section 4. It is implemented with a discrete-time sample interval of 1 min, which is fast compared to time constants of the controlled process but allows scope

"If it hurts, press the button"

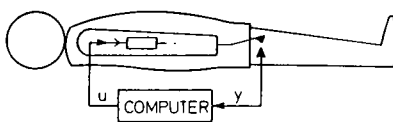


FIG. 2. Demand analgesia.

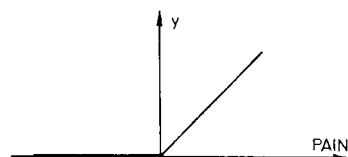


FIG. 3. Demand nonlinearity.

for substantial on-line computations. The work reported and the numbers quoted relate to one particular clinical situation in which Fentanyl was intravenously infused to relieve pain in patients recovering from total hip-replacement. However, the conclusions are applicable to demand analgesia in general.

The method of research combined computer simulations and clinical trials. A first use of simulation was to validate the model; life-like simulated time-histories were produced corresponding to those obtained in clinical trials using conventional demand analgesia. The next use of simulation was to demonstrate that the model-based controller, with its nonlinear estimator, could be expected to give improved performance. Clinical trials then confirmed that the simulated improvements could be achieved in practice, and thus provided further validation of the model. Finally, multiple simulations provided comparative data on 500 simulated patients: these data have statistical significance which would be virtually unobtainable in real, ethically acceptable, clinical trials using only a single machine.

Section 5 describes the experimental work, discusses modelling errors and indicates what computing hardware was used. Section 6 presents experimental results from the simulations and the clinical trials. Conclusions and directions for further work are summarized in Section 7.

2. MATHEMATICAL MODEL

Figure 4 is a block diagram of the mathematical model of the patient and his pain. It shows that analgesic, administered at a rate u (measured in $\mu\text{g s}^{-1}$) affects first the blood plasma level x and then the drug concentration x_1 (ng ml^{-1}) in brain tissue, to give a pain-killing effect which is described here as "comfort" c and is measured in units (p) to be defined below. Little is known about the relationship between comfort and tissue-level except that it is determined by neurophysiological and

psychological factors and is highly patient-dependent.

Perceived pain y , measured in button-pressing units (s^{-1}), is assumed to depend on the difference q between "comfort" c due to administration of the analgesic drug and the "discomfort" d which is to be relieved. The variables c, d, q are measured in units called "pangs" (p), defined as "the amount of pain which, when perceived by button-pressing, as here, produces presses at the rate of 1 s^{-1} ". The working unit is the centipang (cp). The perception mechanism relating y to q includes the demand nonlinearity of Fig. 3 which clips off all negative values of y .

Components of the model, represented as blocks in Fig. 4, are more fully specified as follows.

2.1 Pharmacokinetics

Intravenously infused analgesic is thought (Wagner, 1976) to affect drug concentration x in the blood plasma according to pharmacokinetics which can be approximated by a linear transfer function usually consisting of not more than three exponential terms in the form

$$\frac{X(s)}{U(s)} = \sum_{j=1}^3 \frac{K_j}{1 + sT_j} \quad (1)$$

Typical values of the six parameters K_j, T_j for some common analgesics are shown in Table 1. In the literature these values are sometimes quoted explicitly and sometimes implicitly in the form of coefficients of a compartmental model (Reasbeck, 1982). Drug tissue-level x_1 in the brain follows the blood plasma level x with a transport delay τ of order 1 min which is smaller than most of the time constants in Table 1 and is sometimes neglected.

2.2 Physiology/psychology

So little is known about the relationship between comfort and drug concentration in the brain tissue that it is assumed here to be a simple proportional

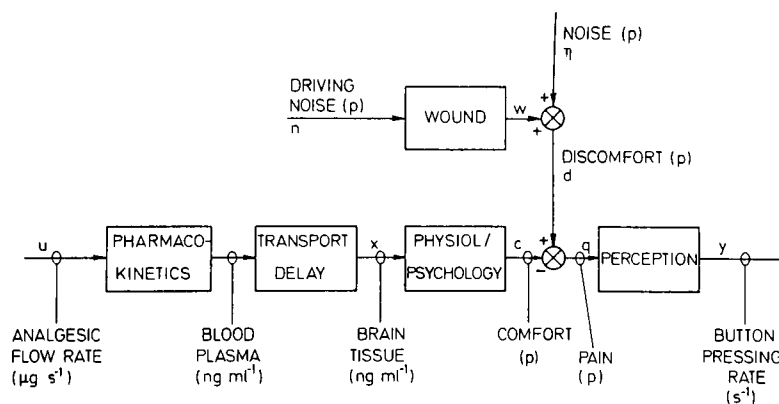


FIG. 4. Structure of mathematical model of patient.

TABLE 1. TYPICAL VALUES OF PHARMACOKINETIC COEFFICIENTS

Analgesic	K_1 (min l^{-1})	K_2 (min l^{-1})	K_3 (min l^{-1})	T_1 (min)	T_2 (min)	T_3 (min)
Buprenorphine	0.413	0.227	0.536	1.87	15.4	162
Diamorphine	0.771	0.134	0.37	5.81	13.3	165
Fentanyl	0.087	0.072	0.908	2.33	18.2	313
Methadone	0.24	0.816	4.771	3	350	1300
Morphine	0.133	0.188	0.5	1.92	18.9	244

scale factor, called 'relief'. There have been clinical indications (McQuay *et al.*, 1981) that relief decreases in proportion to the total amount of analgesic infused, as the patient becomes habituated to the drug. Nevertheless relief is, for the sake of simplicity, assumed here to have a constant, if unknown, value represented below by a state variable x_5 (in units $\text{p}(\text{ng ml}^{-1})^{-1}$). To estimate this value, which is probably highly patient-dependent, would be a worthwhile scientific objective in its own right.

2.3 Wound

The transient component of discomfort is characterized by a Wiener process having transfer function

$$\frac{W(s)}{N(s)} = \frac{1}{1 + sT_4} \quad (2)$$

with time constant T_4 . Clinical experience shows that for total hip-replacements T_4 may have a value of about 8 h and that for abdominal or back surgery the value may be several days. Standard deviations of the driving noise n and of the additive random component η are assumed to decay exponentially with the time constant T_4 of healing. Initial values for variances of the random variables n , η and for the transient component w were determined by tuning simulations of demand analgesia to give life-like behaviour (Reasbeck, 1982).

More extensive evaluation of parameters in the wound model corresponding to various operations is a subject for further research.

2.4 Perception

So little is known about the mechanisms whereby patients perceive and report pain that this part of the model can consist of nothing more than plausible and simple assumptions. A model of perceived pain as arithmetic difference between discomfort and comfort

$$q = d - c \quad (3)$$

is used because it is familiar to control engineers as a way of representing the effect of an external load; it does not yet have much known physiological or psychological basis.

Discomfort d is modelled as the sum of a transient stochastic term w representing a healing surgical wound plus a random noise η which accounts for other incidents such as coughing or commotion in the ward. The transient term w is modelled as a Wiener process forced by white noise n and starting from some positive initial value $w(0)$. Different wounds resulting from different surgical operations could be characterized by Wiener processes having different transfer functions. It is assumed that the button-pressing frequency y is a continuous variable, proportional to q and subject to the demand nonlinearity according to

$$y = \begin{cases} q \text{ pangs} & \text{if } q \geq 0 \\ 0 & \text{if } q < 0. \end{cases} \quad (4)$$

The crudeness in this representation of what might more accurately be described as very slow PFM (Jones *et al.*, 1961; Pavlides and Jury, 1965) is less significant than that of other aspects of the mathematical model, particularly assumptions, implicit in the definition of the pang and in equation (3), that response y to pain q can be characterized by linear equations. Better representations of the PFM in demand analgesia could be a subject for further research.

3. LINEAR CONTROLLERS

The above mathematical model provides a basis for discussing control algorithms. Figure 5 represents demand analgesia as a classical feedback control system in which the patient model of Section 2 is summarized by its perception nonlinearity, by the disturbing discomfort d , and by an apparently linear block having transfer function

$$G(s) = \frac{K(1 + b_1s + b_2s^2)e^{-s\tau}}{(1 + sT_1)(1 + sT_2)(1 + sT_3)} \quad (5)$$

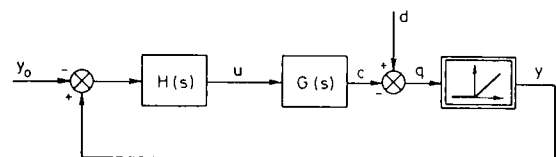


FIG. 5. Demand analgesia as classical feedback control.

where the coefficients b and T are determined by the pharmacokinetics of equation (1) but the gain K depends on relief as defined in Section 2 and is therefore unknown. The desired value y_0 of perceived pain is usually zero. The block with transfer function $H(s)$ represents linear feedback control and corresponds to what has been used in previous demand analgesia systems.

The regime used in most clinical trials (Sechzer, 1971; Forrest, Smethurst and Kienitz, 1970; Keeri-Szanto and Heaman, 1972; Evans *et al.*, 1976; Tamsen *et al.*, 1979) can readily be seen to correspond to simple proportional control, as follows. The regime is based on giving a bolus of constant size ($B\mu g$) in response to each button-push. This corresponds to the action of a simple proportional-control transfer function in Fig. 5, of the form

$$H_1(s) = B\mu g s^{-1}/s^{-1}. \quad (6)$$

The resulting feedback system contains no integration and so its steady-state response y_{ss} to a constant disturbance d would be non-zero by an amount

$$y_{ss} = d/(1 + BK). \quad (7)$$

In many control systems it is acceptable to have nonzero steady-state error proportional to the disturbance d but reduced by a large factor $(1 + BK)$ as in equation (7). However error in the pain-relief system corresponds to pain experienced by the patient and should be eliminated as far as possible by using a controller which is able to match the drug tissue-level in the patient to his analgesic need, without requiring continual demands from him. The classical way to eliminate steady-state error would be to add an integral term to the control of Fig. 5 to give a proportional plus integral (PI) controller having transfer function

$$H_2(s) = B(1 + 1/sT_i). \quad (8)$$

The resulting steady-state response to a constant disturbance d would have zero error, if only there were no demand nonlinearity on the output of the system. The effect of the nonlinearity is that negative error signals are clipped off so that there is no way in which the integral component of controller output could ever be decreased. This difficulty is met in one commercially available system (White, Pearce and Norman, 1979) by making the infusion rate u the sum of two components. One component u_1 is proportional to the demand rate y , as in $H_1(s)$ of equation (6). The other component u_2 is a background infusion similar to an integral control term; it is incremented at a fixed rate μ whenever y exceeds a threshold y_2 and is decremented at the

same rate whenever y falls below a lower, but still positive, threshold y_1 . The resulting infusion rate $By + u_2$ can be regarded as a practical approximation to what would be generated by a PI controller having the transfer function $H_2(s)$ of equation (8) with integral time constant

$$T_i = 2\mu/y_0B \quad (9a)$$

where

$$y_0 = (y_2 - y_1)/2. \quad (9b)$$

This control, when properly stabilized, would produce zero steady-state error in the system of Fig. 5 but the steady-state value of perceived pain would then be y_0 rather than zero. Simulations showed the performance of such a system to be slightly better than that of simple proportional control (Reasbeck, 1982) although not as good as that reported below.

4. SEPARATED ESTIMATION AND CONTROL

Further improvement in performance is obtainable by using a nonlinear controller designed to match the nonlinearities of the controlled process. The demand nonlinearity in particular is so severe that classical linear feedback cannot be expected to be close to optimal. The nonlinear controller uses separate algorithms to estimate states of the controlled process and to implement control action; it has the separated structure of a theoretically optimal stochastic controller (Jacobs, 1974) and can therefore be expected to give good suboptimal performance. The algorithms are implemented using an on-line microcomputer and are based on a discrete-time state-variable representation of the model of Section 2, which is given in Section 4.1. The novel nonlinear state-estimator is summarized in the Appendix. The possibility that not all five states would be simultaneously observable is discussed in Section 4.2.

4.1 State-variable representation

Five state-variables are needed to represent the model specified in Section 2. They are:

- x_1 tissue-level (in $ng\ ml^{-1}$)
- x_2, x_3 two more dynamic states in the third-order response of x to u [equation (1)] (also in $ng\ ml^{-1}$)
- $x_4 \equiv w$ transient component of wound (in p)
- x_5 relief (in $p(ng\ ml^{-1})^{-1}$).

In terms of these variables the comfort c , discomfort d and net pain q are

$$\begin{aligned} c &= x_1 x_5 \\ d &= x_4 + \eta \\ q &= x_4 - x_1 x_5 + \eta. \end{aligned} \quad (10)$$

The corresponding dynamical equations are linear and can be written in discrete-time vector-matrix form (Reasbeck, 1982)

$$x(i+1) = Ax(i) + bu(i) + \xi(i) \quad (11)$$

where i is the discrete-time integer and the elements a in

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 \\ 0 & 0 & 0 & a_4 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (12a)$$

depend on the sampling rate and on the coefficients K , T of equations (1) and (2), as also do the coefficients b in

$$b = [b_1 \ b_2 \ b_3 \ 0 \ 0]^T \quad (12b)$$

The non-zero element $n(i)$ in

$$\xi(i) = [0 \ 0 \ 0 \ n(i) \ 0]^T \quad (12c)$$

is the decaying noise which drives w .

Two nonlinearities affect relationships between the above states x and the output y . One is in equation (10) which is nonlinear because of uncertainty about the relief x_5 . The other is the severe demand-nonlinearity of Fig. 3. This is, in addition, quantized by discrete-time implementation so that the feedback signal $y(i)$ becomes the number of button-presses in one sample period Δ and is the largest integer less than or equal to $\Delta\bar{y}$ where $\bar{y}$ is the average button-rate over the past sample period.

4.2 Nonlinear state-estimator and observability

Each of the two nonlinearities in the mathematical model requires its own nonlinear estimation algorithm. The demand-nonlinearity of Fig. 3 is handled by an on-line implementation of Bayes's rule which generates the conditional mean and variance of the predictable component z of pain q , where z is

$$z \equiv q - \eta = x_4 - x_1x_5. \quad (13)$$

These conditional statistics are weighted so as to provide a pseudomeasurement y' of z which can be used to drive an extended Kalman filter (EKF) handling the nonlinearity due to uncertainty about x_5 and generating estimates of the states x .

Neither estimator is optimal. The nonoptimality of each arises because it is assumed that probability distributions for the random variables can be approximated by normal distributions. This is the

standard simplifying assumption about non-linear states x estimated by an EKF: it is also applied here to the random variable z estimated by Bayes's rule, although the distribution for z could not really be normal if the states x in equation (13) were normal. Details of the estimation are summarized in the Appendix.

Potential observability problems can be discussed with reference to nonsingularity of the matrix

$$M = [c^T \ A^T c^T \ (A^T)^2 c^T \ (A^T)^3 c^T \ (A^T)^4 c^T] \quad (14)$$

where A and c are given by equations (12a) and (A11d). This observability matrix of the EKF has the form

$$M = \begin{bmatrix} \text{Various terms } (3 \times 5) \\ 1 & a_4 & a_4^2 & a_4^3 & a_4^4 \\ -\hat{x}_1 & -\hat{x}_1 & -\hat{x}_1 & -\hat{x}_1 & -\hat{x}_1 \end{bmatrix}. \quad (15)$$

This matrix M would be singular, indicating unobservability, if a_4 had the value unity (corresponding to constant x_4) and x_1 were also constant. Unobservability arises here because the wound x_4 and the comfort x_1x_5 both satisfy the same dynamic equation, and there is then no way to discriminate between them from a single measurement of their difference z .

A similar observability problem would arise under good control where the comfort x_1x_5 was made to track the transient component x_4 of the wound. It follows that good estimation of all five state variables is not to be expected under conditions of good control. This limitation can be met by using a "hybrid" control in which relief x_5 is estimated over a short initial period and the resulting estimate $\hat{x}_5$ is subsequently used but not updated, as described in Section 5.1.

4.3 Control law

The stochastic control law was designed to make the comfort c greater than the predictable component w of discomfort by an amount proportional to the magnitude of the unpredictable component η of discomfort. A suitable specification, in terms of one-step-ahead predictions, is

$$\tilde{c}(i+1) - \tilde{w}(i+1) \equiv -\tilde{z}(i+1) = k \times v_\eta(i+1) \quad (16)$$

where k is a positive constant of proportionality which can be tuned in the light of experience: a typical acceptable value was found to be 2. The required value of $u(i)$, based on current estimates $\hat{x}(i)$ from the EKF and determined from the dynamic state equations (11), is

$$u = \{kv_\eta(i+1) + a_4\hat{x}_4 - a_{11}(\hat{x}_1\hat{x}_5 + \sigma_{15}) - a_{12}(\hat{x}_2\hat{x}_5 + \sigma_{25}) - a_{13}(\hat{x}_3\hat{x}_5 + \sigma_{35})\}/b_1\hat{x}_5 \quad (17)$$

where the index (i) has, for greater clarity, been omitted from $u(i)$, $\hat{x}(i)$ and the elements $\sigma(i)$ of $\Sigma(i)$.

5. EXPERIMENTAL WORK

The experimental work consisted partly of simulations and partly of clinical trials. All this work used the mathematical model of Section 2, as further specified in Sections 5.1 and 5.2, and common control laws as specified in Section 5.3. Hardware is briefly summarized in Section 5.4.

5.1 The clinical regime which was investigated

The trials reported here are restricted to a single clinical regime where the pain of patients in a recovery ward after total hip-replacement is relieved by two sequential modes of analgesia:

- (i) A single large bolus given by an anaesthetist whilst the patient is still under general anaesthetic in the operating theatre. This is intended to provide analgesic tissue-levels sufficiently high to maintain comfort during and immediately after transfer of the patient to the recovery ward.
- (ii) Demand analgesia using Fentanyl, at a concentration of $20 \mu\text{g ml}^{-1}$, in the recovery ward for periods of order 24 h.

Nominal values of parameters in the model of Section 2, for the investigated regime, were chosen (Reasbeck, 1982) to give life-like simulations, as follows:

5.1.1 *Pharmacokinetics.* K_j , T_j ($j = 1, 2, 3$) for Fentanyl from Table 1.

5.1.2 *Initial values* of the three pharmacokinetic states x_j from the nominal K_j , T_j and the known large [mode (i)] bolus which was usually $5 \mu\text{g}$ per kg of patient body weight.

5.1.3 *Wound.*

Time constant $T_4 = 480 \text{ min}$

Initial value $x_4(0) = 1 \text{ cp}$

Initial standard deviation of driving noise n
 $\sqrt{v_n(0)} = 0.05 \text{ cp}$

Initial standard deviation of additive noise η
 $\sqrt{v_\eta(0)} = 2 \text{ cp}$.

5.1.4 *Relief*

$x_5 = 2 \text{ cp} (\text{ng ml}^{-1})^{-1}$.

5.2 Modelling errors

Substantial differences between nominal values, such as the above, and correct values for any individual patient are the inevitable consequence of crudeness in the model and of patient variability. The modelling errors were characterized here by a single number Q such that each correct parameter

value P was assumed to differ from its nominal value P_0 according to

$$P = P_0(1 + Q\xi) \quad (18)$$

where ξ is an independent normal random variable with zero mean and unit variance. In the simulation studies each patient parameter was evaluated by randomizing simulated parameters away from their nominal value according to equation (18) but with re-randomization if the result was a physiologically improbable outlier. It was thus possible to simulate a population of individual patients each having the same nominal values.

The estimation and control algorithms were all designed to match the nominal patient values: equation (A11b) allows the EKF to account for some of the uncertainty due to modelling errors. The estimates $\hat{x}$ were initialized to the nominal values of x and the covariance matrix Σ was initialized to account both for modelling errors and for explicitly assumed uncertainty about x_4 and x_5 (Reasbeck, 1982).

The simulations reported here used modelling error Q of 50% and the clinical trials used an estimator with assumed Q of 20%.

5.3 Controls investigated

Three control laws for demand analgesia were investigated.

5.3.1 *Simple proportional control* as used by other investigators of demand analgesia. This was discussed in Section 3 and uses a controller having the transfer function of equation (6).

5.3.2 *The separated stochastic controller* of Section 4 with control computed according to equation (17).

5.3.3 *A hybrid combination* of proportional control plus state-estimation and separated stochastic control, designed to avoid the observability problems mentioned in Section 4.2. Proportional control is used for a short initial period of demand analgesia whilst the nonlinear estimator generates estimates of all five states. After 10 demands or 2 h, whichever is sooner, the estimate $\hat{x}_5$ of relief is frozen and thereafter separated stochastic control is used but with only four states ($x_1, \dots, x_4$) having their estimates updated.

All controls were subject to safety limits on the total amounts of analgesic which could be infused in any 1 min, in any 5 min and in any 1 h. These were of order respectively 50, 100 and 200 μg of Fentanyl. They were seldom activated during the investigation.

5.4 Implementations

Simulations were done on a VAX 11/780 in Pascal.

The clinical trials used a general-purpose 380Z microcomputer with 56 kbytes of usable RAM, a detachable keyboard, dual floppy discs (8 in. diameter) and an inexpensive VDU. It was interfaced to the patient-activated button, to a counter-timer circuit and to a motorized syringe driven by its own 8085 microprocessor (McGraghan *et al.*, 1980), and programmed in Control Basic (Clarke and Frost, 1979). Figure 6 shows this prototype demand analgesia computing system. Cheaper, smaller, more robust and more portable equipment has been subsequently developed (Lammer *et al.*, 1984).

6. EXPERIMENTAL RESULTS

Results are presented here of clinical trials on a group of 20 patients and of simulated trials on 500 randomized mathematical models from the population specified in Sections 5.1 and 5.2.

The real patients were subdivided into three groups, subject to the different controls as follows:

- (i) Five patients under simple proportional control
- (ii) Ten patients under separated stochastic control
- (iii) Five patients under hybrid control.

The simulated patients were subjected to repeated trials under each of the three controls. They were also subjected to a further control regime in which analgesic was infused at a constant rate ($40 \mu\text{g h}^{-1}$) close to the average under stochastic control but not matched to the analgesic need of any individual patient. The purpose of this fourth simulated regime

was to investigate whether the average infusion rate for a population could give good pain relief in individual members of the population. In this simulation the button-pressing had no effect other than to indicate the amount of pain; corresponding to a clinical experiment which would be ethically unacceptable.

Data recorded include time-histories of state estimates $\hat{x}$, time-histories of the actual states x in the simulations, time-histories of button-pressing and of infusion, the average demand rate and average infusion rate over the period of demand analgesia, and the maximum drug tissue-level during simulated demand analgesia.

Figure 7 shows time-histories of estimated tissue-level $\hat{x}_1$ for three of the real patients, one under each control, and for one simulated patient under each of the four controls. The similarity between clinical and simulated time-histories provides some validation of the mathematical model and of the simulation results.

Figure 8 shows the distributions of average demand rate in each of the three clinical groups, together with corresponding distributions from the first 20 simulated patients. The stochastic and hybrid control laws give significant improvements in quality of pain relief, as indicated by demand rates clustered around average values of 2 or 3 dem h^{-1} , compared to the values scattered around a mean of about 10 under proportional control. These results indicate that the model-based estimation can lead to improved performance, and is robust for purposes of control even though observability problems may prevent prolonged simultaneous estimation of all five states.

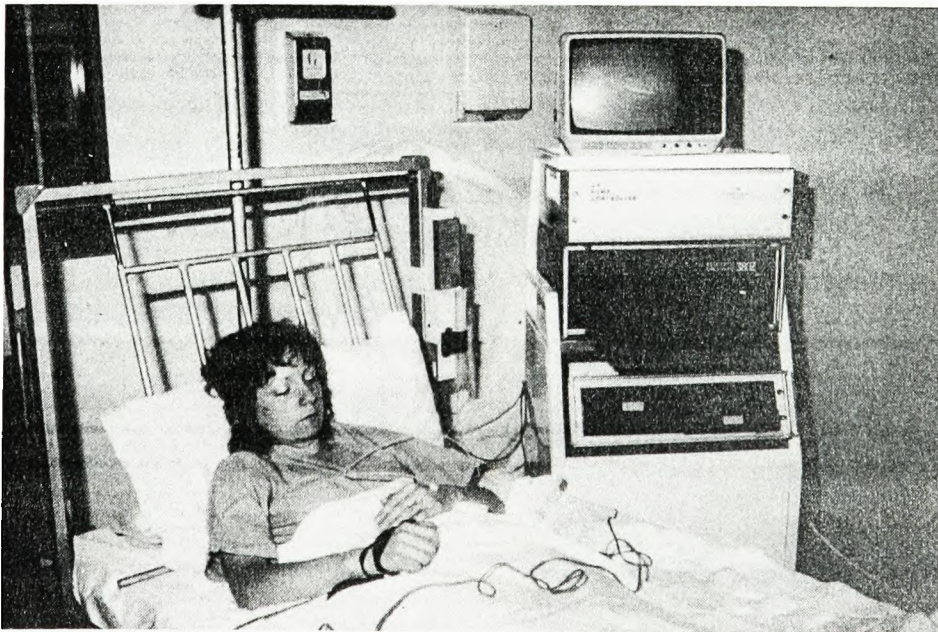


FIG. 6. Demand analgesia system used for clinical trials.

Modelling and relief of pain

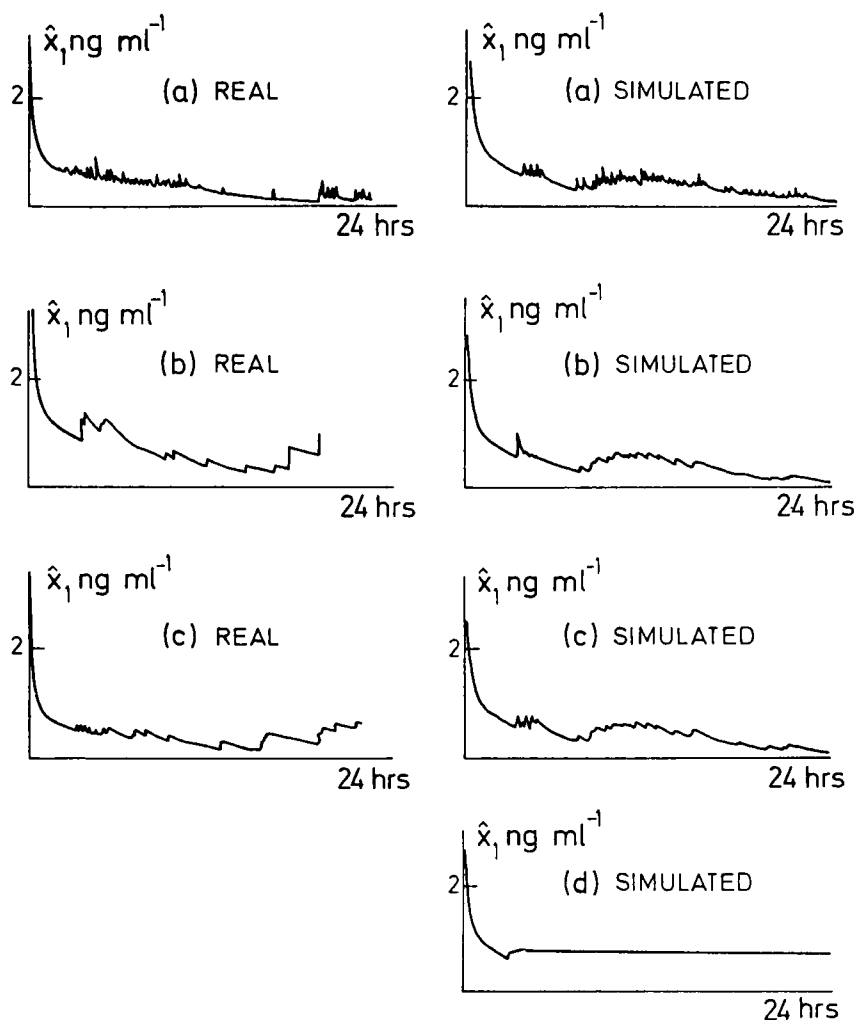


FIG. 7. Time-histories of estimated analgesic concentration in brain tissue. (a) Proportional control. (b) Stochastic control. (c) Hybrid control. (d) Constant infusion.

Comparisons of performance under the different control laws are further quantified in Table 2 which summarizes the results of four 24-h trials on each of 500 simulated patients by quoting the mean and the

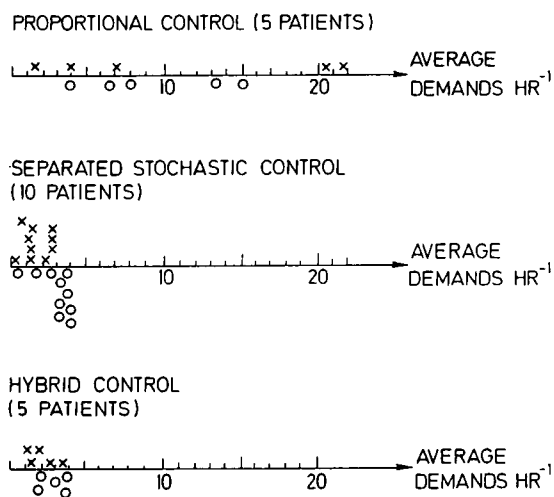


FIG. 8. Distributions of average demand rates. $\times$: Real patients. $\circ$: Simulated patients.

standard deviation of statistics obtained from individual trials, as follows.

Average demand rate in the first pair of columns indicates quality of pain relief; the smaller the better. The three mean values under feedback control (13.61, 4.86, 5.17 dem h^{-1} for proportional, stochastic and hybrid control respectively) correspond well with those from the smaller number of clinical trials quoted above, and indicate that patients under stochastic or hybrid control can expect to experience significantly less pain than those under proportional control. The large mean value (42.98 dem h^{-1}) under constant infusion indicates very poor pain relief. The large standard deviations indicate skewed distributions caused by about 3% of the simulated patients who had abnormal parameter values.

Average infusion rate in the second pair of columns indicates the amount of analgesic used. It can be seen that the significant improvement in pain relief under stochastic or hybrid control was achieved with an acceptably small increase (from

TABLE 2. MEAN AND STANDARD DEVIATION (SD) OF RESULTS FROM 500 SIMULATED PATIENTS

Control law	Average demand (dem h^{-1})		Average infusion ($\mu\text{g h}^{-1}$)		Max. tissue-level (ng ml^{-1})	
	Mean	S.D.	Mean	S.D.	Mean	S.D.
Proportional	13.61	31.04	24.66	23.28	1.05	0.57
Stochastic	4.86	29.46	31.01	29.51	1.13	0.72
Hybrid	5.17	29.50	31.07	28.67	1.05	0.70
Constant	42.98	85.71	40	0	0.76	0.21

about 25 to about $31 \mu\text{g h}^{-1}$) of average infusion. Under demand analgesia the standard deviation of infusion rate is comparable to the mean value; this indicates that infusion is being successfully matched to the variable analgesic needs of individual patients.

Maximum tissue-level indicates danger of poisoning the patient with the drug; the smaller the better. It can be seen that the mean values under stochastic and hybrid control (1.13 and 1.05 ng ml^{-1} respectively) are similar to that (1.05) under proportional control. This shows that the increased average infusion of analgesic under stochastic and hybrid control can be achieved without loss of safety.

Discussion of the work from a medical point of view is given elsewhere (Jacobs and Bullingham, 1984). The conclusion here is that the model of Section 2 is sufficiently accurate and robust that it can be used, in conjunction with an appropriate nonlinear estimator, to give improved pain relief by matching analgesic infusion rates to the needs of individual patients. The hybrid control is recommended for practical purposes, in preference to the stochastic control, because its performance is almost as good and it is free of potential observability problems.

7. CONCLUSIONS

This paper has presented a novel mathematical model of a patient and his pain in demand analgesia. The model explicitly accounts for nonlinearity which is inherent to demand-activated systems. A combination of computer simulations and clinical trials validates the model and indicates that, although crude, it provides a basis for improving the performance of demand analgesia systems.

The work reported includes 20 clinical trials and extensive simulations of one particular regime; demand analgesia by intravenous infusion of Fentanyl for patients recovering from total hip-replacement. Improved performance results primarily from introduction of the model-based nonlinear estimator which uses Bayes's rule to give good estimates in spite of the severe demand-nonlinearity. Similar improvements should therefore be obtainable in other systems with demand-

driven feedback or with severe output-nonlinearity: further research is needed to investigate this possibility.

In the field of demand analgesia, work is in progress to obtain mathematical models and clinical results for more patients and for patients with other forms of severe pain, relieved by other analgesics. A first step in this direction has been the development of improved, cheaper hardware, as mentioned in Section 5.4, which is being used for further extensive clinical trials.

It is possible that the resulting system might eventually become viable as a commercial product but at the time of writing immediate and widespread use of demand analgesia is not expected because it lacks the intrinsic safety of conventional nursing, it requires provision of special equipment, and it is novel. These difficulties may be overcome by designing fail-safe controllers incorporating some device to monitor ventilatory depression, by reducing equipment costs, and by extensive clinical trials to demonstrate efficiency and safety under a variety of circumstances.

The mathematical models, time-histories and other data which arise in connection with demand analgesia could be of wider scientific value in the understanding of pain. The control engineering methodology used here could also contribute to other drug-delivery systems.

Acknowledgements—The authors are grateful to many people at Oxford, in their own Departments and at the Nuffield Orthopaedic Centre who cooperated with them. Authors PL and MPR were supported by the SERC. The 380Z and peripheral hardware were provided by the MRC who also supported author HJM during part of the work.

REFERENCES

- British Medical Journal* (1978). Editorial: Postoperative analgesia. *Br. Med. J.*, 517–518.
- Clarke, D. W. and P. J. Frost (1979). *IEE Conf. Publ.*, 172, 53–57.
- Evans, J. M., J. McCarthy, M. Rosen and M. I. J. Hogg (1976). Apparatus for patient-controlled administration of intravenous narcotics during labour. *Lancet*, i, 17–18.
- Forrest, W. H., P. W. R. Smethurst and M. E. Kienitz (1970). Self-administration of intravenous analgesics. *Anesthesiology*, 33, 363–365.
- Jacobs, O. L. R. (1974). *Introduction to Control Theory*. Chap. 15. (Oxford University Press).
- Jacobs, O. L. R. (1983). Nonlinear estimation by on line computation of Bayesian statistics. *Oxford Univ. Eng. Lab. Rept.* 1471.

- Jacobs, O. L. R. and R. E. S. Bullingham (1984). Modelling estimation and control for demand analgesia. PAA Workshop, Leeds Castle. *Eur. J. Anesth.*
- Jacobs, O. L. R., R. E. S. Bullingham, W. L. Davies and M. P. Reasbeck (1981). Feedback control of post-operative pain. *IEE Conf. Publ.*, **194**, 52–56.
- Jacobs, O. L. R., R. E. S. Bullingham, W. L. Davies and M. P. Reasbeck (1982). Online estimation in the control of post operative pain. *Proc. IFAC Symp. Ident.*, Arlington, Virginia, June 1982.
- Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. (Academic Press).
- Jones, R. W., C. C. Li, A. U. Meyer and R. B. Pinter (1961). Pulse modulation in physiological systems. *IRE Trans. Biomed. Electron.*, **8**, 59–67.
- Keeri-Szanto, M. and S. Heaman (1972). Postoperative demand analgesia. *Surg. Gynecol. Obstet.*, **134**, 647–651.
- Lammer, P., R. E. S. Bullingham and O. L. R. Jacobs (1984). PRODAC demand analgesia computer. PAA Workshop, Leeds Castle. *Eur. J. Anesth.*
- McGrath, T., S. Weller, M. P. Reasbeck and W. L. Davies (1980). A systems and safety problem: an intravenous syringe pump as a computer peripheral. *Microcomputer Applications*, **4**, 167–176.
- McQuay, H. J., R. E. S. Bullingham and R. A. Moore (1981). Acute opiate tolerance in man. *Life Sci.*, **28**, 2513–2517.
- Pavlidis, T. and E. I. Jury (1965). Analysis of a new class of PFM feedback systems. *IEEE Trans. Aut. Control*, **10**, 35–43.
- Reasbeck, M. P. (1982). Modelling and control of postoperative pain. D.Phil. thesis, University of Oxford.
- Sechzer, P. H. (1968). Objective measurement of pain. *Anesthesiology*, **29**, 209.
- Sechzer, P. H. (1971). Studies in pain with analgesic-demand system. *Anesth. Analg.*, **50**, 1–10.
- Tamsen, A., P. Hartvig, B. Dahlstrom and M. Hison Holmdahl (1979). Patient controlled analgesic therapy in the early postoperative period. *Acta Anaesth. Scand.*, **23**, 462–470.
- Tamsen, A., P. Hartvig, C. Fagerlund, B. Dahlstrom and U. Bondesson (1982). Patient-controlled analgesic therapy: clinical experience. *Acta Anaesth. Scand. (Suppl.)*, **74**, 157–160.
- Wagner, J. G. (1976). Linear pharmacokinetic equations. *J. Pharmacokinet. Biopharmaceut.*, **5**, 443–451.
- White, W. D., D. J. Pearce and J. Norman (1979). Postoperative analgesia. *Br. Med. J.*, **ii**, 166–167.

APPENDIX: NONLINEAR STATE-ESTIMATOR

A novel nonlinear estimation scheme (Jacobs, 1985) is summarized here, with application to estimating states x of the model in Section 2. The scheme cascades an on-line implementation of Bayes's rule with an extended Kalman filter.

A.1. Bayes's rule

Bayes's rule is the principal result in probability theory governing estimation of random variables observed through noisy measurements. It gives an equation (Jazwinski, 1970), for a conditional distribution $p(z|y)$ in terms of a prior distribution $p_0(z)$ and a likelihood function $l(y|z)$

$$p(z|y) = \frac{p_0(z)l(y|z)}{\int p_0(z)l(y|z)dz} \quad (A1)$$

which is directly applicable to estimating the variable z of equation (13).

The assumed normal prior distribution for z is written

$$p_0(z) = N(m_0, v_0) \quad (A2)$$

and the white noise η is also assumed to be normally distributed according to

$$p_\eta(\eta) = N(0, v_\eta). \quad (A3)$$

This distribution p_η is used, together with the equations (3) and (4) specifying $y(q)$ and equation (13) defining z , to compute the likelihood $l(y|z)$ as follows:

$$l(y(i)|z) = \text{Prob}(q_{\max} > z + \eta > q_{\min}) = \text{Erf}\{(q_{\max} - z)/(v_\eta)^{1/2}\} - \text{Erf}\{(q_{\min} - z)/(v_\eta)^{1/2}\} \quad (A4)$$

where

$$q_{\max} = y(i) + 1$$

$$q_{\min} = \begin{cases} y(i) & \text{if } y(i) > 0 \\ -\infty & \text{if } y(i) = 0 \end{cases}$$

and $y(i)$ is a positive integer or zero. In equation (A4) $\text{Erf}(\cdot)$ is the standard integral of the normal distribution and can be numerically computed using power series approximations. The conditional distribution $p(z|y)$ can then be computed numerically using an algorithm obtained by substituting from equations (A2) and (A4) into equation (A1). Within the 1-min sample interval Δ of demand analgesia there is ample time for these computations at values of z discretized into say 20 points over a range

$$m_0 \pm \text{constant} \times (v_0)^{-1/2}. \quad (A5)$$

The resulting conditional distribution $p(z|y)$ would be the statistically optimal estimator of z if all the above assumptions were true. Values of the conditional mean m_z and variance v_z are readily computed from $p(z|y)$, they provide the coupling between Bayes's rule and the EKF.

Coupling is achieved by summarizing the combination shown in Fig. A1(a), of additive noise η followed by demand-nonlinearity followed by the Bayes's algorithm, as though it were a linear pseudomeasurement

$$y' = z + \eta' \quad (A6)$$

as shown in Fig. A1(b), with independent normal additive noise η' of variance v_η' . In a real linear measurement the conditional mean and variance would be given by the well-known Bayes's or Kalman result

$$m_z = (m_0 v_\eta + y v_0)/(v_0 + v_\eta) \quad (A7a)$$

$$v_z = v_0 v_\eta/(v_0 + v_\eta). \quad (A7b)$$

These equations (A7) are inverted here to give expressions for the pseudomeasurement y' and its error variance v_η' in terms of the prior and conditional means and variances for z

$$y' = m_z + (m_z - m_0)v_\eta'/v_0 \quad (A8a)$$

$$v_\eta' = v_0 v_z/(v_0 - v_z). \quad (A8b)$$

This pseudomeasurement drives the extended Kalman filter.

The conditional mean m_z computed from $p(z|y)$ would be a better estimator of z than is y' but it cannot be regarded as a measurement under purely independent noise. This is demonstrated from equation (A7a) which shows that for a truly linear measurement the errors e_0, e_z in the prior and conditional means m_0, m_z of z , denfied by

$$e_0 = m_0 - z$$

$$e_z = m_z - z$$

would be related by

$$e_z = (v_\eta e_0 - v_0 \eta)/(v_0 + v_\eta). \quad (A9)$$

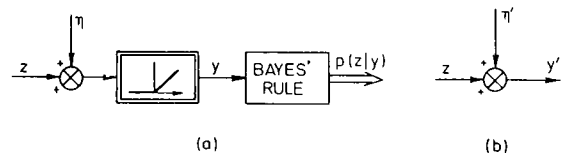


FIG. A1. Pseudomeasurement to couple Bayes's algorithm to EKF. (a) How the measurement is actually processed. (b) Linearization of (a).

Equation (A9) shows that the error e_z in the conditional mean m_z as a measure of z depends on the prior error e_0 . Thus m_z should not be used as the pseudomeasurement y' because the corresponding pseudonoise e_z would not be independent as is assumed in deriving the Kalman filter.

Simulations confirmed that bad estimation results from using the conditional mean as the pseudomeasurement.

A.2 Extended Kalman filter

The EKF is driven by the pseudomeasurement y' of z . It generates estimates $\hat{x}$ of the state-variables x on the assumptions that these satisfy a normal distribution

$$p(x) = N(\hat{x}, \Sigma) \quad (\text{A10})$$

which they do not, and that they are observable via y' . The estimates $\hat{x}$ and their estimated covariance matrix Σ are updated

according to

$$\tilde{x}(i) = A\tilde{x}(i-1) + bu(i-1) \quad (\text{A11a})$$

$$\tilde{\Sigma}(i) = A\Sigma(i-1)A^T + \text{diag}[\sigma_1, \sigma_2, \sigma_3, \sigma_4 + v_\eta(i-1), 0] \quad (\text{A11b})$$

$$\tilde{y}(i) = \tilde{x}_4(i) - (\tilde{x}_1(i)\tilde{x}_3(i) + \tilde{\sigma}_{15}(i)) \quad (\text{A11c})$$

$$c(i) = [-\tilde{x}_3(i) \quad 0 \quad 0 \quad 1 \quad -\tilde{x}_1(i)] \quad (\text{A11d})$$

$$K(i) = \tilde{\Sigma}(i)c^T(i)(c(i)\tilde{\Sigma}(i)c^T(i) + v'_\eta(i))^{-1} \quad (\text{A11e})$$

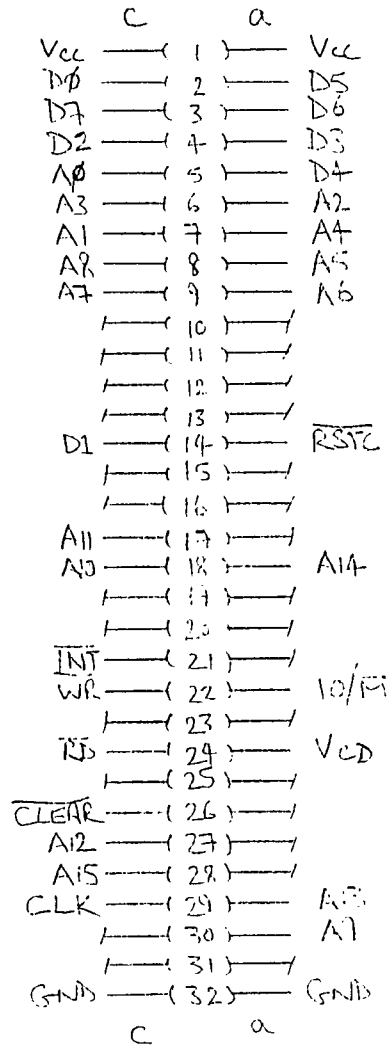
$$\hat{x}(i) = \tilde{x}(i) + K(i)(y'(i) - \tilde{y}(i)) \quad (\text{A11f})$$

$$\Sigma(i) = \tilde{\Sigma}(i) - K(i)c(i)\tilde{\Sigma}(i). \quad (\text{A11g})$$

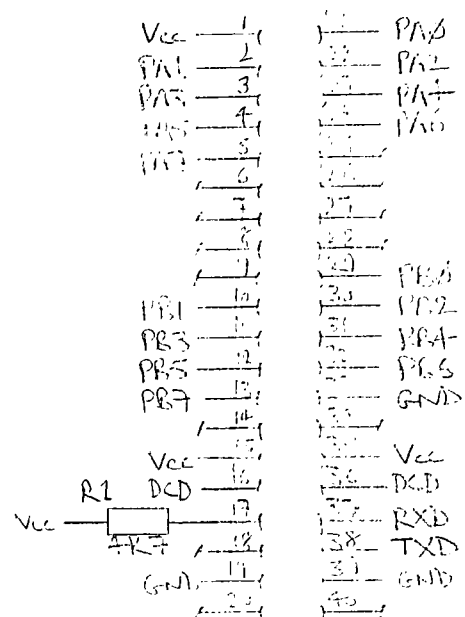
Here the pseudomeasurement y' and its pseudonoise variance v'_η from the Bayes's algorithm appear in equations (A11f) and (A11e), respectively. The nonlinearity of equations (11) and (13) is reflected in equation (A11c) and in the time-varying elements of $c(i)$ in equation (A11d). The elements σ of the diagonal matrix in equation (A11b) account for modelling errors (Reasbeck, 1982).

APPENDIX B
PRODAC CIRCUIT DIAGRAMS

.

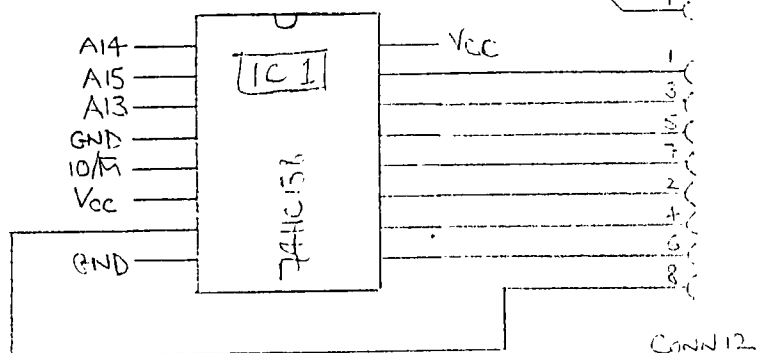
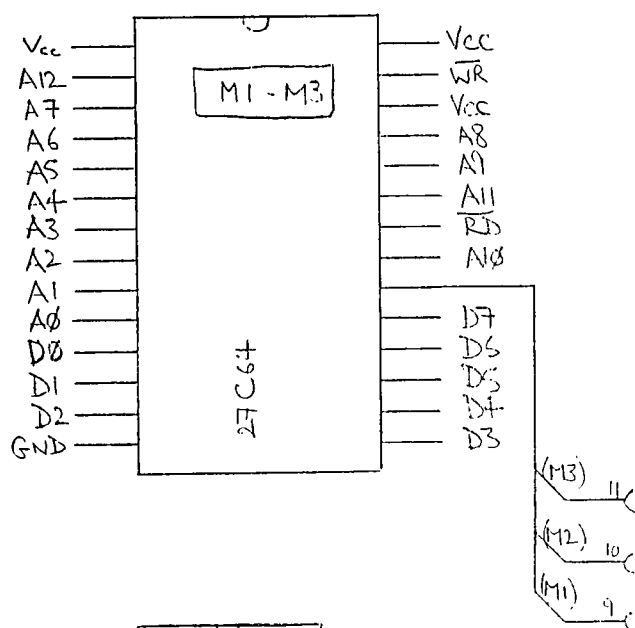


CONN 1

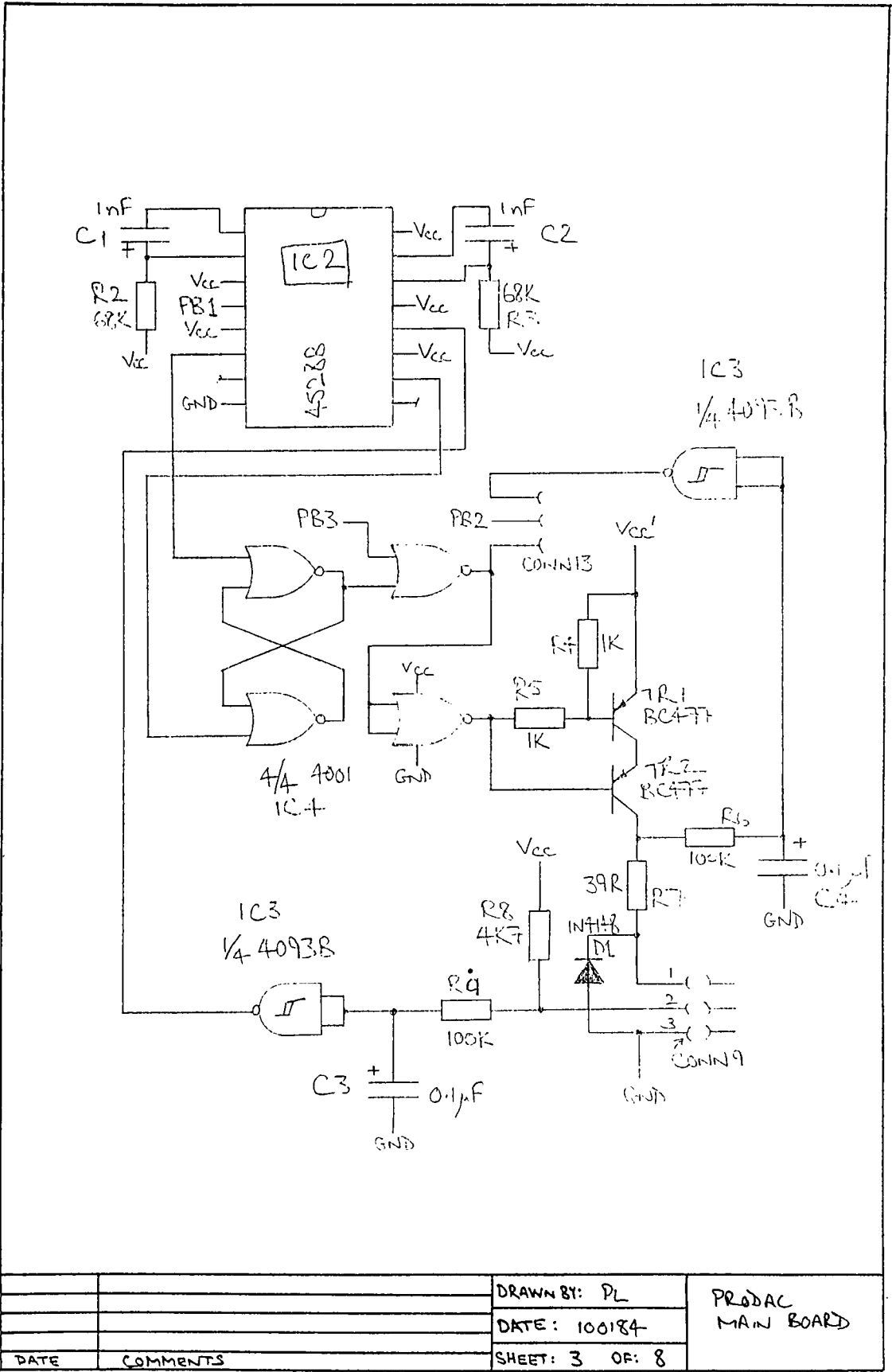


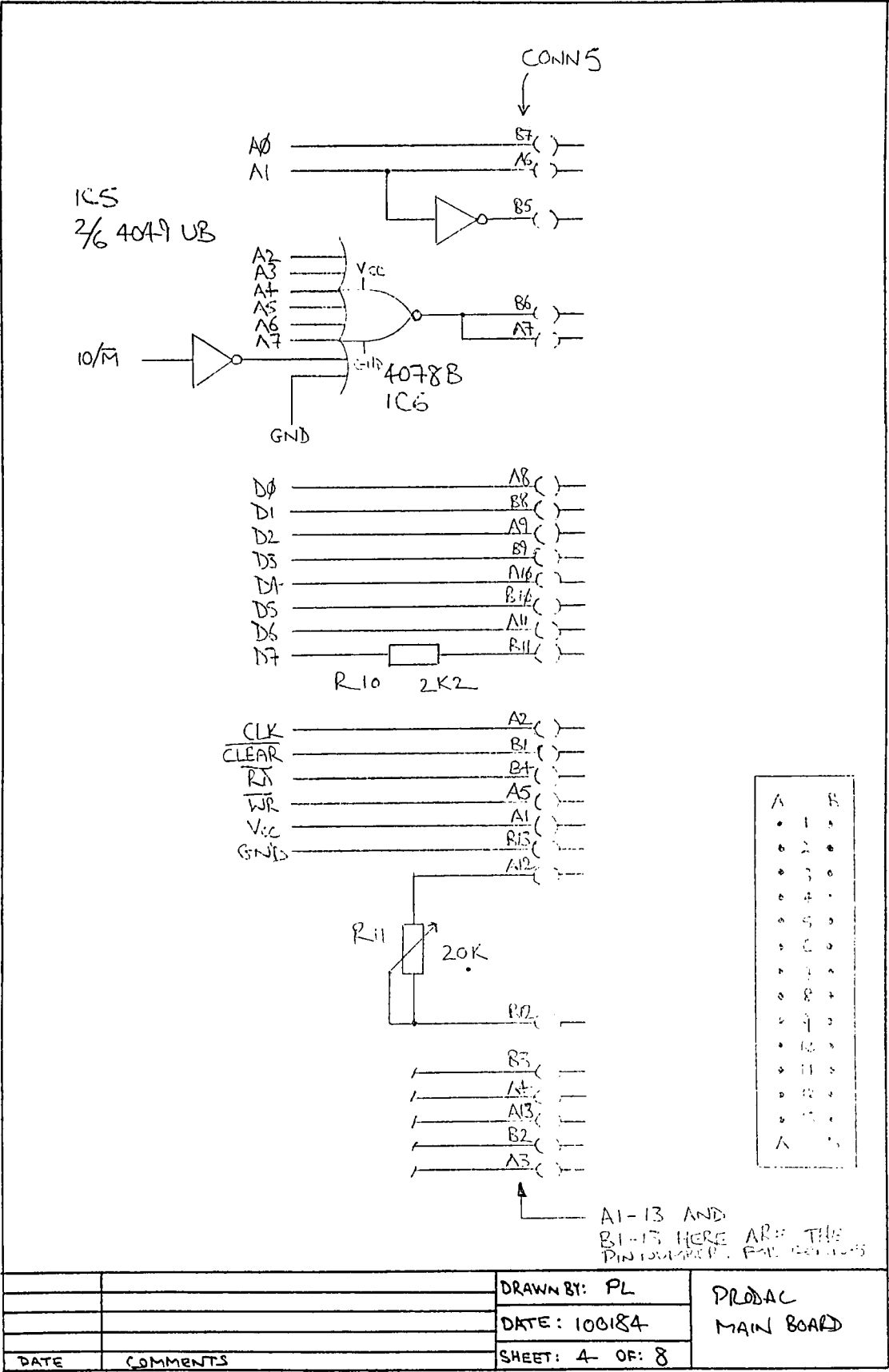
CONN 2

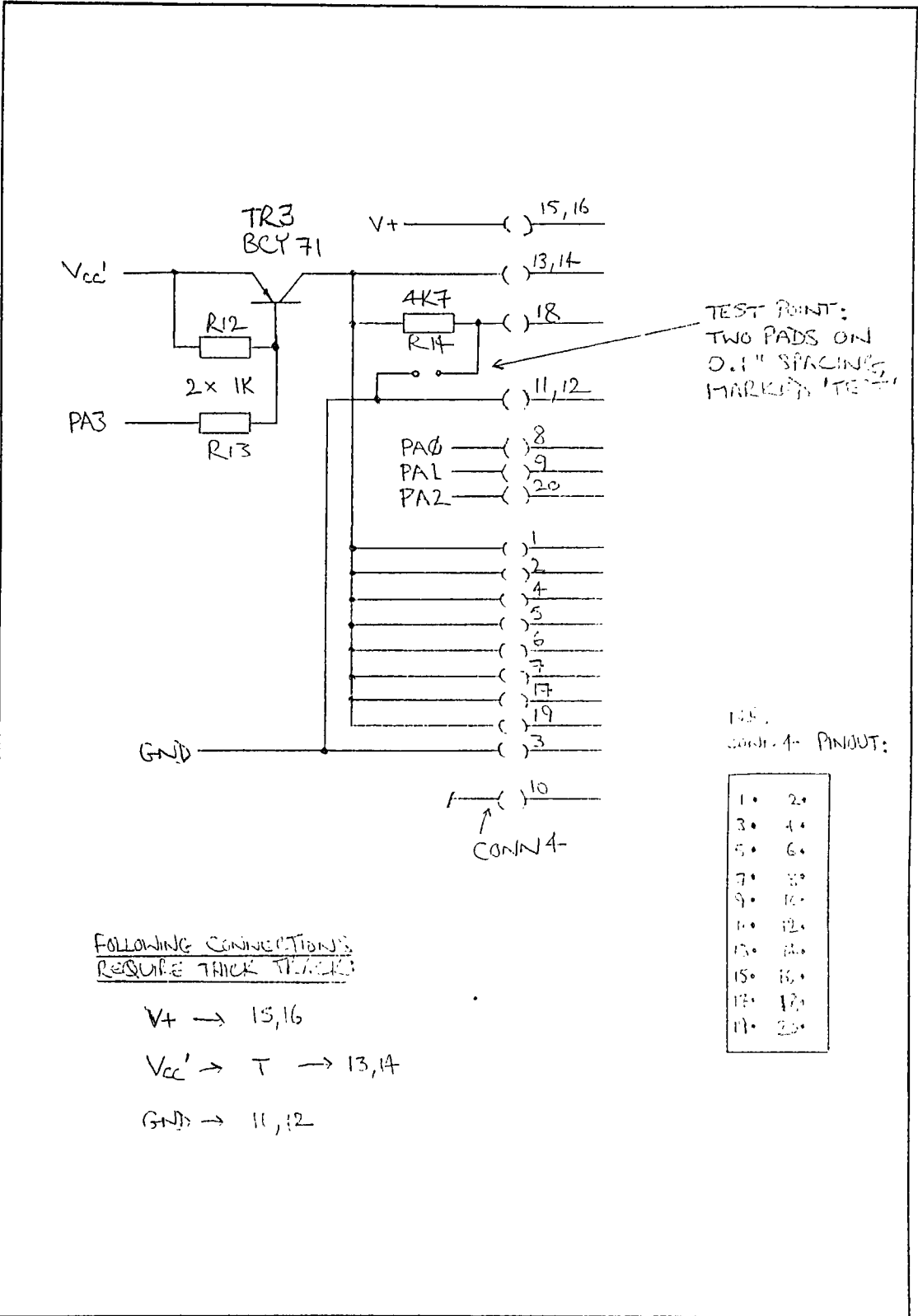
		DRAWN BY: PL	PRODAC MAIN BOARD
		DATE: 10/01/84	
DATE	COMMENTS	SHEET: 1 OF: 8	



		DRAWN BY: PL	PRODAC MAIN BOARD
		DATE: 100184	
1904-K4	IC1 PIN 5 = 10/15 (NOT 1/15)	SHEET: 2 OF: 8	
DATE	COMMENTS		







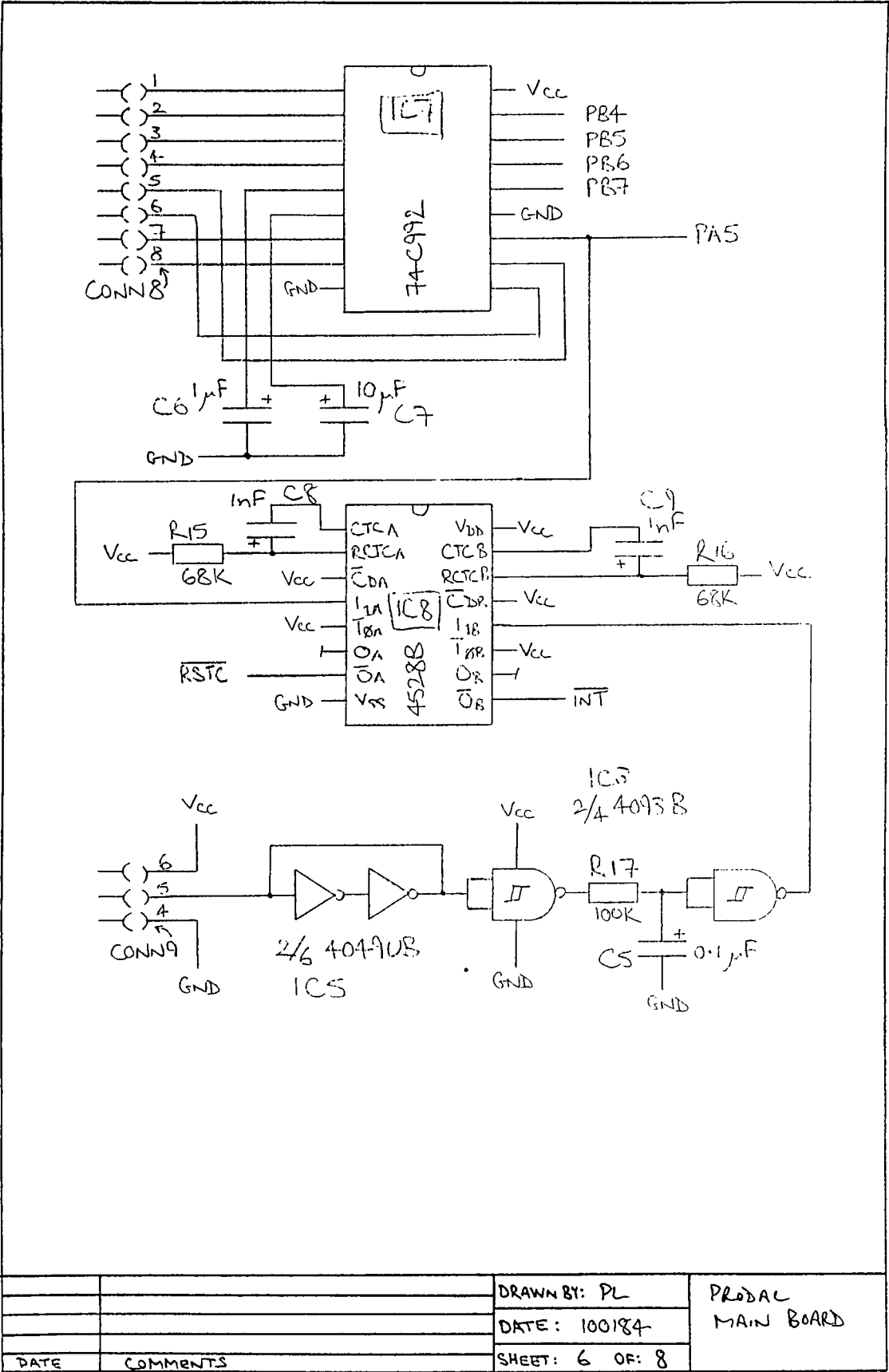
FOLLOWING CONNECTIONS
 REQUIRE THICK TRACKS

V+ → 15,16
 Vcc' → T → 13,14
 GND → 11,12

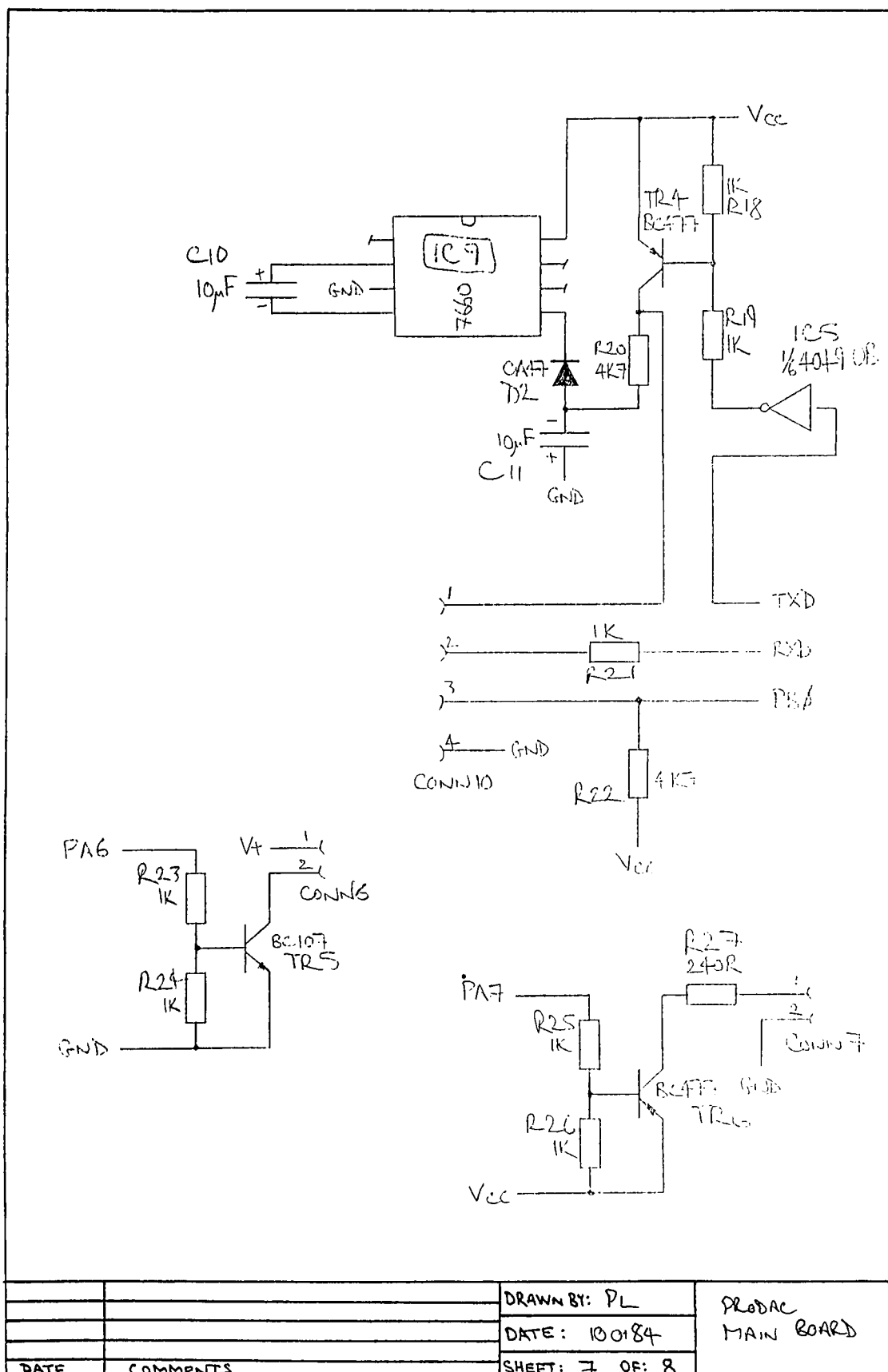
125.
 CONN 4- PINOUT:

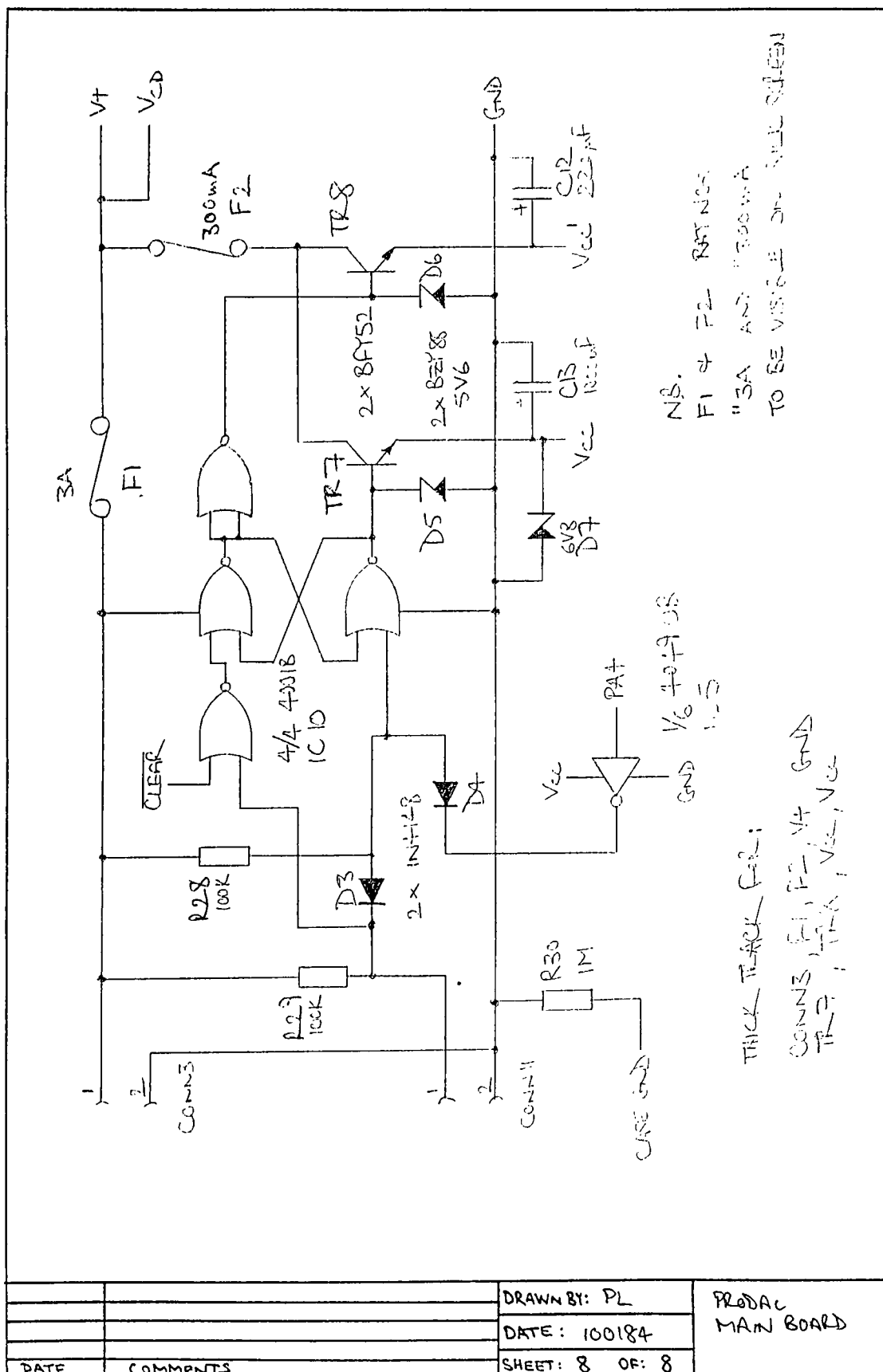
1.	2.
3.	4.
5.	6.
7.	8.
9.	10.
11.	12.
13.	14.
15.	16.
17.	18.
19.	20.

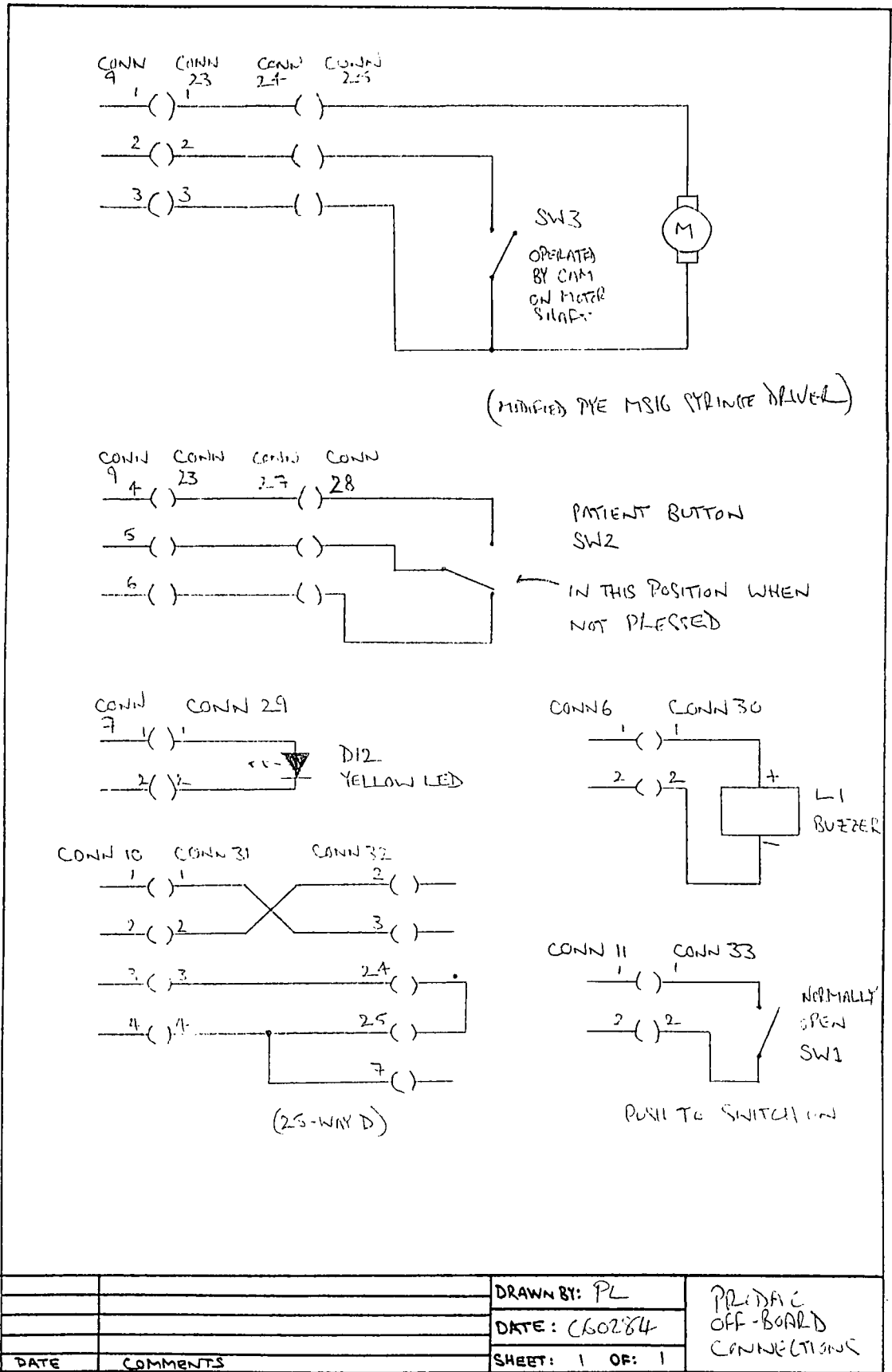
		DRAWN BY: PL	PRODAC MAIN BOARD
		DATE: 100184	
		SHEET: 5 OF: 8	
DATE	COMMENTS		

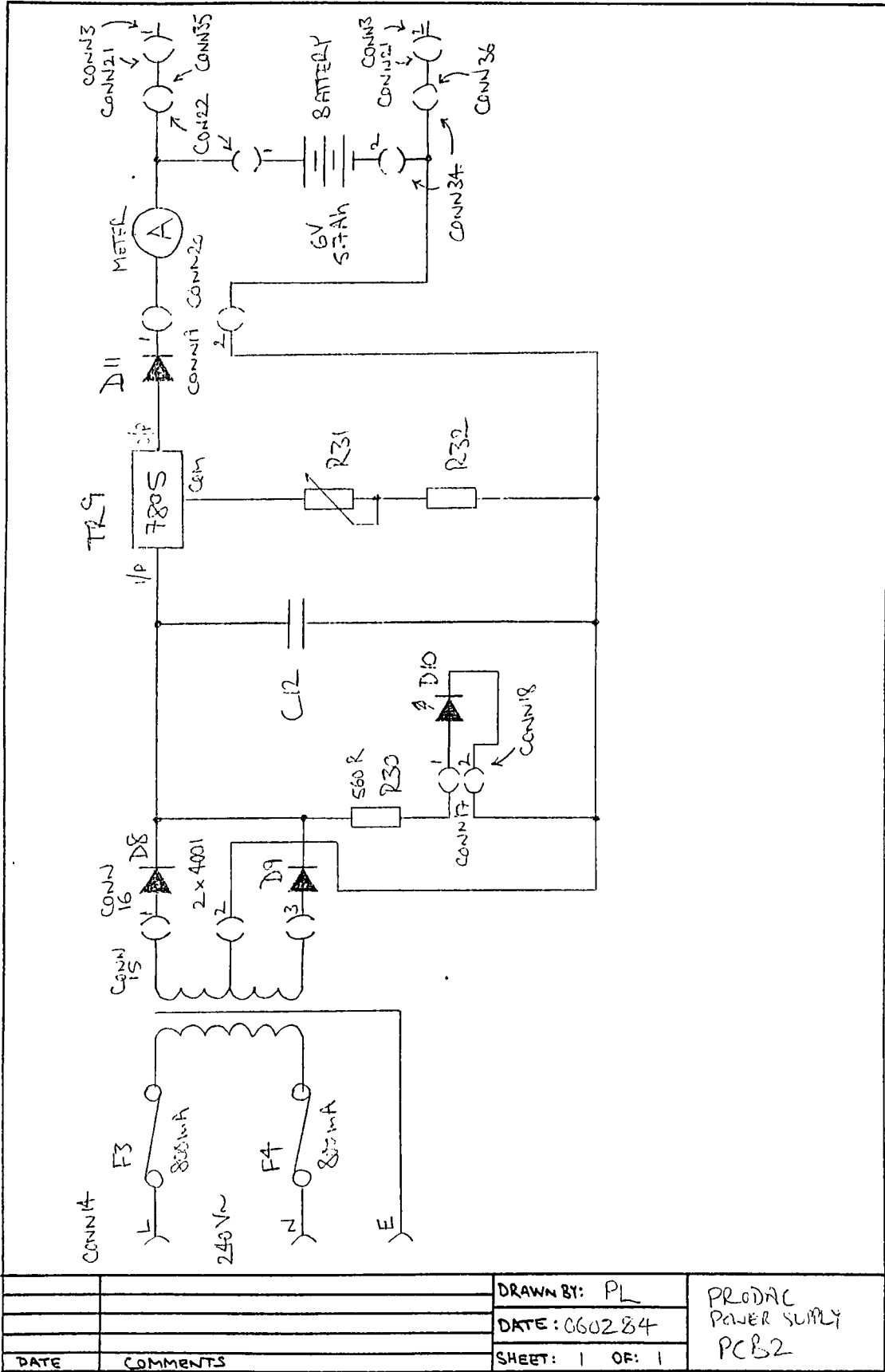


		DRAWN BY: PL	PRODAC MAIN BOARD
		DATE: 100184-	
		SHEET: 6 OF: 8	
DATE	COMMENTS		









APPENDIX C
PRODAC SOFTWARE

```

/*****
/*
/*          control algorithm software
/*          _____
/*
/*
*****/

static int minute;          /* up-counter for number of minutes
                             since start of run          */
static int total;           /* cumulative total of pump-movements
                             over entire run              */
static int total04;         /* total number of pump-movements
                             made in last 4 minutes        */
static int total59;         /* total number of pump-movements
                             made in last 59 minutes       */

/*    the following limits are for pethidine/meptazinol/morphine
    in a 5ml syringe, at concentrations 100, 100 and 15 mg/ml
    respectively. The limits are in numbers of plunger-
    movements, each of volume 0.0258 ml ( = 4.5ml/174 )    */

#define LIMIT01 20          /* max number of pump-movements
                             allowed in one minute         */
#define LIMIT05 40          /* max number of pump-movements
                             allowed in five minutes       */
#define LIMIT60 120         /* max number of pump-movements
                             allowed in one hour           */

static int demands;         /* number of demands made over previous
                             minute                        */
static int givenfix;        /* control output actually made in
                             previous minute               */
static int fixesdue;        /* control output for next minute
static int events;          /* events occurring in past minute
static char record[59];     /* holds ring of delivered doses for
                             safety-limit implementation   */

totals( )
{
    total+=givenfix;
    total04+=givenfix;
    if(minute>4)
        total04-=record[(minute-4)%59];
    total59+=givenfix;
    if(minute>59)
        total59-=record[minute%59];
    record[minute%59]=givenfix;          /* sneaky! */
}

control( )
{
    prop_con( );          /*    proportional control
}

prop_con( )
{
    fixesdue=demands+demands;          /* proportional control gives
- 170 -

```

```

                                0.0517 ml per button-press    */
}

limits()                        /* impose 1-min, 5-min and 1-hour limits
*/
{
    clip(&fixesdue,LIMIT01);
    clip(&fixesdue,LIMIT05-total04);
    clip(&fixesdue,LIMIT60-total59);
}

clip(p,n)
int *p,n;
{
    if(*p>n)*p=n;
}

```

```

/*****
/*
/*      system software: background routines      */
/*      -----
/*
/*
/*****

/*      hardware initialisation      */
/*      -----

fasm
.Z80
;      set up five NSC800 restart/interrupt
;      vectors (not including the NMI, which
;      is not used here):
CSEG
ORG      0000H
JP        .SET01
ORG      002CH
JP        .RSTC
ORG      0034H
JP        .RSTB
ORG      0038H
JP        .INTR          ;mode 1
ORG      003CH
JP        .RSTA

;
;      set up the stack to grow down from the bottom of the rest of RAM:
.SET01: LD      SP,0A7FFH
;
;      set up the correct values in the registers
;      of the NSC810 parallel I/O chip (mapped on
;      I/O locations 040H-05FH):
LD        A,098H
OUT       (040H),A        ;Data - Port A
LD        A,00H
OUT       (041H),A        ;Data - Port B
LD        A,00H
OUT       (042H),A        ;Data - Port C
LD        A,0DDH
OUT       (044H),A        ;DDR - Port A
LD        A,0AH
OUT       (045H),A        ;DDR - Port B
LD        A,03FH
OUT       (046H),A        ;DDR - Port C

;
;      initialise the Epson EA-Y20080AT LCD display
;      mapped on I/O locations:
;
;          00H:      LSI £2 commands
;          01H:      LSI £2 data
;          02H:      LSI £1 commands
;          03H:      LSI £1 data
;
;      the routines .COM1 and .COM2 are in the file
;      iodriver.c as part of the function EAY20x8()
LD        A,10H          ;reset displays
CALL      .COM1

```

```

        LD      A,10H
        CALL    .COM2
        LD      A,01H          ;clear displays
        CALL    .COM1
        LD      A,01H
        CALL    .COM2
;
;   initialise RCA CDP1854A UART for data transfer
;   with 8 data bits, no parity, 1 stop bit:
        LD      A,019H
        OUT     (05H),A
;
        JP      .SET04
;
.RSTC:
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    AF
        PUSH    IX
        PUSH    IY
        EI
        CALL    RSTC_
        POP     IY
        POP     IX
        POP     AF
        POP     HL
        POP     DE
        POP     BC
        RETI
;
.RSTA:
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    AF
        PUSH    IX
        PUSH    IY
        IN      A,(01FH)      ;reset interrupt
        IN      A,(01FH)
        IN      A,(01FH)
        EI
        CALL    RSTA_
        POP     IY
        POP     IX
        POP     AF
        POP     HL
        POP     DE
        POP     BC
        RETI
;
.RSTB:
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    AF
        PUSH    IX

```

```

        PUSH    IY
;
;      NB.: RSTB is the powerlow interrupt. Interrupts are not enabled
;      again once this occurs, and restoring the registers after the
;      call is made is a formality, since there is no return!!
        CALL    RSTB_
        POP     IY
        POP     IX
        POP     AF
        POP     HL
        POP     DE
        POP     BC
        RETI
;
.INTR:
        PUSH    BC
        PUSH    DE
        PUSH    HL
        PUSH    AF
        PUSH    IX
        PUSH    IY
        EI
        CALL    INTR_
        POP     IY
        POP     IX
        POP     AF
        POP     HL
        POP     DE
        POP     BC
        RETI
;
;      initialise real time clock (reset the interrupt)
.SET04: LD      A,00H
        OUT     (01FH),A
        IN      A,(01FH)
        IN      A,(01FH)
        IN      A,(01FH)
        LD      A,15
        OUT     (0BBH),A
        IM      1
        EI
;
        CALL    MAIN_
;
.8080
$endasm

/*                      definitions                      */
/*                      -----                      */

$define EOF '\0'
$define EOT '\004'
$define ACK '\006'
$define CR '\015'
$define XOF '\023'
$define XON '\021'

```

```

#define START 'A'
#define STOP 'B'
#define DELETE 'C'
#define ENTER 'D'
#define MAXREC 4000
#define MAXPAT 50

```

```

/*          declarations of system variables          */
/*          -----          */

static char screen0[8][20];
static char *screen[8];
static char window[8];          /* window for input()          */
static int wpt;                 /* pointer for window          */
static char kbq[10];            /* keyboard queue, FIFO with kbq[0] as
                                front of queue          */
static int kbpnt;               /* pointer to back of queue, equal to
                                number of keys already in queue */
static char date[9];            /* ASCII string containing date in the
                                format 'DD.MM.YY'          */
static char time[9];            /* ASCII string containing time in the
                                format 'HH:MM'          */
static char cumvol[9];          /* ASCII string containing total
cumulative                      volume of drug delivered in ml.
static int ctr_dems,            /* up-counter for patient demands
                                pumpctr,          /* up-counter for time pump runs
                                ctr_duefix,        /* down-counter for number of pump
                                movements still outstanding
                                ctr_fixgave,       /* up-counter for number of pump
                                movements made in the current
                                sample-interval
                                looptime,         /* up-counter for number of RSTB
                                stopctr;          /* up-counter for the time the
                                STOP key is held pressed
static char string[20];        /* utility string
static int stopped,            /* logical flag, true when pump
                                has been stopped either due to
                                overrun or on request
                                finish,           /* logical flag, true when the
                                entire run is terminated
                                empty,            /* logical flag, true when the
                                pump is stopped due to overrun
                                running,          /* logical flag, true during main
                                part of run (ie when control action
                                is taking place)
                                powerlow;         /* logical flag, true when powerlow
                                interrupt has occurred

/*          data storage          */

static char code[6];            /* patient identifier (digital code)
static char startdate[9];       /* DD.MM.YY start of demand analgesia
static char starttime[6];       /* HH:MM start of demand analgesia
static char sinceimd[6];        /* HH:MM since i.m. dose
static char flg_events;         /* events for each minute (bitwise)

```

```

static int file;          /* pointer to next available patient
                           record header block. file==0 if no
                           records are present, file==MAXPAT+1
                           when memory is "full". */
static int recptr;        /* points to next free record in data[] */
static int norecs;        /* logical flag, true if record-keeping is
                           inhibited (i.e. when file>MAXPAT
                           or recptr>MAXREC) */
static struct {           /* record space for all patients */
    int minute;           /* serial no. of minute */
    char dems;            /* integer no. of demands */
    char given;           /* dose delivered (= givenfix) */
    char events;          /* bitwise as follows:
                           7:  syringe emptied
                           6:  STOPped on request
                           5:  reSTARTed
                           4:  run terminated
                           3:  termination by powerdown
                           2:
                           1:
                           0:
                           */
} data[MAXREC], *rp;      /* max number of records MAXREC,
                           rp is a pointer for efficient
                           access */
static struct {           /* record header fields */
    char code[4];         /* 3-digit patient code */
    char stime[6];        /* run start-time */
    char sdate[9];        /* run start-date */
    char imdtime[6];      /* time since i.m. dose */
    int duration;         /* duration of run in minutes */
    int firstrec;         /* pointer to first record */
    int lastrec;          /* pointer to last record */
} patient[MAXPAT], *fp;   /* max number of patients MAXPAT,
                           fp is a pointer for efficient
                           access */

main()
{
    initial();            /* initialisation of system variables */
    body();               /* pre-run dialogue, initialisation of
                           algorithm variables, and run-time
                           control algorithm */
    final();              /* shutdown routines */
}

initial()
{
    int i,j;
    time[2]=': ';
    date[2]='. ';
    date[5]='. ';
    time[5]=EOF;

```

```

date[8]=EOF;
running=0;
ctr_dems=0;          /* up-counter for patient demands      */
pumpctr=0;           /* up-counter for time pump runs        */
ctr_duefix=0;        /* down-counter for number of pump
movements still outstanding            */
ctr_fixgave=0;       /* up-counter for number of pump
movements made in the current
sample-interval                      */
minute=0;            /* up-counter for minutes of run        */
looptime=0;          /* up-counter for number of RSTB
(ie 1/2 second) interrupts           */
flg_events=0;        /* bitwise event flags                  */
strcpy(cumvol," 0.00");
total=0;
total04=0;
total59=0;
stopctr=0;           /* up-counter for the time the
STOP key is held pressed              */
demands=0;
givenfix=0;
fixesdue=0;
events=0;
stopped=0;           /* logical flag, true when pump
has been stopped either due to
overrun or on request                */
finish=0;            /* logical flag, true when the
entire run is terminated              */
empty=0;             /* logical flag, true when the
pump is stopped due to overrun        */
powerlow=0;          /* logical flag, true when powerlow
interrupt has occurred                */
wpt=0;
for(i=0;i<8;i++){    /* initialise screen pointer */
    screen[i]=screen0[i];
}
window[7]=EOF;
clearscr();
kbpnt=0;
cursrow=0;
curscol=0;
curson=0;
if(HX20())
    service();
norecs=0;
if(file>MAXPAT||recptr>MAXREC){
    clearscr();
    print(1,"memory is full !");
    print(3,"to run PRODAC");
    print(4,"without retention");
    print(5,"of new data");
    print(6,"press ENTER");
    print(7,"(else STOP)");
    dumpscr();
    zeroop();
    norecs=1;
}

```

```

}

body( )
{
    char c;
    int i;
    buzz(100);
    dialogue( );
    rundispl( );
    timeron( );                /* enable 0.5 sec interrupts */
    running=1;                 /* set flag */
    file++;                     /* new record in storage */
    ctr_dems=0;                 /* cancel any presses made
                                during dialogue */
    while(!finish){
        while(looptime<117);    /* wait for next minute */
        minute++;
        if(minute%8)             /* correction for timing */
            looptime=1;         /* (RSTA interval is 31/60 sec) */
        else
            looptime=0;
        demands=ctr_dems;
        ctr_dems=0;
        ctr_duefix=0;
        givenfix=ctr_fixgave;
        ctr_fixgave=0;
        events=flg_events;
        flg_events=0;
        records( );              /* records in event of event(s) */
        totals( );               /* records for last hour */
        control( );              /* calculate control for this
minute */
        limits( );               /* impose safety limits */
        ctr_duefix=fixesdue;      /* output control calculated this
                                minute */
        if(minute%60==0 || finish){
            printon( );
            print(8,"\\015");
            strtim( );
            print(8,string);
            print(8," ");
            print(8,cumvol);
            print(8," ml\\015");
            if(!finish)
                printoff( );
        }
    }

    timeroff( );                /* disable 0.5 sec interrupts */
    running=0;
    printon( );
    strtim( );
    print(8,"\\015finished at ");
    print(8,string);
    print(8,"\\015\\015*****\\015\\015\\015\\015");
    if(recptr>=MAXREC||file>MAXPAT){
        - 178 -
    }
}

```

```

        print(8,"memory is now full: no\015further storage
possible");
        print(8,"\015please dump data now!\015\015\015\015");
    }
    else if(file==MAXPAT){
        print(8,"memory has room for only one more patient record
");
        print(8,"\015please dump data now!\015\015\015\015");
    }
    printf();
}

final()
{
    char c;
    buzflash(200);
    kbpnt=0;                /* clear keyboard buffer          */
    shutdown();
}

records()
{
    int i;
    char c;
    if(!norecs){
        if(demands||givenfix||events){                /* if any demand
or drug or event */
            rp=&data[recptr];
            rp->minute=minute-1;
            rp->dems=demands;
            rp->given=givenfix;
            rp->events=events;
            recptr++;
            if(recptr>MAXREC){
                norecs=1;
                strtim();
                prnton();
                print(8,"\015internal memory full
at\015");
                print(8,string);
                print(8," , no more storage\015");
                printf();
            }
            fp->duration=minute;
            fp->lastrec=recptr-1;
        }
    }
}

dialogue()
{
    char c;
    clrscr();
    print(0,"please ensure:      ");
    print(1,"1 syringe loaded and");
    print(2,"2 connected to      ");
    print(3," patient      ");
    - 179 -

```

```

print(4,"3 patient has button");
print(6,"then press ENTER  ");
dumpscr();
while((c=getc())!=ENTER){
    if(c==STOP)
        shutdown();
}
clearscr();
print(0,"enter patient's  ");
print(1,"identity code:  ");
print(2,"    (3 digits)    ");
dumpscr();
input(code,3,4);
clearscr();
print(0,"how long ago was the");
print(1,"i.m. dose given?");
print(2,"please enter:");
print(4,"  hours?");
dumpscr();
input(sinceimd,2,4);
print(5,"minutes?");
dumpscr();
input(sinceimd+3,2,5);
*(sinceimd+2)='\0';
clearscr();
print(1,"demand analgesia");
print(2,"will start when");
print(3,"printing finishes");
dumpscr();
printon();
print(8,"\015");
print(8,"*****");
print(8,"\015");
print(8,"patient code: ");
print(8,code);
print(8,"\015");
print(8,"\015");
print(8,"demand analgesia started");
print(8,"on ");
newdate(date);
newtime(time);
strcpy(startdate,date);
strcpy(starttime,time);
print(8,startdate);
print(8," at ");
print(8,starttime);
print(8,"\015");
print(8,"\015");
print(8,"i.m. dose given\015");
print(8,sinceimd);
print(8," hours  ");
print(8,sinceimd+3);
print(8," minutes\015earlier\015");
print(8,"\015");
print(8,"time:          total volume:");
print(8,"====          =====\015");
*(sinceimd+2)=': ';

```

```

        printf( );
        if (!norecs){
            fp=&patient[file];
            strcpy( fp->code,code);
            strcpy( fp->stime,starttime);
            strcpy( fp->sdate,startdate);
            strcpy( fp->imdtim,sinceimd);
            fp->firstrec=recptr;
        }
    }

rundispl( )
{
    clrscr( );
    print(2,"demand analgesia");
    print(3,"started at ");
    strcpy(screen[3]+11,starttime);
    print(5,"total volume of drug");
    print(6,"infused =          ml");
    strcpy(screen[6]+10,cumvol);
    dumpscr( );
}

strtim( )
{
    strcpy(string,time);
    string[2]=': ';
}

datadump( )
{
    int i,j,m,n;
    char c;
    clrscr( );
    print(3,"please connect HX20");
    dumpscr( );
    while( ! HX20( ))
        if(getc( )==STOP)
            return;

    clrscr( );
    print(1,"start data dump      ");
    print(2,"program on HX20 and");
    print(3,"then press 'START' ");
    dumpscr( );
    kbpnt=0;
    while((c=getc( ))!=START)
        if(c==STOP)
            return;

    clrscr( );
    print(3,"transmitting...");
    dumpscr( );
    txstring("*****");
    while(rxchar( )!=ACK)
        if(getc( )==STOP)
            return;
    for(n=0;n<file;n++){

```

```

        fp=&patient[n];
        txstring(fp->code);
        while( rxchar( )!=ACK)
            if( getc( )==STOP)
                return;
        txstring(fp->sdate);
        while( rxchar( )!=ACK)
            if( getc( )==STOP)
                return;
        txstring(fp->stime);
        while( rxchar( )!=ACK)
            if( getc( )==STOP)
                return;
        txstring(fp->imdttime);
        while( rxchar( )!=ACK)
            if( getc( )==STOP)
                return;
        txhexint(fp->duration);
        txchar(CR);
        while( rxchar( )!=ACK)
            if( getc( )==STOP)
                return;
        for(m=fp->firstrec;m<=fp->lastrec;m++){
            rp=&data[m];
            txhexint(rp->minute);
            txchar(' ');
            hexout(rp->dems);
            hexout(rp->given);
            hexout(rp->events);
            txchar(CR);
            flash(10);
            while( rxchar( )!=ACK)
                if( getc( )==STOP)
                    return;
        }
        txstring("*****");
        while( rxchar( )!=ACK)
            if( getc( )==STOP)
                return;
    }
    txchar(EOT);
    txchar(CR);
    clrscr( );
    print(2,"press DELETE to");
    print(3,"clear memory,");
    print(5,"any other key to");
    print(6,"retain records");
    dumpscr( );
    while(( c=getc( )!=EOF);
    if( c==DELETE){
        file=0;
        recptr=0;
    }
}

txstring(s)
char *s;

```

```

{
    txchar( ' ');
    while( *s)
        txchar( *s++);
    txchar( ' ');
    txchar( CR);
}

hexout(c)
char c;
{
    char c1,c2;
    c1=(c>>4)&15;
    if(c1<10)
        c1=c1+'0';
    else
        c1=c1-10+'A';
    if( rxchar( )==XOF)
        while( rxchar( )!=XON);
    txchar(c1);
    c1=c&15;
    if(c1<10)
        c1=c1+'0';
    else
        c1=c1-10+'A';
    if( rxchar( )==XOF)
        while( rxchar( )!=XON);
    txchar(c1);
}

txhexint(i)
int i;
{
    char c;
    c=i>>8;
    hexout(c);
    c=i;
    hexout(c);
}

dumpscr( )
{
    EAY20x8( screen[0]);
}

clearscr( )
{
    int i,j;
    for(i=0;i<8;i++)for(j=0;j<20;j++)screen[i][j]=' ';
}

print(i,s)
int i;
char *s;
{
    if(i<8)
        /* print to screen */

```

```

        strcpy(screen[i],s);
    if(i==8){                /* printer */
        while(*s){
            while(in(0x40)&2){
                for(i=0;i<100;i++);
            }
            squirt(*s++);
            for(i=0;i<100;i++);
        }
    }
}

strcpy(s,t)                /* copies *t (terminated by NULL) to *s
                           without copying the NULL */
char *s, *t;
{
    while(*t)*s++=*t++;
}

strcpy(s,t)                /* copies *t to *s including NULL */
char *s, *t;
{
    while(*s++=*t++);
}

getc()                    /* definition of getc for keyboard I/O */
{
    int i;
    char c;
    if(kbpnt==0)c=EOF;
    else{
        c = kbq[0];
        for(i=0;i<10;i++)
            kbq[i]=kbq[i+1];
        --kbpnt;
    }
    return c;
}

input(s,j,k)              /* inputs a string of up to j characters from
                           the keypad into *s, using screen line k */
int j,k;
char *s;
{
    char c;
    int i;
    if(j>6) j=6;          /* maximum string size is 6 characters */
    if(j<1) j=1;          /* minimum string size is 1 character */
    for(i=0;i<j+2;i++)
        window[i]=' ';
    window[j+2]=0;
    wpt=0;
    for(;;){
        while((c=getc())==EOF);
        if(c<START){
            if(wpt<j)

```

```

        window[wpt++]=c;
    else{
        window[j]=c;
        for(i=1;i<j+2;i++)
            window[i-1]=window[i];
    }
}
if(c==DELETE){
    if(wpt)
        window[--wpt]=' ';
    else
        window[wpt=0]=' ';
}
if(c==ENTER){
    window[j]=EOF;
    strcpy(s,window);
    return;
}
if(c==STOP){
    shutdown();
}
strcpy(screen[k]+9,window);
dumpscr();
}

wait(n)
unsigned n;
{
    int i;
    for(i=0;i<n;i++);
}

startfix()                /* set pump going:                */
{
    out(0x4d,0x02);        /*      set PB1                */
    out(0x49,0x02);        /*      clear PB1              */
}

stopppump()               /* disable pump:                */
{
    out(0x4d,0x08);        /*      set PB3                */
}

enabpump()                /* enable pump:                */
{
    out(0x49,0x08);        /*      clear PB3              */
}

flash(n)                  /* flash led:                    */
int n;
{
    out(0x48,0x80);        /*      clear PA7              */
    while(--n>0){          /*      wait n units of time   */
        out(0x4c,0x80);    /*      set PA7                */
    }
}

```

```

buzz(n)                                /* sound buzzer:                */
int n;
{
    out(0x4c,0x40);                    /*      set PA6                  */
    while(--n);                        /*      wait n units of time    */
    out(0x48,0x40);                    /*      clear PA6               */
}

buzflash(n)                            /* sound buzzer and flash led:   */
int n;
{
    out(0x48,0x80);                    /*      clear PA7               */
    out(0x4c,0x40);                    /*      set PA6                 */
    while(--n);                        /*      wait n units of time    */
    out(0x4c,0x80);                    /*      set PA7                 */
    out(0x48,0x40);                    /*      clear PA6               */
}

printon( )                            /* switch on printer:           */
{
    out(0x48,0x08);                    /*      clear PA3               */
    out(0x4c,0x05);                    /*      stop paper-feed and set */
                                     /*      serial data high        */
    wait(5000);
}

printoff( )                           /* switch off printer:          */
{
    int i;
    wait(10000);
    out(0x4c,0x08);                    /*      set PA3                 */
    out(0x48,0x05);                    /*      clear PA2 and PA0       */
    wait(1000);
}

shutdown( )
{
    clearscr( );
    print(3,"PRODAC shutdown");
    dumpscr( );
    wait(10000);
    buzz(200);
    shut_down( );
}

zeroop( )
{
    char c;
    for(;;){
        while((c=getc())==EOF);
        if(c==STOP){
            shutdown( );
        }
        if(c==ENTER)
            return;
    }
}

```

```

}

service( )
{
    int year;
    char c;
    buzz(400);
    c=EOF;
    while(c!=ENTER){
        clearscr( );
        print(0,"press:");
        print(1,"1 to set the time");
        print(2,"2 to set the date");
        print(3,"3 to erase records");
        print(4,"4 to dump records");
        print(5,"5 to replace paper");
        print(6,"ENTER to continue");
        print(7,"STOP to shut down");
        dumpscr( );
        while((c=getc( ))==EOF);
        if(c=='1'){
            clearscr( );
            print(0,"please enter");
            print(2,"(24-hour clock)");
            print(3,"  hours?");
            dumpscr( );
            input(time,2,3);
            print(4,"minutes?");
            dumpscr( );
            input(time+3,2,4);
            time[2]=': ';
            RTctime(time);
            newtime(time);
            clearscr( );
            print(0,"the time is now");
            print(2,time);
            dumpscr( );
            wait(10000);
        }
        if(c=='2'){
            clearscr( );
            print(0,"please enter");
            print(2,"  day?");
            dumpscr( );
            input(date,2,2);
            print(3,"month?");
            dumpscr( );
            input(date+3,2,3);
            print(4,"  year?");
            dumpscr( );
            input(date+6,2,4);

            /*      the following obscure expression converts a year into the
                    form required by the 58174A real time clock leap-year
                    register. Note that the year will still need to be set
                    manually once per year (unavoidable with this chip)
                    but that 28th/29th February will be handled correctly:

```

```

*/
    year=0x08>>((date[6]-'0')*10+date[7]-'0')%4;
    out(0xld,year%4);
    date[2]='.';
    date[5]='.';
    RTCdate(date);
    newdate(date);
    clearscr();
    print(0,"the date is now");
    print(2,date);
    dumpscr();
    wait(10000);
}
if(c=='3'){
    clearscr();
    if(file){
        print(3,"are you sure ??");
        print(5,"press 1 for 'YES'");
        print(6,"      0 for 'NO'");
        dumpscr();
        while((c=getc())==EOF);
        clearscr();
        if(c=='1'){
            file=0;
            recptr=0;
            print(3,"records erased");
        }
        else
            print(3,"      coward!");
    }
    else
        print(3,"no records present");
    dumpscr();
    wait(10000);
    c='3';
}

if(c=='4'){
    if(file)
        datadump();
    else{
        clearscr();
        print(3,"no records present");
        dumpscr();
        wait(10000);
    }
}

if(c=='5'){
    clearscr();
    print(3,"just a moment...!");
    dumpscr();
    printon();
    clearscr();
    print(1,"press:");
    print(3,"1 for paperfeed");
    print(4,"2 to continue");
    dumpscr();

```

```

        while((c=getc())!='2'){
            if(c=='1'){
                pfeedon();
                while(key());
                pfeedoff();
            }
        }
        clearscr();
        print(3,"just a moment...!");
        dumpscr();
        printoff();
        c='5';
    }
    if(c==STOP){
        shutdown();
    }
}

```

fasm

.Z80

```

PUBLIC EAY20x8_                                ;screen dump for Epson EA-Y -
series                                           ;20 x 8 liquid crystal display

```

```

EAY20x8_:
; the routine receives on the stack the address of the
; virtual screen in RAM. the automatic cursor advance
; feature of the display is then used, allowing a
; simple sequential writing of characters to the two
; controllers.
    PUSH    BC                                ;save BC
    LD      HL,4                               ;get screen address from stack
    ADD     HL,SP
    LD      C,(HL)
    INC     HL
    LD      B,(HL)
    LD      A,010H                             ;software reset
    CALL    .COM1
    LD      A,010H
    CALL    .COM2
    LD      A,0DH                               ;switch whole display ON
    CALL    .COM1
    LD      A,0DH
    CALL    .COM2
    LD      A,080H                             ;set cursor
    CALL    .COM1
    CALL    .DLIN1                             ;output contents of line 0
    LD      A,0COH                             ;set cursor
    CALL    .COM1
    CALL    .DLIN1                             ;output contents of line 1
    LD      A,080H                             ;set cursor
    CALL    .COM2
    CALL    .DLIN2                             ;output contents of line 2
    LD      A,0COH                             ;set cursor
    CALL    .COM2
    CALL    .DLIN2                             ;output contents of line 3
    LD      A,094H                             ;set cursor

```

```

CALL    .COM1
CALL    .DLINE1      ;output contents of line 4
LD      A,0D4H        ;set cursor
CALL    .COM1
CALL    .DLINE1      ;output contents of line 5
LD      A,094H        ;set cursor
CALL    .COM2
CALL    .DLINE2      ;output contents of line 6
LD      A,0D4H        ;set cursor
CALL    .COM2
CALL    .DLINE2      ;output contents of line 7
POP      BC          ;restore BC
LD      A,H
OR      L
RET

.DLINE1:
LD      D,20          ;counter for 20 characters
.DL1LP:
LD      A,(BC)        ;get character from screen
CALL    .DAT1         ;send to master
INC      BC
DEC      D
JR      NZ,.DL1LP     ;repeat for 20 characters
RET

.DLINE2:
LD      D,20          ;counter for 20 characters
.DL2LP:
LD      A,(BC)        ;get character from screen
CALL    .DAT2         ;send to slave
INC      BC
DEC      D
JR      NZ,.DL2LP     ;repeat for 20 characters
RET

.COM1:
                                ;commands to master
OUT      (02H),A
CALL    .DISWAIT
RET

.COM2:
                                ;commands to slave
OUT      (00H),A
CALL    .DISWAIT
RET

.DAT1:
                                ;data to master
OUT      (03H),A
CALL    .DISWAIT
RET

.DAT2:
                                ;data to slave
OUT      (01H),A
CALL    .DISWAIT
RET

.DISWAIT:
                                ;wait for master and slave
                                ;to be ready
IN      A,(02H)
BIT     7,A
JP      NZ,.DISWAIT
IN      A,(00H)
BIT     7,A
JP      NZ,.DISWAIT

```

```

    RET

    PUBLIC squirt_           ;sends a character at 2400 baud
                             ;to the printer
                             ;(data transfer is serial via pin
                             ;PA0 of the NSC810)

squirt_:
    DI                     ;ensure timing not corrupted
    LD     HL,2            ;get argument from stack
    ADD    HL,SP
    LD     E,(HL)          ;E contains character to be sent
    LD     D,7
    CALL   .SPACE          ;send start bit
.LOOP:
    SRL    E               ;put next data bit in carry
    CALL   C,.MARK         ;send as appropriate
    CALL   NC,.SPACE
    DEC    D
    JP     NZ,.LOOP        ;repeat until finished
    CALL   .MARK           ;send parity substitute
    CALL   .MARK           ;send stop bits
    CALL   .MARK
    EI                     ;restore interrupts
    LD     HL,1            ;comply with Aztec CII conventions
    LD     A,1
    OR     A
    RET

.MARK:
    EX     AF,AF';'
    LD     A,01H
    OUT    (04CH),A        ;NSC810 Port A bit set
    CALL   .TIME           ;to give total time of 512 t-states
                             ;(yielding 2400 baud @ 1.2 MHz)
    EX     AF,AF';'
    RET

.SPACE:
    EX     AF,AF';'
    LD     A,01H
    OUT    (048H),A        ;NSC810 Port A bit clear
    CALL   .TIME           ;to give total time of 512 t-states
                             ;(yielding 2400 baud @ 1.2 MHz)
    EX     AF,AF';'
    RET

.TIME:
    LD     A,28            ;takes 409 t-states to make
                             ;the total of 512
.TIME1:
    DEC    A
    JP     NZ,.TIME1
    RET

    PUBLIC pfeedon_        ;paperfeed on (requires previous
printon())
pfeedon_:
    LD     A,4
    OUT    (048H),A        ;clear PA2
    LD     H,A
    OR     L
    RET

```

```

;
    PUBLIC pfeedoff_      ;paperfeed off
pfeedoff_:
    LD      A,4
    OUT     (04CH),A      ;set PA2
    LD      H,A
    OR      L
    RET

;
;   the following timer routines are to drive the
;   National Semiconductor MM58174A real time clock
;   chip, mapped on I/O locations 10H to 1FH
;   PUBLIC newtime_      ;updates the string 'time[]' when
;                           ;invoked as 'newtime(time)'
newtime_:
    LD      HL,2
    ADD     HL,SP
    LD      E,(HL)        ;get start address of string into DE
    INC     HL
    LD      D,(HL)
    PUSH    BC
newtime3:
    LD      C,017H        ;get tens of hours
    CALL    newtime1
    LD      C,016H        ;get units of hours
    CALL    newtime1
    INC     DE
    LD      C,015H        ;get tens of minutes
    CALL    newtime1
    LD      C,014H        ;get units of minutes
    CALL    newtime1
    LD      A,0
    LD      (DE),A        ;set terminator
    POP     BC
    LD      A,H
    OR      L
    RET
newtime1:
    IN      A,(C)         ;get character
    AND     0FH           ;mask it
    ADD     A,'0'         ;and convert to ASCII
    CP     '?'
    JR      NZ,newtime2
    POP     AF
    JP      newtime3
newtime2:
    LD      (DE),A        ;place in string
    INC     DE
    RET

;
    PUBLIC RTctime_      ;loads the string 'time[]' into
;                           ;the real time clock when
;                           ;invoked as 'RTctime(time)'
RTctime_:
    LD      HL,2
    ADD     HL,SP
    LD      E,(HL)        ;get start address of string into DE

```

```

        INC     HL
        LD      D,(HL)
        PUSH    BC
        LD      A,0                ;reset seconds, stop clock
        OUT     (01EH),A
        LD      C,017H            ;set tens of hours
        CALL    RTctime1
        LD      C,016H            ;set units of hours
        CALL    RTctime1
        INC     DE
        LD      C,015H            ;set tens of minutes
        CALL    RTctime1
        LD      C,014H            ;set units of minutes
        CALL    RTctime1
        LD      A,1                ;reset seconds, start clock
        OUT     (01EH),A
        POP     BC
        LD      A,H
        OR      L
        RET

RTctime1:
        LD      A,(DE)            ;get from string
        SUB     '0'               ;and convert from ASCII
        OUT     (C),A            ;set character
        INC     DE
        RET

;
        PUBLIC newdate_          ;updates the string 'date[]' when
                                   ;invoked as 'newdate(date)'

newdate_:
        LD      HL,2
        ADD     HL,SP
        LD      E,(HL)           ;get start address of string into DE
        INC     HL
        LD      D,(HL)
        PUSH    BC

newdate3:
        LD      C,019H           ;get tens of days
        CALL    newdate1
        LD      C,018H           ;get units of days
        CALL    newdate1
        INC     DE
        LD      C,01CH           ;get tens of months
        CALL    newdate1
        LD      C,01BH           ;get units of months
        CALL    newdate1
        POP     BC
        LD      A,H
        OR      L
        RET

newdate1:
        IN      A,(C)            ;get character
        AND     0FH              ;mask it
        ADD     A,'0'            ;and convert to ASCII
        CP      '?'
        JR      NZ,newdate2
        POP     AF

```

```

        JP      newdate3
newdate2:
        LD      (DE),A      ;place in string
        INC     DE
        RET

;
        PUBLIC RTCdate_    ;loads the string 'date[]' into
                           ;the real time clock when
                           ;invoked as 'RTCdate(date)'
RTCdate_:
        LD      HL,2
        ADD     HL,SP
        LD      E,(HL)      ;get start address of string into DE
        INC     HL
        LD      D,(HL)
        PUSH    BC
        LD      C,019H      ;set tens of days
        CALL    RTCdate1
        LD      C,018H      ;set units of days
        CALL    RTCdate1
        INC     DE
        LD      C,01CH      ;set tens of months
        CALL    RTCdate1
        LD      C,01BH      ;set units of months
        CALL    RTCdate1
        POP     BC
        LD      A,H
        OR      L
        RET

RTCdate1:
        LD      A,(DE)      ;get character from string
        SUB     '0'         ;and convert from ASCII
        OUT     (C),A       ;set character
        INC     DE
        RET

;
        PUBLIC timeron_
timeron_:
        LD      A,9
        DI
        OUT     (01FH),A
        IN      A,(01FH)
        IN      A,(01FH)
        IN      A,(01FH)
        EI
        LD      A,H
        OR      L
        RET

        PUBLIC timeroff_
timeroff_:
        LD      A,0
        DI
        OUT     (01FH),A
        IN      A,(01FH)
        IN      A,(01FH)
        IN      A,(01FH)

```

```

        EI
        LD      A,H
        OR      L
        RET

        PUBLIC shut_down_      ;switch off power, kill micro:
shut_down_:
        DI          ;disable interrupts
        LD      A,010H      ;set PA4
        OUT     (048H),A
        HALT

;
        PUBLIC key_           ;returns 1 if a key is pressed:
key_:
        IN      A,(040H)
        AND     020H          ;test PA5 of NSC810
        LD      H,0
        LD      L,A
        OR      H
        RET

;
;   the routines rxchar() and txchar() are designed
;   to drive an RCA CDP1854A UART mapped on I/O
;   locations 04H (data) and 05H (control/status)
;   PUBLIC rxchar_           ;get character from RS232 port
rxchar_:
        LD      HL,0
        IN      A,(05H)
        BIT     0,A           ;check for received character
        JP      NZ,rx1
        RET                  ;if none return EOF

rx1:
        IN      A,(04H)      ;else get character
        RES     7,A          ;clear msb
        LD      L,A
        OR      H
        RET                  ;and return it

;
        PUBLIC txchar_       ;send character via RS232 port
txchar_:
        IN      A,(05H)      ;wait for tbre
        RLA
        JR      NC,txchar_
        LD      HL,2          ;get argument
        ADD     HL,SP
        LD      A,(HL)
        OUT     (04H),A      ;and send it
        LD      A,H
        OR      L
        RET

;
        PUBLIC HX20_         ;returns 1 if the HX20 is connected
                                ;to the RS232 port
HX20_:
        IN      A,(041H)
        CPL
        AND     1

```

```

        LD      L,A
        LD      H,0
        OR      H
        RET

        PUBLIC newkey_
newkey_:
        IN      A,(041H)      ;get data from NSC810 PB4-PB7
        SRL     A
        SRL     A
        SRL     A
        SRL     A
        LD      L,A
        LD      H,0
        LD      DE,.new1
        ADD     HL,DE
        LD      L,(HL)      ;get character from table
        LD      H,0
        LD      A,L
        OR      H
        RET

.8080
fendasm

```

```

/*****
/*
/*      system software: foreground routines      */
/*      -----      */
/*      (these routines are interrupt-driven)      */
/*      */
*****/

                INTR      from patient button
                RSTC      from keypad
                RSTA      from real-time clock
                RSTB      from power-down signal      */

INTR( )
{
    ctr_dems++;
    buzflash(50);
}

RSTB( )
{
    powerlow=1;
    /* printer off: */
    out(0x4c,0x08);
    out(0x48,0x05);
    stoppump( );
    flg_events|=0x18;
    if(running && (!finish)){
        rp=&data[recptr];
        rp->minute=minute-1;
        rp->dems=ctr_dems;
        rp->given=ctr_fixgave;
        rp->events=flg_events;
        recptr++;
        fp->duration=minute;
        fp->lastrec=recptr-1;
    }
    clearscr( );
    print(0,"  BATTERY FLAT!!  ");
    print(1,"  demand analgesia  ");
    print(2,"    terminated  ");
    print(4,"    press STOP to  ");
    print(5,"  shut down and then ");
    print(6,"RECHARGE FROM MAINS!");
    dumpscr( );
    while(newkey( )!=STOP){
        buzflash(500);
        wait(500);
    }
    shutdown( );
}

RSTA( )
/* real-time clock interrupt:
- 197 -
*/

```

```

{
    char c;
    int i;
    unsigned n,n1;
    n=total+ctr_fixgave;          /* scale and convert to ml      */
    n=n*75;
    n1=n/29;                      /* note careful order of
                                   evaluation to achieve
                                   required numerical
                                   precision      */

    cumvol[6]=EOF;
    for(i=5;i>=0;i--){
        cumvol[i]=n1%10+'0';
        n1/=10;
    }
    for(i=1;i<4;i++){
        cumvol[i-1]=cumvol[i];
    }
    cumvol[3]='.';
    if(cumvol[0]=='0'){
        cumvol[0]=' ';
        if(cumvol[1]=='0')
            cumvol[1]=' ';
    }
    if(running && !stopped){
        newtime(time);
        if(time[2]==' '){
            time[2]=': ';
        }
        else
            time[2]=' ';
        strcpy(screen[0]+15,time);
        strcpy(screen[6]+10,cumvol);
        dumpscr();
    }
    looptime++;                  /* increment counter for sample int. */
    if(!stopped){
        if(pumpctr>0){           /* if pump was set running...      */
            if(in(0x41)&0x04){    /* ...and still is,                */
                if(pumpctr>10){  /* then if it has overrun,        */
                    stoppump(); /* stop it                          */
                    stopped=1; /* set flag                          */
                    pumpctr=0; /* reset counter                    */
                    empty=1;
                    flg_events|=0x80; /* set bit 7                      */
                    buzflash(200); /* let there be light              */
                    clearscr();
                    print(1,"please check");
                    print(2,"syringe driver");
                    print(4,"press STOP to");
                    print(5,"silence bleeper");
                    print(6,"then");
                    print(7,"resume with START");
                    dumpscr();
                }
            }
            else
                pumpctr++;        /* otherwise...                    */
        }
        else{
            /* ...but no longer is, */

```

```

        pumpctr=0;                /* then reset counter */
        ++ctr_fixgave;            /* mod PL 19.x.1984 */
    }
}
else( /* if pump was not set running, */
    if(in(0x41)&0x04){ /* but nevertheless is, */
        /***** stoppump(); *****/
        /***** panic(); *****/
    }
    if(ctr_duefix>0){ /* but further delivery is due, */
        startfix(); /* start pump */
        --ctr_duefix; /* and adjust counters */
        pumpctr=1; /* accordingly */
        /* "++ctr_fixgave" moved to new
place
                                in code by PL 19.x.1984 */
    }
}
if(key()){ /* if a key is pressed */
    if(newkey()==STOP){ /* and it is 'STOP' */
        if(stopctr>8){ /* and has been held */
            /* for long enough */
            buzz(100);
            stoppump();
            flg_events|=0x40; /* set bit 6 */
            clearscr();
            print(0,"drug delivery");
            print(1,"halted on request");
            print(2,"at");
            print(4,"resume with START");
            print(6,"or hold DELETE to");
            print(7,"stop altogether");
            time[2]=': ';
            strcpy(screen[2]+4,time);
            dumpscr();
            kbpnt=0;
            stopctr=0;
            stopped=1;
        }
        else
            stopctr++; /* otherwise increment
                        counter */
    }
}
else
    stopctr=0;
}
else if(! finish){ /* if stopped but not finished */
    if(empty){
        buzflash(100);
    }
    while((c=getc())!=EOF){ /* NB: this way the foreground
                            routine checks for a keypress
                            each time it is entered, and
thus
                                waits for input without robbing
                                CPU time from the background

```

```

                                task!                                */
if(c==START){
    enabpump( );
    kbpnt=0;
    ctr_duefix=0;
    stopped=0;
    empty=0;
    flg_events|=0x20; /* set bit 5 */
    rundispl( );
}
else if(empty && (c==STOP)){
    empty=0;
    flg_events|=0x40;
    clearscr( );
    print(2,"refill the syringe  ");
    print(3,"and then press START ");
    dumpscr( );
}
}
if(key( )){
    if(newkey( )==DELETE){
        if(stopctr>12){
            flg_events|=0x10; /* set bit 4 */
            finish=1;
            kbpnt=0;
            buzz(100);
            clearscr( );
            print(0,"demand analgesia");
            print(1,"now terminated");
            print(3,"please wait up to");
            print(4,"one minute for");
            print(5,"PRODAC to complete");
            dumpscr( );
            stopctr=0;
        }
        else
            stopctr++;
    }
    else
        stopctr=0;
}
else
    stopctr=0;
}
}

RSTC( ) /* interrupt routine which reads the
        value of a new key from the NSC810,
        converts it to the correct ASCII
        code and places it in the keyboard
        buffer. If the key was pressed during
        dialogue, audio feedback is given.
{
    char c;
    c=newkey( );
    if(c){
        if ( ! running || (stopped&&(c==START)))
            - 200 -
    }
}

```

```
        buzz(100);
    if ( kbpnt < 10)
        kbq[kbpnt++] = c;
    }
}

/* end of PRODAC software */
```

APPENDIX D
SOFTWARE FOR 2ND ORDER FIT

.

```

/*****
/*
/*      Program to find 2nd-order fit for 4th-order pharmacokinetics
/*
/*
/*
/*****

#include stdio
#include math

double T1, T2, T3, K1, K2, K3, Tr;
int n;          /*      no. of independent variables      */
int ibound;     /*      set to zero, specify all bounds      */
double bl[4];   /*      lower bounds on x[]      */
double bu[4];   /*      upper bounds on x[]      */
double x[4];    /*      contains initial guess on entry to NAG routine,
               position of minimum on exit      */
double f;       /*      contains minimum value on exit      */
int iw[6];
int liw;        /*      equal to size of workspace iw[]      */
double w[62];
int lw;         /*      equal to length of w[]      */
int ifail;

main()
{
    n=3;
    ibound=0;
    liw=6;
    lw=62;

    printf("enter K1 K2 K3\n");
    if (scanf("%f%f%f",&K1,&K2,&K3)!=3)
        error(1);
    printf("enter T1 T2 T3\n");
    if (scanf("%f%f%f",&T1,&T2,&T3)!=3)
        error(1);

    for(;;)go();
}

go()
{
    printf("enter Tr: ");
    if (scanf("%f",&Tr)!=1)
        error(1);

    x[0]=(Tr+T2)/2.0;
    x[1]=(T2+T3)/2.0;
    x[2]=(K1+K2+K3)/3.0;

    bl[0]=0.00001;
    bu[0]=100.0;

    bl[1]=2.0;
    bu[1]=500.0;

```

```

    bl[2]=0.00000001;
    bu[2]=10000000;

    do{
        ifail=1;
        e04jaf(&n,&ibound,bl,bu,x,&f,iw,&liw,w,&lw,&ifail);
    }while(ifail==2);

    printf("\nminimum of %f at\nl_alpha=%f, t_beta=%f, k=%f\n"
           ,f,x[0],x[1],x[2]);
    f=x[0]*x[1]*log(x[1]/x[0])/(x[1]-x[0]);
    printf("giving peak effect at t=%f\n\n",f);
}

FUNCT1(n,xc,fc)
int *n;          /*      no. of variables, do not alter      */
double xc[];     /*      real array, contains point for          */
                /*      function evaluation                      */
double *fc;      /*      place value here before exit           */
{
    double alpha, beta, a[10], b[10], v, t1, t2, k1k2, temp1, temp2;
    int nl, j, jl, ierr, i;

/*      here evaluate the coefficients of the polynomials A(s) and B(s) */

    nl=6;
    t1=xc[0];
    t2=xc[1];
    k1k2=xc[2];

    temp1=(T2*T3*K1+T1*T3*K2+T1*T2*K3);
    temp2=(K1*(T2+T3)+K2*(T1+T3)+K3*(T1+T2));

    b[1]=0.0; /*      s**5      */
    b[2]=t1*t2*temp1-k1k2*T1*T2*T3*Tr; /*      s**4      */
    b[3]=temp1*(t1+t2)+temp2*t1*t2
          -k1k2*(T1*T2*T3+T1*T2*Tr+T2*T3*Tr+T1*T3*Tr); /*      s**3      */
    b[4]=t1*t2*(K1+K2+K3)+(t1+t2)*temp2
          +temp1-k1k2*(T1*T2+T1*T3+T1*Tr+T2*T3+T2*Tr+T3*Tr); /*      s**2      */
    b[5]=(t1+t2)*(K1+K2+K3)+temp2-k1k2*(T1+T2+T3+Tr); /*      s**1      */
    b[6]=K1+K2+K3-k1k2; /*      s**0      */

    a[1]=T1*T2*T3*t1*t2*Tr; /*      s**6      */
    a[2]=T1*T2*T3*t1*(t2+Tr)+T1*T2*t2*Tr*(T3+t1)+T3*t1*t2*Tr*(T1+T2); /*      s**5      */
    a[3]=T1*T2*T3*(t1+t2+Tr)+(T1*T2+T1*T3+T2*T3)*(t1*t2+t1*Tr+t2*Tr)
          +t1*t2*Tr*(T1+T2+T3); /*      s**4      */
    a[4]=T1*T2*(T3+t1+t2+Tr)+T3*t1*(T1+T2+t2+Tr)+t2*Tr*(T1+T2+T3+t1)
          +(T1+T2)*(T3+t1)*(t2+Tr); /*      s**3      */
    a[5]=T1*(T2+T3+t1+t2+Tr)+T2*(T3+t1+t2+Tr)+T3*(t1+t2+Tr)
          +t1*(t2+Tr)+t2*Tr; /*      s**2      */
    a[6]=T1+T2+T3+t1+t2+Tr; /*      s**1      */
    a[7]=1.0;

```

```

/*      and then calculate the integral of

          1/(2*i*pi) * B(s)*B(-s)/(A(s)*A(-s))

a la Astrom (1970) ch.5 "evaluation of loss functions":      */

ierr=0;
v=0.0;
if(a[1]>0.0){
    for(j=1;j<=n1;j++){
        if(a[j+1]>0.0){
            alpha=a[j]/a[j+1];
            beta=b[j]/a[j+1];
            v+=beta*beta/alpha;
            j1=j+2;
            if(j1-n1<=0)
                for(i=j1;i<=n1;i+=2){
                    a[i]-=alpha*a[i+1];
                    b[i]-=beta*a[i+1];
                }
        }
        else
            error(2);
    }
    v/=2.0;
    *fc=v;
    return;
}
else
    error(3);
}

error(n)
int n;
{
    printf("\007\007\nerror %d\n",n);
    exit();
}

```

APPENDIX E
BIBLIOGRAPHY

Adair,E.R., J.C.Stevens and L.E.Marks (1968), Thermally induced pain, the dol scale, and the psychophysical power law, American Journal of Psychology, vol LXXXI no. 2, pp.147-164.

Arnsperger,J.M., B.C.McInnis, J.R.Glover et al. (1983),Adaptive Control of Blood Pressure, IEEE Trans. on Biomedical Engineering, BME-30, pp.168-176

Astrom,K.J. (1970), Introduction to Stochastic Control Theory, Academic Press

Auer,L.M. and H.Rodler (1981), Microprocessor control of drug infusion for automatic blood pressure control, Med. and Biol. Eng. and Comput., 19, pp.171-174

Barrier,G. and M.Harmer (1985),Patient-controlled Analgesia in Obstetrics, Patient Controlled Analgesia, Blackwell Scientific Publications, pp.170-179

Bennett,R.L. (1985a), Patient-controlled Analgesia for the control of postoperative pain: Experience at the University of Kentucky, Patient Controlled Analgesia, Blackwell Scientific Publications, pp.149-154

Bennett,R.L. (1985b), Patient-controlled Analgesia for the treatment of the pain of terminal cancer: Description of an initial clinical experience, Patient Controlled Analgesia, Blackwell Scientific Publications, pp.160-169

Berger,C.S. and W.A.Brown (1984), Adaptive Control of Muscle Relaxation, Proc. IFAC/IFIP Symposium on Real Time Digital Control Applications, Guadalajara, Mexico, pp.109-119

Chapman,C.R. (1985), Psychological factors in Postoperative Pain and its Treatment, Acute Pain, Butterworths, pp.22-41

Collins,S.M. and R.C.Arzbaeher (1979), Computer Control of Cardiac Arrhythmia, Frontiers of engineering in health care: Proc. 1st Annual IEEE Conference on Engineering in Medicine and Biology, Denver, pp.276-279

Cross,D.V., B.Tursky and M.Lodge (1975), The role of regression and range effects in determination of the power function for electric shock, Perception and Psychophysics, vol 18(1), pp.9-14

Dahlstrom,B., A.Tamsen, L.Paalzow and P.Hartvig (1982), Patient-controlled Analgesic Therapy Part IV, Clinical Pharmacokinetics 7, pp.266-279

Evans,J.M., M.Rosen, J.MacCarthy and M.I.J.Hogg (1976), Apparatus for patient-controlled administration of intravenous narcotics during labour, The Lancet, 3rd Jan. 1976, pp.17-18.

Garrett,E.R., A.J.Agren and H.J.Lambert (1967), Pharmacokinetic Analysis of Receptor Site Models in Multicompartmental Systems, Int. J. Clin. Pharm., 1/67, pp.1-14

Godfrey,K.R. (1983), Compartmental Models and their Application, Academic Press

Grafox Ltd. (1985), Logistix, 65 Banbury Rd Oxford

Graves,D.A., T.J.Baumann, R.L.Bennett, R.L.Batenhorst and T.S.Foster (1983), Patient-controlled Analgesia, Annals of Internal Medicine 99, pp.360-366

Holford,N.H.G. and L.B.Sheiner (1981), Understanding the Dose-Effect Relationship: Clinical Application of Pharmacokinetic-Pharmacodynamic Models, Clinical Pharmaceutics, 6, pp.429-453

Horowitz,P. and W.Hill (1980), The Art of Electronics, Cambridge University Press

Hull,C.J. (1979), Pharmacokinetics and Pharmacodynamics, Br. J. Anaesth., 51, pp.579-594

Hull,C.J. and A.Sibbald (1981), Control of postoperative pain by interactive demand analgesia, Br. J. Anaesth., vol 53, pp.385-391.

Jacobs,O.L.R., R.E.S.Bullingham, P.Lammer, H.J.McQuay, G.O'Sullivan and M.P.Reasbeck (1983), On-line estimation in the Relief of Postoperative Pain, OUEL Report no. 1452/83.

Jacobs,O.L.R., R.E.S.Bullingham, P.Lammer, H.J.McQuay, G.O'Sullivan and M.P.Reasbeck (1985), Modelling, Estimation and Control in the Relief of Postoperative Pain, Automatica, vol. 21 no. 4, pp.349-360

Jacobs,O.L.R. and P.Lammer (1986), Modelling Postoperative Pain and its Relief in Demand Analgesia, Biomedical Measurement Informatics and Control, vol.1 no.2

Keeri-Szanto,M. (1979), Drugs or drums: what relieves postoperative pain?, Pain, vol 6, pp.217-230

Keeri-Szanto, M. and S. Heaman (1972), Postoperative demand analgesia, Surg. Gynaecol. Obstet., 134, pp.647-651

Kernighan, B.W. and D.M. Ritchie (1978), The C Programming Language, Prentice-Hall

Lammer, P., R.E.S. Bullingham and O.L.R. Jacobs (1985), PRODAC demand analgesia computer, Patient Controlled Analgesia, Blackwell Scientific Publications Ltd, pp.102-108

Lehmann, K.A. (1984), On-Demand Analgesie: Neue Möglichkeiten zur Behandlung akuter Schmerzen, Arzneim.-Forsch./Drug Res. 34(II) Nr.9a, pp.1109-1114

Lehmann, K.A. (1985a), Discussion to lecture by C.J. Hull, Patient Controlled Analgesia, Blackwell Scientific Publications Ltd, pp.18-29

Lehmann, K.A. (1985b), Practical Experience with Demand Analgesia for Postoperative Pain, Patient Controlled Analgesia, Blackwell Scientific Publications Ltd, pp.134-139

Linkens, D.A., A.J. Asbury, S.J. Rimmer and M. Menad (1982), Self-tuning control of Muscle Relaxation during Anaesthesia, IEEE Conference on applications of adaptive and multivariable control, Hull, pp.96-102

Mackintosh,N. (1974), Psychology of Animal Learning, Academic Press

Mather,L.E., M.J.Lindop, G.T.Tucker and A.E.Pflug (1975), Pethidine revisited: plasma concentrations and effects after intramuscular injections, Br. J. Anaesth., 47, 1269-1275.

McQuay,H.J., R.A.Moore, P.Poppleton, J.Uppington, R.E.S.Bullingham and P.Lammer (1985), Postoperative single-dose comparison of intramuscular meptazinol, morphine and pethidine, compared in the same patients with intravenous meptazinol, morphine and pethidine given by demand analgesia, (awaiting submission).

Melzack,R. (1977), The problem of Pain, Penguin Books.

Paalzow,L.K. (1982), Pharmacokinetic Aspects of Optimal Pain Treatment, Acta anaesth. scand., Suppl. 74, pp.37-43

Rametti,L.B. (1985), On-line Control of D-Tubocurarine Induced Muscle Relaxation, Ph.D. thesis, University of Capetown

Reasbeck,M.P. (1982), Modelling and Control of Postoperative Pain, D.Phil thesis, University of Oxford.

Schwartz,B.J. (1984), Psychology of Learning and Behaviour, New York & Co.

Sechzer,P.H. (1968/1970), Objective measurement of pain Proceedings of 4th world congress of anaesthesiologists Sep. 1968, pub. Excerpta Medica, 1970.

Slattery,P.J., M.Harmer, M.Rosen and M.D.Vickers (1983), An open comparison between routine and self-administered postoperative pain relief, Annals of the Royal College of Surgeons, 65, pp.18-19.

Stanski,D.R., D.J.Greenblatt and E.Lowenstein (1978), Kinetics of Intravenous and Intramuscular Morphine, Clin. Pharm. Therap., 24, pp.52-59

Sternbach,R.A. and B.Tursky (1974), On the psychophysical power function in electric shock, Psychonomic Science, vol 1, pp.217-218.

Stevens,S.S. (1957), On the psychophysical power law, The Psychological Review, vol 4, pp.153-181.

Stevens,S.S. and H.B.Greenbaum (1966), Regression effect in psychophysical judgement, Perception and Psychophysics, vol 1, pp.439-446.

Tamsen,A., P.Hartvig, B.Dahlstrom, B.Lindstrom and M.Holmdahl (1979), Patient-controlled analgesic therapy in the early postoperative period, Acta. anaesth. scand., vol 23, pp.462-470

Tamsen,A., P.Hartvig, C.Fagerlund and B.Dahlstrom (1982), Patient-controlled Analgesic Therapy Parts I & II, Clinical Pharmacokinetics 7, pp.149-175

Wagner,J.G. (1968), Kinetics of Pharmacological Response, J.Theoret.Biol 20, pp.173-201

Wagner,J.G. (1971), Relations between Drug Concentration and Response, J. Mond. Pharm., 4, 14, pp.279-310

Wallenstein,S.L. and R.W.Houde (1975), The Clinical Evaluation of Analgesic Effectiveness, Methods in Narcotics Research, ed. S.Ehrenpreis and A.Neidle, Marcel Dekker (New York), pp.127-145

.