MEASUREMENT AND PREDICTION OF SURFACE ROUGHNESS IN FINISH TURNING

BY

DINGHONG YAN

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE

DEPARTMENT OF MECHANICAL AND INDUSTRIAL ENGINEERING

THE UNIVERSITY OF MANITOBA

WINNIPEG, MANITOBA, CANADA

ISBN 0-315-99009-0

Name JINGHONG YAN

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

0548 Mechanical Engineering

SUBJECT TERM

0548

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523
Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES

American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences

Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

MEASUREMENT AND PREDICTION OF SURFACE ROUGHNESS IN FINISH TURNING

BY

DINGHONG YAN

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

ACKNOWLEDGMENTS

First of all, I would like to express my heartfelt appreciation and gratitude to my advisors, Dr. N. Popplewell and Dr. S. Balakrishnan. They introduced me into this challenging area of research and have been most and very supportive throughout this study. Their responsible attitude in academic research has been and will always be a good example in my professional life. I wish also to thank Dr. E. Shwedyk in the Department of Electrical and Computer Engineering for his informative comments and substantive help during this study.

My indebtedness is also towards my friend and colleague, Dr. J. E. Kaye for his work in building the interface board for the CCD measuring system and many hours of discussion on tool wear monitoring. Many thanks are due to Mr. M. Cheng for his contributions to the software integration of neural networks.

Finally, I owe a very large debt of gratitude to my wife, Jingmin and children, Qian and Nan. Without their moral support and patience in the past four years, I would not have finished this work.

The author was supported financially by the Natural Sciences and Engineering Research Council of Canada through his advisors' research grants and the University of Manitoba Graduate Fellowship.

ABSTRACT

Surface finish produced during machining is an important aspect of modern manufacturing. It determines the fit and reliable functioning of mating parts. The on-line, real-time assessment (measurement or prediction) of surface roughness has attracted many efforts in the research community for about 30 years. However, little success has been achieved. This challenging topic is examined in this study. Three new approaches are proposed to provide a framework which allows a practical realization of an on-line, real-time assessment of the surface roughness produced in finish turning.

The first approach permits the surface roughness to be measured by solving the inverse problem of light scattering. A laser measuring system is developed to capture the Angular Scattered Light Pattern (ASLP) reflected from a turned surface. The maximum peak-to-valley roughness, R_{max}, is estimated using a linear, least-square algorithm together with an iterative search to determine a correction factor for the measured light. The effects of the ambient room light as well as the workpiece's rotational speed are investigated and removed. A 14% maximum measurement error is obtained between the averaged optical and stylus-measured R_{max}.

Further efforts involve real-time measurements using the same laser measuring system. However, instead of directly solving the inverse problem of light scattering, a three-layer neural network which incorporates a back-propagation learning algorithm is constructed to recognize a measured ASLP pattern and, hence, the corresponding R_{max}. The sampling of the ASLP data and the determination of R_{max} are combined into an

integrated software module. This approach is shown to be robust and quick and gives an average error of less than 10% in a real cutting environment.

The final approach represents an alternative strategy of developing a comprehensive model for predicting R_{max} by utilizing the nominal cutting variables and the cutting tool's geometry. The effect of a cutting tool's progressive flank wear as well as the relative vibration between the tool and workpiece are considered. The change in a tool's nose radius due to flank wear is estimated on-line and incorporated in the model. All the predicted R_{max} are within the given bound of conventional stylus measurements under different cutting conditions and a maximum error of 13% is obtained in comparison with the average stylus data. The model provides a straightforward relationship between the surface roughness and cutting variables which offers the potential for implementing an adaptive control of surface finish in finish turning.

LIST OF FIGURES

		Page
Fig. 1.1	Surface roughness-a link between the turning process and performance of a machined part.	2
Fig. 2.1	Typical texture of a machined surface.	8
Fig. 2.2	A section of a surface profile depicting vertical measures in the mean-line system.	9
Fig. 2.3	Diagram of a conventional turning operation.	11
Fig. 2.4	Roughness profile for V=119.8 m/min, d_o=0.5 mm, workpiece's material: SPS 4340 steel, HRc=36, Carbide tool: 7,7,7,7,75,15,0.40, (a) f_o=0.061 mm/rev. (b) f_o=0.122 mm/rev.	13
Fig. 2.5	A magnified view of the tool bit showing flank and crater wear.	14
Fig. 3.1	Block diagram of a stylus instrument.	18
Fig. 3.2	Cutaway view of the fringe-field capacitive sensor on a one-dimensional surface.	20
Fig. 3.3	Schematic of ellipsometry in which the detector and analyzer are usually positioned in the specular direction.	27
Fig. 4.1	Profile height distribution of the finish-turned surface shown in Fig.2.4(a).	37
Fig. 4.2	Light scattered from a rough surface.	38
Fig. 4.3	Variations of the correction factor, K, with the estimated R_{max}.	41
Fig. 4.4	Flow chart of the algorithm for the estimation of R_{max}.	44
Fig. 4.5	Variation of the correction factor, K, and computational error, ε, with the estimated R_{max} for Section 5 of Specimen 1.	45

Fig. 4.6 Variation of correction factor, K, and computational error, ε, with the estimated R_{max} for Section 3 of Specimen 1. 45
Fig. 4.7 Diagram of the laser measuring system. 49
Fig. 4.8 Overview of the experimental set-up. 50
Fig. 4.9 Structure of MOS capacitor. 51
Fig. 4.10 Functional diagram of CCD sensor. 52
Fig. 4.11 Electronic interface. 54
Fig. 4.12 Block diagram of electronic interface. 55
Fig 4.13 A/D electronic circuitry for CCD. 56
Fig. 4.14 Output inversion and level adjustment circuitry for CCD. 57
Fig. 4.15 Transport clock synchronization, lockout circuit and timing diagram. 58
Fig. 4.16 Influence of a cutting fluid on the measured ASLP. 60
Fig. 4.17 Strategy to compensate for the effect of the workpiece's rotational speed. 63
Fig. 4.18 Sampled ASLP from Section 5 of Specimen 1 triggered by:
(a) the index pulse but discarded at 660, 1,156 and 2,320 rpm; and
(b) the software READ and recorded at 660, 1,156 and 2,320 rpm. 64
Fig. 4.19 Comparison of the sampled ASLP from six finish-turned surfaces having different R_{max}. 65
Fig. 4.20 Correlation between the optically-determined and stylus-measured R_{max}. 66
Fig. 4.21 Change in the tool's nose radius, observed after a 2.15 min. cutting period (V=144.0 m/min, f_o=0.0601 mm/rev., d_o=0.38 mm). 68
Fig. 4.22 A point-to-point comparison between the optically-determined and stylus-measured R_{max} for cutting test 5. 69
Fig. 5.1 A perceptron which forms a weighted sum of N_l inputs and passes the result through a nonlinearity. 75

Fig. 5.2 A typical multi-layered neural network. 77

Fig. 5.3 The back-propagation algorithm learns by changing its weights to follow the steepest path toward the bottom of a bowl-shaped error surface [H93b]. 79

Fig. 5.4 A three-layered neural network constructed for the measurement of R_{max}. 82

Fig. 5.5 Variation of CPU and data preparation times with the number of training patterns. 84

Fig. 5.6 (a) Raw ASLP data, and (b) the compensated ASLP data input to the neural network. 86

Fig. 5.7 The integrated software package in (a) the training phase, and (b) the testing phase. 88

Fig. 5.8 The R_{max} predicted by the neural network on the specimens used in training. 90

Fig. 5.9 The R_{max} predicted by the neural network on the unknown specimens. 91

Fig. 5.10 The results of the real-time measurements from ten cutting tests. 93

Fig. 6.1 Geometrical details needed to calculate $\Delta\rho(t)$. 97

Fig. 6.2 Workpiece-cutting tool idealization. 99

Fig. 6.3 Variation of uncut chip area, A(t), during turning. 102

Fig. 6.4 Flow chart for predicting R_{max} . 104

Fig. 6.5 Variations of the unit cutting force, K_s, with feedrate, cutting speed, and hardness of the workpiece's material. 109

Fig. 6.6 Comparison between the measured and predicted tangential cutting forces for a steel workpiece having HRc=36. 110

Fig. 6.7 Comparison between the measured and predicted tangential cutting forces for a steel workpiece having HRc=22. 110

Fig. 6.8 Temporal changes in the spindle speed drop, the measured and estimated flank wear for Test 1. 111

Fig. 6.9 Comparison between the measured and predicted R_{max} for Test 1. 113

Fig. 6.10 Comparison between the measured and predicted R_{max} for Tests 2 through 6. 115

Fig. 6.11 Comparison between the measured and predicted R_{max} for Tests 27 through 31. 115

Fig. A1 The maximum peak-to-valley roughness, R_{max}. 124

Fig. A2 Ten-point average peak-to-valley roughness, R_z. 125

Fig. A3 Amplitude probability density function of a surface profile. 125

Fig. A4 Crest measures. 126

Fig. A5 Envelope measures. 127

Fig. A6 A random surface profile. 128

Fig. A7 Different surface profiles, $f(x)$, and their autocorrelation, $R(\beta)$ for (a) sine wave; (b) line; (c) wide-band random noise; and (d) mixed profile. 130

Fig. C1 Geometrical details needed to determine the RMS slope of a surface profile. 192

Fig. D1 Evaluation board for linear CCD imagers. 197

Fig. D2 Input and output voltage waveforms for the operation of the CCD evaluation board . 198

Fig. D3 Schematic of the CCD evaluation board. 199

Fig. D4 Logic timing waveforms for the operation of the CCD evaluation board. 200

Fig. D5 External time exposure circuit of the CCD evaluation board. 201

Fig. E1 Calibration curve of the dynamometer in the radial direction. 203

Fig. E2 Calibration curve of the dynamometer in the tangential direction. 204

Fig. E3 Calibration curve of the dynamometer in the feed direction. 204

LIST OF TABLES

		Page
Table 3.1	Overview of the methods used to describe light scattering [BT85].	24
Table 3.2	Summary of surface roughness measurement techniques.	29
Table 4.1	Stylus-measured surface roughness of specimens.	46
Table 4.2	Effect of a cutting fluid on the estimation of R_{max}.	60
Table 4.3	Effect of ambient light level on the estimation of R_{max}.	61
Table 4.4	Cutting conditions used for on-line measurements.	67
Table 4.5	Comparison between stylus and optical data for on-line measurements.	70
Table 5.1	Conditions used for real-time measurements.	92
Table 6.1	Cutting conditions used for on-line predictions.	106
Table 6.2	Mass, damping ratio and spring constant values.	107
Table E1	Crosstalk between the measured cutting force components.	202
Table E2	Natural frequencies.	203

NOMENCLATURE

ASLP	Angular Scattered Light Pattern
A(t)	uncut chip area at time t (mm^2)
a(t),b(t)	thickness and width of uncut chip at time t (mm)
BP	back propagation
CCD	Charge-Coupled Device
C_p	proportional constant for determining force increment due to tool flank wear
D_1	delay time between the completion of the ASLP triggered by encoder pulse and READ pulse (μs)
$D(t_i)$	diameter of the workpiece at instant t_i, (mm)
D_x	variance of the surface profile
d_0	nominal depth of cut (mm)
$d_q(x_p)$	target value of the qth output node for the pth training pattern
d(t)	depth of cut at time t (mm)
f_0	nominal feed rate (mm/rev)
F(t)	dynamic cutting force (N)
f(t)	feed rate at time t (mm/rev)
$F_p(t),F_d(t)$	component of F(t) due to variation of chip area and damping (N)
$F_f(t)$	component of F(t) due to rubbing effect between the tool's flank and machined surface (N)
G_E	surface roughness within the sampling length in envelope measures (μm)

G_M	surface roughness within the sampling length in crest measures (μm)
H	hardness of workpiece (found on Rockwell Scale C)
h_{f0}	tool's initial flank wear (mm)
$h_f(t)$	tool's time varying flank wear at time t (mm)
$h_f(t_i)$, $h_f(t_{i-1})$	tool flank wear estimated at instant t_i and t_{i-1}, respectively (mm)
$I(\theta_1,\theta_2)$	intensity of reflected light
$I_m(\theta_1,\theta_2,t)$	measured intensity of reflected light at time t
I_s	normalized specular intensity
J	moment of inertia of chuck (kg m^2)
j	representative location of a CCD pixel
$\bar{j}$	CCD pixel location corresponding to the mean light level
K	correction factor for the measured light intensity
k	number of iterations in training
K_a, K_b	starting and ending point, respectively, within the search range for K
K_c, K_d	two points within (K_a, K_b), determined by the golden section search method
K_f	proportional constant for determining the rubbing force
K_n	constant depending on the lathe's setting
K_s	unit cutting force of workpiece material (N/mm^2)
K_δ	proportional constant for determining the contact length during penetration
L	sampling length of stylus measurement (mm)
L_o	sampling length associated with the laser spot on the workpiece's surface
L_w	cutting length of one cut (mm)
M	number of layers

MOS	metal-oxide-semiconductor
M_t, C_t, K_t	mass, damping and spring constant of tool
M_w, C_w, K_w	mass, damping and spring constant of workpiece
N	number of pixels of CCD array
N_c	number of predictions with one cut
N_l	number of nodes in layer *l*
$n(t_i)$	spindle speed at instant t_i (rpm)
$o_{l,j}$	output of node j in layer *l*
OS	output signal of CCD array
P	number of training patterns
p_j	density function of the light intensity at the *j*th pixel
Q(w)	sum of squared error
$Q_p(w)$	total squared error for the *p*th training pattern
R	total depth within the sampling length (µm)
r	ratio between the radial and tangential force
R_a	arithmetically averaged roughness within the sampling length (µm)
RCK	reset clock
R_{max}	maximum peak-to-valley roughness within the sampling length (µm)
$R_{max}(t)$	maximum peak-to-valley roughness within the sampling length at time t (µm)
R_o	optical roughness of the surface
R_p	leveling depth within the sampling length (µm)
R_q	RMS roughness within the sampling length (µm)
R_{th}	theoretical component of $R_{max}(t)$, (µm)
R_v	complementary depth within the sampling length (µm)

$R_{wr(t)}$	component of $R_{max}(t)$ due to tool wear at time t, (μm)
R_z	ten point average peak-to-valley roughness within the sampling length (μm)
$R(\beta)$	autocorrelation function of the surface profile
S_c	standard deviation of the ASLP calculated from an estimated R_{max}
S_m	standard deviation of the measured ASLP
$S(\omega)$	power spectrum of the surface profile
T	correlation distance of the surface, defined as the distance in which the correlation coefficient, $C(\tau)$, equals e^{-1}
TCK	transport clock
T_m	cutting time of a section on the specimen (min)
T_r	time required for one revolution of the workpiece (s)
u	RMS slope of the surface within the sampling length
u(t)	relative displacement between the workpiece and cutting tool at time t (m)
$\dot{u}(t)$	relative velocity between the workpiece and cutting tool at time t (m/s)
V	cutting speed (m/min)
v	scattering vector of reflected light
V_{cr}	critical cutting speed above which built-up edge is negligible
v_x, v_z	components of **v** in the x and z direction, respectively
$w_{l,j,i}$	connection weight between node *j* in layer *l* and node i in layer (*l-1*)
WRCK	white reference clock
XCK	transfer clock
α	tool's relief angle (°)
β_f, β_d	indices related to f_o and d_o, respectively
γ	tool's rake angle (°)

ΔK	control step size of K
$\Delta n(t_i)$	spindle speed drop at instant t_i (rpm)
Δt_i	estimation interval $(t_i - t_{i-1})$ (s)
$\Delta\rho(t)$	uncut chip area at time t (mm)
δ	tool's wedge angle (°)
ε	computational error, defined as $\varepsilon=\lvert S_m-S_c\rvert/S_m$
ζ	momentum
η	learning parameter
θ_1	laser beam's angle of incidence[1]
θ_2	scattering angle defining a CCD pixel location[1]
λ	wavelength of laser light (μm)
μ	coefficient of friction on the cutting tool's flank
μ_k	overlapping factor used to incorporate the vibration effect due to the previous cut
ρ_o	nominal tool's nose radius (mm)
$\rho(t)$	tool's time varying nose radius at time t (mm)
σ	standard deviation of surface amplitude distribution
σ_y	yield stress of workpiece material (N/mm^2)
σ_z	RMS height of the surface (μm)
τ	gain of sigmoid nonlinearity
ϕ	tool's principal cutting edge angle (°)
ϕ_1	tool's auxiliary cutting edge angle (°)
ω_n	natural frequency (Hz)

1 θ_1 and θ_2 are measured with respect to the normal of the workpiece's surface coincident with the center of the laser beam. (See Fig. 4.7.)

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS ii

ABSTRACT iii

LIST OF FIGURES v

LIST OF TABLES ix

NOMENCLATURE x

TABLE OF CONTENTS xv

CHAPTER 1. INTRODUCTION.. 1

1.1 Objectives of the research 3

- Measurement
- Prediction

1.2 Organization of the thesis 4

CHAPTER 2. SURFACE FINISH GENERATED IN TURNING.................... 6

2.1 Surface texture 6

2.2 Characterization of a machined surface 8

2.2.1 Mean-line system 8

2.2.2 Two-dimensional assessment of a machined surface 10

2.3 Surface finish generated in finish turning 10

2.3.1 Finish turning 10

2.3.2 Generated surface finish 11

2.3.3 Factors affecting surface finish 14

CHAPTER 3. LITERATURE REVIEW.. 17
3.1 Techniques for surface roughness measurement 17
3.1.1 Comparative method 17
3.1.2 Stylus method 17
3.1.3 Pneumatic method 19
3.1.4 Capacitance method 19
3.1.5 Ultrasonic method 21
3.1.6 Optical method 21
(1) Reflectance 22
(2) Light scattering 23
(3) Speckle pattern 25
(4) Ellipsometry 27
3.2 Models for surface roughness prediction 28
3.2.1 Empirical model 30
3.2.2 First- or second-order model 31
3.2.3 A model including the effect of the cutting tool's wear and vibration 32
3.2.4 Dynamic generation of the topography of a machined surface 33
3.3 Remarks 34

CHAPTER 4. ON-LINE MEASUREMENT USING LIGHT SCATTERING....... 35
4.1 Introduction 35
4.1.1 The inverse problem of light scattering 35
4.1.2 Objectives 36
4.2 Analysis 37
4.2.1 Formulation of the model 37

4.2.2 Correction factor, K 40

4.2.3 Estimation of R_{max} 42

4.3 Optical measuring system 47

4.3.1 Design objectives 47

4.3.2 Description of the measuring system 48

4.3.3 The CCD linear image sensor 51

4.3.4 The CCD evaluation board 52

4.3.5 Electronic interface for data acquisition 53

4.4 Experimental studies and results 59

4.4.1 Influence of a cutting fluid 59

4.4.2 Influence of the ambient light 61

4.4.3 Influence of the workpiece's rotational speed 62

4.4.4 On-line measurements 65

4.5 Remarks 70

CHAPTER 5. REAL-TIME MEASUREMENT USING NEURAL NETWORKS ..72

5.1 Introduction 72

5.1.1 Objectives 73

5.2 Neural networks 73

5.2.1 What are neural networks? 73

5.2.2 The perceptron 75

5.2.3 The multi-layered neural network 76

5.3 Development of the neural network 81

5.3.1 Architecture 81

5.3.2 Training 83

5.3.3 Ambient light compensation 85

5.3.4 Software integration 87

5.4 Experiments and results 88

5.4.1 Preparation of specimens 89

5.4.2 Test on the specimens used in the training 89

5.4.3 Test on the unknown specimens 90

5.4.4 Real-time measurements 91

5.5 Remarks 94

CHAPTER 6. ON-LINE PREDICTION.. 95

6.1 Introduction 95

6.2 Model's formulation 96

6.3 Experiments 103

6.4 Results and discussion 108

6.4.1 Determination of K_s 108

6.4.2 Prediction of cutting force 109

6.4.3 Estimation of h_f 111

6.4.4 Prediction of R_{max} 113

6.5 Remarks 116

CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS........................ 117

7.1 Conclusions 117

7.1.1 On-line measurements 117

7.1.2 Real-time measurements 118

7.1.3 On-line prediction 119

7.2 Recommendations 119

7.2.1 Calibration of the laser system 119

7.2.2 Extension of the measuring range beyond 20 μm 120

7.2.3 Compensation for the ambient light 120

7.2.4 Converting Rmax to R_a or R_q 121

7.2.5 Effect of the concentrated groove wear at the cutting tool's trailing edge on surface finish 121

7.2.6 Cutting fluid 122

7.2.7 Comments on C_p and K_s 122

APPENDIX A: Numerical assessment of surface roughness.................................. 123

APPENDIX B: Program code.. 132

Program 1 Snapshot.c 133

Program 2 INVERSE.FOR 149

Program 3 Neural.c 161

Program 4 New_flan.c 172

Program 5 Est_Rmax.c 182

APPENDIX C: Relationship between the RMS slope and R_{max} [S87]........................ 192

APPENDIX D: PC401 CCD evaluation board.. 194

APPENDIX E: Performance of dynamometer.. 202

BIBLIOGRAPHY.. 205

CHAPTER 1

INTRODUCTION

There are a number of outputs in a turning operation, such as productivity, dimensional accuracy and surface integrity, which represent the end-goal of the operation. In rough turning, productivity (i.e. the material removal rate (MRR)) is the major factor to consider. To maximize MRR, high values of the cutting variables (i.e. the cutting speed, feedrate or depth of cut) are usually recommended. However, the effect of these variables on the life of the cutting tool as well as on the capacity of the machine tool have to be considered. On the other hand, in finish turning, dimensional accuracy and surface integrity as well as the overall productivity are the key factors. Surface integrity is a term that involves several considerations, such as surface texture, freedom from cracks, chemical change, thermal damage (burn, transformation, and overtempering), and adverse, tensile residual stress in the surface layer. For a finish turning operation, the surface texture is a critical factor for determining the production cost, wear, fatigue strength, load capacity, heat transfer, noise, appearance and life of a machined part. Finish turning is usually involved in producing a high quality surface texture, .

All finish-turned surfaces depart, to some extent, from perfect smoothness and geometrical form (i.e., these are both macro- and micro-geometrical aberrations). Indeed, the surface of a turned part has a succession of hills and valleys which may vary in amplitude, spacing and form. Surface texture is a characteristic of a machining operation and represents a combination of irregularities of various kinds and magnitudes, the presence of which is due to machining error, machine vibrations, tool conditions, warping strains, etc. Surface texture includes components such as the surface roughness, waviness, as well as lay and flaws. The surface roughness provides a link between a turning process and the tribological performance of a machined part, as illustrated in Fig. 1.1. When assessed during the machining process, surface roughness can be employed to control the cutting process. On the other hand, the tribological performance of a part can be predicted if surface roughness information is available. The route sheet which contains the machinability data could be modified if the performance of the part is not satisfactory. However, the key point is to have the surface roughness serve as the link when it is available during the turning process.

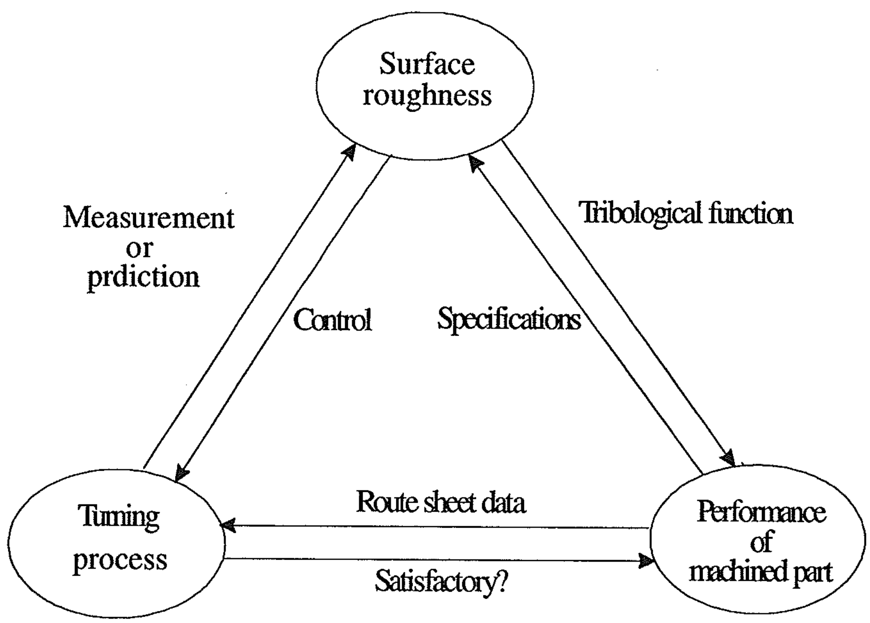

Fig. 1.1 Surface roughness -a link between the turning process and performance of a machined part.

Today, more and more turning operations employ fully automated machines in comparison to manually operated lathes. A Computer Numerically Controlled (CNC) lathe is becoming standard equipment for turning. Even though most of the parameters that control the surface roughness can be programmed in CNC lathes, surface roughness is also affected by non-programmable parameters like tool wear and vibration which vary during turning. By monitoring the surface roughness of a part during finish turning, it should be possible to develop techniques that allow the cutting variables to be optimized in order to maintain a desired surface roughness. Hence, the on-line assessment of surface roughness has become the focus of much research in the past thirty or so years. An on-line assessment can be performed routinely and without disruption between the stages of a machining process. If performed while a part is being machined and the assessment is completed before the next sampling set, real-time assessment is realized. Real-time assessment offers the potential for the adaptive control of surface finish.

1.1 Objectives of the research

The basic motivation of the present research is to explore new approaches suitable for the on-line, real-time assessment of the surface roughness produced in finish turning. The research consists of a measurement and a prediction phase. The goals of the two phases are described next.

- **Measurement**

  (1) To develop a laser measuring system capable of capturing the light scattered from a rotating surface as the turning operation proceeds.

  (2) To propose an optical approach to quantify the maximum peak-to-valley roughness, R_{max}, of a finish-turned surface.

(3) The measurement is required to be performed both on-line and in real-time in the feed direction because changes in surface roughness in the direction of the surface lay, i.e. along the workpiece's circumference, are insignificant in a turning operation [S81].

(4) The measuring range must cover the roughness range encompassed by a finish-turned surface, i.e., R_{max} varies between 4.0 and 20 μm.

- **Prediction**

(1) To propose a comprehensive model to predict the maximum peak-to-valley roughness, R_{max}, of a finish-turned surface.

(2) The prediction is required to be performed on-line by using only the nominal values of the cutting variables and the tool geometry as input.

(3) The effects of progressive tool wear, the relative vibration between the tool and workpiece as well as the changing tool's nose radius on generated surface roughness need to be incorporated into the model.

1.2 Organization of the thesis

Chapter 2 contains a brief description of the texture and characterization of a machined surface. (A detailed description of the numerical assessment of surface roughness is given in Appendix A.) Also discussed in Chapter 2 are the complications involved in finish turning as well as the influential factors . Chapter 3 is a literature review which covers almost 35 years (from 1958 to 1993) of efforts to develop different measurement techniques as well as predictive models for assessing the surface roughness in a machining operation. A comparison of different techniques and models is also presented. Chapter 4 presents an optical approach for the on-line measurement of the maximum peak-to-valley roughness, R_{max}, in finish turning. A laser measuring system, employing a Charge-Coupled Device (CCD) as the image sensor, is developed to capture the Angular Scattered Light Pattern (ASLP) reflected from a finish-turned surface in the feed direction. A description of the system is detailed in Chapter 4. The effects of the ambient light and the workpiece's rotational speed and methods for their compensation are also discussed in this chapter. Chapter 5 describes the application of this optical measuring system in a real-time cutting environment. A three-layer neural network is constructed, in conjunction with a back-propagation learning algorithm, to predict R_{max} by quickly recognizing the captured ASLP. A comprehensive model for predicting the R_{max} on-line is derived in Chapter 6. The nominal cutting variables and the cutting tool's geometry are used as input. The effects of tool flank wear, the relative vibration between the tool and workpiece as well as the tool's changing nose radius are all incorporated into the model. Conclusions and recommendations for future work are given in the final chapter.

CHAPTER 2

SURFACE FINISH GENERATED IN TURNING

2.1 Surface texture

Surface texture, or surface topography, describes the outermost layer of the workpiece which displays numerous macrogeometrical and microgeometrical deviations from the ideal geometrical surface. Deviations of the first order include errors of form which are macrogeometrical. Deviations of the second order are waves while the deviations of the third or higher orders refer to surface roughness. The principal elements of surface texture are defined below:

- Surface: The surface of an object is the boundary which separates that object from another substance. A nominal surface means the intended surface contour, the shape and extent of which are usually defined by a drawing or descriptive specifications.
- Surface texture: Surface texture results from repetitive or random deviations from the nominal surface to form a surface pattern. Surface texture includes roughness, waviness, lay and flaws.
- Profile: The contour of a surface which is in a plane perpendicular to the surface unless another angle is specified.
- Roughness: Relatively fine irregularities which occur in the surface texture. They usually include those irregularities produced by the inherent action of the production

process like the cutting action of tool edges and the feed of the machine tool. Roughness may be considered as short-wavelength surface irregularities superposed on a wavy surface.

- Waviness: Surface irregularities which have greater spacing than the roughness. Waviness may be considered to be long-wavelength surface irregularities and may result from such factors as machine or workpiece deflection, heat treatment or warping strains.
- Lay: The direction of the predominant surface pattern, ordinarily determined by the production method used.
- Flaws: Surface defects which occur at one place or at relatively infrequently or widely varying intervals on the surface, e.g., scratches, checks, ridges, blow holes, cracks, etc.

A typical surface texture, generated during machining, is illustrated in Fig. 2.1. The identification of the macro- or micro-geometrical deviations of the surface are based on a ratio between the width, l_r or l_W, and height, h_r or h_W, as shown in Fig. 2.1. When $(l_r/h_r) < 50$, the micro-geometrical deviations characterizes the surface roughness and when $500 > (l_W/h_W) > 50$, the deviation represents the waviness.

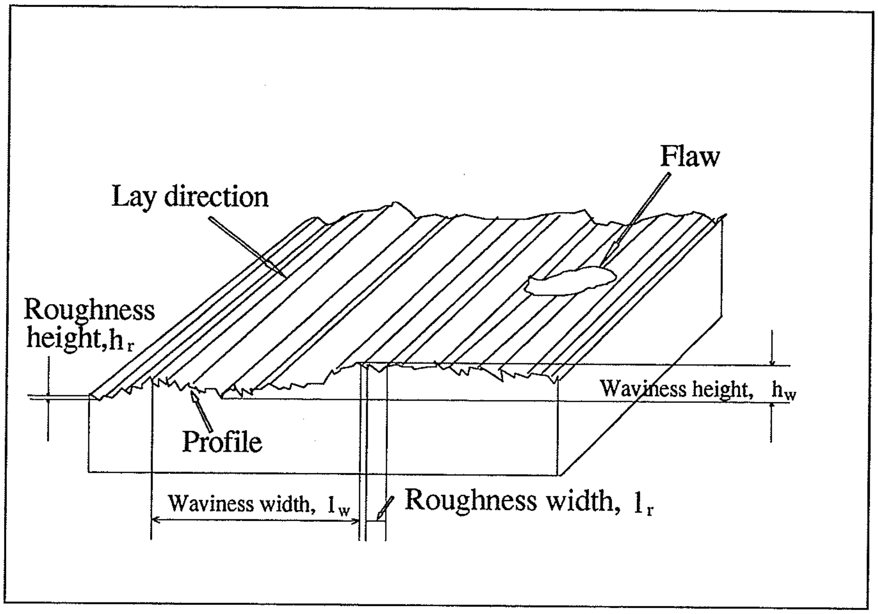

Fig. 2.1 Typical texture of a machined surface.

2.2 Characterization of a machined surface

Many methods have been developed for the characterization of a machined surface. A detailed description of these parameters can be found in Appendix A. One of the most frequently used methods is the mean-line system. Based on the mean-line system, a number of parameters are introduced for the functional assessment of surface roughness. The autocorrelation function, for example, is used for a two-dimensional assessment of a surface profile.

2.2.1 Mean-line system

The mean line, M.L., is defined as "a line having the form of the geometrical profile within the limit of the meter cut-off, and so placed that within the meter cut-off, the sum of the squares of the deviations of the profile from the mean line is minimum" [BSI61]. Given the surface profile $y=f(x)$ shown in Fig. 2.2, M.L. represents the mean line within the sampling length, L. The upper and lower reference lines are represented by U.L.

and L.L., respectively. Based on the mean line, a number of roughness parameters, such as the center line average roughness, R_a, the maximum peak-to-valley roughness, R_{max}, the root-mean-square roughness, R_q, etc., are defined.

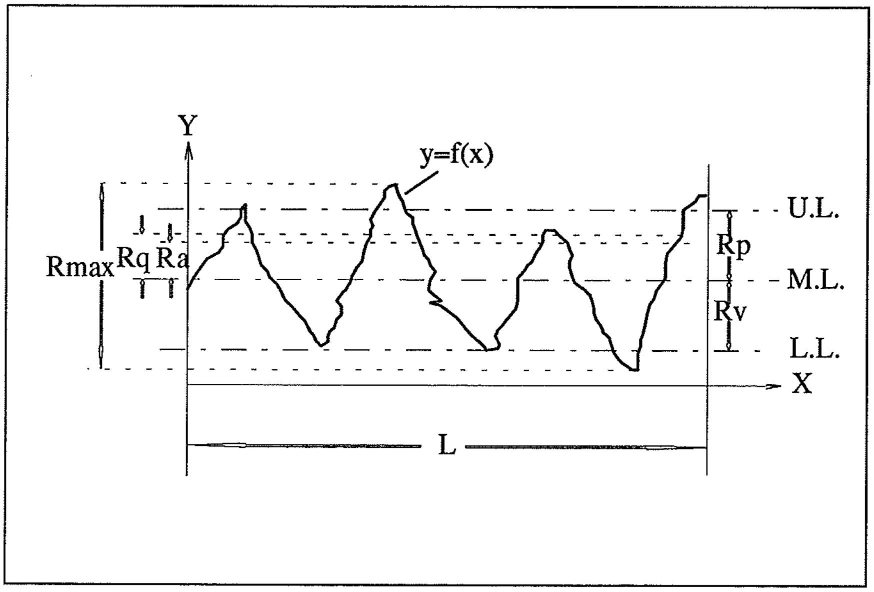

Fig.2.2 A section of a surface profile depicting vertical measures in the mean-line system.

All the parameters defined in the mean-line system are used to determine the statistical characteristics of a traced profile. They represent the expected values of the first or second central moment and characterize the average properties of a surface profile along the direction perpendicular to the surface or *y* co-ordinate axis. Previous investigations [PK67] show that the structure of a surface profile may be quite different when the height-oriented parameters are similar. This is due to the difference in the spacing of the surface profile which can be represented as a function, *y(x)*. To describe the characteristics along the second *x* co-ordinate axis, the surface texture is needed to be assessed in a two-dimensional plane.

2.2.2 Two-dimensional assessment of a machined surface

The profile of a machined surface has been observed to contain periodic and random components. It is of great interest to characterize a surface by detecting the surface's deterministic component as well as the randomness. The concepts of autocorrelation function and power spectral density function are useful to describe such a surface profile in a two-dimensional plane. The autocorrelation function of a surface, $y(x)$, is defined as the expected value of $y(x)$ multiplied by $y(x+\beta)$, where β is an offset distance along the x direction. The autocorrelation function can be used to detect the periodic and random components in a surface profile. The power spectral density function is the Fourier cosine transform of the autocorrelation function and it is used to describe y(x) in the frequency domain.

2.3 Surface finish generated in finish turning

2.3.1 Finish turning

Finish turning is frequently the final phase in a machining operation. It is characterized usually by relatively small depths of cut and light feedrates. A depth of cut less than 1.5 mm and a feedrate less than 0.15 mm/rev. are normally used. The items of major interest in finish turning are:

- dimensional accuracy
- surface finish
- tool life.

Dimensional accuracy depends mainly upon the stiffness and stability of the machine tool. It includes errors in both the longitudinal and circumferential directions. Errors in the longitudinal direction result from the deflection of the workpiece and machine tool's spindle caused by the cutting forces. A thermally induced stress in the machine tool can also cause this type of error. Errors in the circumferential direction result from run out of the spindle, the inaccuracy of spindle bearings, and from vibrations of the tool and workpiece. The geometry of the nose radius and auxiliary cutting edge of the cutting tool produces a significant contribution to the type of surface finish obtained. The range of R_a values to be expected in turning is 0.75-6.3 μm, depending upon the application. The corresponding values of R_{max} will be about 4 to 30 μm [CSA62].

2.3.2 Generated surface finish

In a conventional turning operation, the surface finish produced on a workpiece is the result of the relative movement between the workpiece and cutting tool. The auxiliary cutting edge, which is separated from the primary cutting edge by the nose radius as shown in Fig. 2.3, and the nose radius itself play an important role in generating the surface finish.

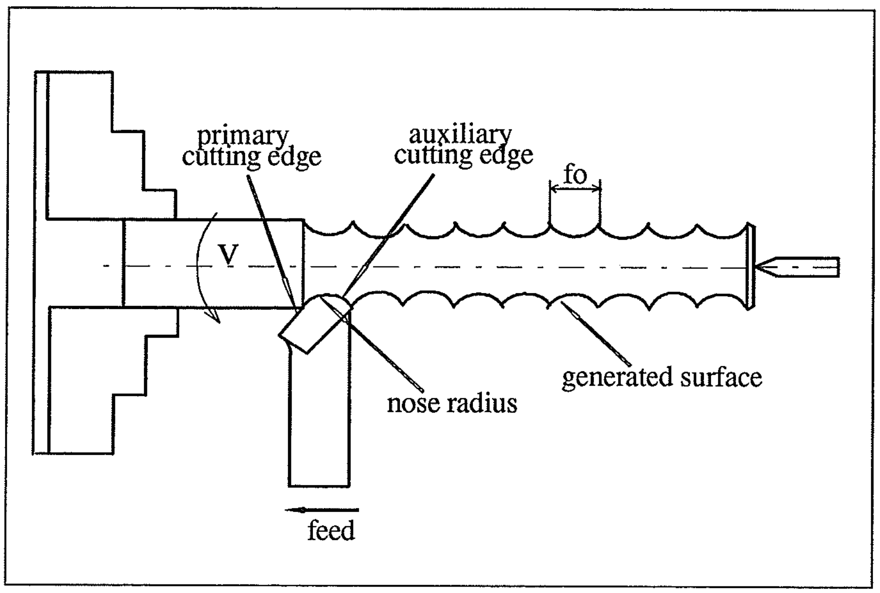

Fig. 2.3 Diagram of a conventional turning operation.

Two roughness profiles of finished-turned surfaces produced by using a carbide tool having a geometry: 7,7,7,7,75,15,0.40 [The American Standards Association, S89] are shown in Figs. 2.4(a) and (b). It can be seen from these figures that feed marks with shapes similar to the nose of the tool and with a pitch equal to the feedrate are produced on the finished surface. As the cutting proceeds, progressive wear occurs on the tool's rake and flank face. A magnified view of a tool bit that shows flank and crater wear is given in Fig. 2.5. The parameter h_f in this figure represents the value of the tool's flank wear.

The following complications are introduced because of the use of the auxiliary cutting edge and the nose radius.

(1) Concentrated groove wear can be observed on the tool's flank face, as seen in Fig.2.5, at both_leading and trailing edge of the tool. The groove thus formed on the trailing edge of the tool acts as a forming tool and leaves a highly cold-worked ridge on the generated surface.

(2) The undeformed chip thickness decreases gradually to zero at the auxiliary cutting edge which causes uncertainty in the geometry of the cut at the trailing edge.

(3) The metal at the trailing edge of the tool is subjected to an unusually high normal stress and will flow to the side to relieve this stress. This produces, in turn, a furrow that contributes to the roughness.

(4) The nose radius, together with the auxiliary cutting edge, is the last portion of the cutting edge which contacts the workpiece's surface and it has a direct effect on the generated profile.

(5) The effect of tool vibrations on the roughness profile is more significant at a smaller feedrate. (Compare Figs. 2.4(a) and (b).)

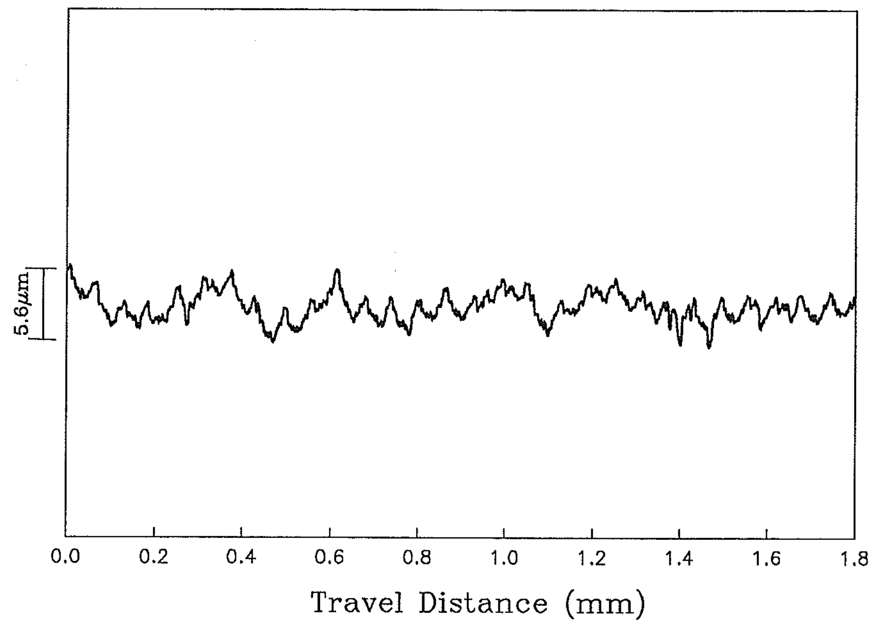

Fig. 2.4(a) Roughness profile , V=119.8 m/min, f_0=0.061 mm/rev, d_0=0.5 mm, workpiece: SPS 4340 steel, HRc=36, carbide tool: 7,7,7,7,45,15,0.40.

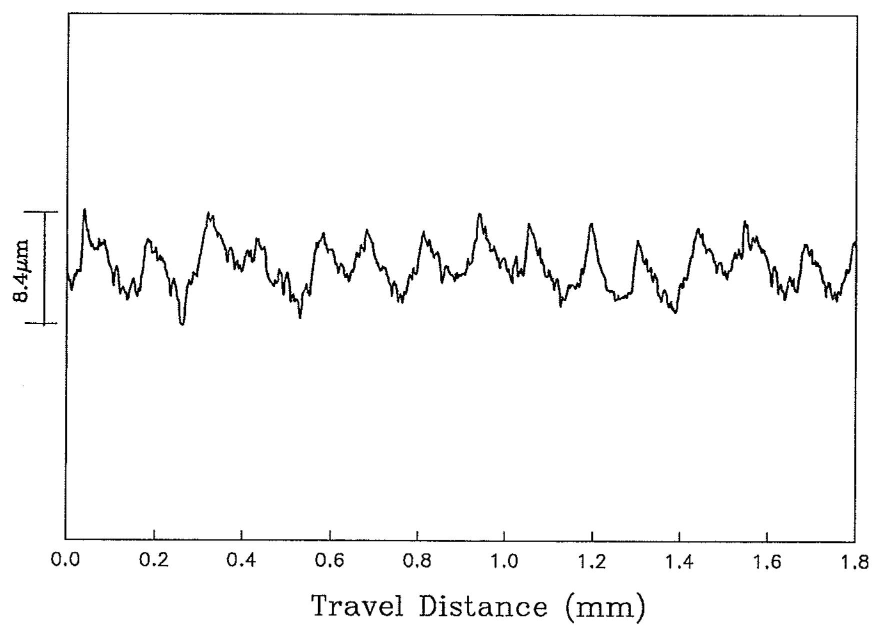

Fig. 2.4(b) Roughness profile , V=119.8 m/min, f_0=0.122 mm/rev, d_0=0.5 mm, workpiece: SPS 4340 steel, HRc=36, carbide tool:7,7,7,7,45,15,0.40.

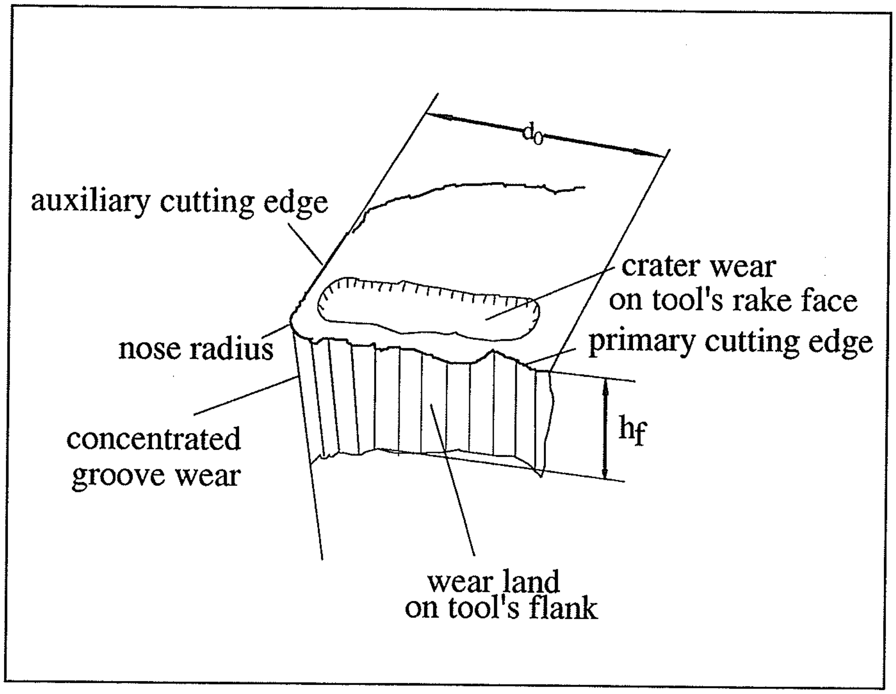

Fig. 2.5 A magnified view of a tool bit showing flank and crater wear.

The value of surface roughness measured on a turned surface will vary, depending on the direction of the measurement. The direction normally employed to define this value is that corresponding to the maximum roughness. For a surface finish generated by an auxiliary cutting edge, the roughness will be greatest in a direction perpendicular to the cutting speed vector and, hence, the roughness values should be obtained by using a stylus traversed in an axial direction.

2.3.3 Factors affecting surface finish

The major factors affecting a surface finish produced in finish turning are:

- cutting variables: feedrate, cutting speed and depth of cut,
- conditions of cutting tool (type, geometry, tool wear),

- rigidity of the machine and tool,
- property of the workpiece's material, and
- application of cutting fluid.

Of the factors influencing the surface roughness, the depth of cut has been found to have the least effect [S89]. Another factor which affects the surface roughness is the built-up edge (BUE). The BUE is a separated layer from the bulk of chip and is formed due to a considerably high frictional resistance at the tool-chip interface. This separated layer is built up on the rake face of the tool as the cutting proceeds. Discontinuous or semi-discontinuous chips as well as the BUE formation may occur at cutting speeds less than 25 m/min. Built-up edges are strongly deformed strain-hardened material and they form a zone of stagnation which increases tool life at the expense of a poor surface finish. It was found that BUE forms with increasing friction and diminishes with increased temperature [S63,B84]. For example, the BUE may form when the temperature is below 300^{o}C while cutting steel. In addition, the influence of the BUE is less at a large rake angle of the cutting tool. There is usually a critical cutting speed, V_{cr}, corresponding to a certain rake angle. When the cutting speed is higher than V_{cr}, the BUE is negligible. Hence, it is very important to carefully choose the cutting speed so that the influence of the BUE is minimized. The auxiliary cutting edge and Brinell number of the workpiece's material also affect the surface roughness. On the other hand, the tool flank wear causes the surface finish to deteriorate rapidly and it has a direct effect on the maximum peak-to-valley roughness [MFY87]. Moreover, the tool's vibratory motion will become significant at a small feedrate appropriate to finish turning. It was reported [ZK91b] that reducing the feedrate leads to major consequences on the dynamic generation of surface irregularities.

The smaller is the feedrate, the larger is the variance of the random excitation [ZK91a]. This random excitation, which is caused by a random distribution of the material hardness of the workpiece, powerfully impacts the cutting tool. This, in turn, introduces vibrations during cutting. Moreover, when the feedrate is reduced, the overlap factor of the regenerative mechanism is increased and the tool vibration is effectively transmitted from one revolution to the next revolution as the cutting tool is traversed along the workpiece's surface.

CHAPTER 3

LITERATURE REVIEW

3.1 Techniques for surface roughness measurement

3.1.1 Comparative method

The basic form of sensory feedback to assess surface roughness before 1930 was the use of tactual standards. In this approach, the operator in the shop ran a fingernail across a standard series of specimens having different finishes and compared them with the surface to be assessed. When two surfaces were felt to have the same roughness and the reference surface was sufficiently smooth, then the workpiece's roughness was considered acceptable. Such a measurement obviously depended mainly on the operator's experience and, hence, was very subjective.

3.1.2 Stylus method

In the early 1930s, surface roughness was commonly measured off-line by using a stylus type of instrument that traversed a surface with a diamond stylus (Fig. 3.1). Transducers were used to reproduce the unknown surface profile by relating the vertical and horizontal motions of the stylus to the convolution of the geometry of the stylus-tip and the microgeometry of the surface under inspection. This method has been internationally accepted as standards for surface roughness measurements [ASA62; CSA62; JIS76; BSI61]. It is often used as the reference to calibrate other techniques. A variety of statistical parameters and functions (which are discussed in Appendix A) are used as measures. They may be displayed on a screen or output on a line printer in order to obtain a permanent record. The stylus technique has a fine resolution and wide range. The ultimate vertical resolution can be as high as 0.05 μm and the range can be as great as 100 μm. The horizontal resolution is limited by the radius of the stylus tip which is typically 5 μm but can be as small as 1 μm. These devices have been modified to be compatible with computer-aided measurements. However, the measurement procedure is still impractical for on-line applications because the workpiece has to be removed from the machine for the measurements. The transducer and stylus tip are fragile and, as a result, the instrument must be used in a laboratory-like environment.

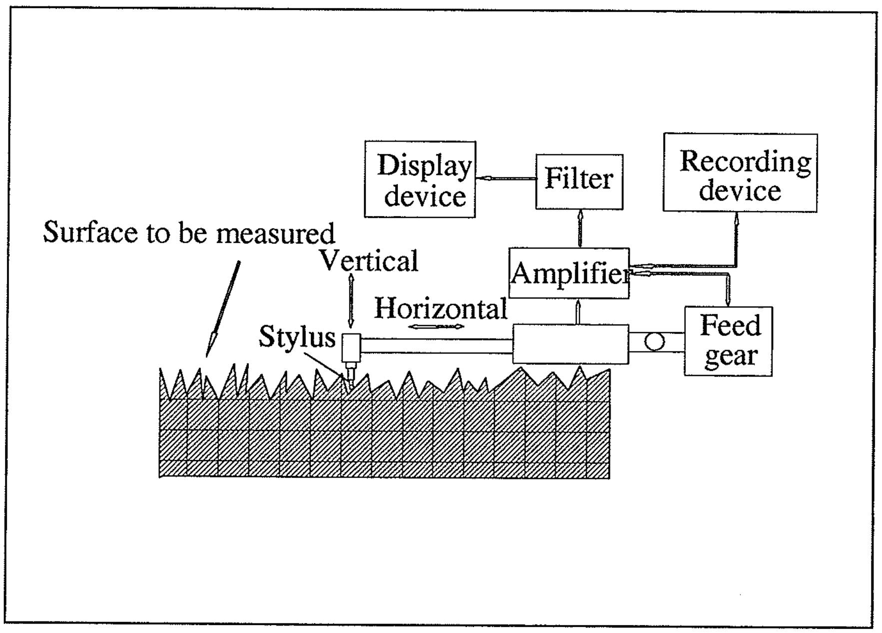

Fig. 3.1 Diagram of a stylus instrument.

The speed with which measurements are obtained is also very slow and the pressure exerted by the stylus tip may also leave a mark on the workpiece's surface. In response to this challenge, many attempts have been made to develop alternative *non-contact* methods. These methods can be classified generally into pneumatic, electrical , ultrasonic and optical devices.

3.1.3 Pneumatic method

With a pneumatic method, a current of air flows between a gauge head and the surface to be measured. The volume of air flowing through the gap and thus the pressure gradient created varies with the roughness of the surface. This method was applied to different machined wood and paper surfaces as well [B79] and a roughness volume, which represents the volume between the gauge head and surface, was introduced to replace the absolute value of the surface roughness. As a result, the output is a comparative or relative roughness value of the surface. The measuring range varies from 20 to 240 μm for R_{max}. However, this method suffers from poor resolution and instability [YDL90].

3.1.4 Capacitance method

Conventional capacitance measurement of surface roughness uses a wide, flexible, metal electrode located parallel to, and in close proximity with, the conducting surface to be measured. The capacitance generated between these two conductors is inversely proportional to the average roughness of the surface. However, this *area-method* capacitance is unable to trace the local height of the surface due to the thickness of the metal electrode employed in the device. A modified capacitive profilometry was developed for turning applications by Garbini *et al* [GKJR92]. In this technique, a translating probe uses a thin metallic strip (typically 0.3 μm), laminated between layers of insulating ceramic material. The sensing probe is oriented so that the metallic strip is perpendicular to both the surface and to the direction of motion of the probe. The surface acts as a second electrode. The arrangement is shown in Fig. 3.2. A "fringe" electric field is produced between these perpendicular surfaces. This perpendicular arrangement of the electrode results in a highly *localized* electric field which is sensitive to only the local surface height. As the probe is moved along the surface, variations in the capacitance of the field are generated by virtue of the changing local distance. These capacitance changes modulate an electronic detector providing a signal related to the local distance. The sensing element remains in contact with the surface throughout the measurement so that machine and workpiece vibrations do not contribute to the measured height. It is reported [GKJR92] that this method is particularly suitable for a harsh environment and real-time measurement. It was claimed [GKJR92] that a R_a measuring range of 0.08 to 4.0 μm could be obtained.

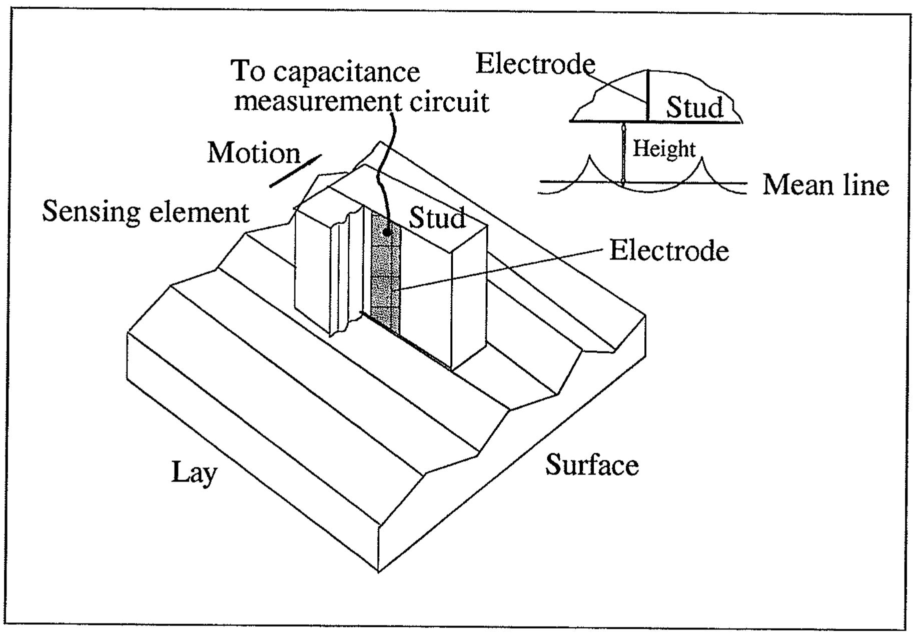

Fig. 3.2 Cutaway view of the fringe-field sensor on a one-dimensional surface.

3.1.5 Ultrasonic method

Huynh *et al* [HF92] have described an ultrasonic sensor used to measure the surface roughness in water. The measuring range for R_q was extended from 1 to 40 μm with a resolution of 0.5 μm.

3.1.6 Optical method

Optical methods for surface roughness measurement can be classified into two categories: parametric and profiling techniques [VT80]. A profiling technique, such as interferometry or holography, enables the surface topography to be derived from a point-to-point scan of the surface height, y, as a function of the travel distance, x, along a line on the surface. This procedure is similar to that for a conventional stylus measurement. The resulting profile can be used to derive various roughness parameters. In this method, data is sampled sequentially, one element at a time, and, hence, the method is very slow. Consequently, the method is not suited to on-line measurements and, hence, it will not be discussed further. A parametric technique, on the other hand, directly produces a measurable parameter representing some property of the surface topography averaged over the illuminated area. Parametric techniques are much faster than profiling techniques but the measurements are difficult to relate to the roughness parameter. Several methods employing a parametric technique are discussed below.

When collimated laser light is projected onto a rough surface, the intensity and pattern of the scattered reflection depends upon the roughness, the spatial wavelength of the surface as well as the wavelength of the light. For a very smooth surface (RMS surface height, $\sigma_z \ll$ wavelength of light, λ), most of the reflected light propagates in the specular direction in which the scattering angle, θ_2, equals the angle of incidence, θ_1. As σ_z increases, the specular intensity decreases but the reflection increases in intensity and becomes more diffuse. It is also found that the diffused reflection has a fine grainy structure called a speckle pattern, which represents an intensity contrast between neighboring points in the scattered pattern. Additionally, the polarization of light wave may change upon reflection from the surface. All five quantities: specular intensity, total intensity of the scattered light, diffuseness of the angular scattering pattern, speckle contrast, and the polarization, depend upon the surface roughness and, hence, can be employed potentially to measure the surface roughness.

(1) Reflectance

In this method, the specular intensity or diffusely scattered light reflected from a surface is used to measure the surface roughness. The easiest way is to measure the intensity of the specular beam. If σ_Z is much smaller than λ and the surface has a Gaussian height distribution, the correlation between the normalized specular intensity, I_S, and σ_Z is described by [BS63]:

$$I_S = e^{-\frac{4\pi\sigma_Z \cos\theta_1}{\lambda}} . \tag{3.1}$$

For $\sigma_Z \geq \lambda/10$, the true specular beam effectively disappears so that σ_Z is no longer measurable. The limitation of the method arises from the size of the detector's aperture as well as that of the illuminating beam and optical wavelength. If a He-Ne laser with λ=0.6328 μm is used, only smooth surfaces (σ_Z < 0.1 μm) can be measured. For most engineering surfaces, σ_Z varies from 0.1 to 2.0 μm so that an infrared light source (for which 1μm < λ < 20 μm) must be employed. A high-speed instrument based on this principle was developed by Jansson *et al* [JRB84] to measure ground and honed surfaces in a 0.05 to 0.5 μm R_a range. A 3.39-μm, infrared laser and a 1-mm, lead solenoid detector were employed to produce a roughness measurement every 20 ms.

Another approach [S81] used two He-Ne laser beams projected onto a turned workpiece perpendicular to the surface. Two pairs of photodiodes were arranged to detect the images of the reflected light. The movement of the two images depends on the displacement of the laser spots which are a function, in turn, of the surface roughness. The maximum roughness in both the feed and circumferential directions can be measured and chippings or chatter can be detected very sensitively. A similar approach was developed by Yoo *et al* [YDL90] to correlate the positional variation of the reflected laser light to changes in the surface roughness. This laser detector consisted of an OPEYE detector having a 1 cm x 1 cm two-dimensional photodiode. When the light spot was imaged onto the detector, the photocurrent generated depended on the distance of the light spot from the respective electrodes of the detector. The procedure has been successfully used for measurements on wood surfaces.

(2) Light scattering

The variations in the microgeometrical structures of a surface will produce different scattered light patterns and, therefore, it can be used for surface roughness measurements. The basic theory can be found in Beckmann and Spizzichino [BS63]. Broadmann gave an overview of different models [BGT85]. The characteristic roughness values and characteristic surface functions determined from the models given by [BGT85] are summarized in Table 3.1.

An optical measuring system was developed by Broadmann [BGT85] to measure the light scattering pattern, the variance of which is correlated with R_a. An infrared LED (Light-Emitting Diode) having a wavelength of 0.8 μm was projected onto a surface by means of a collimator lens and an image lens. The scattered light was collected and directed to a diode array. Good accuracy was found within a range of R_a=1 to 2 μm and the sensitivity is especially high in a range of 0.02 to 0.5 μm for R_a. The sensitivity decreases with larger roughness values. An investigation of the relationship between R_q and characteristics of the light scattering pattern reflected

Table 3.1 Overview of the Methods Used to Describe Light Scattering [BT85].

Description	Model	Roughness parameter	Restriction
Total integrated scattering	Diffraction	R_q	$R_q < 0.01$ μm Gaussian profile
Fourier method	Diffraction	$S(\omega)$, R_q	$R_q < 0.1$ μm
Beckmann method	Diffraction + Facet	R_q, T	Gaussian profile
Profile angle method	Facet	$p(\xi)$, R_q	No diffraction

$S(\omega)$ = Power spectrum, $p(\xi)$ = Probability density, ξ = Profile angle.

from a ground surface was made by Tsukada *et al* [TN86]. It was reported that a strong relationship exists between R_q and the mean value, variance, standard deviation or kurtosis of the pattern for R_q up to 0.35 μm. Similar work on internal ground surfaces was reported by Yanagi *et al* [YN87] and a good linear relationship was found between R_a (up to 0.2 μm) and the standard deviation of the scattering pattern. A distinctive feature is that there is no restriction on the curvature of the workpiece's surface.

An on-line approach was proposed by Yim *et al* [YK90] to measure R_q of a ground surface and a less than 5% error was reported in the range of 0.025 to 1.6 μm . A 4096-pixel CCD linear array was used as the image sensor. The approach consisted of two phases: for a surface whose R_q roughness value is below 0.1 μm, only the specular intensity was measured; for a R_q roughness in the range of 0.1 to 1.6 μm, the standard deviation of the scattering pattern was used.

The gray-level histogram of the optical Fourier transform pattern of a surface has also been used [LHN89; CH92] to derive an optical roughness parameter. One of these methods utilized a machine vision system [LHN89]. A collimated, white light source was used to illuminate the surface under inspection. The image of the surface was captured, after being passed through a microscope, by a CCD camera having 491x 384 pixels and digitized by using a microcomputer. An optical Fourier transform was employed to obtain a gray-level histogram of the digitized image. An optical roughness parameter, R_O, was derived from

$$R_O = \frac{SD}{RMS_S} \tag{3.2}$$

where *SD* and RMS_S are the standard deviation and RMS height of the distribution of the histogram, respectively. A unique relationship was established between R_O and the average stylus-measured R_a for different materials. The measurement was also performed on specimens immersed in a cutting oil and encouraging results suggested that this method could be applied in a production environment.

(3) Speckle pattern

When a rough surface is illuminated with partially coherent light, the reflected light consists partly of random patterns of bright and dark regions known as speckle. This pattern is caused by the interference of wavelets, scattered from different points within the illuminated area, with their phases randomized by height variations of the surface. Basic information of the statistical properties of laser speckle patterns can be found in [G75]. The spatial pattern and contrast of the speckle depends on the surface roughness, the coherence condition of the illumination, and the optical system. The average contrast, defined as the normalized standard deviation of the speckle intensity variations at the observation plane, was used by Fujii *et al* [FAS76; FA77] to measure the surface roughness of polished glass or metal. Good correlation was found between the maximum average contrast of the image speckle intensity variations produced in the far-field region and R_q or R_a of the surfaces in a range of 0.03 to 0.25 μm. The measuring range in which the coherent monochromatic speckle contrast is sensitive, can be increased by changing the angle of incidence or by increasing the illumination wavelength. The average contrast was also found to be a function of the extent of the illuminated surface area.

Another widely used technique for relating speckle with surface roughness is the speckle pattern decorrelation measure. In this technique, two speckle patterns are obtained from a surface by using different angles of incidence or wavelengths of light. The correlation between two patterns is studied by double exposure of the patterns on the same photographic plates or by photoelectric detection. The primary attribute of this technique is its large measuring range. Leger *et al* [LP76] developed a real-time technique using two Michelson interferometers to measure the ensemble-averaged coherence function of the light scattered from a surface illuminated simultaneously by two coherent plane waves. A measurement could be performed in less than 30 s and with a R_q measuring range up to 30 μm .

(4) Ellipsometry

The change in the polarization state of a beam of light reflected from a surface can be measured by ellipsometry. The ellipsometry measurements of the index of refraction are sensitive to a number of surface properties, including the surface roughness. As exemplified in Fig. 3.3, a complex ratio between the two components, R_p and R_s, of the electric field is defined by:

$$\frac{R_p}{R_s} = \tan \Psi e^{i\Delta} \tag{3.3}$$

where the R_p and R_s are the components parallel and perpendicular to the plane of incidence, respectively. The tan ψ and Δ are the amplitude and phase of the complex ratio, respectively.

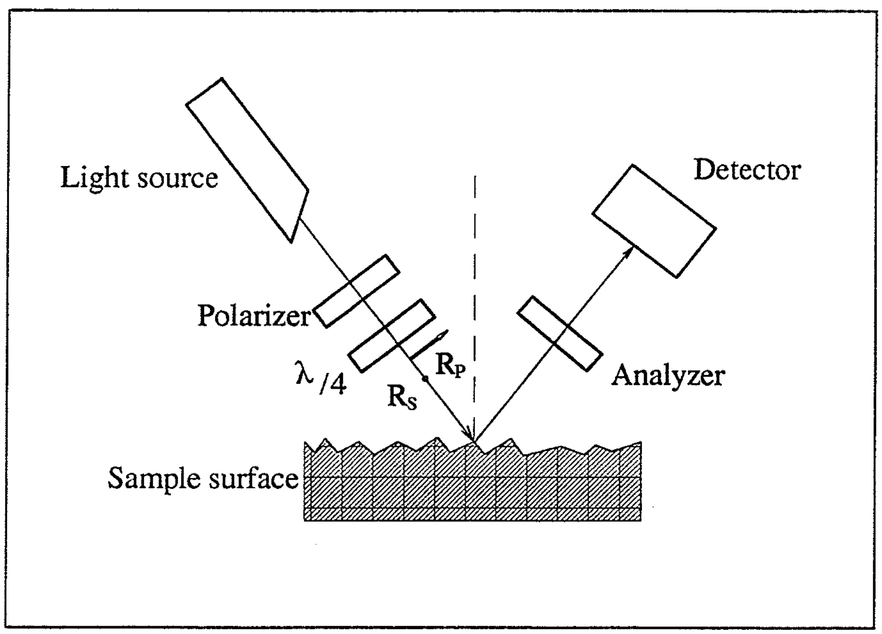

Fig. 3.3 Schematic of ellipsometry in which the detector and analyzer are usually positioned in the specular direction.

The amplitudes of the ratio, tan Ψ , and the phase shift between the two components, Δ, or a combined parameter including the effects of both tan Ψ and Δ are found to vary linearly with R_a.

Lonardo *et al* [L78] employed ellipsometry to measure the surface roughness of ground and polished steel parts having a range of R_a from 0.01 to 1.1 μm . It was found that both Ψ and Δ decreased with increasing R_a. However, the use of ellipsometry for surface roughness measurements has been largely empirical so far and it is not clear how the parameters of the surface topography affect the ellipsometry results. Moreover, two key problems remain to be solved. The first is the quantification of the measured optical data related to the stylus roughness parameter such as R_a. The other requires the ellipsometric measurements to be insensitive to fluctuations of the scattered light intensity caused by vibrations of the surface.

A comparison of the attributes of different measurement techniques is summarized in Table 3.2. These attributes include the capability for quantification, the resolution, range, measurement speed, applicable area , operating cost and availability of a working device.

3.2 Models for surface roughness prediction

The assessment of surface roughness can be performed from on-line measurements by using optical, electrical or pneumatic techniques as described in the previous section. This, however, needs relatively complex instrumentation and does not provide a straightforward relationship between the surface roughness and cutting variables which is especially useful when implementing adaptive control strategies for machining. The last deficiency is overcome usually by employing the alternative strategy of modeling the cutting process as a way of predicting the surface roughness on-line.

Table 3.2 Summary of Surface Roughness Measurement Techniques*.

Technique	Quantified?	Resolution (μm)	Range (μm)	Speed	Area	Cost	Working device?
Stylus	Yes	1.0 (H) , 0.05 (V)	<100	Slow	Laboratory	Medium to high	Yes
Resistance	N/A	1	< 50, R_q	Medium	On-line	Low	Yes
Capacitance	0.3-4 μm	1	< 50, R_q	Medium	On-line	Low	Yes
Pneumatic	<22.9 μm	Poor	20-240	Medium	On-line	Low	Yes
Ultrasonic	Yes	0.5	<40, R_q	Fast	In-process	High	N/A
Specular	Rq< λ	1	< λ	Fast	On-line	Low	Yes
Scattering	Rq< 1.6 μm	< 0.001	1-3	Fast	On-line	Low	Yes
Speckle contrast	Rq< 0.1 λ	0.01	0.01-30	Fast	On-line	Medium	No
Speckle correlation	λ<Rq<10λ	10% of range	1-30	Medium	On-line	Medium	No
Optical Fourier transform	Rq< λ	0.01	<0.1	Medium	On-line	Medium	No
Ellipsometry	N/A	<0.001	1	Medium	On-line	Medium	No

* Source : [HF92;VT80]

H - Horizontal

V - Vertical

N/A - not available

Four approaches have been used to predict the surface roughness in finish turning and they are described next.

3.2.1 Empirical model

The first approach defines an empirical relationship between the surface roughness, cutting variables and the cutting tool's geometry. The most commonly used relationship is [S63]:

$$R_{max} = \frac{f_o^2}{8\rho_o} \tag{3.4}$$

for $f_o < 2\rho_o \sin\phi$. Olsen [O68] investigated the combined effect of the cutting speed and feedrate on the surface roughness from numerous finish turning tests on plain carbide steel. Two empirical equations were obtained to predict R_a and R_{max} (μin), as:

$$R_a = 600\left(\frac{V}{75}\right)^{(62.5f_o - 1.6)} \tag{3.5}$$

and

$$R_{max} = 2400\left(\frac{V}{75}\right)^{(62.5f_o - 1.6)} \tag{3.6}$$

where V is in ft/min and f_o is in in/rev. It was proved that eqs. (3.5) and (3.6) had fairly good accuracy in the feedrate range of 0.002 (0.051 mm/rev) < f_o < 0.026 (0.66 mm/rev) in/rev. and the speed range of 225 (68.2 m/min) < V < 750 (227.3 m/min) ft/min.

Hasegawa *et al* [HSL76] developed a predictive model in terms of the four variables V (m/min), f_o (mm/rev), d_o (mm), and ρ_o (mm). A relationship was derived from turning tests by using a response surface methodology as:

$$R_{max} = 624.2432 \cdot V^{-0.4349} \cdot f_o^{0.8129} \cdot d_o^{0.0385} \cdot \rho_o^{-0.4680} (\mu m). \tag{3.7}$$

Based on the experimental data given in [HSL76], a dynamic simulation was utilized by Jang *et al* [JS89] to develop a similar equation which included the influence on the surface roughness of the machine tool stiffness and tool vibration . No experimental validation was reported. All these predictive models apply to only a limited range of cutting conditions whilst the cutting tool's nose radius is assumed constant.

3.2.2 First- or second-order model

The second approach is to establish a first- or second-order model from a multiple regression analysis. The procedure to obtain a model is even more time-consuming and costly than the first approach. Sundaram *et al* [SL81] developed a first-order predictive model for a finish-turned surface. Variables included in the model were the cutting speed, feedrate, depth of cut and cutting time. A similar second-order model was also provided whenever a better prediction was required. The models given in [SL81] can be applied to predict R_q for AISI 4140 steel using TiC coated or uncoated Tungsten carbide throw-away tools in the following cutting conditions:

Cutting speed (ft/min):	285 - 680
Feedrate (in/rev):	0.0075 - 0.0104
Depth of cut (in):	0.017 - 0.050.

Mital et al [MM88] developed a general, non-linear model for three materials: ductile cast iron, medium carbon leaded steel 10L45, and medium carbon alloy steel 4130. The models proposed take into account the cutting speed, feedrate, tool's nose radius, and their interactions. A randomized complete block factorial design was used for the test. For each kind of material, three nose radii (0.794, 1.190, and 1.587 mm), four feedrates (0.0508, 0.127, 0.2032, and 0.3048 mm/rev), and three cutting speeds (ranging from 76.2 to 365.7 m/min, depending on the material) were used. The model has the form :

$$R_a = -69.16 - 2.59\rho_o - 57.92f_o - 14.04\rho_o f_o + 0.013mVf_o + 0.007mVf_o\rho_o - 0.374\log V. \quad (3.8)$$

The *m* in equation (3.8) is a constant whose value is unity for ductile cast iron, two for 10L45 steel, three for 4130 alloy steel. The performance of the model was evaluated on the test data and good correlation was found. The second model, like the first approach, can only be used off-line and the effect of progressive tool wear is ignored [S64]. The tool wear gradually alters the feed marks generated by the cutting tool on the machined surface and changes the uncut chip area which causes, in turn, undesirable fluctuations in the cutting force. Consequently, efforts were made to develop a model that included the effect of tool wear and vibration.

3.2.3 A model including the effect of the cutting tool's wear and vibration

The effect of tool wear is included in the third modeling approach. Such a modification provides a basis for the adaptive control of a finish turning operation in which the process variables can be adjusted as the tool wears in order to maintain a desired surface roughness. Dontamsetti *et al* [D85] developed a model by using experimental data from finish turning gray iron with an uncoated tungsten carbide tool. The surface roughness generated is a function of the cutting speed, feedrate, tool's nose radius, and the amount of tool wear. It is given by:

$$\begin{aligned}\mathrm{Log}(R_a) = 3.945 + 196.23f_o + 0.00057V - 3.335\rho_o + 37.73h_{fo} - \\ 0.957\cdot 10^{-5} f_o V^2 - 7243.8h_{fo}f_o - 0.0673h_{fo}V + \\ 20.61h_{fo}\rho_o - 1.503f_o^2 V - 7.86\cdot 10^{-7} V^2\end{aligned} \quad (3.9)$$

where R_a is in μin, V is in fpm, f_o is in in/rev, ρ_o and h_{fo} are in in. This formula gave an error which is less than 10% of stylus measured data. Solaja [S59] found that the concentrated groove wear on a tool's flank had a significant influence on the surface finish in turning steel. A mathematical relationship connecting this groove wear and the tool's flank wear with R_{max} (μin) was derived as:

$$R_{max} = (h_{ft} - h_f)\tan\alpha + \frac{(h_{ft}^2 - h_f^2)}{D} + \frac{(f_o - 0.065h_{ft}\frac{\rho_o}{f_o})^2}{8(\rho_o - h_f\tan\alpha)} \quad (3.10)$$

where all dimensions are in in. and h_{ft} is the height of the concentrated groove wear on the tool's flank. Good correlation was found between the data estimated from equation (3.10) and stylus measurements. Sata [S63] found that the roughness generated by a tool-nose cut surface consists mainly of four components: the ideal roughness determined by the tool's nose shape and feed, a component due to vibration, swelling of the workpiece's material, and a component due to tool wear. Experimental results were presented but no mathematical derivation was given. The influence in finish turning of the cutting forces on the surface roughness along the lay was investigated by Rakhit *et al* [RSO76]. A relationship was obtained to determine the resulting surface roughness, R_a. It consisted of a theoretical roughness, R_{th}, and component, σ_f which represents the mean-square displacement of the relative dynamic motion between the tool and workpiece. The relation is:

$$R_a = \sqrt{0.865R_{th}^2 + 0.640\sigma_f^2}. \quad (3.11)$$

The predicted R_a agree closely with actual measurements in the case of finish turning. However, the change in the tool's condition was not considered in the above models due to the difficulty of monitoring tool wear on-line.

3.2.4 Dynamic generation of the topography of a machined surface

The fourth and most recent approach for predicting surface roughness involves a mechanistic model to dynamically generate the surface texture. Zhang *et al* [ZYL91;

ZK91a,b] employed a computer simulation to generate the surface topography during intermittent turning. A mathematical model was developed to characterize intermittent turning as an alternating sequence of forced and free vibration. The tool's vibratory motion was integrated with its geometric motion to form a basis for constructing the surface topography produced during machining. The roughness parameter could be assessed , therefore, by using the calculated surface topography. Experiments showed good agreement between the simulated R_a values with stylus-measured data. However, the effect of tool wear was neglected although the geometrical aspects as well as tool vibrations were accommodated. A surface generation model was developed for milling which included both the cutter vibration and the effects of tool wear [IEDU93]. The surface roughness parameters, and characteristic features of the surface topography were examined by using computer simulations. Predicted R_a and R_{max} values were compared with stylus measurements and good agreement was shown. However, the effect of tool wear was considered merely by using a constant flank wear value measured off-line before the cut.

3.3 Remarks

The literature review indicates that an optical-based, on-line technique for the surface roughness measurement still remains to be developed for finish turning. A comprehensive model which incorporates the cutting variables and tool geometry as well as the effect of progressive tool wear and the dynamic characteristics of the machine-workpiece-tool system needs to be derived for on-line prediction of surface roughness in finish turning. These two topics are addressed in the following chapters.

CHAPTER 4

ON-LINE MEASUREMENT USING LIGHT SCATTERING

4.1 Introduction

Laser light scattered from a rough surface has been used in the past mainly as a comparator method in which changes in the roughness properties of a machined surface were monitored by using scattered light. A surface roughness parameter might also be determined in an empirical way. However, the measurement range was found to be far less than that encountered usually in finish turning. Hence, an on-line measurement system using light scattering still remains to be developed for *quantifying* the surface roughness in a range applicable to finish turning. To deal with this topic, the inverse problem of light scattering has to be considered first.

4.1.1 The inverse problem of light scattering

Given a surface having known roughness values, the determination of the intensity distribution of the scattered light reflected from the surface is called the direct problem of light scattering. On the other hand, the indirect determination of the surface roughness from measurements of the light intensity scattered from that surface is termed the "inverse problem of light scattering". This is a research topic of great interest. To-date, only Marx and Vorburger [MV90] have attempted to solve this problem. However, their approach determines the surface roughness straightforwardly from the reflected light intensity only when R_q is much lower than the wavelength of the He-Ne laser light source. For situations when R_q is equal or slightly less than this wavelength, R_q is found by employing a least-squares curve fitting method. This method, however, needs stylus-measured data of the surface to inconveniently supplement the measured light intensity. Furthermore, an R_q can be determined between merely 0.01 and 0.50 μm which is much lower than the roughness range encountered in finish turning. Nevertheless, if the inverse problem of light scattering could be solved, it would have several important advantages over stylus methods. The optical sensor can respond much faster to the scattered light which would provide a quick, real-time measurement for surface finish control. Moreover, it integrates the roughness measurement over a small region on the surface, giving a more area measure of the surface roughness. In summary, it is of great interest to uniquely determine the roughness parameters of a finish-turned surface without having to remove the workpiece when machining is in progress.

4.1.2 Objectives

The formulation of a methodology to quantify, on-line, the relationship between the light scattered from a finish-turned surface and the roughness parameters is the focus of this chapter. An optical approach, based on solving the inverse problem of light scattering, is proposed to determine the maximum peak-to-valley roughness, R_{max}, when the effects of chatter and a built-up edge on the tool are avoided [S63]. The theory supporting the methodology is presented in Section 4.2. A measuring system employing a He-Ne laser light source and a linear Charge-Coupled Device (CCD) sensor are developed to capture the scattered light and verify the methodology. Details of the design of the measuring system are given in Section 4.3. A description of the experimental procedure and experimental results are presented in Section 4.4. Optical methods for measuring the light reflected from a rotating surface are usually affected by the ambient room light as well as by the rotational speed of the surface. The influence of these two parameters on the estimated R_{max} and procedures to correct the estimation are also studied in Section 4.4. Results from on-line measurements without and with cutting occurring are also discussed in this section. Section 4.5 presents the concluding remarks.

4.2 Analysis

4.2.1 Formulation of the model

The roughness profile, produced in turning by using a tool having an auxiliary cutting edge, is generally considered to contain a periodic component superposed with

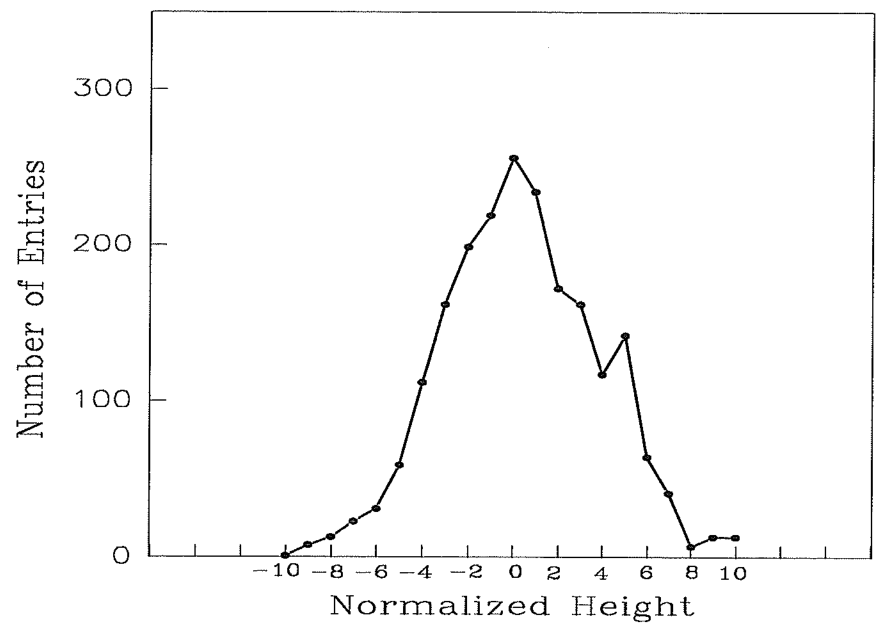

Fig.4.1 Profile height distribution of the finish-turned surface shown in Fig.2.4(a).

random components. The height distribution of the surface profile shown in Fig.2.4(a) is given in Fig.4.1. The height of the surface profile is normalized relative to the maximum height and more than 95% of the amplitudes were found to be within $\pm 3\sigma$, where σ is the standard deviation. Hence, the distribution illustrated in Fig.4.1 is reasonably Gaussian. When a laser light beam is projected onto such a surface, it will be scattered by microscopic surface irregularities (Fig.4.2).

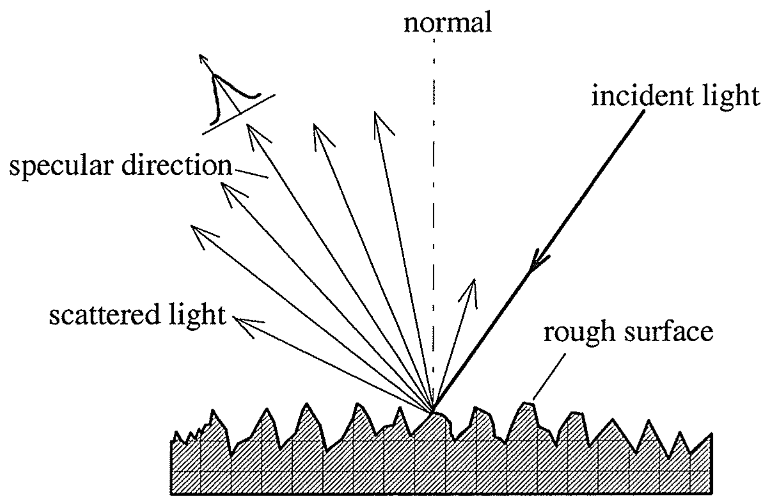

Fig. 4.2 Light scattered from a rough surface.

Beckmann and Spizzichino [BS63] derived the intensity distribution, $I(\theta_1,\theta_2)$, of the light scattered from a one-dimensional, randomly rough surface, whose RMS surface height, σ_z, is far greater than the wavelength of the light source, λ, as:

$$I(\theta_1,\theta_2) = \frac{\sqrt{\pi} F_2^2 T}{L_o v_z \sigma_z} \cdot e^{-\frac{v_x^2 T^2}{4 v_z^2 \sigma_z^2}} \tag{4.1}$$

where

$$F_2 = \frac{[1+\cos(\theta_1+\theta_2)]}{[\cos\theta_1(\cos\theta_1+\cos\theta_2)]}, \tag{4.2}$$

$$v_x = \frac{2\pi}{\lambda}(\sin\theta_1 - \sin\theta_2), \tag{4.3}$$

and

$$v_z = \frac{2\pi}{\lambda}(\cos\theta_1 + \cos\theta_2). \tag{4.4}$$

Various parameters in equations (4.1) through (4.4) are defined in the nomenclature.

The intensity distribution given by equation (4.1) represents the "Angular Scattered Light Pattern" (ASLP). An ASLP can be defined as the distribution of light intensity reflected from a small area on the surface and registered by an imaging system located perpendicular to and symmetrically about the specular direction of the reflected light. Equation (4.1) suggests that the ASLP contains information about the ratio T/σ_z rather than about T or σ_z individually [BS63]. Moreover this ratio is proportional to the RMS value of the slope of the surface, *u*. An equation relating *u* to R_{max} and the tool's nose radius, ρ_o, was given for a turned surface by Shiraishi [S87] as:

$$u = \sqrt{2}\frac{\sigma_z}{T} = \sqrt{\left(\frac{R_{max}}{\rho_o - R_{max}}\right)}. \tag{4.5}$$

By using this relationship, equation (4.1) can be modified to [S87]:

$$I(\theta_1,\theta_2) = \frac{\sqrt{2}\pi F_2^2}{v_z L_o}\sqrt{\left(\frac{\rho_o - R_{max}}{R_{max}}\right)} \cdot e^{-\frac{v_x^2}{2v_z^2}\left(\frac{\rho_o - R_{max}}{R_{max}}\right)} \tag{4.6}$$

Now ρ_o changes due to progressive tool wear occurring in turning. Consequently, not only R_{max} but, for the last equality to hold, $I(\theta_1,\theta_2)$ actually varies over time. Later,

$R_{max}(t)$ and $I(\theta_1,\theta_2,t)$ will be used to represent the values of R_{max} and $I(\theta_1,\theta_2)$, respectively, at time t, during cutting.

$R_{max}(t)$ in equation (4.6) can be determined by solving the inverse problem of light scattering in which all the other variables including the light intensity of the scattered light, $I(\theta_1,\theta_2,t)$, are known. More specifically, by using a logarithmic transform to linearize equation (4.6), it follows that:

$$\ln\left[\frac{I(\theta_1,\theta_2,t)}{X_1(\theta_1,\theta_2)}\right] = \ln K_1 - \frac{L_o{}^2}{2\pi} K_1^2 X_2(\theta_1,\theta_2) \tag{4.7}$$

where

$$K_1 = \frac{\sqrt{2\pi}}{L_o}\sqrt{\frac{\rho(t)-R_{max}(t)}{R_{max}(t)}}. \tag{4.8}$$

with $x_1(\theta_1,\theta_2)$ and $x_2(\theta_1,\theta_2)$ calculable from the optical parameters by employing

$$X_1(\theta_1,\theta_2) = \frac{F_2^2}{v_z}, \tag{4.9}$$

and

$$X_2(\theta_1,\theta_2) = \frac{v_x^2}{2v_z^2}. \tag{4.10}$$

In principle, K_1 can be evaluated from equation (4.7) by using, say, a standard linear least-squares estimation. Then $R_{max}(t)$ can be found from equation (4.8) providing the value of tool's nose radius, $\rho(t)$, is known at that time t.

4.2.2 Correction factor K

The developed model given by equation (4.7) does not account for discrepancies between the theoretical and measured light intensities, $I(\theta_1,\theta_2,t)$ and $I_m(\theta_1,\theta_2,t)$, respectively. These discrepancies arise because Beckmann's model assumes a randomly distributed surface profile whilst a turned surface contains periodic as well as random components. Moreover, differences may be introduced by (i) the use of a collecting lens in the optical system causes pattern distortion; and (ii) the electronic processing of the measured signal also introduces a scaling effect. Therefore a correction factor, K, has to be introduced into equation (4.7) which becomes:

$$\ln[\frac{K \cdot I(\theta_1,\theta_2,t)}{X_1(\theta_1,\theta_2)}] = \ln K_1 - \frac{L_o^2}{2\pi} K_1^2 X_2(\theta_1,\theta_2). \tag{4.11}$$

K was found to change with R_{max} during a preliminary investigation so that a single value of K does not exist for the entire R_{max} range of 4 to 30 μm reported for turning [CSA62]. Typical variations of K found at various *stylus-measured* R_{max} are illustrated by the experimental points of Fig.4.3. The R_{max} can be seen to noticeably influence K so that an iterative procedure is needed to find the K that will minimize the difference between the theoretical and measured light intensity distributions, $I(\theta_1,\theta_2,t)$ and $I_m(\theta_1,\theta_2,t)$ respectively.

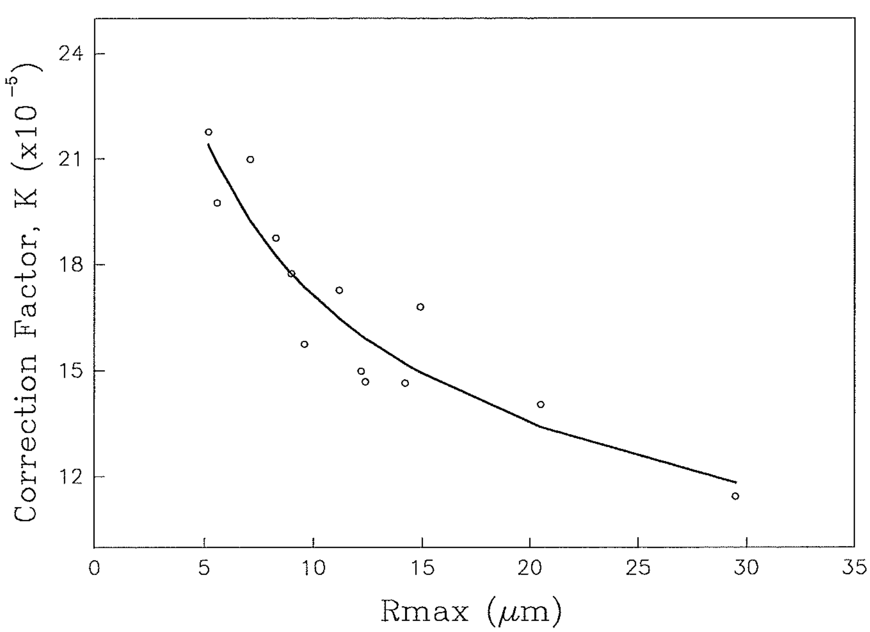

Fig.4.3 Variations of the correction factor, K, with the estimated R_{max}.

It is useful to first determine an appropriate search range for K for finish-turned surfaces. This can be established from two reference surfaces whose R_{max} are known, say from stylus measurements, and encompass the extremes of 4 and 30 μm for turned, steel specimens [CSA62]. The corresponding values of K, i.e. K_a and K_b, can be approximated from equation (4.8) through (4.11) and the resulting K_a and K_b can be obtained to be about $11x10^{-5}$ to $22x10^{-5}$, respectively. However, a larger range of K is recommended to accommodate the experimental scatter noted in the figure. In this study, the search range for K is set from $4x10^{-5}$ to $40x10^{-5}$ for a measuring range of R_{max} from 4 to 30 μm.

4.2.3. Estimation of R_{max}

The objective for a non-categorized surface is to precisely find its intermediate value of K in order to be subsequently used in estimating R_{max}. The search is accomplished by employing the standard golden section method [BPW79] to progressively reduce the range of K considered from that defined by K_a and K_b. Details are presented in Fig.4.4.

A new subrange, K equal to K_c or K_d, is tested by first using equation (4.11). The two corresponding K_1 from which the R_{max} can be computed are found by employing equation (4.8). The R_{max} are substituted into equation (4.6) to calculate each instantaneous light intensity distribution, $I_c(\theta_1,\theta_2,t)$, whose standard deviation, S_c, can be found conventionally and used to describe the characteristic of scattered light pattern [VT80, BGT85] which, therefore, can be compared to the standard deviation of the measured ASLP, S_m, to assess the difference between $I_c(\theta_1,\theta_2,t)$ and $I_m(\theta_1,\theta_2,t)$ at a given instant, t. Now $I_m(\theta_1,\theta_2,t)$, unlike $I_c(\theta_1,\theta_2,t)$, is not continuous but reflects the discrete nature of the CCD sensor. Consequently, the discrete analogy:

$$S_m = \sqrt{\sum_{j=1}^{N} p_j (j - \bar{j})^2} \quad (4.12)$$

to the conventional standard deviation calculation must be utilized. In equation (4.12), N (equal 3456 in this study) is the number of CCD pixels whilst $\bar{j}$ is the nearest pixel location corresponding to the mean light level [BGT85]. The p_j, on the other hand, is the probability density of the light intensity at the representative jth pixel. As reported in [BGT85], the standard deviation of a scattered light pattern can be used to describe the characteristics of the pattern. Thus, an error, ε, defined as:

$$\varepsilon = \frac{|S_m - Sc|}{S_m} \quad (4.13)$$

is used to indicate the difference between computed and corresponding measured ASLP data. The "c" or "d" is added to the ε shown in Fig.4.4 to denote, by the letter, whether $K=K_c$ or $K=K_d$ is being considered.) The required K is the value which minimizes the ε. In Fig.4.5, for example, the minimum ε occurs at K equal to 15×10^{-5} whose substitution into equation (4.11) enumerates K_1 which, combined with equation (4.8), gives a R_{max} of 14.0 μm. This estimation is very close to the average stylus-measured 14.9 μm.

However, a minimum is never truly attained with computations involving finite length arithmetic. Consequently, a search for K can be terminated when the current width $|K_d - K_c|$ is less than a preset value, ΔK. Moreover, an examination of Fig.4.5 shows that R_{max} is less sensitive to a given variation in K when R_{max} is in the neighbourhood or less than 10 μm.

The algorithm, which combines the least-squares estimation with the iterative search for K, is summarized in the flow chart presented in Fig.4.4. A program, written in FORTRAN, can be found in Appendix B (See Program 2.).

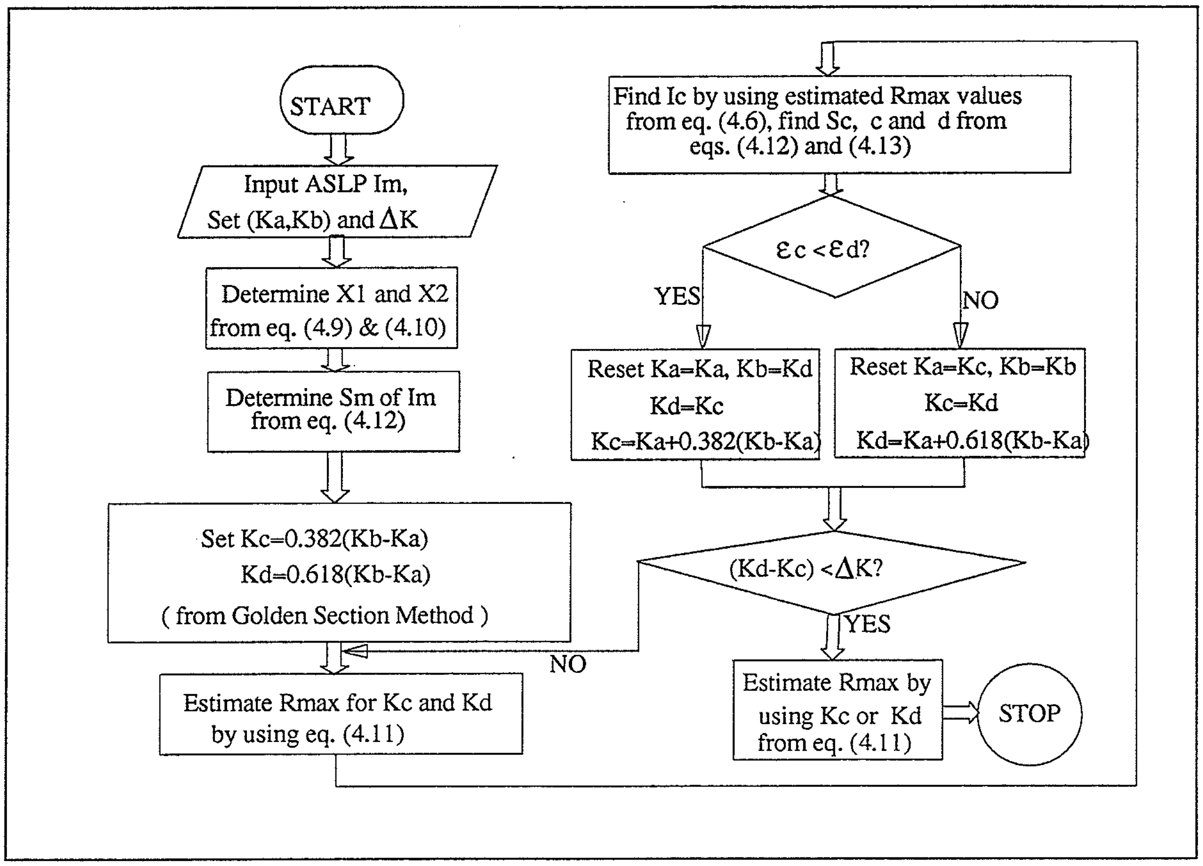

Fig.4.4 Flow chart of the algorithm for the estimation of R_{max}.

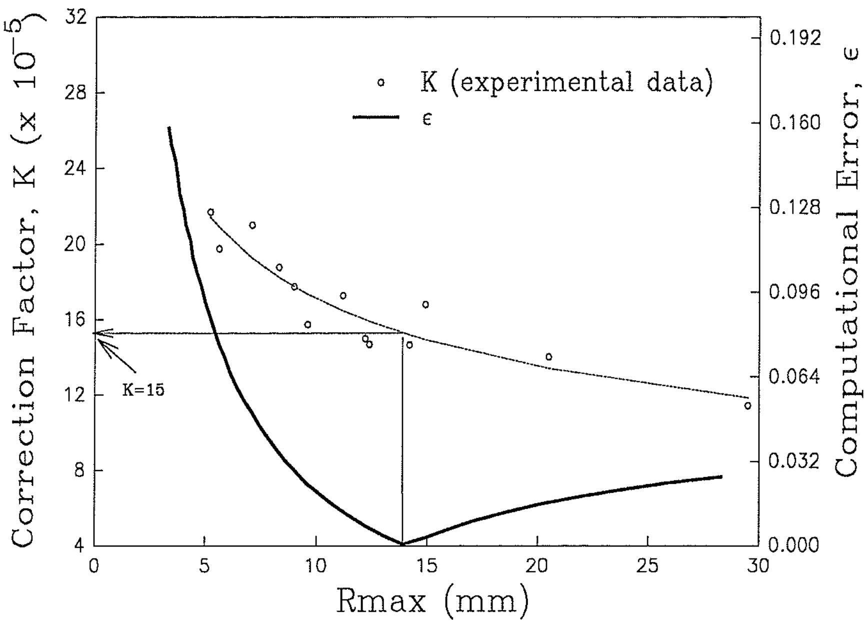

Fig.4.5 Variation of the correction factor, K, and computational error,ε, with the estimated R_{max} for Section 5 of Specimen 1.

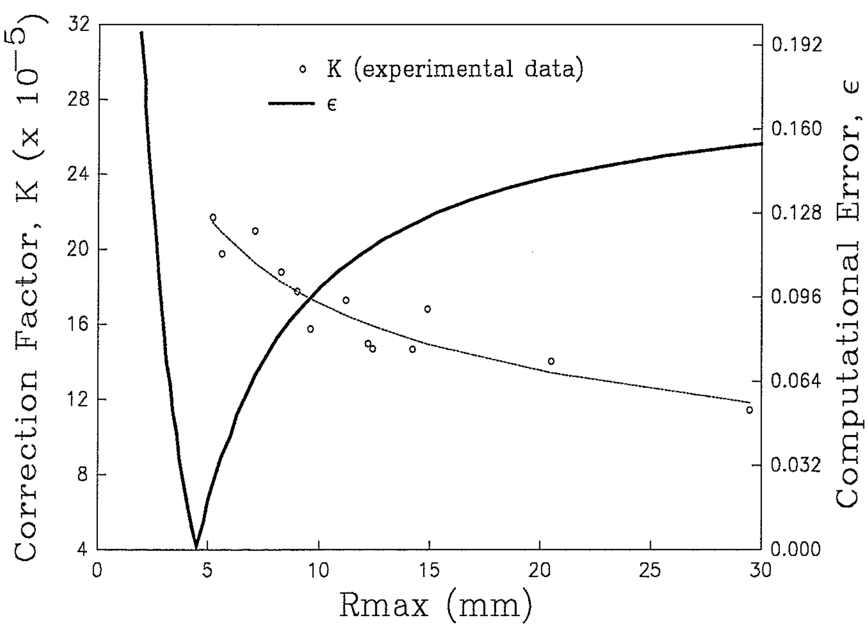

Fig.4.6 Variation of K and ε with the estimated R_{max} for Section 3 of Specimen 1.

Table 4.1 Stylus-measured Roughness of Specimen (μm).

Specimen 1

Section #	Average R_{max}	Range of R_{max}
1	11.8	10.4-14.4
2	18.0	16.4-20.4
3	5.5	5.0-6.2
4	8.3	7.4-9.4
5	14.9	14.2-15.8
6	29.2	28.4-30.2
7	14.2	13.6-15.0

Specimen 2

Section #	Average R_{max}	Range of R_{max}
1	10.5	9.8-11.0
2	10.1	8.6-11.2
3	9.8	8.2-11.2
4	7.4	6.8-9.2
5	11.9	10.4-12.8
6	11.8	11.0-13.6
7	7.5	7.0-8.4
8	11.8	11.0-12.8
9	20.2	20.1-24.2
10	11.4	10.6-12.4
11	9.6	8.4-11.2

Therefore the preset ΔK can be doubled in this region from the normally employed 0.5×10^{-6} to beneficially reduce computational time by around 15%.

Sections 5 and 3 of Specimen 1, whose specifications are given in Table 4.1, were selected arbitrarily to verify the search strategy. The iteratively determined K of 15×10^{-5} and 24×10^{-5} can be seen from Figs.4.5 and 4.6 to produce R_{max} of 14.0 μm and 4.9 μm, respectively. These R_{max} are within a reasonable 90% of corresponding stylus-measured values found by using a Mitutoyo Surftest 401 stylus instrument with a skid on the stylus. A roughness-width cutoff value of 0.8 mm was chosen to meet the CSA standard [CSA62] for stylus measurements. The instrument has two built-in R-C filters to yield a 80% amplitude transferability at the cutoff value [JIS76].

4.3 Optical measuring system

4.3.1. Design objectives

An optical measuring system suitable for on-line use in finish turning should have the following features.

(1) It should be capable of sampling at least the light reflected from spots located along a line in the feed direction in order to be comparable with the standard stylus measurement of surface roughness.

(2) The system must be able to collect the light reflected within a range of angles, relative to the specular direction, to provide adequate ASLP information to quantify R_{max}.

(3) It should produce a small L_o/D ratio, where L_o is the light's spot size on the surface of the workpiece in the feed direction and D is the diameter of the workpiece, to minimize light scatter caused by the workpiece's curvature.

(4) A very high sampling speed should be incorporated for measurements in the feed direction to reduce the effect of lateral scattering of light as a result of the workpiece's rotation.

(5) The sensing system must not interfere with the cutting process.

(6) Measurement accuracy and repeatability must be acceptable.

(7) The system must be insensitive to environmental disturbances (such as vibration, ambient light and temperature variations) and it should be inexpensive as well as easy to operate.

4.3.2 Description of the measuring system

A laser measuring system was developed to incorporate the design objectives listed above. The system is capable of capturing the ASLP reflected from the workpiece's surface at a very high speed. When the turning proceeds, the measuring system assembled on a moving board is traversed together with the tool post at the feedrate. The diagrammatic sketch of the system and an overview of the experimental set-up are shown in Figs.4.7 and 4.8 respectively.

A 0.5 mw He-Ne laser (Uniphase 155SL) was used as the light source to produce a parallel light beam having a diameter of 0.75 mm and a wavelength of 6348 Å. The laser beam was split equally by a beam splitter and aligned on the workpiece's surface just behind the cutting tool. A Texas Instruments, TC104 Charge-Coupled Device (CCD) linear image sensor of 3489 pixels together with its evaluation board was installed beside the lathe, as shown in Fig. 4.7, in an orientation capable of capturing the reflected light from the workpiece. Of the 3,489 pixels, 3,456 were image pixels, eighteen provided a black level reference for the video image, with the remainder acting as isolation pixels to provide a separation between two adjacent sets of ASLP data. The scattered light reflected off the surface and focused on the CCD linear sensor using a collecting lens having a focal length of 85 mm.

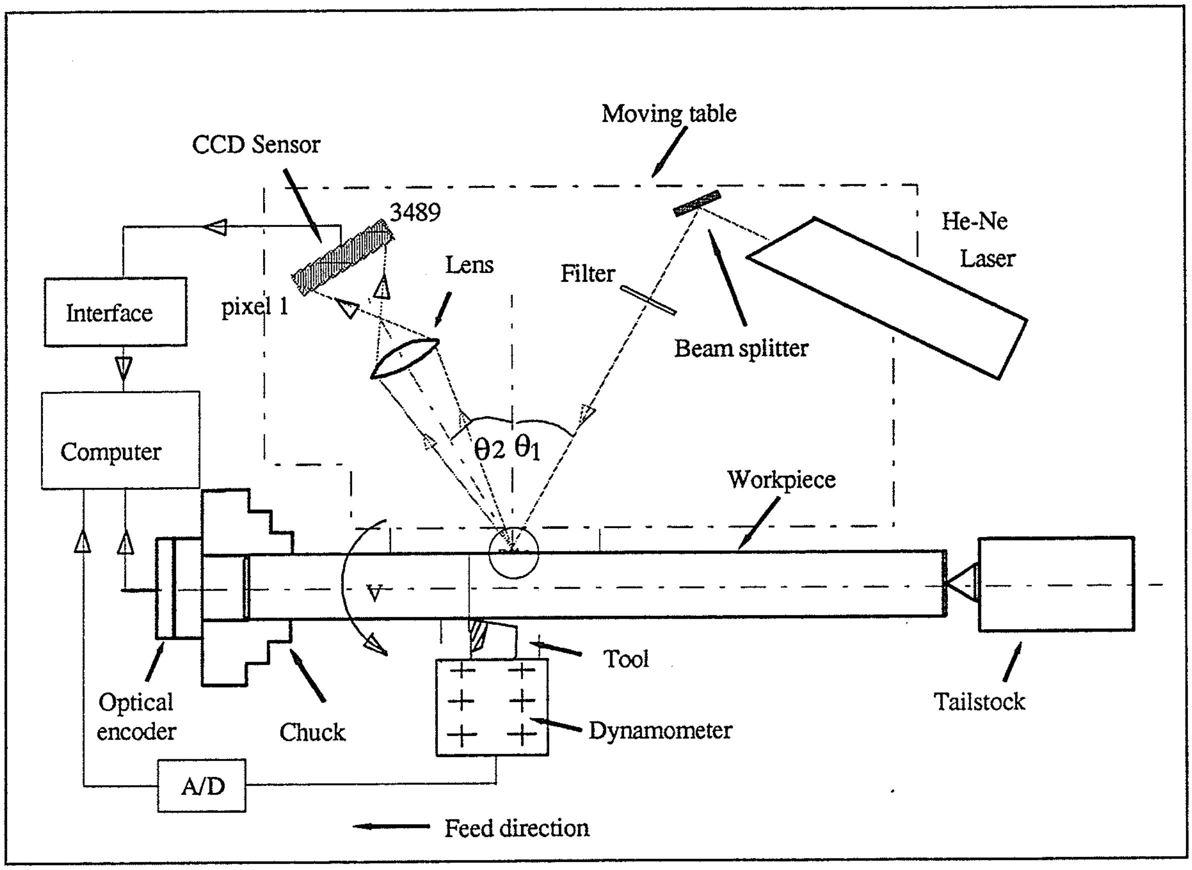

Fig. 4.7 Diagram of the laser measuring system.

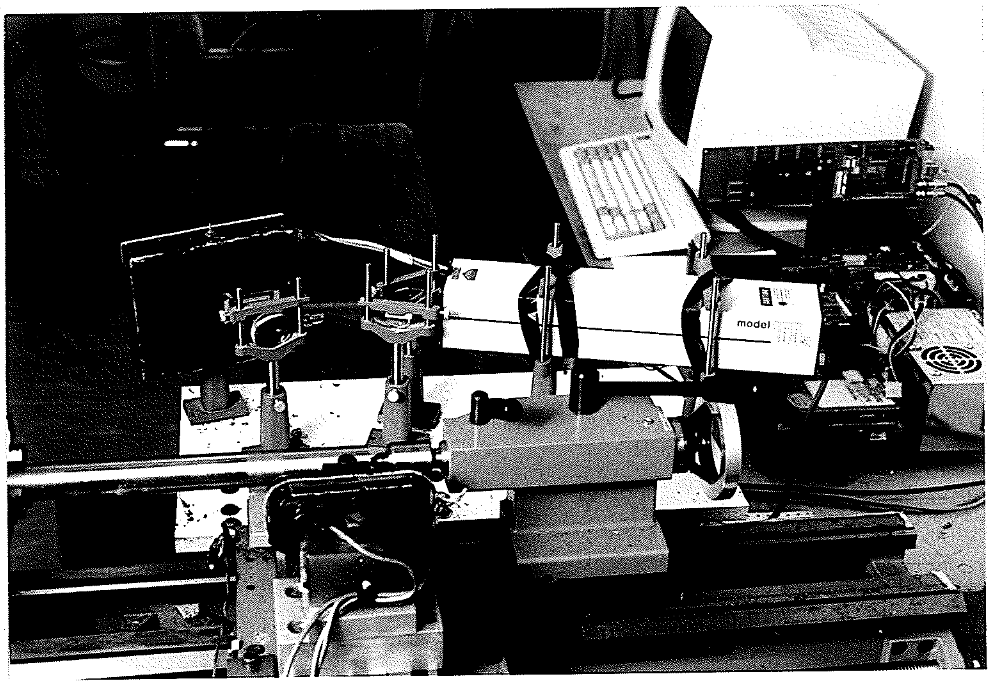

Fig. 4.8 Overview of the experimental set-up.

As stated previously, the CCD mounting mechanism moved at the same rate as the lathe's feedrate with the incident laser beam projected on the workpiece just behind the tool bit. The angle of incidence,θ_1, has a major effect on the scattering pattern. For the roughness range considered, the intensity of the reflected light basically increases with an increase in angle, θ_1 [FH92]. However, a large θ_1 will limit the length of traverse of the measuring system. An angle of incidence of 15^o relative to the normal of the workpiece's surface was chosen to cover the maximum length of workpiece used in this study. The angles of reflection, θ_2, observed at the CCD imagers varied from 10^o to 20^o with respect to the normal of the workpiece's surface. For this setting, the collecting angle of the CCD array was set to cover -5^o to $+5^o$ with respect to the specular reflection. Although a wider collecting angle is preferred, it will increase the possibility of interference with the cutting process. A wide angle lens system will locate the CCD sensor closer to the reflected roughness range of 4 to 20 μm. The central axes of the laser, CCD array and workpiece were adjusted to be on the same horizontal plane as the base of the tool post. The CCD sensor unit was interfaced to electronic logic circuitry to correctly synchronize and drive the trigger signals for continuous operation. An optical encoder, mounted on the axis of the lathe's spindle, produced an index pulse every spindle revolution. Each index pulse was employed by the associated electronics [KBPM90] to externally trigger the CCD sampling circuit.

4.3.3 The CCD linear image sensor

The CCD linear image sensor is used for high-resolution image scanning applications. In this study, the TC104 is used together with its evaluation board for sampling the ASLP. The TC104 is a one-dimensional array of 3,489 discrete detectors called pixels. Each pixel consists of a metal-oxide-semiconductor (MOS) capacitor shown in Fig.4.9.

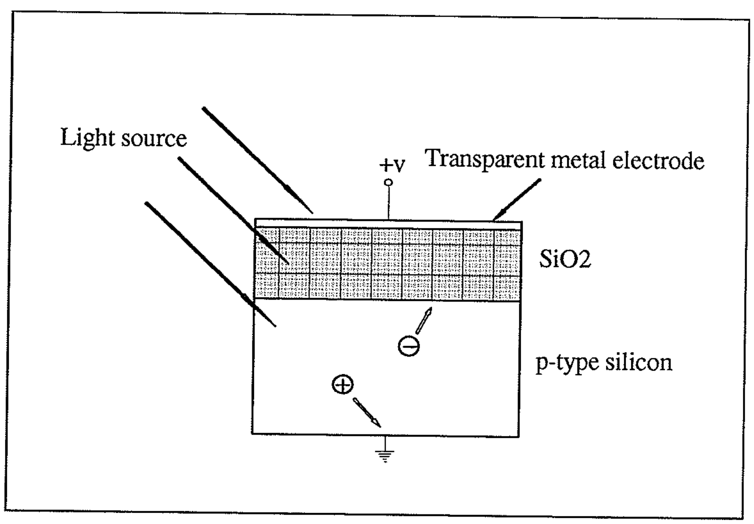

Fig.4.9 Structure of MOS capacitor.

When light falls on the detector, electron-hole pairs become separated and the electrons move towards the positive terminal and accumulate beneath the oxide layer. The amount of charge trapped is proportional to the total light flux falling onto the cell. Information in each pixel is sent to its corresponding register by a clock signal and then transferred serially out of the registers to become the output signal (OS) of the CCD. The functional diagram of the CCD is shown in Fig.4.10.

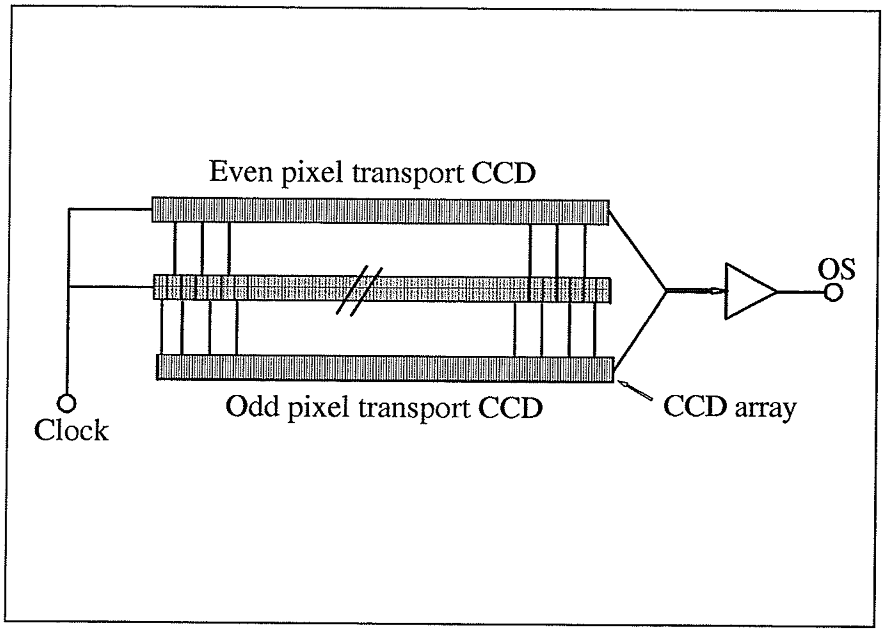

Fig.4.10 Functional diagram of CCD sensor.

4.3.4 The CCD evaluation board

The CCD linear image sensor comes with an evaluation board made by Texas Instruments Corporation and can operate with its own control signals. Four control signals are discussed below [Refer to TI83].

(1) Output signal (OS)

The output signal from the CCD is a pulse train. It contains light information in each pixel and a period of 0.82 μs is required for data of each pixel to transfer. This results in a total of about 3 ms for a complete scan of 3,489 pixels.

(2) Transport clock (TCK)

The TCK runs at half of the data rate (OS). This signal is used to transport the pixel data serially to the output amplifier.

(3) White reference clock (WRCK)

The WRCK runs at the transfer clock frequency and indicates the start of conversion in each scan. The interface uses this signal to determine when to start the conversion. However, a biased voltage is present in the WRCK. Signal conditioning must be provided to reduce the bias to zero.

(4) Reset clock (RCK)

The RCK operates at the same frequency as the OS. This signal is used to synchronize the operations between the evaluation board and the data acquisition system. Signal conditioning is also required for the biased voltage.

A detailed description of the test procedure and requirements for setting up the evaluation board can be found in Appendix D.

4.3.5 Electronic interface for data acquisition

A custom designed, electronic interface [KYPBT93] (Fig.4.11) was built to sample, convert, store and transfer the data from the CCD to the main memory of an IBM compatible computer. This interface samples the data from the CCD sensor at 1.2 MHz and stores it in a static RAM of the interface. An 8-bit, high speed, analog-to-digital converter (Analog Device 7831), coupled with a RCA CMOS 6382 chip having 8kx1 static RAM, digitizes and stores the signals. The A/D converter operates in the read mode (RD) and has a sampling frequency of 1.2 MHz, which is the same as the OS of the CCD. Data is read at bus speed (baud rate: 52 Kbytes) into the computer. The built-in filtering network gives a signal-to-noise ratio of about +30 dB which was found adequate to avoid noise amplification caused by the logarithmic transformation associated with equation (4.7). The block diagram given in Fig.4.12 shows the complete electronic interface.

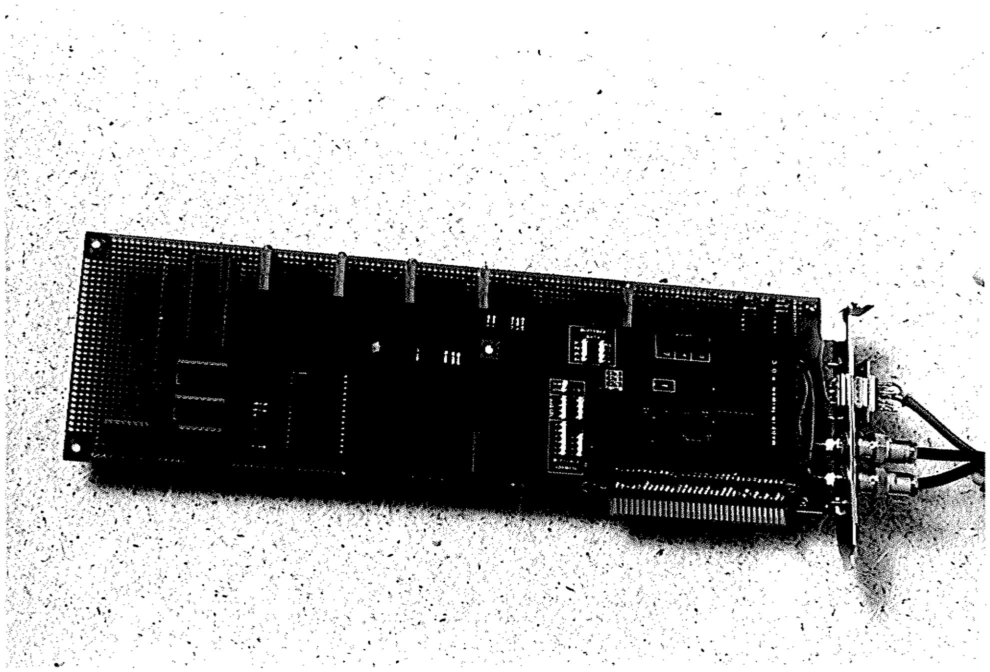

Fig. 4.11 Electronic interface.

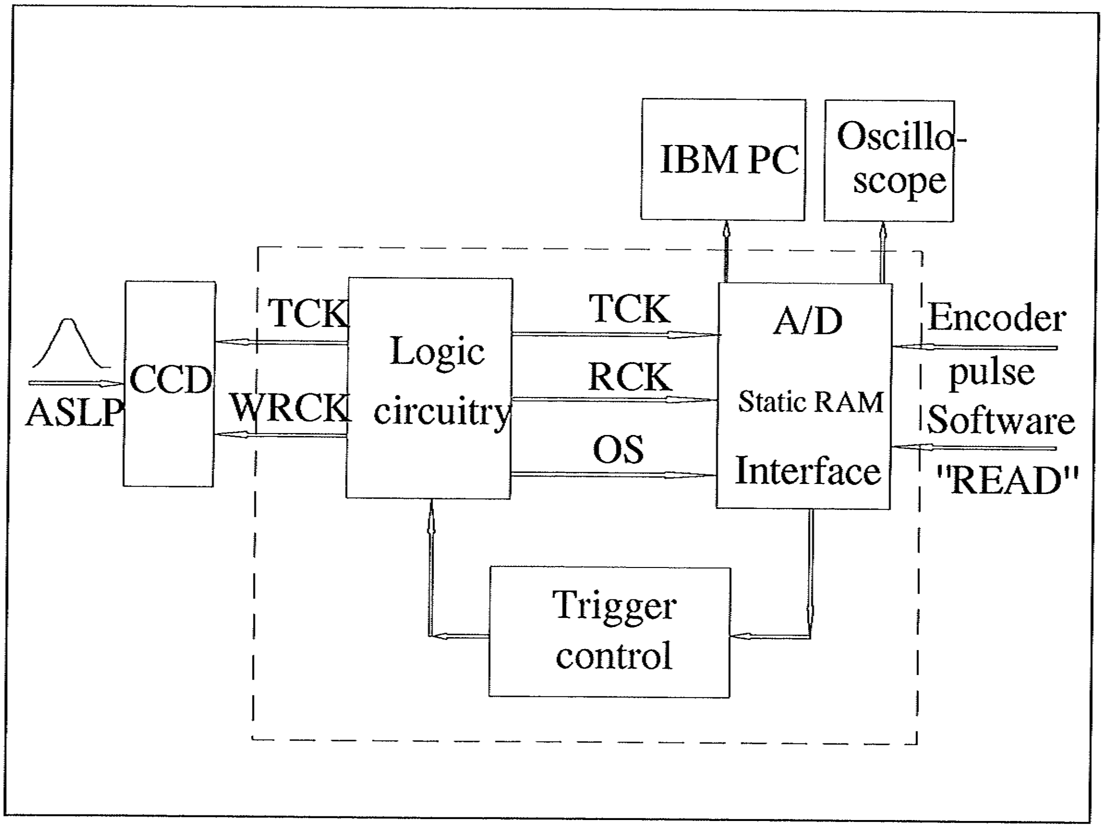

Fig.4.12 Block diagram of electronic interface (in the dashed box).

To correlate the data measured by the laser system with the conventional stylus measurements, modifications had to be made to the interface circuitry. It was desirable to have the electronics trigger the CCD at precisely the same circumferential location on the workpiece's surface so that the ASLP is sampled along a line on that surface for comparison with the stylus measurements. The selective triggering was provided by the once per revolution index pulse from an optical encoder mechanically mounted on the lathe spindle. This scheme ensured that the sampled ASLP data was captured, like that of a stylus instrument, along a longitudinal line on the workpiece's surface. (However, the optical system can be adapted easily to any desired sampling strategy, for example sampling along the circumferential direction by changing the selective triggering.)

The heart of the electronic design is shown in Fig.4.13. Each pixel level is sampled by the A/D converter in synchronization with the falling edge of the RCK (Reset Clock) pulse from the CCD. The A/D input was referenced at +5 volts and, hence, it was necessary to invert and level shift the output signal (OS) of the CCD to have a full scale of 0 to +5 volts (corresponding to the 256 levels, 0 to 255, of the CCD) as shown in Fig. 4.14.

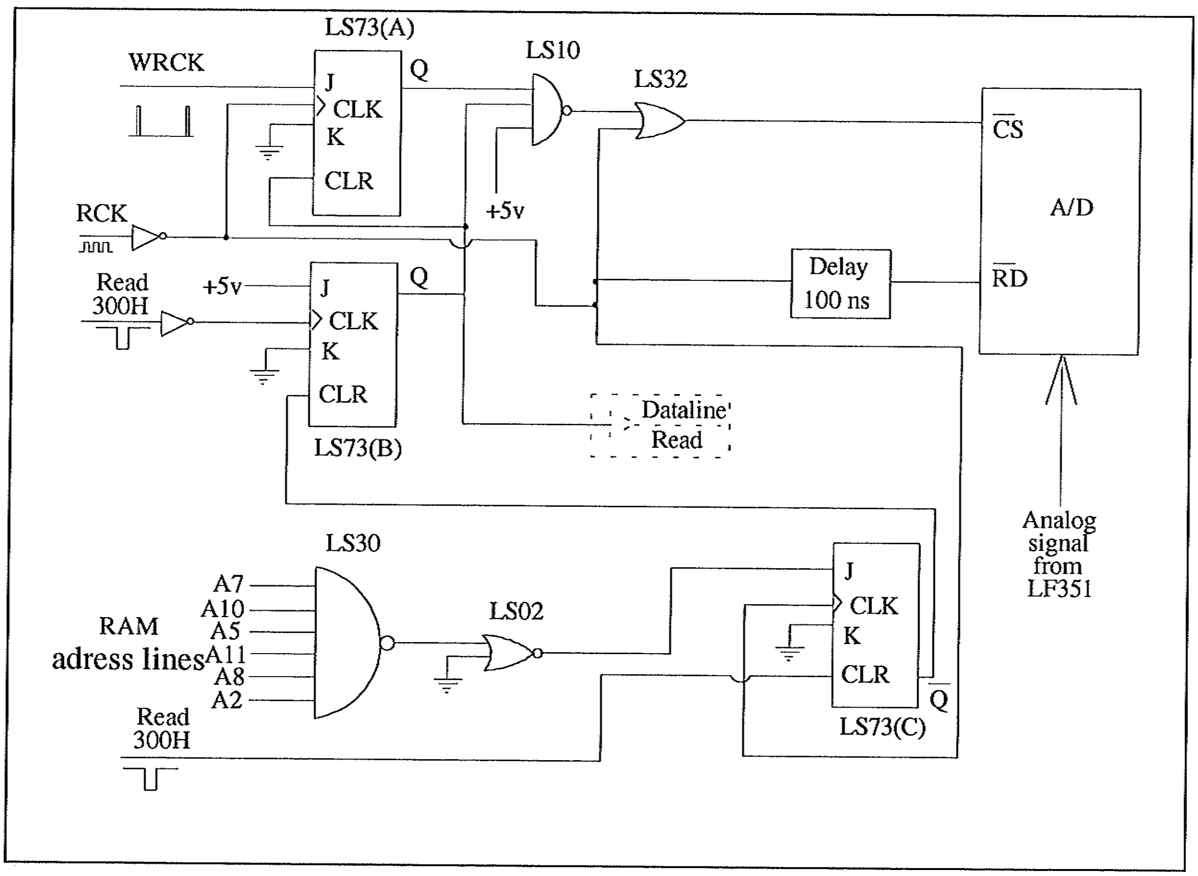

Fig.4.13 A/D electronic circuitry for CCD.

The key point to the successful capture of a *particular* image pixel array in the correct sequence(1-3,489), is the synchronizing of a WRCK pulse (White Reference Clock pulse indicating the *start* of the image measurement) and the immediate low setting of the

A/D chip select with the falling edge of a RCK pulse. The sequence of operations to accomplish this task begins with the arrival of the Read pulse at address 300H. The J-K flip-flops, LS73(B) and (C), are not triggered until the Read pulse appears at the Clock and Clear inputs, respectively. Then $\overline{Q}$ of LS73(C) sets the Clear high on LS73(B) which is triggered by the same Read pulse on its Clock input. The LS10 NAND gate only requires the Q output of LS73(A) to go high upon the arrival of a WRCK pulse which indicates that a burst of image pixel data is to follow. Chip Select of the A/D converter then goes low in sequence with the falling edge of each RCK pulse. As each pixel is sampled by the A/D, a counter is incremented and the bit outputs are connected indirectly to a 6 input NAND gate which goes low when the count reaches DA1H(3489) terminating the A/D conversions. If multiple WRCK pulses trigger the LS73(A), due to incorrect triggering of the CCD by an index pulse of the encoder, the A/D output will not be in the correct sequence and cannot be used.

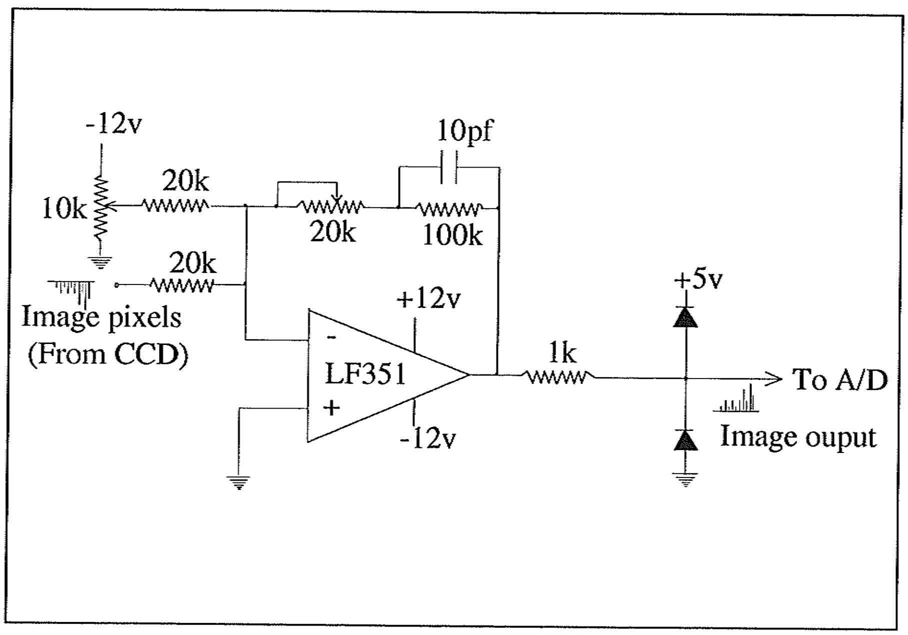

Fig.4.14 Output inversion and level adjustment circuitry for CCD.

To ensure that the CCD is triggered at the selected time and thereby over-ride continuous CCD triggering, the encoder pulse is synchronized with the rising edge of a TCK (transport clock) pulse. This clock transports the linear image signal's charges from the sensor elements (pixels) to the output amplifier. The transport clock's pulse width is very much smaller than the lone pulse width, t_1, of the encoder or read pulse. Therefore, it is necessary to lock out all subsequent transport pulses after one rising edge is detected by ensuring that $t_2 > t_1$, where t_2 is the exposure pulse width as set by a resistive potentiometer on the evaluation board. Otherwise multiple CCD triggering occurs. The lock out is accomplished by using a D-type flip-flop and introducing the exposure time pulse from the evaluation board into the circuitry at the preset input. The circuit and timing is shown in Fig.4.15.

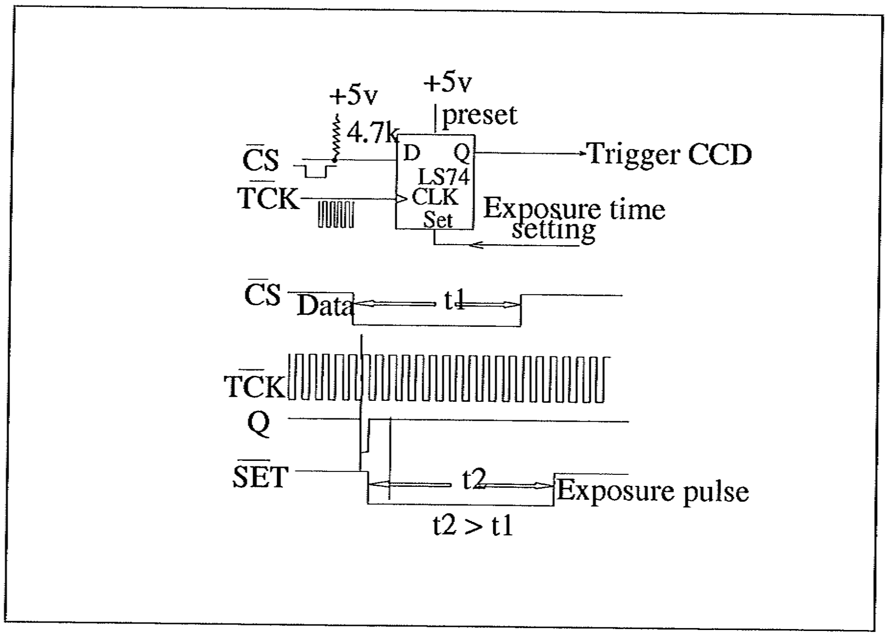

Fig.4.15 Transport clock synchronization, lockout circuit and timing diagram.

4.4 Experimental studies and results

The laser measuring system described in the last section was utilized to : (i) investigate the influence of the cutting fluid on the measured ASLP data; (ii) study the influence of the ambient room light as well as the rotational speed of the workpiece on the measured ASLP data, and (iii) compare the on-line measurements obtained from the optical system under different cutting conditions with those found from stylus measurements. Details are given next.

4.4.1 Influence of a cutting fluid

A thin film of fluid is expected to be introduced on the workpiece's surface when a cutting fluid is used. This covering film functions to reduce the peak-to-valley roughness of the surface. To investigate its effect on the measurement, an experimental investigation was conducted. The ASLP data, reflected from a rotating finish-turned surface, was first sampled without using any cutting fluid. Then two types of cutting fluid were sprayed on the surface and the two ASLP data were sampled, respectively. All three ASLP data are shown in Fig.4.16. It can be seen from the figure that the measured ASLPs appear different due to the use of the fluid. A distinct specular reflection is present in the ASLP sampled when using an oil-base cutting fluid (DoALL #150, medium-low viscosity cutting oil). On the other hand, the ASLP data sampled using a water-based fluid (ESSO #45 Kutwell cutting oil, 50% diluted with water) appears a little bit lower than that without using any cutting fluid. Consequently, these differences in the ASLP will affect

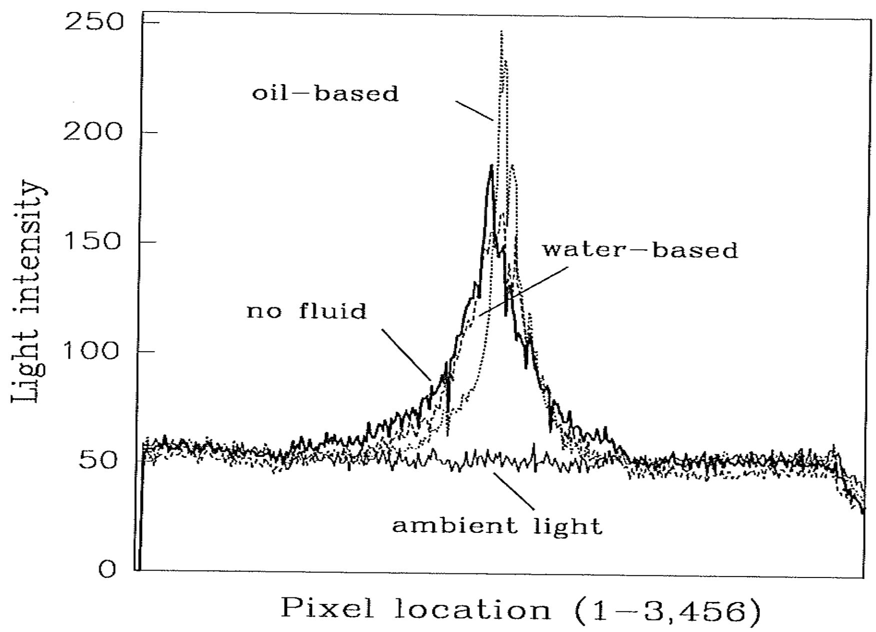

Fig.4.16 Influence of a cutting fluid on the measured ASLP.

the accuracy of the estimated R_{max} as given in Table 4.2. It can be seen that the estimated R_{max} from the ASLP data without using any cutting fluid is very close to the stylus measurement in comparison to the lower R_{max} obtained when the oil-based cutting fluid was used. Therefore, the effect of the viscosity of the cutting fluid on the ASLP is significant and needs further investigation. In the present study, no cutting fluid is used .

Table 4.2 Effect of a Cutting Fluid on the Estimation of R_{max}.

Case	Dry cutting	Water-based (ESSO #45 Kutwell)	Oil-based (DoALL #150)	Stylus data Range	Average
R_{max} (µm)	4.5	5.0	2.8	3.8--5.6	4.8

4.4.2 Influence of the ambient light

Optical measurements can be expected to be influenced by the level of the ambient light in the room enclosing the measured surface. To verify this hypothesis, the effect of ambient light changes observed during a working day was investigated. The changes are reported in column 2 of Table 4.3. For the 8-bit interface used, levels 0 and 255 (arbitrary unit) represent the output of the CCD sensor when exposed to totally dark and bright light conditions, respectively. The sensor is saturated for levels above 255. The effect on a mid-valued R_{max} of 7.1 ± 1.3 μm can be seen from column 3 of Table 4.3 to be significant. Consequently, the following versatile compensation strategy should be adopted.

Table 4.3 Effect of Ambient Light Level on the Estimation of R_{max}.

Test #	Ambient light level (0-255)	Uncompensated R_{max} (μm)	Compensated R_{max} (μm)
1	21	3.3	5.0
2	31	5.9	6.4
3	38	7.2	7.0
4	42	7.6	6.9
5	55	11.2	7.9

A reference ambient light level of 38 was chosen at which the optically-determined R_{max} and that measured by the stylus instrument were almost equal. It was found that an ambient light change from the reference level merely produced a dc offset proportional to the difference between the actual ambient light intensity and the reference level. Hence, a straightforward compensation strategy was to simply subtract this offset from the measured ASLP before data was analyzed by using the least-squares procedure. The ambient light compensated R_{max} values shown in column 4 of Table 4.3 compare much better than given in column 3. This improved agreement demonstrates the effectiveness of the compensation technique.

4.4.3 Influence of the workpiece's rotational speed

It is well known [TI83] that the charge of a CCD pixel increases linearly with time for a given incident light intensity within the CCD's operating range. Now, the sampling of the CCD is triggered externally by an index pulse from the optical encoder of every spindle revolution. The corresponding duration that each pixel is exposed to light is controlled by the period between two consecutive index pulses. This period is inversely proportional to the rotational speed of the spindle and, hence, changes when the speed is altered. This behavior is illustrated in Fig.4.17. Two different speeds of n_1 and n_2 (rpm) are used with $n_1 > n_2$. This means that the exposure duration at n_1 is less than that at n_2. Therefore, a pixel's charge is modified so that the resulting ASLP data sampled at n_1 can be expected to be lower than that at n_2, as shown in Fig.4.17. Consequently, a given roughness measured at one spindle speed will not be identical to that obtained at a different speed. To remedy this discrepancy, a compensation strategy was adopted as shown in Fig.4.17. A second pulse (a software READ) following the encoder index pulse was introduced in order to again trigger the CCD. A small delay D1 (D1 > 0 to allow for the settling time after the final A/D conversion) between the completion of the transfer of the first 3,489 pixels and the READ pulse was required to minimize the effect of varying exposure times. This predetermined delay was found to be about 3.75 μs by trial and error and was critical in limiting the effective CCD charge time in order to make the image's response independent of the spindle speed. This value of D1 was controlled by a count loop in the software. At the highest lathe speed of 2,320 rpm,

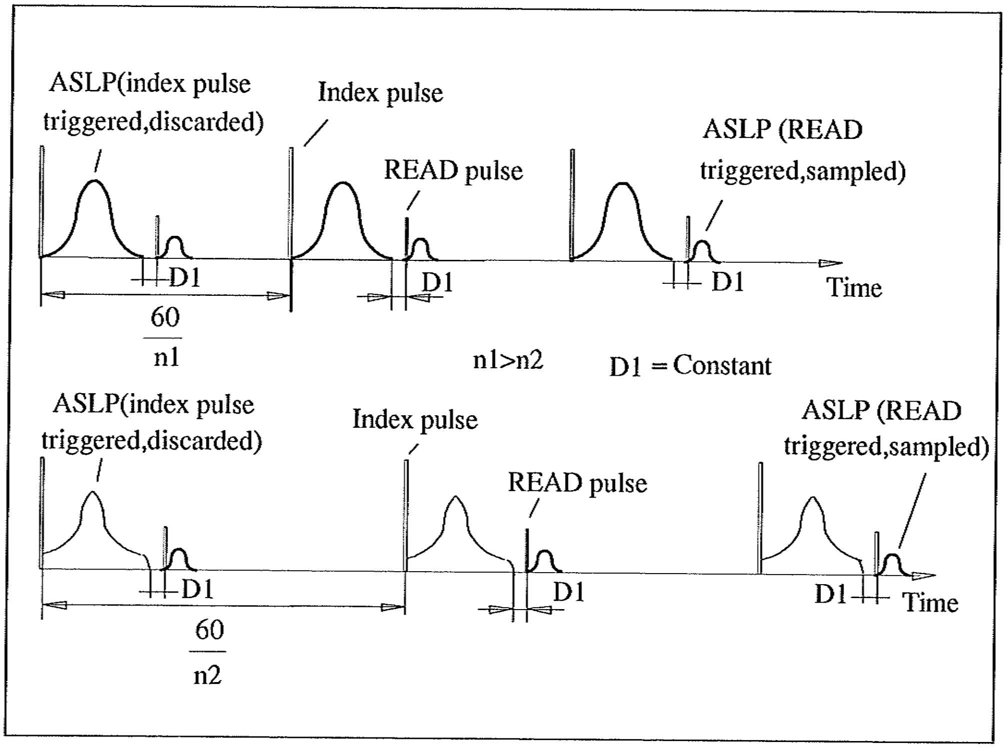

Fig.4.17 Strategy to compensate for the effect of the workpiece's rotational speed.

the real-time delay represented a small but acceptable shift of about 9 mm. on the workpiece's circumference. The electronics rejects the first ASLP snapshot, which in many cases showed saturated pixels due to a lengthy exposure time. As shown in Fig.4.17, the second selected CCD image following this rejected data is sampled and stored for analysis. To distinguish between the encoder index pulse and the READ pulse, a polling loop waits for the arrival of an encoder pulse and only triggers the CCD once it has been latched. A second polling loop waits for the completion of the first transfer of 3,489 pixels and discards the data. Subsequently, the READ pulse is initiated after the delay D1 by using a software "Read" followed by the sampling and storing routines which capture the data. Fig.4.18(a) shows the measured ASLP when the specimen considered in the previous section was rotated at 660, 1,156 as well as 2,320 rpm, respectively. Although the peak of the ASLP remains almost unchanged, the spread at 660 rpm is increased by about 27%

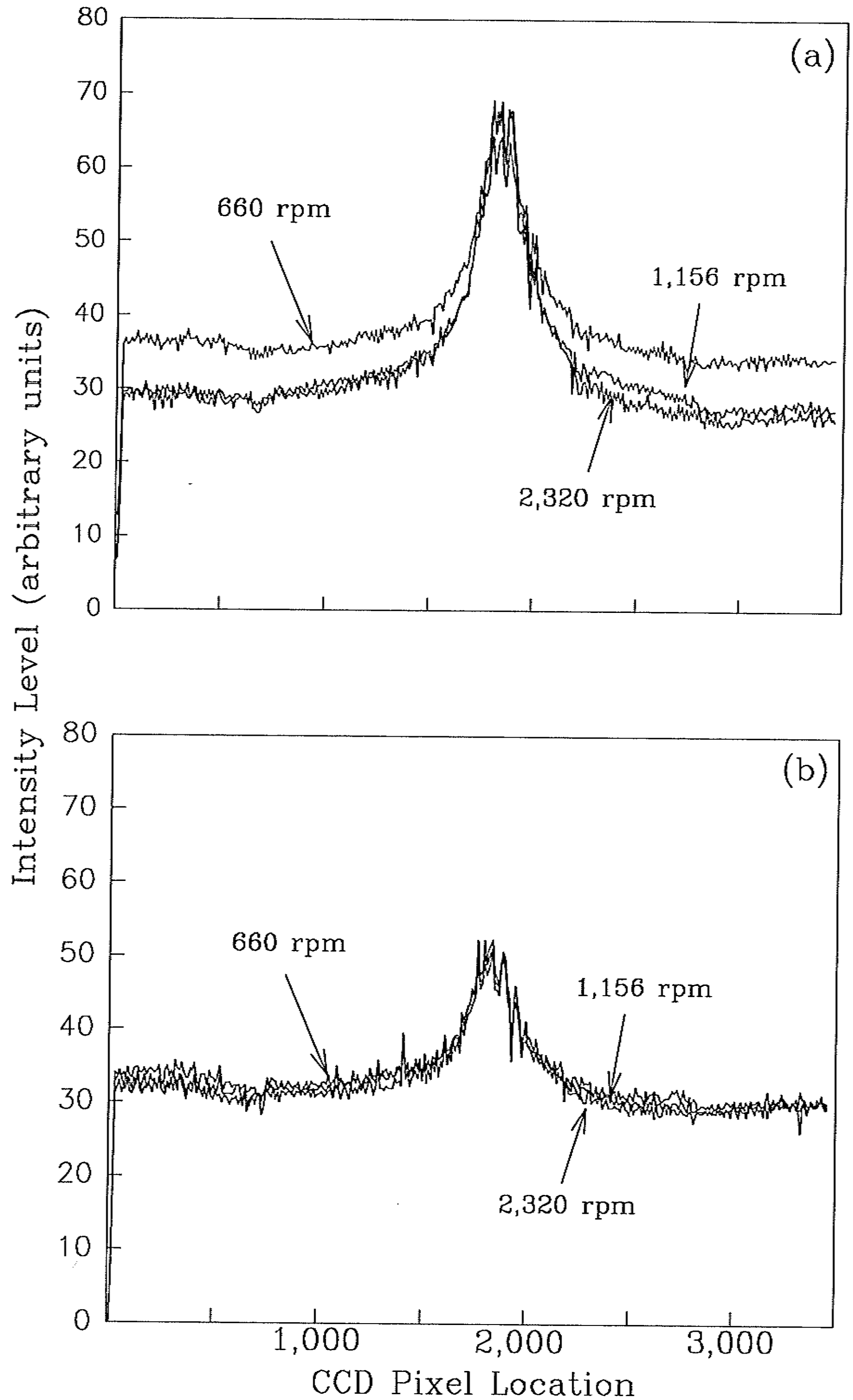

Fig.4.18 Sampled ASLP from Section 5 of Specimen 1 triggered by:
(a) the index pulse but discarded at 660, 1,156 and 2,320 rpm; and
(b) the software READ and recorded at 660, 1,156 and 2,320 rpm.

and 35% compared to that observed at 1,156 and 2,320 rpm. The resulting analogous data to Fig.4.18(a) is presented in Fig.4.18(b) and can be seen to be completely independent of the change in the spindle speed.

4.4.4 On-line measurements

To verify the strategy proposed in the last section, on-line measurements were first obtained from two specimens prepared to have known R_{max} values. There was no cutting involved. An initial check was performed to make sure that the surface profile does not contain any waviness, as defined in [B84]. A specimen was rotated at 660 rpm whilst the laser spot was traversed longitudinally from one end of the specimen to the other. A total of eighteen equally spaced, optical "snap-shots" were captured from each section of the specimen. The test

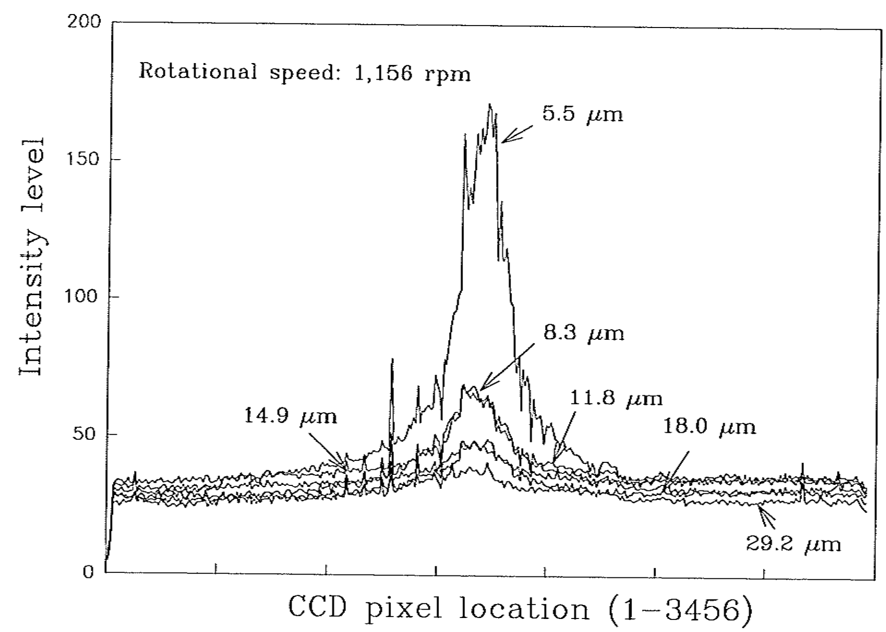

Fig.4.19 Comparison of the sampled ASLP from six finish-turned surfaces having different R_{max}.

was repeated but with the specimen rotating at 1,156 rpm and the same procedure was duplicated for another specimen. Shown in Fig.4.19 are the ASLP data sampled from six sections of Specimen 1 rotating at 1,156 rpm. A distinct difference in the ASLP can be observed. A typical comparison between the optical measurements at 1,156 rpm and stylus-measured R_{max} is given in Fig.4.20. Both the optically-determined and stylus-measured R_{max} were averaged within each section. Good correlation can be seen. Moreover, in comparison to the stylus-measured data, which shows variations as high as 25% relative to the average value (see column 3 of Table 4.1), most of the optically-determined R_{max} values for the measured surfaces are within the given ranges of the stylus data.

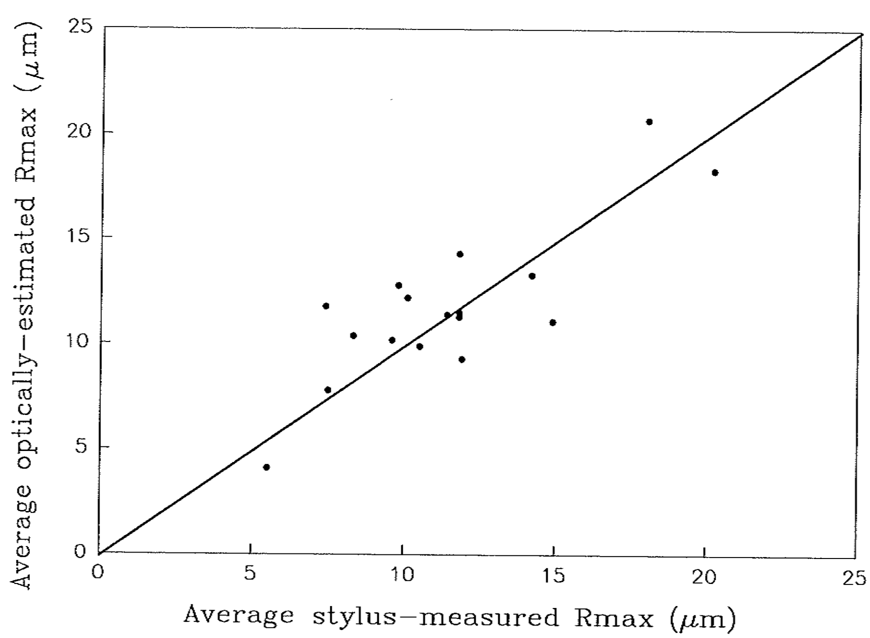

Fig.4.20 Correlation between the optically-determined and stylus-measured R_{max} .

Further on-line measurements were made to evaluate the performance of the laser system under actual cutting environments. The five different cutting conditions listed in Table 4.4 were used. The heat-treated 4340 SPS steel workpieces had a diameter range from 28.1 to 39.7 mm with a whole length of 300 mm. Finish turning was performed without a cutting fluid by using a BR6C6 carbide tool having the designation of 7,7,7,7,75,15, 0.40 according to the American Standards Association [S89].The laser spot on the workpiece's surface was located diametrically opposite the tool-bit but with a constant offset of 5 mm to enable the spot to closely follow the cutting tool. During each cut, a total of thirty-six equally distributed, optical "snap-shots" were taken on a cutting length ranged from 100 to 150 mm. The tool's nose radius was determined by averaging the values measured before and after a cut. A Nikon Epiphot-Time optical microscope producing a

Table 4.4 Cutting Conditions Used for On-line Measurements.

Cut #	V (m/min)	f_o (mm/rev)	d_o (mm)	H (Rc)	T_m (min)
1	80.0	0.0457	0.38	34	3.31
2	66.4	0.0810	0.20	34	2.80
3	102.0	0.0457	0.25	32	2.83
4	144.0	0.0601	0.38	36	2.15
5	102.0	0.0810	0.25	32	1.60

Tool's nose radius ρ (mm): 0.40 .

T_m - cutting duration.

the tool's nose radius. The shape of the tool's nose radius for cutting test 4 (See Table 4.4) is shown in Fig. 4.21. It can be seen from the figure that the contact area on the cutting edge is within the tool's nose corner due to a very light feedrate which features finish turning. A uniform frontal wear on tool's nose radius can be observed.

From the measured ASLPs, an estimate of R_{max} was obtained at each of the illuminated spots. For comparison, a total of 36 stylus measurements were also made on each machined surface after the workpiece was removed from the lathe. A point-to-point comparison between the optically-determined and stylus-measured R_{max} is shown in Fig.4.22 for Test 5 obtained during cutting. For the cutting conditions listed in Table 4.4, the stylus-measured R_{max} ranged from 7.0 to 12.2 μm and the optically-determined R_{max}

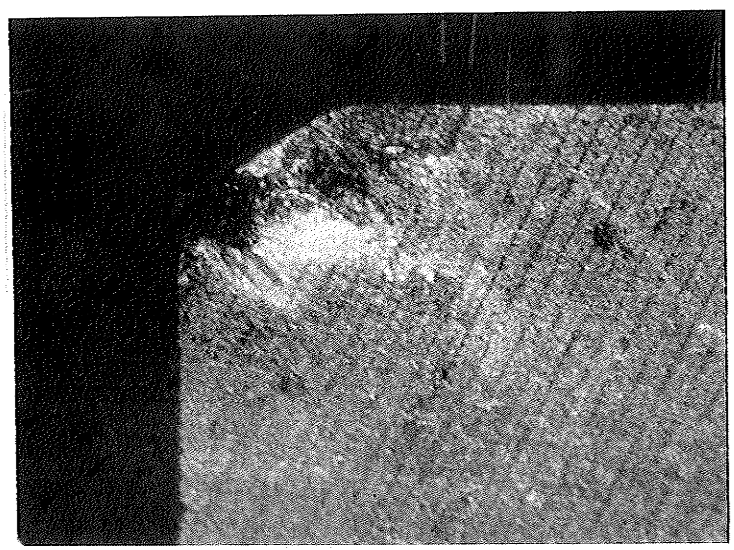

Fig. 4.21 Shape of the tool's nose radius observed, after a 2.15 min. cutting period (v=144.0 m/min, f_o=0.0601 mm/rev, d_o=0.38 mm).

basically behaved similarly. However, several large differences can be observed in Fig.4.22. An optical value represents an estimation of the local maximum peak-to-valley roughness over a small illuminated area defined by its diameter L_0. The stylus measurement, conversely, gives the maximum peak-to-valley roughness within a sampling length which is much larger than the illuminated diameter. Therefore, more marked fluctuations are produced by the stylus measurements. This inherent difference suggests that a comparison based on the average R_{max} is more reasonable. Such a comparison is given in Table 4.5 for all five cuts. It can be seen that the difference between the two techniques is within a reasonable 15% under the cutting conditions used.

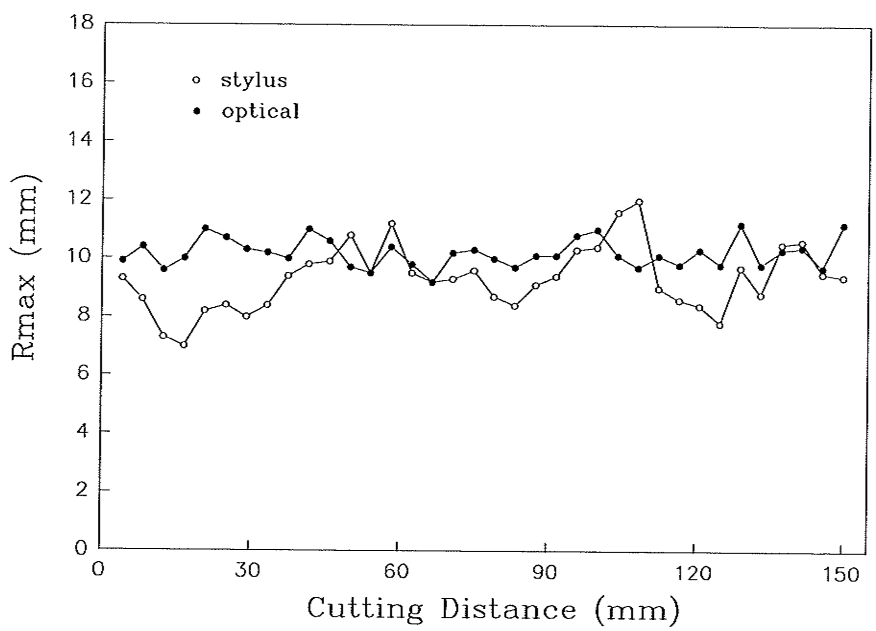

Fig. 4.22 A point-to-point comparison between the optically-determined and stylus-measured R_{max} for cutting test 5.

Table 4. 5 Comparison between Stylus and Optical Data for On-line measurements.

Cut #	Range of stylus R_{max} (μm)	Average R_{max} (μm) stylus	optical	* %
1	6.2 - 11.6	8.1	7.0	13.6
2	8.8 - 18.2	12.1	12.3	1.9
3	5.6 - 11.4	7.5	7.4	1.3
4	9.0 - 14.4	11.8	11.9	1.0
5	7.0 - 12.2	9.4	10.2	8.5

$$* \% = \frac{|\text{Average stylus } R_{max} - \text{Average optical } R_{max}|}{\text{Average stylus } R_{max}} \times 100\%$$

4.5 Remarks

The ASLP contains information about T/σ_z so that it can be used as a measure of the surface roughness. The CCD-based laser system developed in this chapter is capable of capturing, on-line, the Angular Scattered Light Pattern (ASLP) in the feed direction. The R_{max} is quantified by solving the inverse problem of light scattering. The results from on-line measurements shows that the measuring range has been extended to about 20 μm and the maximum difference between the average optically-determined and standard off-line measured R_{max} with a stylus instrument was found to be less than 15%. Moreover, the influences of the ambient light and spindle speed changes on the optical data have also been investigated and minimized. However, the approach involves an iterative search for a measurement correction factor, K which is required to fit the model to a turned surface. Although no *a priori* knowledge of its value is needed, the computational time required to obtain one reading was found to be about 75 seconds when using an IBM/PC 486 computer. This measurement speed is not suitable for application in a real-time environment. It is also found that an oil-based cutting fluid has a significant influence on the measured ASLP data which, in turn, results in a large discrepancy in the estimation of R_{max}. This influence needs further investigation.

CHAPTER 5

REAL-TIME MEASUREMENT USING NEURAL NETWORKS

5.1 Introduction

The optical approach proposed in Chapter Four can be employed to quantify, on-line, the value of $R_{max}(t)$ of a finish-turned surface by solving the inverse problem of light scattering. Although the approach is accurate if the tool's nose radius is known at the sampling time t, it is not sufficiently fast for real-time measurements because of the iterative search for the correction factor, K, performed sequentially by a von Neumann computer. However, it is possible to develop an approach that allows the turning operation to be optimized while maintaining a desired surface finish by adjusting the cutting variables *if* the surface roughness information is available *in real-time*. Therefore, developing such an approach which is accurate, reliable, non-destructive, and most importantly, fast enough for real-time measurements, will be explored in this chapter.

The $I(\theta_1,\theta_2,t)$ in equation (4.6) represents a pattern of scattered light at time t as shown in Fig.4.19. Hence, the measurement of $R_{max}(t)$, is analogous to recognizing the different patterns of the sampled ASLP data which correspond to various $I(\theta_1,\theta_2,t)$. Using neural networks is particularly suitable for this type of pattern recognition problem. As an electronic simulation of the biological architecture of the human brain, neural networks simultaneously explore many competing hypotheses by using parallel networks composed typically of analog, nonlinear computational elements connected by links having variable weights [L87]. A network's parallel nature enables it to compute much faster than traditional serial computing. Moreover, a network can adaptively infer subtle relationships even from incomplete data contaminated by noise. All these advantages make the application of neural networks to a pattern recognition problem in a real-time environment appealing.

5.1.1 Objectives

The objective of this chapter is to explore the feasibility of applying a neural network to determine the R_{max} generated during finish turning. The key issue here is to guarantee that the time required for data processing must be less than that required for sampling data. A three-layer, back-propagation neural network is constructed in conjunction with a back-propagation algorithm to predict R_{max} by quickly recognizing the ASLP data, sampled by the laser measuring system developed in the previous chapter. The performance of the new approach will be evaluated in a real turning environment.

5.2 Neural networks

5.2.1 What are neural networks?

Neural networks go by many names such as artificial neural nets, connectional models, parallel distributed processing models, neuromorphic systems. All these models are an implementation of an algorithm, inspired by research into the human nervous system, in which computers are used to achieve good performance through dense interconnections of simple computational elements. Neural networks act on data by detecting some kind of underlying organization, thereby assisting in classification, function estimation, pattern recognition, and similar tasks. Neural networks have emerged as a powerful technique for pattern recognition because they offer valuable characteristics unavailable elsewhere.

(1) Neural networks are adaptive or they can learn. They are able to infer solutions from data without prior knowledge of the relationships in the data. They extract the regularities empirically. Adaptation also provides a degree of robustness by compensating for minor variabilities in characteristics of processing elements. As a result, gathering rather than explaining data is the prime consideration. Conventional programming is also not needed.

(2) Neural networks can generalize, meaning they are able to respond correctly on the data which are broadly similar to the original training patterns. Generalization is useful because real-world data is noisy, distorted, and often incomplete. They provide a great degree of robustness or fault tolerance.

(3) Neural networks are nonlinear, or they can make weaker assumptions concerning the shapes of underlying distributions than traditional statistical classifier [L87], in that they are able to capture complex interactions among the input variable. Nonlinear behavior is common but difficult to handle mathematically.

(4) Neural networks are highly parallel. They explore many competing hypotheses simultaneously using massively parallel nets composed of many computational elements connected by links with variable weights. These identical, independent operations can be executed much faster than alternative methods.

5.2.2 The perceptron

A perceptron is the building block of a multi-layered network. One of the simplest perceptron sums N_l weighted inputs and adds a threshold value called bias, w_o, to form a weighted sum. The result is then passed through a nonlinearity, like those shown in Fig.5.1, to give an output as: $O = f(\sum_{i=1}^{N_l - 1} w_i X_i - \text{bias})$. More complex perceptrons may include other mathematical operations than the summation. The perceptron is characterized by a threshold and by the type of the nonlinearity. There are three common types of nonlinearities: hard limiters, threshold logic, and the sigmoid which is the most popular.

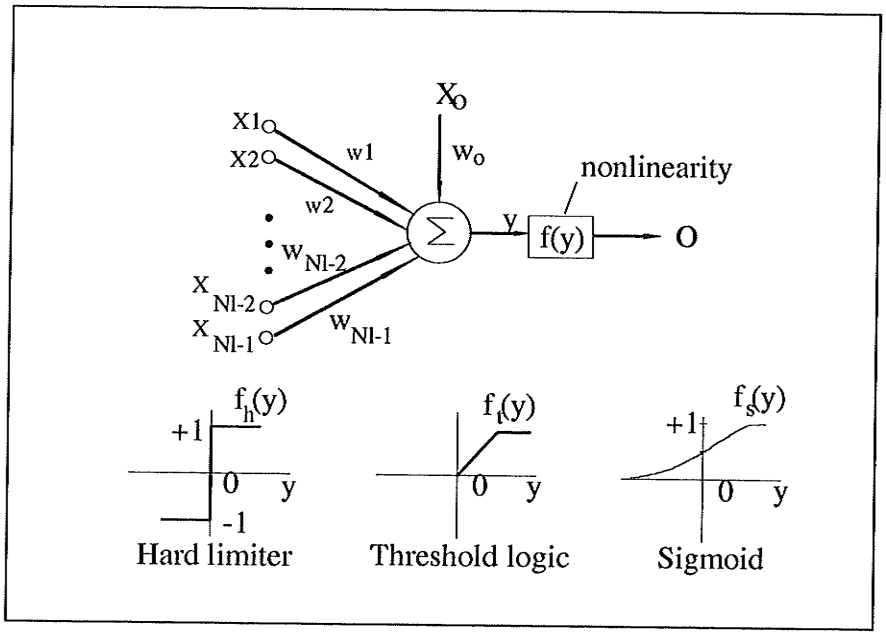

Fig.5.1 A perceptron which forms a weighted sum of N_l inputs and passes the result through a nonlinearity.

When the perceptrons are cascaded together in layers, it is more common to use the sigmoid nonlinearity, $f_s(y)$, given by:

$$f_s(y) = \frac{1}{(1+e^{-\tau y})}. \tag{5.1}$$

This function is continuous and varies monotonically from 0 to 1 as y varies from -∞ to +∞. The gain, τ, of the sigmoid function determines the steepness of the transition region. Note that, as the gain approaches infinity, the sigmoid approaches a hard-limit nonlinearity. Often the gain is set to 1. The sigmoid is differentiable which makes it possible to derive a gradient search learning algorithm for networks with multiple layers. Moreover, the sigmoid produces a continuous valued output between 0 and 1 which can often be interpreted as a probability estimate. This characteristic is especially suitable for a pattern recognition problem.

5.2.3 The multi-layered neural network

The capabilities of a single perceptron, usually called a node, are limited to linear decision boundaries and simple logic functions. When nodes are arranged in layers and each are connected to the nodes in the preceding layer for input and those in the following layer for output, a multi-layer network is obtained. Complex decision boundaries and arbitrary Boolean operations can be implemented by using a multi-layer network.

Data feeds into a neural network which has M layers through the input layer, as illustrated in Fig.5.2. The nodes in the input layer are passive; each simply broadcasting a single data value over weighted connections to the hidden layer. Hidden nodes thus receive all input data but, because each has a different set of weights, the sets of values differ. Introducing these intermediate layers enhances a network's ability to model complex functions. The output nodes also have distinct sets of weights and process input values to produce a result. The hidden and output nodes process their inputs in two steps. Each multiplies every input by its weight, adds the product to a running total, and then passes the sum through the non-linear sigmoid function to produce a result. The attenuation at the upper and lower sections of the S-shape curve of the sigmoid function smoothly constrains the sum smoothly within fixed limits. On the other hand, with the differentiability of the

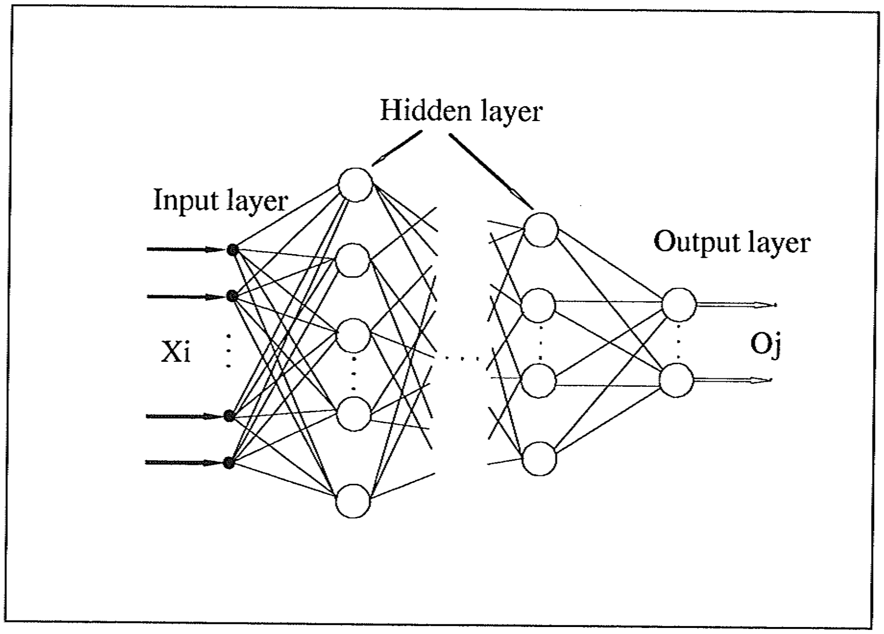

Fig.5.2 A typical multi-layered neural network.

sigmoid function, a gradient search of the weight space can be implemented. The nodes are often connected fully between layers by using an adjustable weight, i.e., every node in layer l is connected to every node in layer(l-1) as well as layer (l+1). Suppose that the output of the ith node in layer (l-1) is $o_{l-1,i}$, then the output of the jth node in layer l, $o_{l,j}$, can be expressed as:

$$o_{l,j} = f\left(\sum_{i=0}^{N_l - 1} w_{l,j,i} \cdot o_{l-1,i} \right) \qquad (5.2)$$

where $w_{l,j,i}$ is the weight which connects the ith node in layer (l-1) to the jth node in layer l. Function $f(\cdot)$ is the sigmoid nonlinearity. When l=0, the zero layer holds the input vector components. Thus, the output becomes $o_{0,j} = x_j$ where x_j is the *j*th component of the input vector. Similarly, the zero component of the input vector to each layer equals to unity, i.e., $w_{l,j,0}$ is the bias weight. When $l=M$, the output layer is reached.

The most popular algorithm for supervised learning is back propagation (BP) which has the ability to adjust the values of its weights to respond to errors, i.e., to minimize a criterion function. A series of training patterns, labeled with their target values, are needed to calculate the errors, The criterion function to be minimized is the sum of the squared error, Q(w), which is defined as [HH93]:

$$Q(w) = \sum_{p=1}^{P} Q_p(w) \tag{5.3}$$

where P is the number of the training patterns and $Q_p(w)$ is the total squared error for the *p*th training pattern given by:

$$Q_p(w) = \frac{1}{2}\sum_{q=1}^{N_M}(o_{L,q}(x_p) - d_q(x_p))^2. \tag{5.4}$$

Here N_M is the number of nodes in the output layer; x_p is the pth training pattern and $d_q(x_p)$ is the target value of the *q*th output node for the *p*th training pattern. An iterative formula is used to adjust the nodal weights as :

$$w_{l,j,i}(k+1) = w_{l,j,i}(k) - \eta \sum_{p=1}^{P} \left.\frac{\partial Q_p(w)}{\partial w_{l,j,i}}\right|_{w(k)} \tag{5.5}$$

where η is a learning parameter and *k* represents the iteration step. Equation (5.5) has an intuitive geometric meaning with all possible sets of weights plotted against the corresponding sum-of-squared error. This results in a bowl-shaped error surface whose bottom represents the best set of weights. This goal is achieved by calculating the instantaneous slope of the error surface with respect to the current weights. It then incrementally adjusts the weights in the direction of the locally steepest path toward the bottom of the bowl. This process resembles rolling a ball down a hill and it is called the gradient descent search, shown in Fig.5.3. The gradient descent search improves the network's overall accuracy as a result of the aggregate corrections during training. However, a real error surface can have complex ravine-like features and dent-like local minima. As a result, the BP may train a network so that a local rather than a global minimum is attained. This effect depends on the exact path down the gradient which, in turn, depends upon the initial values of the weights and factors like the learning parameter. Experimentation is needed to find the least of the minimum errors. Now the partial derivative of $Q_p(w)$ in equation (5.5) with respect to each weight in the network is given by:

$$\frac{\partial Q_p(w)}{\partial w_{l,j,i}} = \frac{\partial Q_p(w)}{\partial o_{l,j}} \cdot o_{l,j} \cdot (1 - o_{l,j}) \cdot o_{l-1,i} \cdot \tag{5.6}$$

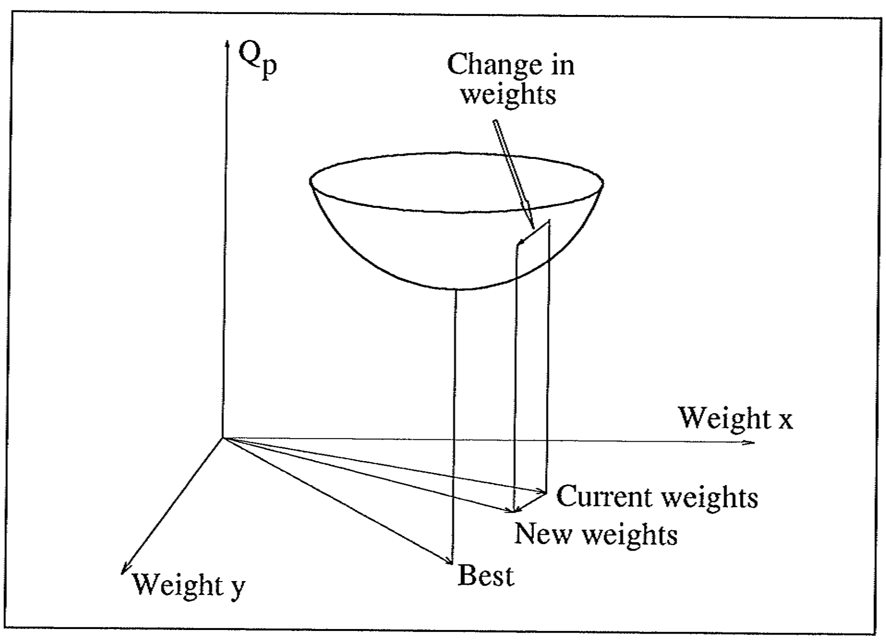

Fig.5.3 The back-propagation algorithm learns by changing its weights to

follow the steepest path toward the bottom of a bowl-shaped error surface. [H93b]
In equation (5.6), the term $\partial Q_p(w)/\partial o_{l,j}$ represents the sensitivity of $Q_p(w)$ to the output node, $o_{l,j}$. $\partial Q_p(w)/\partial o_{l,j}$ can be further expressed as a function of the sensitivities to *the nodes in the next highest layer* as follows [HH93]:

$$\frac{\partial Q_p(w)}{\partial o_{l,j}} = \sum_{q=1}^{N_l+1} \frac{\partial Q_p(w)}{\partial o_{l+1,q}} \cdot o_{l+1,q} \cdot (1-o_{l+1,q}) \cdot w_{l+1,q,j}. \tag{5.7}$$

Equation (5.7) can be continued through all the layers until the output layer, *l=M*, is reached. At the output layer, the target value should be achieved. The sensitivities of the nodes in the output layer are obtained, by using equation (5.4), as:

$$\frac{\partial Q_p(w)}{\partial o_{L,q}} = o_{L,q}(x_p) - d_q(x_p). \tag{5.8}$$

Equations (5.7) and (5.8) are called the hidden layer error and the output layer error, respectively. Then, the sensitivity of a node in the preceding hidden layer is determined backwards from the output layer by using equation (5.7) and so on. The derivative of the output error is passed from the output layer *backwards* and the values of the weights are adjusted, in response to the output errors, by using equation (5.5). Equations (5.6) through (5.8) comprise the back-propagation learning algorithm.

As stated above, it is difficult to determine the value of the best learning parameter. To adapt the learning parameter as a function of the local curvature of the error surface, a momentum term having the form $\zeta(w_{l,j,i}(k) - w_{l,j,i}(k-1))$ is added to each updated weight in equation (5.5), where ζ is called the momentum that varies from 0 to 1. This term makes the current search direction an exponentially weighted average of past directions, and helps keep the weights moving across flat portions of the surface after they have descended from the steep portions. In practice, the unique global minimum is seldom found during learning. What does matter is to find a set of weights that act on data accurately enough for the application. As they are found, the updated weights are saved and the training of the neural network is completed.

Several real concerns have to be considered in developing a neural network for real-time measurement. Choosing an appropriate network size, such as the number of nodes and layers, and extracting key features from the input data are crucial issues. Finding suitable values of the parameters which control how the network learns is another concern. In addition, collecting sufficient samples of *representative* training data is important. All these issues are addressed in the following section.

5.3 Development of the neural network

5.3.1 Architecture

A network is generally incapable of forming a realistic model of a problem if its size is too small. On the other hand, solutions might be obtained with too great a network size but the solution learned during any given training is likely a poor approximation. A guideline used in selecting a network's size is to make the network as simple as possible in order to reduce the processing time without sacrificing a good mapping. A compromise between these two conflicting demands is sought, in practice, through experimentation.

For simplicity, a three-layer network was constructed. The number of nodes in the input layer depended upon the ASLP data from the 3,456 pixels of the CCD sensor. A careful examination of Fig.4.19, however, suggests that the major differences between the ASLP data obtained from those six surfaces is described by only about 1,000 pixels, half of which are located on each side of the pixel corresponding to the peak light intensity. The pixels located outside this range appear to represent the effect of the ambient light and are essentially a *dc* offset. Consequently, the number of nodes in the input layer was reduced significantly to 50 by letting each input represent the arithmetically averaged light intensity over every 20 adjacent pixels.

The hidden layer features the relationship between the input data and the output objective. Experiments involving different numbers of hidden nodes suggested that the constructed network only worked well when six or more nodes in this layer were used so that the lowest number of six was chosen. On the other hand, merely one node was needed in the output layer because the single variable R_{max} described the output objective. The architecture of the neural network is illustrated in Fig.5.4 .

Implementation of the neural network involved the two phases of training and testing. The neural network was trained by utilizing previously measured ASLP data obtained from the specimens. The output R_{max} was compared with the target value which was measured conventionally by using a Mitutoyo Surftest 401 stylus instrument. The weights and bias values were adjusted according to equation (5.5) to reduce the resulting error. In the testing phase, ASLP data from the inspected surface was the only

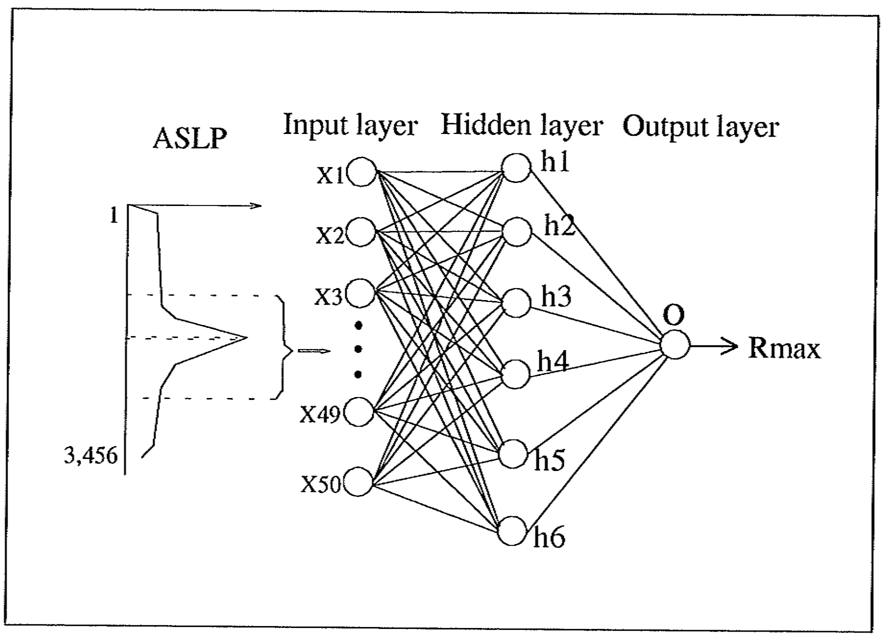

Fig.5.4 A three-layered neural network constructed for the measurement of R_{max}.

information applied to the input layer. The target value was not needed because no comparison was required between the actual and target outputs. The output layer in this phase directly gave R_{max} based on the input ASLP.

5.3.2 Training

Knowledge needed by the neural network to generate the desired output objective comes from its training. One major concern in training is how fast the answers from a particular network will converge as the number of training patterns increases. Generally speaking, the more training patterns that are used, the lower is the predicted error. However, the training time will increase dramatically. Therefore, the selection of the training patterns is critical.

In this study, eight finish-turned surfaces, whose R_{max} values were found to cover from 5 to 20 μm as generated in finish turning [CSA62, S89] by using a Mitutoyo Surftest 401 stylus instrument, were chosen for the training. A 0.8 mm of the roughness-width cutoff value was employed in measurements according to the prescribed standards [CSA62]. An 80% amplitude transferability was obtained from the R-C filter in Surftest 401 according to [JIS76]. A total of 72 sets of ASLP data were sampled from these 8 surfaces. The target value was obtained for each surface by arithmetically averaging the R_{max} values which usually varied within a bound obtained from the stylus measurement. Any overlap of the R_{max} variations for two close average R_{max} values must be avoided in selecting the training samples because the overlap was found to confuse the neural network. As a result, only forty-six out of the 72 ASLP data were selected as the training patterns. The training of the network was conducted iteratively. Part of the selected patterns were employed to train the network. To determine how the network should be trained, the network was tested by recognizing an ASLP obtained from a finish-turned surface the network did not meet before. Then a measurement error, which was obtained by dividing the difference between the target output and stylus-measured data by that data, was found. If the error was greater than a preset value, more training patterns were added and new testing was executed, and so on. When the error converged to the preset value, the solution of the weight was considered acceptable and training was terminated. A comparison of the time for data preparation (which includes manual preparation such as checking the sampled data, selecting the training patterns, and arranging the input ASLP-output target pair) and the training time required for different when the numbers of the training pattern were used is presented in Fig.5.5.

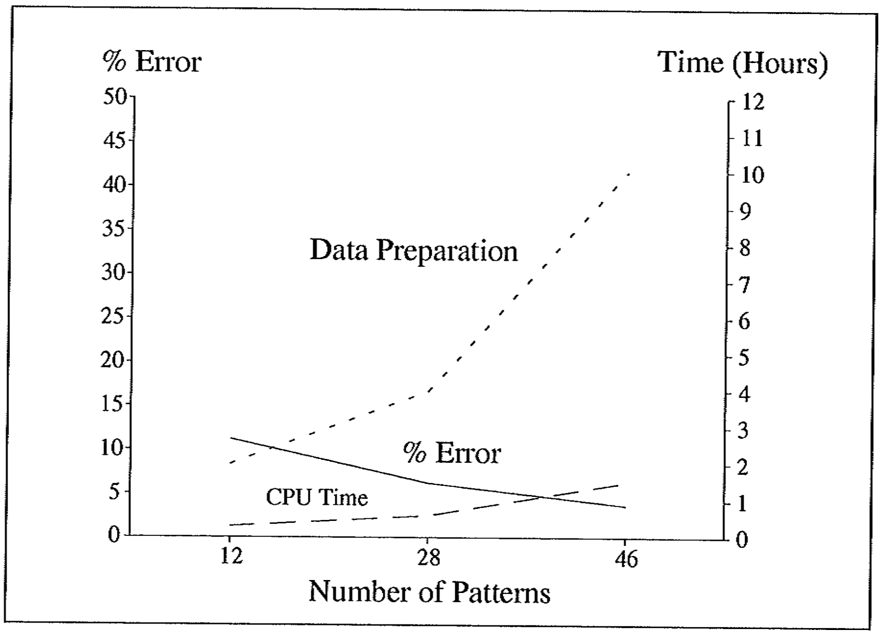

Fig.5.5 Variation of CPU and data preparation times with increasing number of training patterns.

It is apparent from this figure that the CPU time for training grows but the measurement error is reduced when the number of the training patterns increases. The time for data preparation increases particularly quickly because the manual preparation was very time-consuming.

5.3.3 Ambient light compensation

The variation of the ambient light in the workpiece's vicinity has to be compensated for the reasons discussed in Chapter Four. However, the situation here is different from that described in Chapter 4. In Chapter 4, a linear least-squares estimation was employed to solve the inverse problem of light scattering and, hence, a complete set of the sampled ASLP data was required to quantify R_{max}. Now, however, the measurement of R_{max} is a problem of pattern recognition and only about 1,000 pixels located immediately on both sides of the pixel corresponding to the peak light intensity are employed for feature extraction. The pixels located outside this central control region represent the effect of the ambient light and they correspond to essentially a *dc* offset.

A single ambient light level was obtained by arithmetically averaging the light intensities represented by the outputs of the pixels outside the control region. This average was simply subtracted from the values of the 1,000 pixel within the control region and the simple data processing hardly slows the real-time measurements. The compensation strategy is illustrated in Fig.5.6. The ASLP data input into the neural network is shown in Fig.5.6(b). It is obtained from the sampled ASLP of Fig.5.6(a) after the effect of the ambient light has been removed. A small dc offset is still present in Fig 5.6(b) because fluctuations in the ambient light level cannot be totally removed. Fig.5.6(b) indicates that better recognition of R_{max} might be obtained because the response of the nodes in the hidden and output layers are more sensitive to the input ASLP because the amplitude of the activation is around zero and the slope of the sigmoid function is then higher. See Fig. 5.1.

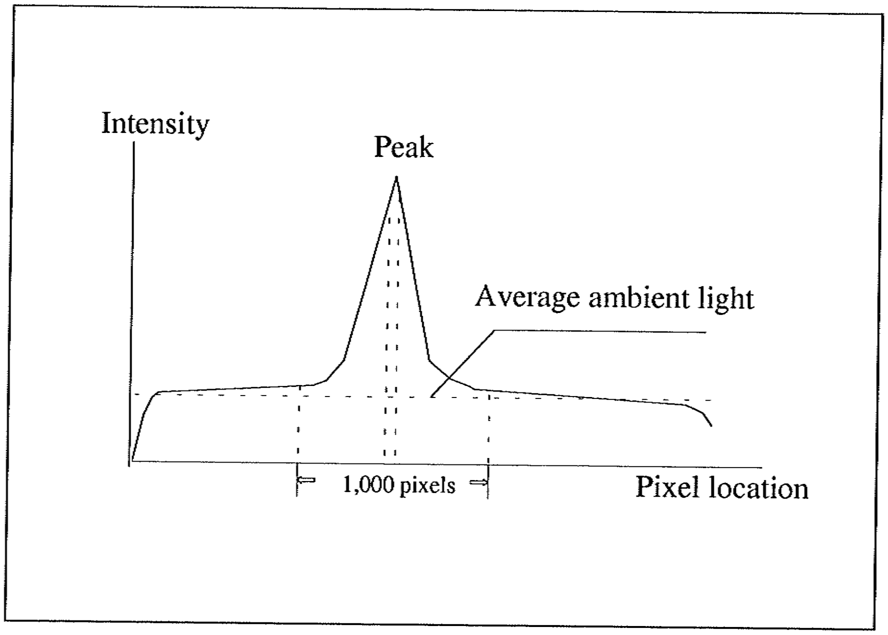

Fig.5.6 (a) Raw ASLP data.

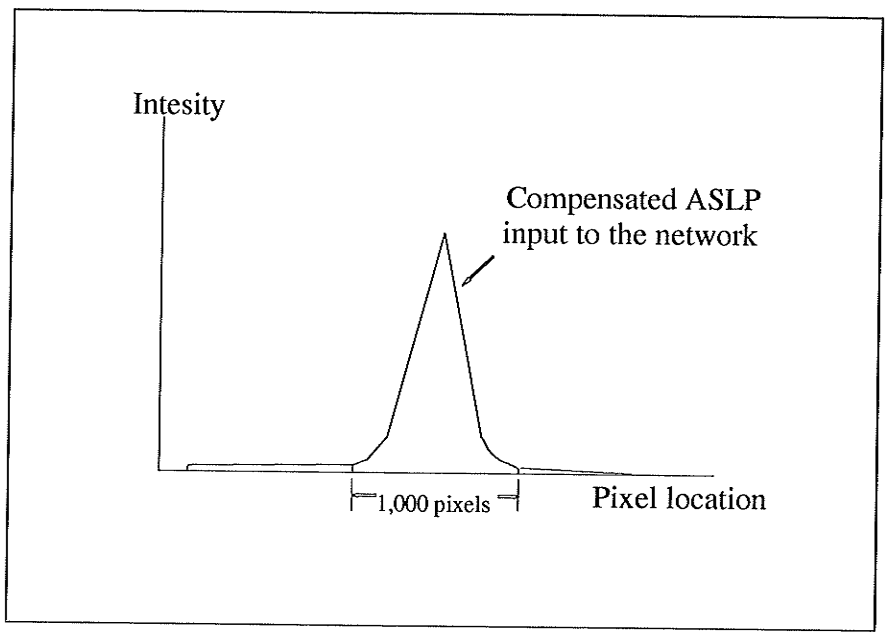

Fig.5.6 (b) The compensated ASLP data input to the neural network.

5.3.4 Software integration

For real-time application, all data processing undertaken in the different software modules, e.g., the A/D conversion, data storage on static RAM, reading data into main memory, and recognizing the R_{max}, must be completed before the next ASLP arrives. This task was accomplished by integrating the modules so that the complete package could be run on an IBM compatible microcomputer as well as a SUN workstation.

The integrated package functioned differently in the training and testing phases. In the training phase, the main program communicated with the BP program [MR88] interactively, as shown in Fig.5.7(a), until a preset 5% error was achieved. The adjusted weighting matrix was saved for the testing phase. Then, the main program continuously received the ASLP data from the laser system and cyclically called the recognition module. Each call would lead to a recognition of R_{max} in the manner shown in Fig.5.7(b). This process was continuous until the completion of the machining operation.

It was found that a slow IBM PC/XT computer having a clock frequency of 8 MHz_was capable of completing one measurement within a reasonable 125 ms with the present integrated package. However, it is recommended that the very time-consuming training be conducted on a high-speed computer like a SUN workstation.

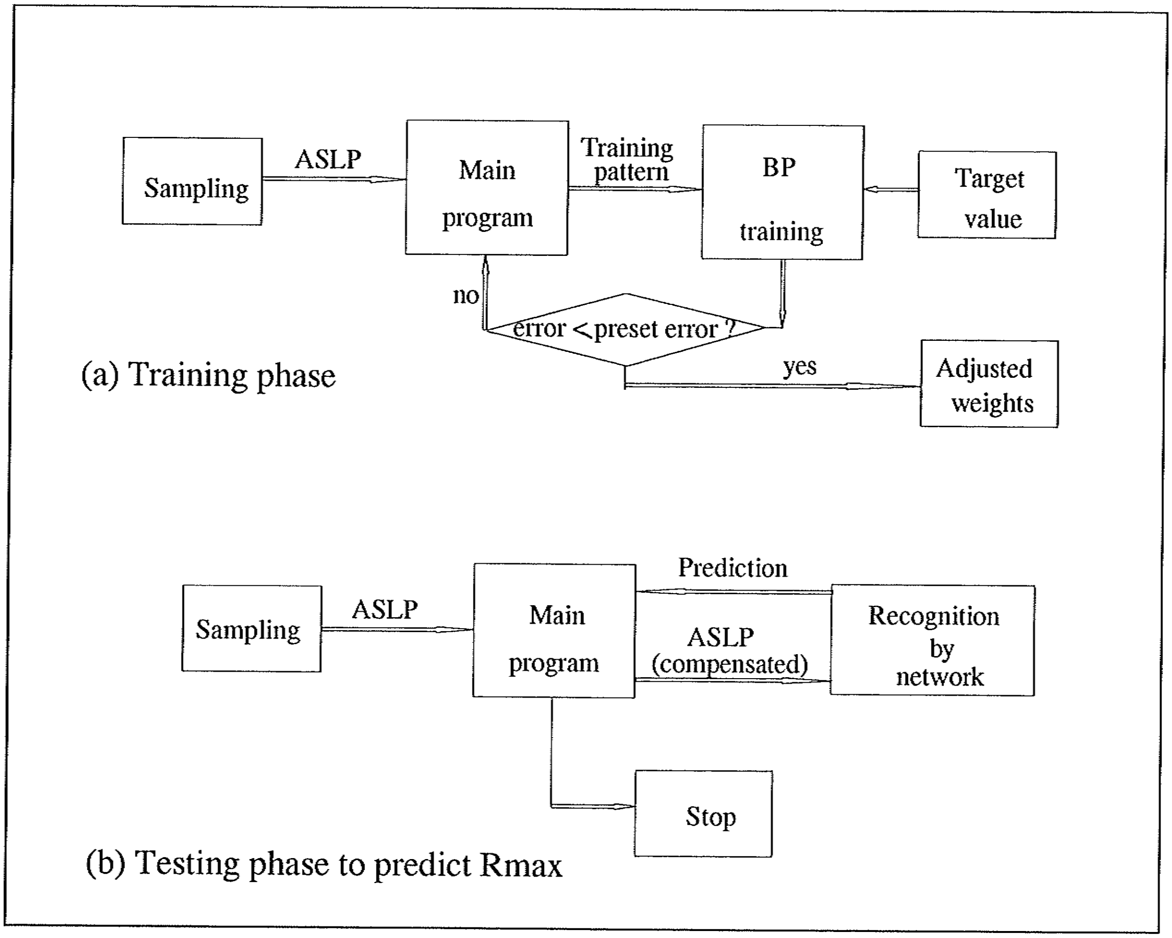

Fig.5.7 The integrated software package in (a) the training phase, and (b) the testing phase.

5.4 Experiments and results

Experiments were conducted in three stages. Finish-turned surfaces, whose ASLP data were used to train the neural network before, were employed in the first stage to check the network's performance. The generalization of the trained network was evaluated in the second stage by employing other finish-turned surfaces which the network had not met before. The measurement error, as defined in Section 5.3.2, was obtained from these two stages by comparing the predicted and stylus-measured R_{max} values. The laser measurement system was verified in the third stage by employing different cutting conditions in a workshop environment.

5.4.1 Preparation of specimens

Thirty-one specimen surfaces, each having the same 20-mm length, were turned on a Supermat 101 lathe. The R_{max} of each specimen's surface was measured by using the stylus instrument. Cutting conditions were chosen in accordance with standard finish turning practice recommended in [MM88, S89] as well as the lathe's specifications. The cutting speed and feedrate ranged from 68.4 to 266.4 m/min. and 0.046-0.122 mm/rev., respectively, but the depth of cut was kept at a constant 0.5 mm because it has little effect on the surface roughness [B84,S89]. The cutting tool had a BR6C6 carbide insert with the geometry: 7,7,7,7,75,15,0.40 according to the American Standards Association given in [S89]. All specimens were made of 4340 SPS steel and they were heat-treated to have two uniform hardnesses of HRc 22±0.5 and 36±0.5.

5.4.2 Test on the specimens used in the training

The objective of this test was to determine the performance of the trained network. A specimen's surface was rotated on the lathe whilst the laser measuring system was traversed in the feed direction. The rotational speed and feedrate were identical to those used when the surface was actually turned. Based upon the captured ASLP, the neural network output a R_{max} reading. A total of three readings was obtained for each surface and these readings were averaged arithmetically to give the predicted R_{max} shown by the solid circles in Fig.5.8. It can be seen from this figure that all the predictions except one are within the dotted R_{max} bounds of the stylus measurement. The exception corresponds

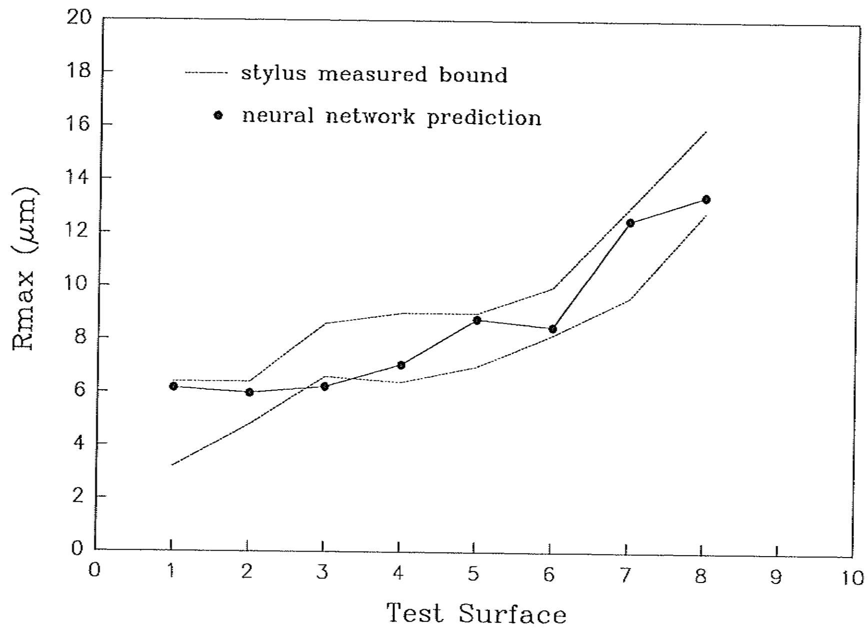

Fig.5.8 The R_{max} predicted by the neural network on the specimens used in training .

to a surface which was turned at the highest cutting speed used of 266.4 m/min and a 0.06 mm/rev. feedrate. It is possible that the lateral scattering [BS63] caused by extraneous vibrations at this high cutting speed could have introduced the higher but still acceptable 6% or so measurement error.

5.4.3 Test on the unknown specimens

The capability of the trained network to recognize unknown_ASLP was evaluated in this test. The procedure is the same as that described in Section 5.5.2. The results shown in Fig.5.9 indicate that eight out of the twenty-three predictions are outside the measured stylus bounds. On the other hand, measurement errors ranging from only 1 to 11% were found for these eight delinquent predictions. One of the largest discrepancies occurs for Surface 15 which was turned at a cutting speed of 239 m/min. and a feedrate

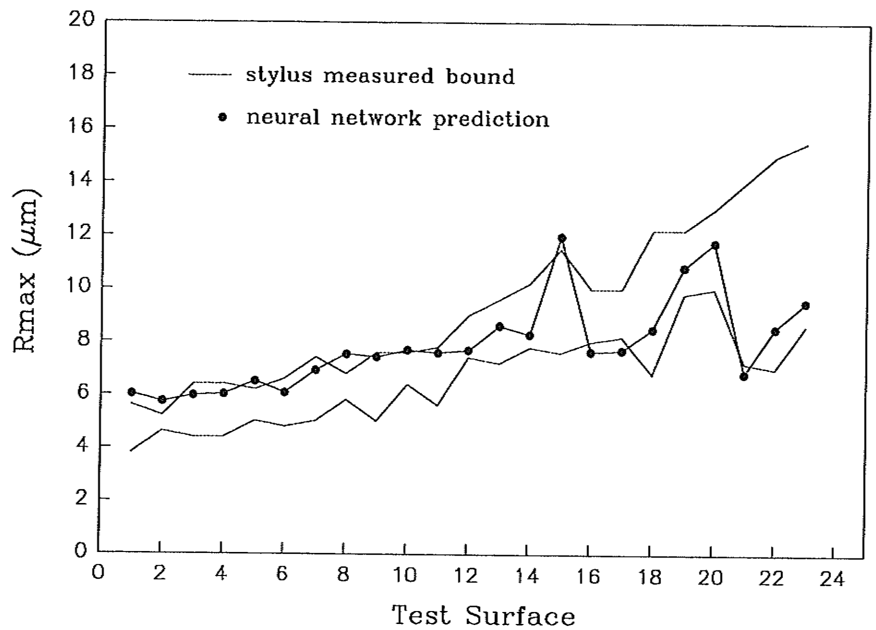

Fig.5.9 The R_{max} predicted by the neural network on the unknown specimens.

of 0.081 mm/rev. Again, the lateral scattering caused by extraneous vibrations at this very high cutting speed affects the sampled ASLP data and , therefore, the performance of the neural network. This effect needs further investigation. Nevertheless, the test shows that the new approach is capable of measuring R_{max} with an acceptable accuracy.

5.4.4 Real-time measurements

The objective of this test was to verify the approach's robustness, accuracy and dynamic response in a real cutting environment. A total of ten cutting tests were performed by employing conditions, listed in Table 5.1, which are typical of finish turning [SL81,M88]. The lathe and cutting tool were the same as those used to prepare the specimen surfaces whilst the workpiece was a 4340 SPS steel having a hardness of HRc22. The depth of cut was a constant 0.5 mm. The remaining cutting variables are given in Table 5.1. All the tests were performed without a cutting fluid because a non-water based coolant influences the measured ASLP as discussed in the previous chapter. Fig.5.10 shows that ten predictions are within the extreme stylus measurements and agree closely with the average stylus data. Better predictions were obtained in these real-time measurements, compared with the results of Fig.5.9, because extraneous vibrations are insignificant at moderate (less than 150 m/min) cutting speeds.

Table 5.1 Conditions Used for Real-time Measurements.

Test No.	V (m/min)	f_0 (mm/rev)	Cutting time (s)
1	82.2	0.046	39.6
2	82.2	0.081	28.7
3	82.2	0.122	18.8
4	144.0	0.046	28.3
5	144.0	0.061	17.4
6	144.0	0.081	16.2
7	51.2	0.046	75.7
8	51.2	0.081	43.2
9	51.2	0.122	28.3
10	144.0	0.122	10.6

It was found that the laser system was capable of completing three sets of measurements within 10.6 s even when using the IBM/PC XT computer mentioned earlier. On the other hand, about 75s was needed previously to obtain only one R_{max} reading by alternatively solving the inverse problem of light scattering on a 33 MHz IBM/PC 486 computer. The advantage of using a neural network for real-time measurements is very obvious.

The question as to how the present system's speed can be improved was investigated further. Real-time measurement can take place only when the time required to complete the A/D conversion, pixel charge transport, bus transfer of data, and the neural network's recognition is less than the designed interval between two adjacent measurements. A high speed 7821 A/D converter, for example, has a conversion time of 660 ns and a delay between consecutive samples of approximately 330 ns. The reset clock speed was set, therefore, to 1.2 MHz or about 0.82 μs per pixel so that approximately 3 ms was required to transport the entire array of 3,489 pixels into the static RAM. The time

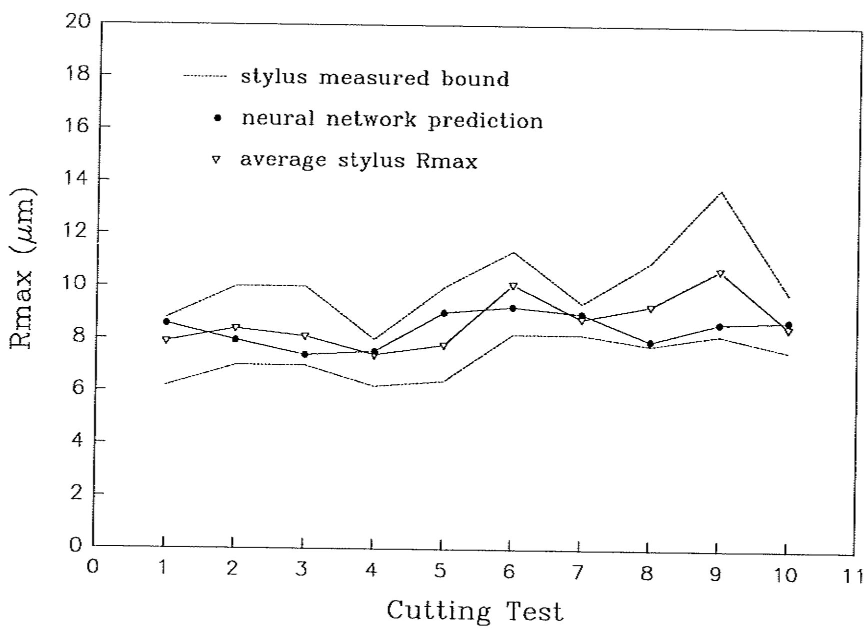

Fig.5.10 The results of the real-time measurements for ten cutting tests.

spent for bus transfer was about 122 ms. Moreover, the processing time of the present neural network was negligible because of its simple architecture. Thus, the total time required to complete one measurement was about 125 ms which was found to be adequate for the real-time surface roughness measurements discussed. This time can be reduced considerably with the use of presently available, higher speed microcomputers.

5.5 Remarks

The new approach proposed in this chapter is suitable for the real-time measurement of the maximum peak-to-valley roughness, R_{max}, produced during finish turning. The ASLP scattered from the workpiece's surface was sampled by using the laser measuring system developed in the previous chapter and input into a three-layer neural network involving back propagation learning for processing. By using these innovations, the R_{max} value can be predicted with a measurement error less than 10% in comparison with average stylus measured R_{max} in a dry cutting environment. The new approach is more robust and responds much faster than the method of solving the inverse problem of light scattering. Moreover, the results from training the neural network converge quickly due to the efficient mapping between the light data and the required R_{max} output. The influence of the ambient light has also been removed by carefully defining the input light data. However, two key points must be noted. First, the selection of the training data is crucial to the performance of the neural network. There should be no overlapping of the R_{max} ranges between adjacent training patterns. Second, an iterative training procedure, which was conducted in this study, is recommended strongly because it can improve the performance of the neural network.

CHAPTER 6

ON-LINE PREDICTION

6.1 Introduction

The approaches proposed in Chapters 4 and 5 can be used to assess the surface roughness by using on-line or real-time measurement. These approaches are feasible and good results were obtained under finish turning conditions but they do not provide a straightforward relationship between the surface roughness generated during turning and cutting variables like the cutting speed, feedrate, and depth of cut. Such a relationship is especially useful when implementing adaptive control strategies for machining. This deficiency is usually overcome by employing an alternative strategy of modeling the cutting process as a way of predicting the roughness on-line. This will be the subject of this chapter.

A comprehensive model for predicting the maximum peak-to-valley surface roughness, R_{max}, on-line is developed in this chapter. The model employs the preset nominal cutting variables as well as the cutting tool's geometry. The progressive tool flank wear as well as the vibratory interaction between the cutting tool and workpiece are considered in the model. The formulation of the model is presented in Section 6.2, where a description of the on-line estimation of the tool flank wear and the dynamic cutting force is also presented. Experimental procedures to validate the model are described in Section 6.3.

Section 6.4 includes a comparison between the predictions and conventional stylus measurements and discusses the experimental results obtained under different cutting conditions. Findings from this approach are given in Section 6.5.

6.2 Model's formulation

A turned surface is generated as a result of the movement of the tool's cutting edge relative to the workpiece. It is generally accepted [S89] that the feedrate and the tool's nose radius significantly influence the surface finish. A better surface finish is expected with a smaller feed rate f_o, auxiliary cutting edge angle ϕ_1, and with a larger nose radius, ρ_o. When $f_o < 2\rho_o \sin \phi_1$, which is usually the case in finish turning, the R_{max} generated can be derived from purely geometrical considerations as [S64]:

$$R_{\max} \equiv R_{th} = \frac{f_0^2}{8\rho_0}. \tag{6.1}$$

This formula is time independent and, hence, does not account for progressive tool wear or extraneous vibrations of the cutting tool. As a result, the R_{max} derived purely from equation (6.1) are found to be considerably lower than experimental values obtained from stylus measurements. Disagreement becomes more obvious for finish turning because of a high tool wear-rate due to the use of a high cutting speed as well as vibrations introduced as a result of using a small feedrate. Therefore equation (6.1), which gives the "theoretical " value, R_{th}, of R_{max} has to be modified.

A worn tool is illustrated in Fig.6.1. If there is no built-up edge formed on the tool's rake face, the change in the tool's nose radius at time t, $\Delta\rho(t)$, can be determined geometrically from the flank wear, $h_f(t)$. The $h_f(t)$ is defined as the length of the wear land on the principal cutting edge, (shown in elevation A-A of Fig.6.1) at the point where the arc of the nose radius meets the cutting edge. Now $\Delta\rho(t)$ can be expressed as [B84]:

$$\Delta\rho(t) = h_f(t)\frac{\tan\alpha}{1-\tan\alpha\tan\gamma}. \qquad (6.2)$$

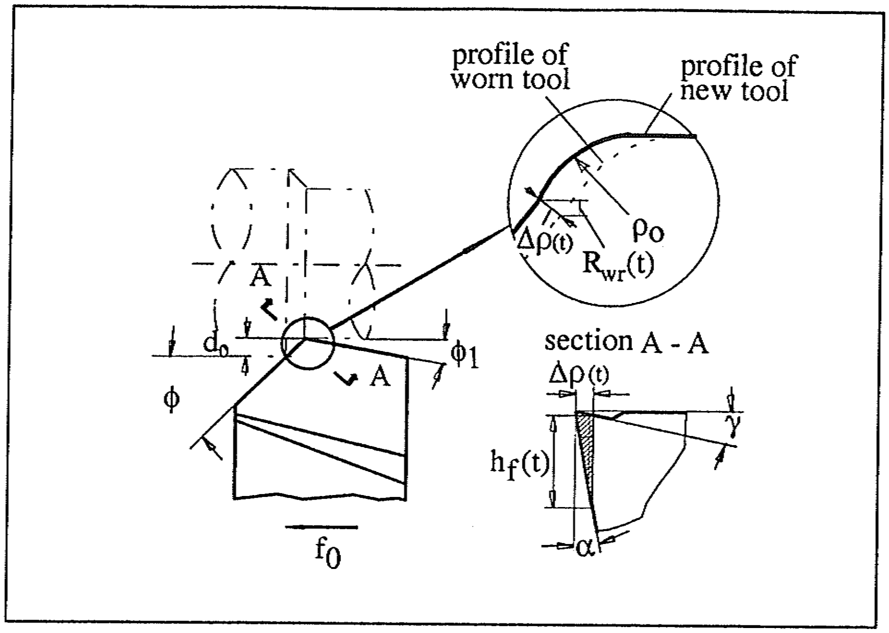

Fig.6.1 Geometrical details needed to calculate $\Delta\rho(t)$.

This change in tool nose radius is reproduced with high fidelity on the turned surface [MKST85]. When resolved perpendicular to the machined surface , $\Delta\rho(t)$ gives the surface roughness, $R_{wr}(t)$, caused purely by the tool flank wear. Thus

$$R_{wr}(t) = \Delta\rho(t)\cos\phi = h_f(t)\frac{\tan\alpha\cos\phi}{1-\tan\alpha\tan\gamma}. \qquad (6.3)$$

The tool's flank wear $h_f(t)$, which is needed to compute the tool wear component, $R_{wr}(t)$, from equation (6.3), can be found indirectly from on-line measurements of the changes in the lathe's spindle speed during the normal tool life phase [KYPB93]. It is well known [K91] that the tangential cutting force increases in proportion to the tool wear. This increase produces a larger resisting torque on the lathe which reduces the spindle speed. Hence, the observed change in spindle speed $\Delta n(t_i)$, at instant t_i, can be used to indirectly monitor the tool's flank wear at that time, $h_f(t_i)$. The relationship proposed in [KYPB9393] involves point-to-point computations which update the flank wear, $h_f(t_i)$, based on the $h_f(t_{i-1})$ calculated at a previous instant $t_{i-1}=t_i-\Delta t_i$, where Δt_i is the ith interval of estimation, by employing:

$$h_f(t_i) = h_f(t_{i-1}) + \frac{66.7\pi K_n J}{C_p f_0^{\beta_f} d_0^{\beta_d}} \frac{\Delta n(t_i)}{D(t_i)\Delta t_i}. \tag{6.4}$$

The C_p is a constant for given workpiece and cutting tool materials and typically has a value of about 5,500 when 4340 steel is cut by a carbide tool [KYPB93]. Moreover, K_n in equation (6.4) is also a constant which depends upon the lathe's setting, such as the number of poles of the motor, the gear transmission ratio, and the type of chuck used. The values of C_p and K_n are obtained from [KYPB93] in which an empirical procedure to determine both constants was detailed. On the other hand, representative β_f and β_d , the indices in equation (6.4), are $0.4 > \beta_f > 0$ and $\beta_d \approx 1.0$ for steel being cut with a carbide tool [K91]. The computations to find $h_f(t_i)$ from equation (6.4) are started by evaluating the initial flank wear, $h_f(t_i=0) \equiv h_{f0}$. The empirical relationship

$$h_{fo} = 0.040 - 0.005V - 0.002f_0 + 0.0003d_0 + 0.006(\text{mm}) \tag{6.5}$$

proposed in [KYPB93] was used to find h_{f0} from the nominal cutting variables.

A light feed rate will easily lead to tool vibrations which are sustained after the completion of the previous cut. The resulting, extraneous relative displacement between the tool and workpiece directly affects the surface finish [T65]. Now the stiffness of a machine tool is generally much larger than that of the workpiece. Consequently, the effect of the vibrations can be assessed by using the simplified theoretical model of Jang and Seireg [JS89] in which the interaction between the tool and workpiece need be determined only in the direction perpendicular to the machined surface. The workpiece-tool system is illustrated in Fig.6.2 from which the equations of motion:

$$M_W \ddot{X}w(t) + C_W \dot{X}w(t) + K_W Xw(t) = F(t) \tag{6.6}$$

and

$$M_t \ddot{X}_t(t) + C_t \dot{X}_t(t) + K_t X_t(t) = F(t). \tag{6.7}$$

can be formulated straightforwardly.

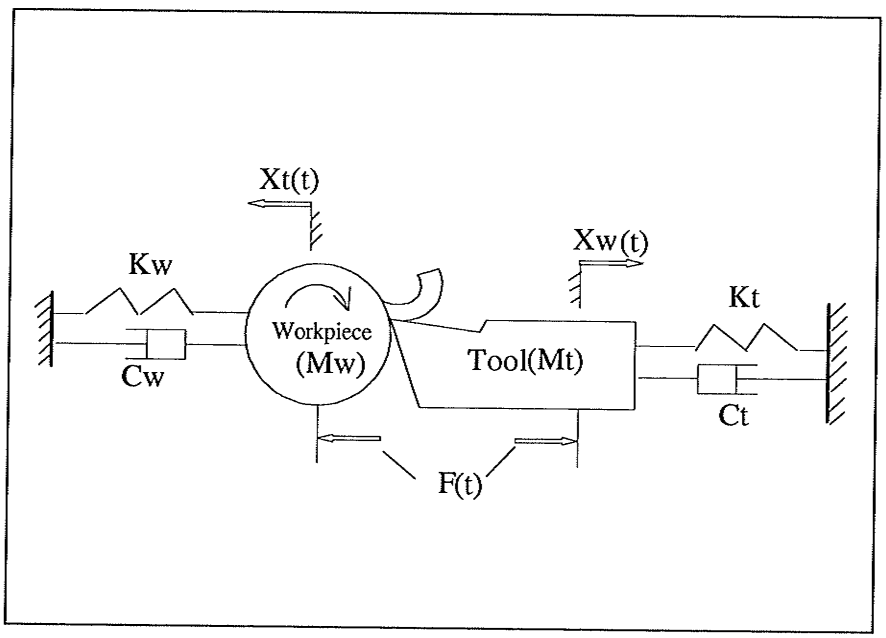

Fig.6.2 Workpiece-cutting tool idealization.

The relative displacement between the workpiece and tool , u(t), can be seen from Fig.6.2 to be the sum:

$$u(t) = X_w(t) + X_t(t). \tag{6.8}$$

This dynamic motion (from the stationary configuration) is perpendicular to the machined surface and represents an increment in the peak-to-valley roughness produced by the extraneous vibrations. Hence, u(t) is defined as the vibration component of the total predicted $R_{max}(t)$.

A standard fourth order, Runge-Kutta integration procedure [BF93] can be used to compute $X_w(t)$ and $X_t(t)$ from equations (6.6) and (6.7), respectively, in order to calculate the u(t) from equation (6.8) once the coupling cutting force, F(t), has been determined. The cutting force depends mainly upon the instantaneous, uncut chip area and the properties of the workpiece's material [K67]. However, F(t) is modified by a rubbing mechanism between the tool's flank and the machined surface which causes the force to increase as the tool's flank wears [K91]. Moreover, a resisting (i.e. opposing) force is also present due to the relative motion between the workpiece and cutting tool [SK69]. Mathematically, F(t) can be reasonably regarded as the superposition of three components: the principal component, $F_p(t)$, which is altered by the rubbing component, $F_f(t)$, and the damping component, $F_d(t)$. Hence,

$$F(t) = F_p(t) + F_f(t) - F_d(t). \tag{6.9}$$

The principal component, $F_p(t)$, is proportional to the instantaneous, uncut chip area, A(t), given as [K67]:

$$F_p(t) = K_s \cdot r \cdot A(t) \tag{6.10}$$

where K_s is the unit cutting force (N/mm^2) which depends upon the uncut chip area and the workpiece material. The value of K_s is usually determined experimentally. For example, the experimental value of K_s increases from 1,400 to 3,000 N/mm^2 for SAE 1045 steel [B84] when the uncut chip area is reduced from 2 to 1 mm^2. The r in equation

(6.10) is the ratio between the radial and tangential cutting forces which, from the second cutting force law [B84], is found to be about 0.5 to 0.6 for the carbide cutting tools used in the current investigation. Now the uncut chip area, A(t), is simply the product of the chip's instantaneous thickness, a(t), and width, b(t), or:

$$A(t) = a(t) \cdot b(t). \tag{6.11}$$

The instantaneous width , b(t), can be seen from the geometry of Fig.6.3 to be:

$$b(t) = \frac{d_0 - [R_{wr}(t) + u(t)]}{\sin \phi}. \tag{6.12}$$

A machined surface, on the other hand, is influenced by the surface conditions generated by the previous cut [T65]. This influence is included in the instantaneous thickness of the chip, a(t), which takes the form (Fig.6.3):

$$a(t) = f_o \sin \phi - \mu_k [u(t) - u(t - T_r)] \cos \phi \tag{6.13}$$

where μ_k is a simple overlapping factor used to incorporate the vibration effect due to the previous cut. The μ_k varies from zero for thread cutting to unity for plunging. T_r is the time required for the workpiece to rotate one revolution. In summary, the principal component of the cutting force may be found by combining equations (6.10) through (6.13).

The rubbing effect is proportional to the contact area between the tool flank and the machined surface. With greater flank wear, both the contact area as well as the rubbing increases. Hence, the rubbing force, $F_f(t)$, is given by:

$$F_f(t) = K_f \cdot h_f(t) \cdot \{d_0 - [R_{wr}(t) + u(t)]\} \tag{6.14}$$

where K_f is a proportional constant and the remaining terms represent the contact area due to the flank wear and vibrations.

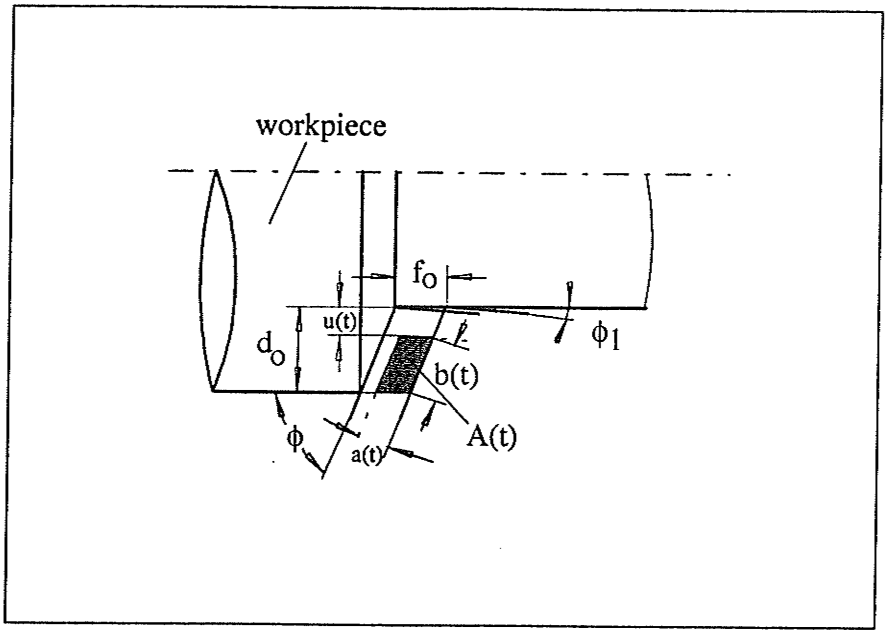

Fig.6.3 Variation of uncut chip area, A(t), during turning.

It is generally observed [SK69] that turning at a higher cutting speed is more stable due to greater damping caused by the penetration of the vibrating tool's cutting edge into the workpiece [T65]. The damping component, $F_d(t)$, is proportional to the relative velocity between the tool and workpiece, $\dot{u}(t)$, so that it can be expressed as:

$$F_d(t) = C_t \dot{u}(t). \tag{6.15}$$

The C_t, on the other hand, is derived from the formula given in [SK69]:

$$C_t = \frac{\mu \sigma_y}{V \alpha^2} K_\delta \delta \{d_0 - [R_{wr}(t) + u(t)]\}, \tag{6.16}$$

where $K_\delta \delta$ corresponds to the contact length between the tool flank and workpiece. The K_δ is found to be 0.25 [SK69] and δ is the wedge angle of the tool. The instantaneous contact width is represented by the term, d_0-[$R_{wr}(t)$+u(t)] during the penetration. Here μ is the coefficient of friction on the tool's flank and α is the relief angle of the tool.

The tool wear component, $R_{wr}(t)$, and the vibration component, u(t), are all defined in the direction perpendicular to the machined surface. Hence, the total predicted peak-to-valley roughness, $R_{max}(t)$, at time t can be expressed as the sum of these two time-dependent components and the theoretical component, R_{th}. The resulting equation for $R_{max}(t)$ is:

$$R_{max}(t) = R_{th} + R_{wr}(t) + u(t). \tag{6.17}$$

The flow chart of the prediction procedure described in this section is shown in Fig.6.4.

6.3 Experiments

Experiments were performed on a workpiece which were supported individually between a three-jaw chuck and a tail stock and which were turned in a Supermat 1011 lathe driven by a 2.7 kW motor having a 2 or a 4 pole configuration. A 1024 pulses per revolution, optical encoder was installed on the lathe's spindle to sample, in conjunction with custom designed electronics [KYPB93], the spindle speed. A three-dimensional, strain-gauge dynamometer [YJPB90] was used to measure the dynamic cutting forces. It is shown in Appendix E that the sampled data of the radial cutting force are not reliable due to very high crosstalk with the other two components. However, the dynamic cutting force depends mainly on the tangential force. Hence, only the tangential force data was used. A low 36 Hz sampling frequency, which is higher than used previously in [K91], was chosen to conserve computer memory. Each workpiece comprised a 4340 SPS steel bar having a diameter of 32±0.5, 33±0.5 or 36±0.5 mm and a 300 ±0.5 mm length. Workpieces were heat-treated to a uniform hardness of either 22±0.5 or 36±0.5 measured on the Rockwell C scale by employing a Versitron hardness tester. As the load torque on the spindle depends upon the workpiece's radius as well as the cutting force,

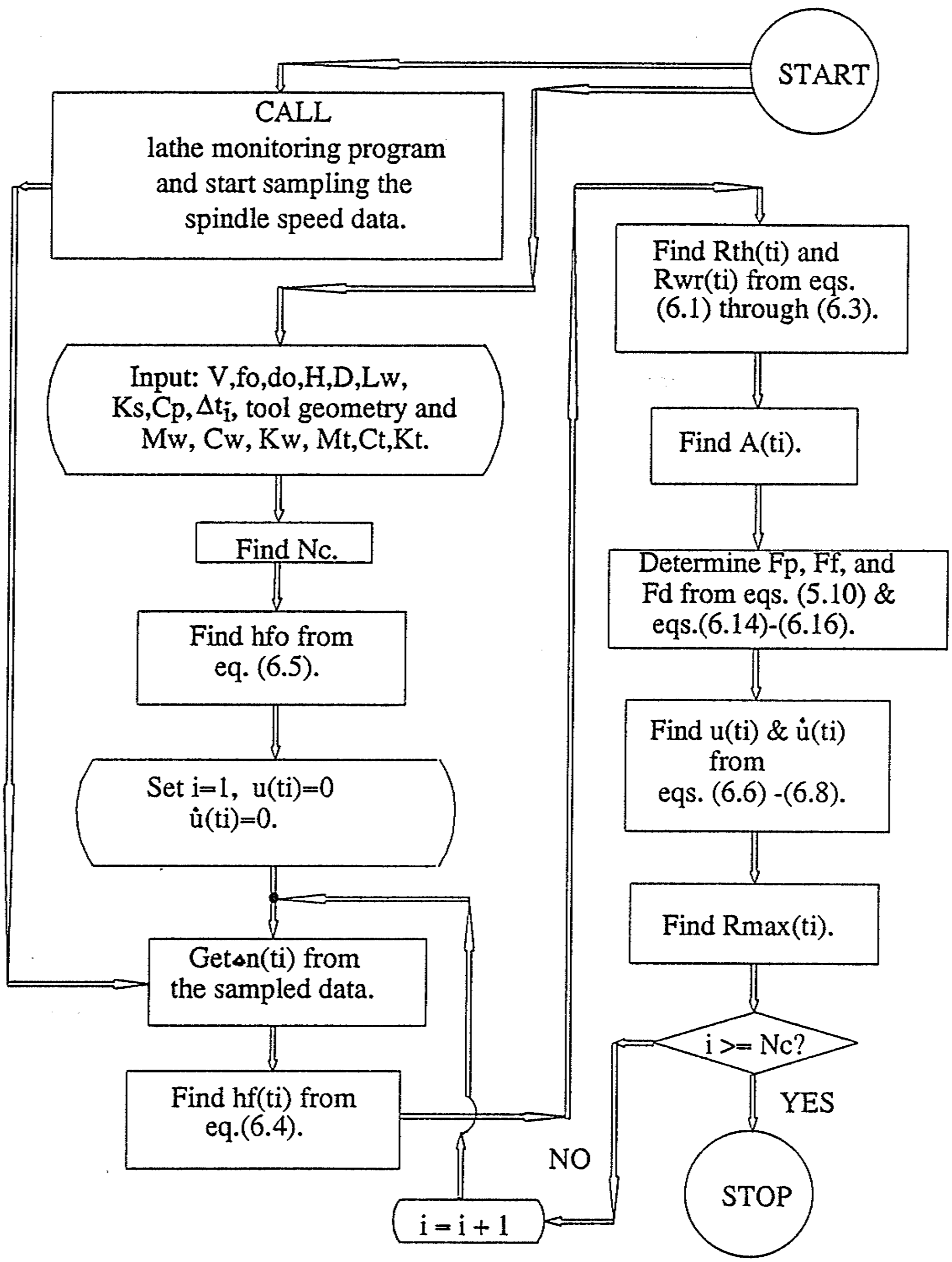

Fig.6.4 Flow chart for predicting R_{max}.

workpiece's diameter was maintained constant for each set of tests. This constraint ensured that torque changes were due purely to cutting force variations.

Thirty-one tests were conducted without cutting fluid under the conditions given in Table 6.1. Test #1 was a tool life test in which the cutting conditions were unchanged for a total of 23 cuts. The cutting length in each test was 150-mm. When a cut was finished, the workpiece was reversed to continue the next cut. This tool life test investigated the effect of progressive flank wear on the predicted R_{max}. The initial flank wear, h_{fo}, was determined by using equation (6.5) and the nominal cutting variables. Subsequent values of $h_f(t_i)$ were obtained from equation (6.4) based on the spindle speed change sampled by the optical encoder mounted on the lathe's spindle, as shown in [KBPM90]. To compare the estimated $h_f(t_i)$ with the actual value, the tool was removed after each 150-mm cut so that its flank wear could be recorded by utilizing a Nikon Epiphot-Time optical microscope having a 210X magnification. The CIRP specifications described in reference [GST87] were used to determine the flank wear. The tool was a BR6C6 carbide insert having the geometrical designation 7,7,7,7,75,15,0.46 according to the U.S. Standards Association as described by [S89].

The remaining tests #2 through 31, each of which involved a cutting length of 20 mm, were subdivided into six groups according to the cutting speed employed. Five feedrates were used but the depth of cut was kept constant because of its restricted range and, hence, limited effect on the surface finish. The cutting time for each test varied from 4 s to 40 s so that the change in the tool's flank wear was negligible during any one cut. The tool flank wear for a given test was found by taking measurements from a magnified photograph before each cut. This wear value was input to the previously developed theory as a known parameter. Consequently, the effects of the feed rate and cutting speed on the surface roughness could be investigated on the predicted R_{max}.

A lathe monitoring program was used to sample the spindle speed. During the 150-mm cut of Test #1, eight values of the tool's flank wear were first estimated on-line by running the program.(See Program 4 given in Appendix B.) Then the estimated flank wear and the nominal cutting variables were processed to produce eight predictions of R_{max} for each cut. For Tests #2 through 31, the flank wear value and nominal cutting variables given in Table 6.1 were used directly to predict R_{max}.

For comparison, a workpiece's surface roughness was measured off-line for each test by using a Mitutoyo Surftest, Model 401 stylus instrument having a cut-off length of 0.8 mm. For each cut of Test #1, measurements were made at 32 equally spaced points in a direction parallel to the feed. Each measurement was repeated three times at every point and averaged arithmetically. Finally, the 32 averaged values were themselves averaged to determine R_{max} for each cut. For Tests #2 through 31, on the other hand, measurements were made at 6 equally spaced points within the cutting length and R_{max} was obtained from averaging these six readings.

Table 6.1 Cutting Conditions Used for On-line Predictions.

Test #	V (m/min)	f_o (mm/rev)	d_o (mm)	H (Rc)	h_f (mm)
1	141.4	0.061	0.75	22	estimated on-line
2-6	68.4	$f_1 = 0.046$	0.50	36	0.048
7-11	75.8	$f_2 = 0.061$	0.50	22	0.026
12-16	119.8	$f_3 = 0.081$	0.50	36	0.028
17-21	132.7	$f_4 = 0.101$	0.50	22	0.035
22-26	239.0	$f_5 = 0.122$	0.50	36	0.035
27-31	266.4	(Used for Tests 2 through 31)	0.50	22	0.044

The workpiece rotated with the lathe's chuck during cutting so that the mass, M_w used in equation (6.6) corresponds to the combined mass of the workpiece and the chuck. Moreover, the dynamometer also served as the tool post so that its mass was added to that of the tool to determine M_t in equation (6.6). Each mass was measured by employing an Accu-Weigh electronic scale which has a capacity of 135 kg with a resolution of 0.1 kg. A computer simulation showed that the relative displacement , u(t), changed less than 1% when the tool's critical damping ratio was varied from 0.01 to 0.1. Hence, the damping ratios taken from reference [JS89] and given in Table 6.2 were substituted in equations (6.5) and (6.6). To determine the spring constant of the workpiece, K_w, a beam idealization was used in which the heavy chuck corresponded to a fixed end and the light weight tail center was represented as a simply supported end. The spring constant of the tool, K_t, depends upon the stiffness of the dynamometer and it was determined from the relationship $K_t=M_t\omega_n^2$, where ω_n is the fundamental natural (circular) frequency of the dynamometer. The ω_n was computed by using the ANSYS finite element package [A89] and was found to be close to the result from an impact hammer test. All data pertaining to the mass, damping ratio and spring constants are listed in Table 6.2.

Table 6. 2 Mass, Damping Ratio and Spring Constant Values.

	Workpiece+Chuck	Tool+Transducer
Mass (kg)	12.65	7.64
Damping ratio	0.003	0.07
Spring (N/m^2)	$4.3x10^7$	$1.1x10^7$

6.4 Results and discussion

The performance of the theoretical model will depend mainly upon the accuracy of the predicted principal cutting force and the tool's flank wear. Now, the unit cutting force, K_s, is a function of the uncut chip area as well as the properties of workpiece's material and, hence, it has a unique value for particular cutting conditions. The values of K_s are not documented in the literature for a wide variety of cutting conditions. As a result, the first step in applying the model is to determine K_s for the cutting conditions chosen. The results will be used to determine F_p from equation (6.10). The h_f is estimated by using equations (6.4) and (6.5). With F_p and h_f as inputs, the accuracy of the model can then be examined for different cutting conditions. The results obtained from these estimations are presented in the following subsections.

6.4.1 Determination of K_s

The actual values of K_s for the chosen cutting condition are obtained directly as a ratio between the measured cutting force and the nominal uncut chip area. The variations found in K_s at various feedrates and cutting speeds are shown in Fig.6.5 for a depth of cut of 0.5 mm. It can be seen from this figure that K_s is larger with a smaller feedrate for the chosen cutting speeds. However, for a constant uncut chip area, K_s will decrease as the cutting speed grows due to the reduction in the cutting force [B84]. When a very small feedrate is used, as in finish turning, the variations in K_s are more pronounced for different cutting speeds. For example, an examination of Fig.6.5 indicates a change of 25% in K_s when the cutting speed is altered from 68.4 mm/min. to 239.0 m/min at a constant feedrate of 0.0457 mm/rev. Also, the hardness of the workpiece's material will influence K_s. A hard material is seen from Fig. 6.5 to produce a higher K_s and a wider range of values than a soft material when a small feed rate is used. Thus, it may be important to accurately determine K_s from a prior-to- finish cutting test and use this value in the model.

It must be pointed out that the contribution to the dynamic cutting force from changes in the tool's flank wear are insignificant in Tests 2 through 31, because h_f has the small values shown in Table 6.1 and the cutting time is short.

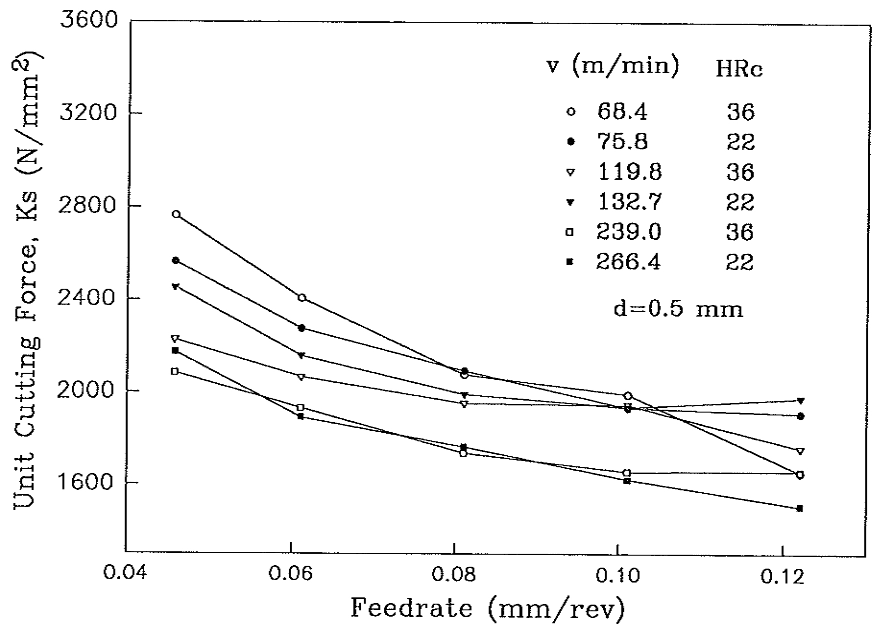

Fig.6.5 Variations of the unit cutting force, K_s, with feedrate, cutting speed, and hardness of workpiece's material.

6.4.2 Prediction of cutting force

Comparisons between the predicted and measured tangential cutting forces are shown in Figs.6.6 and 6.7 for Tests 2 through 31. There is invariably good agreement. Moreover, it can be seen that the tangential force increases linearly with the feedrate but decreases at greater cutting speed. These findings agree with those reported in [B84]. For the two workpieces considered, the tangential cutting force basically increases with workpiece hardness.

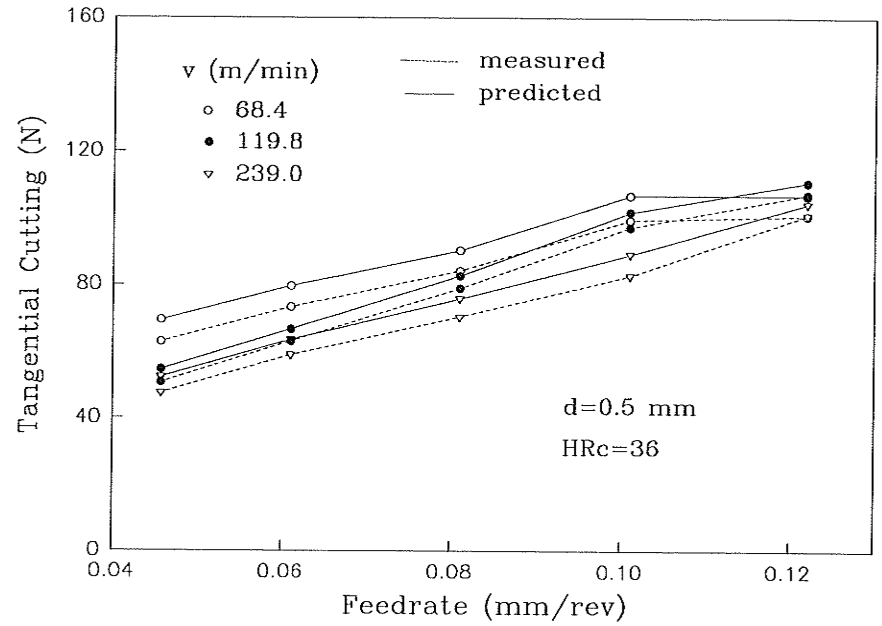

Fig.6.6 Comparison between the measured and predicted tangential cutting force for a steel workpiece having HRc=36.

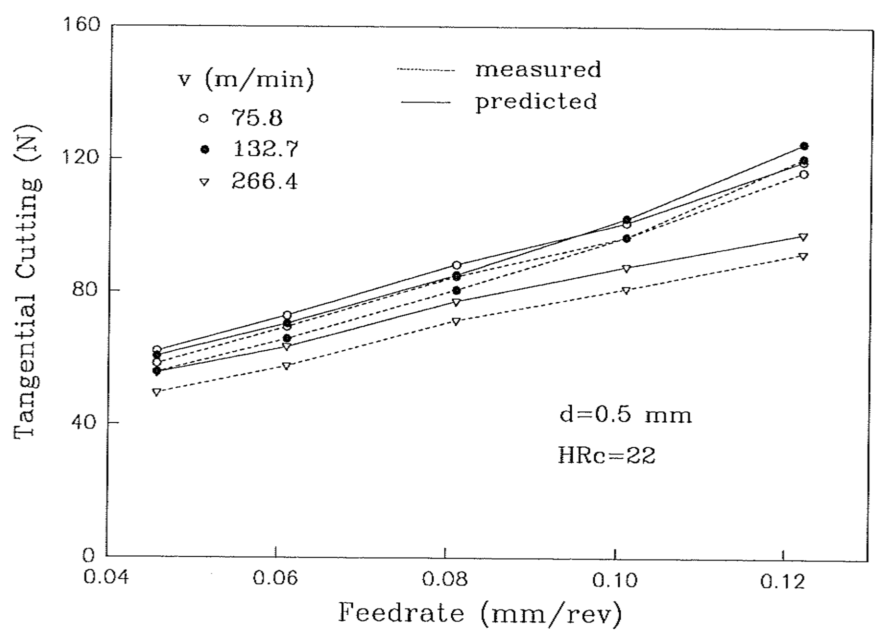

Fig.6.7 Comparison between the measured and predicted tangential cutting force for a steel workpiece having HRc=22.

6.4.3 Estimation of h_f

A comparison between the 29 estimated and measured values of the tool's flank wear is presented in Fig.6.8 for Test 1 . Good correlation can be observed although the estimates are always higher. This higher estimation might be the result of utilizing a somewhat high initial flank wear data, h_{f0}, calculated from equation (6.4). Fig. 6.8 also indicates that it is possible to have small and theoretically unaccounted increases in the spindle speed caused by "soft spots" in the workpiece's material. However, a reduction in the tool flank wear due to this speed increase is physically impossible. Progressive flank wear will always cause a consistent cutting force increase which, in turn, results in a spindle speed reduction. Hence, these observed speed increases were automatically set to zero in the model This discrepancy might also have contributed to the higher estimated h_f.

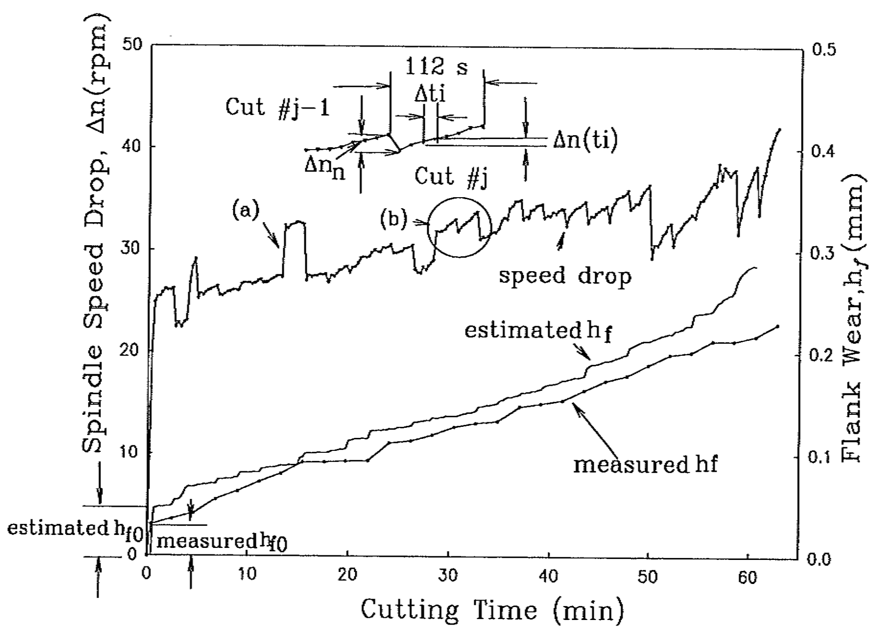

Fig.6.8 Temporal changes in the spindle speed drop, Δn, measured and estimated flank wear, h_f, for Test 1.

A close examination of Fig.6.8 reveals several discontinuities in the spindle speed drop compared with that at the previous sampling point. Test 1 consists of 29 cuts, with a cutting time of 112 s for each cut. A total of 4,032 data points were obtained for the spindle speed during this period. Within each cut, eight speeds were obtained by a moving average on the 4,032 sampled points so that eight estimations of the flank wear were made. At the end of Cut #(j-1), the workpiece was reversed (as explained on Page 105 of in Section 6.3) before Cut #j was initiated. The insert within Fig.6.8 shows the magnified section (b) of Fig.6.8. The discontinuity between Cut #(j-1) and Cut #j is marked by the spindle speed drop, Δn_n, shown in the figure. This speed drop represents a difference in the speed between the first sampled point of Cut #j and the sampled data when the machine is unloaded. It is caused primarily by the loading of the lathe's spindle when the tool contacts the workpiece at the onset of cutting. The tool's flank wear progresses from this point during Cut #j. The continuous reduction in the sampled speed during the cut is then caused by the progressive tool's flank wear during that cut. This effect can be seen clearly by a consistent decrease in the spindle speed shown in Fig.6.8. Therefore, to calculate the speed drop due to the tool wear alone, the speed at the very first sampled point of each cut was subtracted from subsequently sampled speeds associated with that cut.

The jump between adjacent Cuts #(j-1) and #j are not considered in estimating the flank wear. The justification for this strategy can be illustrated by considering point (a) shown in Fig.6.8. This point corresponds to the beginning of Cut #7 where the depth of cut was increased from 0.75 mm to 1.0 mm. As a result of the sudden change in the corresponding load on the motor, an abrupt speed decrease can be observed in the figure. However, this abrupt speed change can be seen to have almost no effect on the estimated flank wear. Consequently, the approach outlined above can be seen to give a good estimate of the tool flank wear and, hence, it forms a reliable basis for predicting the surface roughness.

6.4.4 Prediction of R_{max}

The predictions of R_{max} for the first 50 minutes of Test 1 are shown in Fig.6.9 by discrete points. They generally agree with the measured R_{max} except that the predictions increase more smoothly over time. The increase in R_{max} is largely a consequence of the tool wear component, $R_{wr}(t)$ because the other two components, R_{th} and u(t), remain almost unchanged. On the other hand, u(t) has the largest magnitude of the three components whilst R_{th}, as expected, has by far the least. The contribution of the vibration component, u(t), to the total predicted R_{max} depends upon the structural parameters of the dynamometer used in this study.

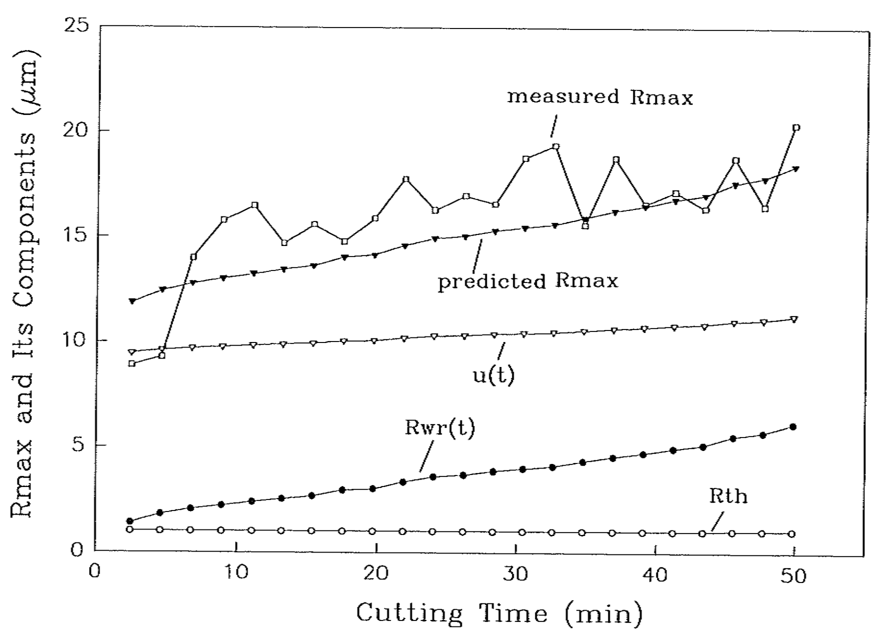

Fig.6.9 Comparison between the measured and predicted R_{max} for Test 1.

Figs. 6.10 and 6.11 give comparisons of the predicted and stylus-measured R_{max} obtained from Tests 2 through 6 and Tests 27 through 31, respectively. In addition to the averaged R_{max}, which are obtained by arithmetically averaging the stylus-measured R_{max}, upper and lower values of the stylus R_{max} are also shown. It can be seen that most predictions fall within the given bounds of the stylus measurements. However, they tend to be larger than the average stylus-measured R_{max} except at a 0.0457 mm/rev. feedrate and a 68.4 m/min cutting speed. At this very light feedrate and low cutting speed, a longer time is available for plastic deformation of the surface layer of the workpiece's material. Such a deformation usually causes swelling of the material towards the free surface [S64]. The present model neglects this effect. On the other hand, an average error of only 10%, in relation to the bounds of the stylus-measured R_{max}, was obtained from Tests 2 through 31. As shown in Figs. 6.9 and 6.10, the difference between the predicted and average stylus R_{max} increases with the feedrate. It was generally observed that the cutting process becomes more stable as the feedrate increases. As a result, the surface roughness caused by the vibration will decrease as the feedrate increases. However, this effect was not included in the model.. This difference might contribute to a higher prediction of R_{max} at a greater feedrate. Although the effect of the cutting speed has been incorporated into equation (6.16), the effect of the feedrate needs further investigation.

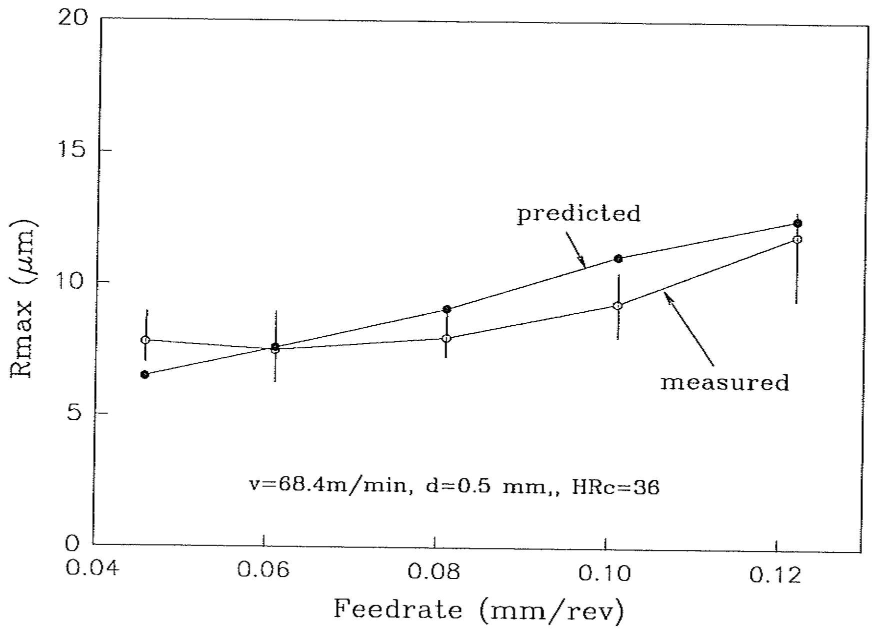

Fig.6.10 Comparison between the measured and predicted R_{max} for Tests 2 through 6.

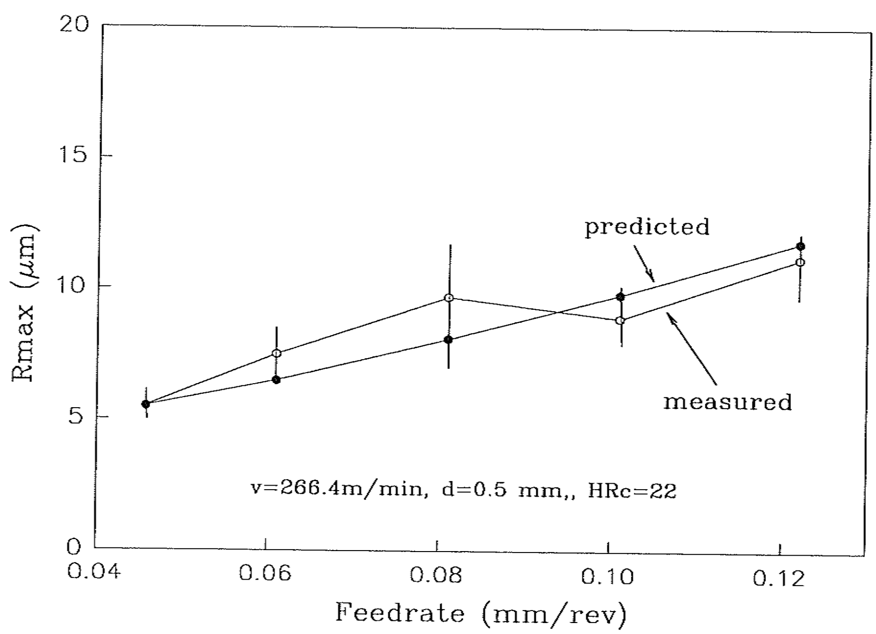

Fig.6.11 Comparison between the measured and predicted R_{max} for Tests 27 through 31.

6. 5 Remarks

The comprehensive model proposed in this chapter incorporates the nominal cutting variables and the tool's geometry as well as its flank wear obtained from on-line monitoring. The R_{max} can be predicted, point by point, as the turning proceeds. Good agreement is obtained with stylus-measured R_{max} under different finish turning conditions, particularly at the lowest feedrates used. The predicted R_{max} is always dominated by the vibration component but the tool wear component increases steadily so that it becomes more important as the cutting approaches the final stage of tool life. On the other hand, swelling, produced by plastic deformation on the workpiece's surface layer at a very low feedrate and cutting speed, should not be ignored and the model should be refined to also accommodate this effect.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

The three approaches proposed in this study represent an effort to provide a framework which allows the practical realization of an on-line and real-time assessment of the surface roughness produced in finish turning. The main achievements will be summarized in the following subsections.

7.1.1 On-line measurements

1. The R_{max} is quantified on-line by solving the inverse problem of light scattering. The maximum difference between the average optically-determined and standard off-line measured R_{max} with a stylus instrument was found to be less than a respectable 15% in the feed direction in a dry cutting environment.
2. The measuring range of R_{max} has been extended to about 20 μm which can be successfully applied to a finish turning operation.
3. The scheme to iteratively search for the measurement correction factor, K, makes it possible not to need *a priori* knowledge of its value. This also avoids the problem of requiring the stylus data as the input as encountered in [MV90].

4. The CCD-based laser system is capable of capturing, on-line, the ASLP in the feed direction. The influences of spindle speed changes on the optical data were also investigated and minimized.

5. The method to compensate for the influences of the ambient light on the ASLP is feasible

6. An oil-based cutting fluid has a significant influence on the measured ASLP data which, in turn, results in a large discrepancy in the estimation of R_{max}. Its influence needs further investigation.

7.1.2 Real-time measurements

1. The approach, which entails the ASLP reflected from the workpiece's surface by using the laser measuring system and employs a simple, three-layer neural network involving back propagation learning, can be used to measure the R_{max}, in real-time, in finish turning. The results from the cutting tests show that the new approach is more robust and responds much faster than other optical methods.

2. The predicted R_{max} values have a maximum error of about 10% when compared with conventional stylus measurements.

3. The training of the neural network converges quickly due to the efficient mapping between the ASLP and the required R_{max} output.

4. The influence of ambient light has been effectively removed by carefully defining the input light data.

5. The performance of the neural network depends mainly on the selection of training patterns. An iterative training procedure is recommended.

6. The lateral scattering caused by extraneous vibrations at a very high cutting speed (e.g. $V > 200$ m/min) will affect the measured ASLP. Its effect on the performance of the neural network needs further investigation.

7.1.3 On-line prediction

1. A comprehensive model is proposed to incorporate the main cutting variables such as cutting speed, feedrate, depth of cut and hardness of workpiece's material as well as the cutting tool's geometry. The effect of a tool's wear on the surface roughness has been taken into account by estimating the flank wear on line.
2. It is easy to implement the proposed model because the nominal cutting variables and tool geometry are the only input. This model can be applied to surface finish control in turning by adjusting the above cutting variables in finish turning. Tool wear monitoring using spindle speed measurement is remote and does not interfere with the cutting process.
3. The R_{max} is predicted point-by-point as the machining proceeds. The prediction agrees well with stylus-measured R_{max} under different finish turning conditions, particularly at the lowest feedrates used.
4. It is found that the predicted R_{max} is always dominated by the vibration component but the tool wear component increases steadily so that it becomes more important as the cutting approaches the final stage of the tool life.
5. The model needs to be refined to accommodate the effect of swelling at a very low feedrate and cutting speed.

7.2 Recommendations

Finally, the author would like to make the following recommendations for future work.

7.2.1 Calibration of the laser system

The range of the correction factor, K, has a direct effect on the estimation of R_{max}. As a result, it is important to calibrate the laser system to check the values of K_a

and K_b from time to time. The calibration has to be done periodically when the initial setting of the laser system is expected to be disturbed. Obviously, this is inconvenient. Hence, it is useful to devise methods to make the present calibration procedure faster, easier and more reliable. Several possible modifications may include:

- fine tuning mechanism
- display device
- zero-point adjustment to compensate for any offset.

7.2.2 Extension of the measuring range beyond 20 μm

It was found that the optically-determined R_{max} above 20 μm were usually lower than the stylus data. The main reason is that the measurement accuracy will deteriorate because the detectability of the measuring system decreases with increasing surface roughness as shown in Fig.4.19. To improve this deficiency, two options are recommended:

- use a 16-bit A/D converter to increase the resolution
- increase the collecting range to enhance the ASLP.

7.2.3 Compensation for the ambient light

Two kinds of compensation technique were adopted in this study. In Chapter Four, a reference ambient light level was used as the comparator in an empirical way but apparently this value can not be considered to be constant because it changes with the system's sensitivity. The sensitivity is affected by the calibration as mentioned in Section 7.2.1. The compensation procedure outlined in Chapter Five was performed simply by subtracting an ambient level obtained by averaging the sampled CCD data outside the 1,000-pixel control region. It was found that this kind of averaging was environmentally dependent because the actual ambient light does not correspond to a dc offset. Hence, a more sensitive and robust compensation technique should be developed. A recommended method would be to use two laser beams with one as the reference and the other measuring the scatter from the surface being inspected. The use of a band pass filter may also be investigated.

7.2.4 Converting R_{max} to R_a or R_q

The R_{max} was chosen as the roughness parameter in Chapters Four and Five because of its straightforward relationship with the ASLP. The predictive model, on the other hand, was used to determine the R_{max} in the direction perpendicular to the surface. The vibration and tool's nose change were also defined in this direction. However, R_{max} is rarely used in industry because it is, by nature, a very unstable parameter. In contrast, R_a and R_q are two roughness parameters recommended by ISO (International Standards Organization). Further effort is required to accommodate R_a (or R_q) into the approaches proposed in this study.

7.2.5 Effect of the concentrated groove wear at the cutting tool's trailing edge on surface finish

It was pointed out [S58] that concentrated groove wear was often observed on the trailing edge of the tool's flank face when turning steel using a carbide tool. This type of concentrated groove wear had a significant influence on the generated surface finish. Although this effect was considered by Solaja [S58], who simply introduced a factor of 1.33, this kind of treatment is not justified because the concentrated groove wear is not uniform in size and its occurrence is also affected by the cutting conditions. To explore this interesting phenomenon and incorporate it into the predictive model, further investigations are needed.

7.2.6 Cutting fluid

The effect of the water-based cutting fluid on the measured ASLP can be minimized by using an air nozzle to blow out the water-based film on the illuminated spot of the workpiece. To deal with the oil-based cutting fluid having a high viscosity, a more sophisticated calibration procedure may be required.

7.2.7 Comments on C_p and K_s

C_p in equation (5.4) and K_s in equation (5.10) are two important constants in estimating the tool's flank wear, h_f, and the principal cutting force, F_p, respectively. Unfortunately, both constants have not been well documented. At this stage, a series of cutting tests have to be performed which should be similar to those conducted by other researchers [K91; B84] to find their values. However, the author believes that it is possible to determine, *in process*, the values of these two constants by indirectly monitoring certain process variables which are observable during cutting, viz. the spindle speed change. There is no doubt that this topic, although very challenging, is worthy of merit.

APPENDIX A: Numerical assessment of surface roughness

1. Mean-line system

The roughness parameters defined in the mean-line system are described below. (See also Fig.2.2.)

(1) Leveling depth, R_p, is the distance between the mean line, M.L., and the upper reference line, U.L.

(2) Complementary depth, R_v, is the distance between M.M. and the lower reference line, L.L.

(R_p and R_v are often necessary for assessing the frictional quality of a surface.)

(3) Total depth, R, is the peak-to-valley value which is perceived by looking at and touching the surface. It can be expressed as: $R = R_p + R_v$.

(4) The center-line-average roughness value, CLA, or arithmetic average roughness, AA or R_a , is defined as "the arithmetical average value of the departure of the whole of the profile both above and below its center line throughout the prescribed meter cut-off in a plane substantially normal to the surface" [BSI61]. Here, the center line is a line below and above which there is an equal area between it and the surface profile within the sampling length, L. Mathematically, R_a is given by

$$R_a = \frac{1}{L}\int_0^L |y|\, dx. \tag{A1}$$

Although R_a does not describe any actual depth of an irregularity on the surface, it leads to easy and reliable instrumentation and demonstrates little scatter. Hence, it provides a useful datum for many functional purposes.

(5) The root-mean-square value, RMS or R_q, is "the geometrical average value of the departure of the whole of the profile both above and below its center line throughout

the prescribed meter cut-off in a plane substantially normal to the surface" [BS61]. Mathematically, it can be expressed as:

$$R_q = \sqrt{\left(\frac{1}{L}\int_0^L y^2 dx\right)}. \tag{A2}$$

(6) The maximum peak-to-valley roughness, R_{max}, is the maximum distance between two reference lines which are parallel to the mean line touching the highest and lowest points within the sampling length, L, as shown in Fig.A1.

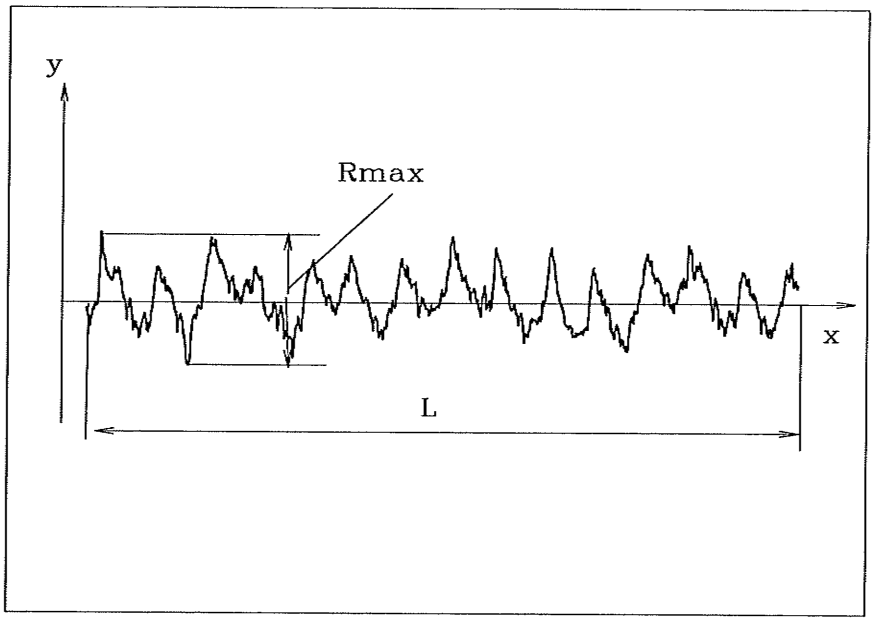

Fig.A1 The maximum peak-to-valley roughness, R_{max}.

(7) Ten-point average height, R_z, is the difference between the average height of the five highest peaks and the five lowest valleys within the sampling length, L, measured with respect to a datum parallel to the mean line, as shown in Fig.A2. It can be written mathematically as:

$$R_z = \frac{(R_1 + R_3 + R_5 + R_7 + R_9) - (R_2 + R_4 + R_6 + R_8 + R_{10})}{5}. \qquad \text{(A3)}$$

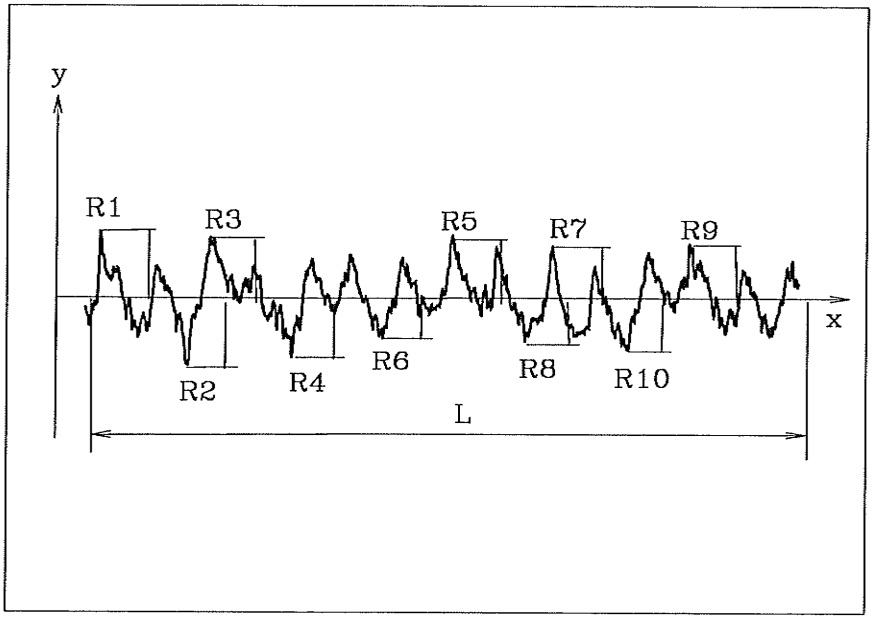

Fig.A2 Ten-point average peak-to-valley roughness, R_Z .

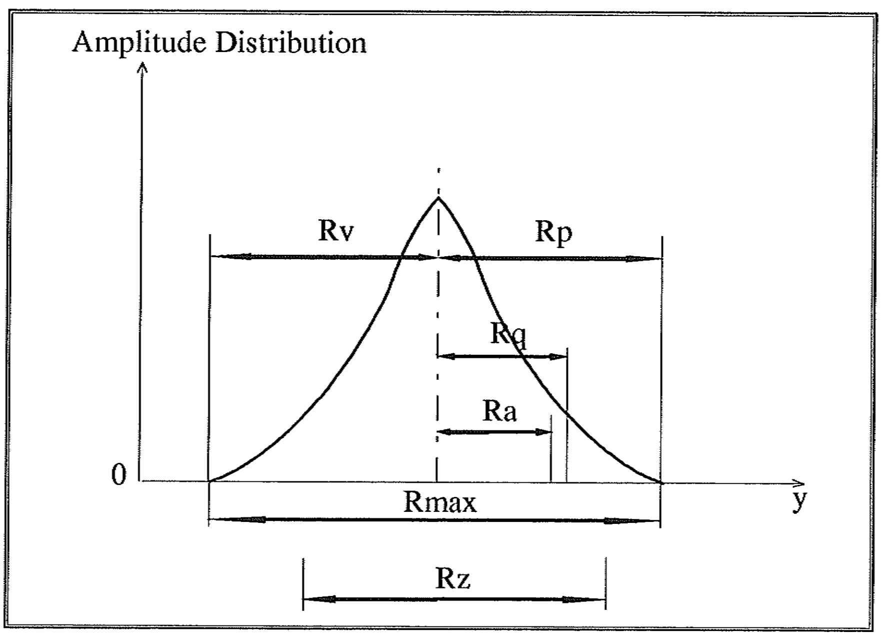

Fig.A3 Amplitude probability density function of a surface profile.

The above roughness parameters can also be described by the amplitude probability density function of the surface profile, represented by a stationary random function with ergodic properties, as shown in Fig.A3.

2. Crest line measures (Nicolau's method)

A datum is obtained by either contacting the highest peaks, as shown in Fig.A4, or the lowest valleys. Then the surface roughness is defined as the mean distance of the surface profile from the datum within the sampling length, L, i.e.,

$$G_M = \frac{1}{L}\int_0^L y\,dx. \tag{A4}$$

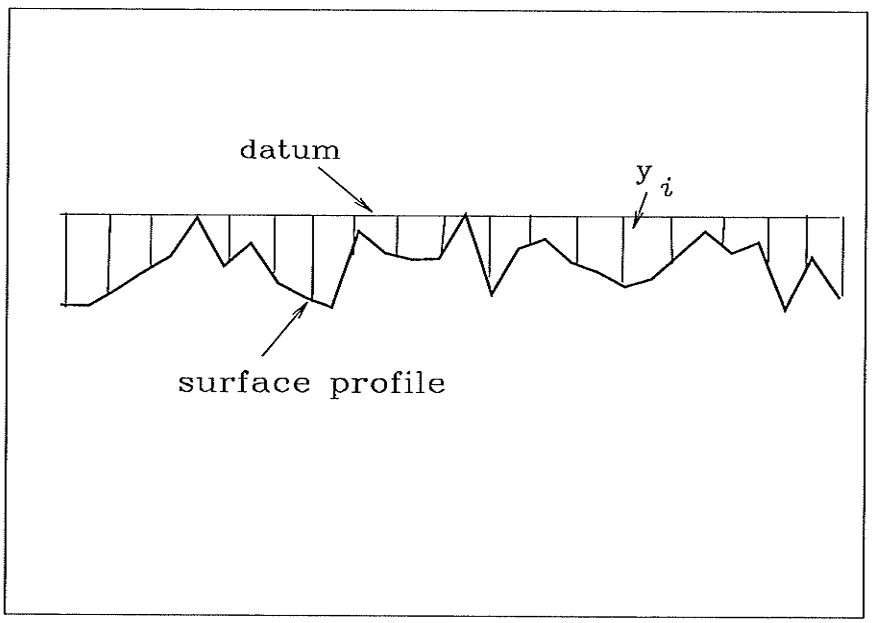

Fig.A4 Crest measures.

3. Envelope method

A crest line is generated from the locus of the center of a rolling circle moving across the surface as shown in Fig.A5. A contacting envelope is obtained by moving the crest line towards the surface until it contacts the crests of the surface. The mean depth of the profile below the contacting envelope and within the sampling length, L, is given by:

$$G_E = \frac{1}{L}\int_0^L y\,dl. \tag{A5}$$

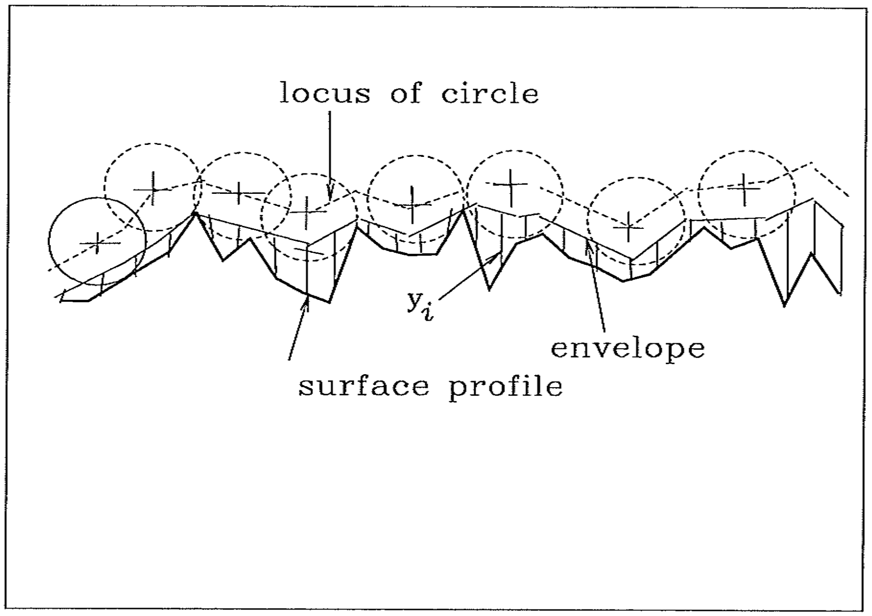

Fig.A5 Envelope measures.

It is obvious from the above definition that either the waviness or the roughness of the surface is evaluated depending upon the selection of the roller's radius. In the German standard DIN-4762(1960), a mean curve is obtained when the contacting envelope is displaced to a position below and above which there is an equal area between it and the effective surface profile. The mean curve in the envelope method corresponds to the mean line obtained from the mean line system. If the contacting envelope is displaced further to pass through the deepest valleys, the lower contacting envelope is obtained. Thus, the envelope system is similar to the mean line system. The main disadvantage of the envelope system is that complicated instrumentation is required to evaluate the surface texture.

4. Autocorrelation function

Assume that the distance along the surface is *x* and that the waveform representing the surface profile is y(x). (See Fig.A6.) The autocorrelation function of the surface, R(β), is defined as the expected value of y(x) multiplied by y(x+β), where β is an offset distance in the *x* direction, i.e.

$$\begin{aligned} R(\beta) &= E(f(x)f(x+\beta)) \\ &= \lim_{L\to\infty}\frac{1}{L}\int_{-\frac{L}{2}}^{\frac{L}{2}} f(x)\, f(x+\beta)\, dx. \end{aligned} \tag{A6}$$

If β=0, the autocorrelation function becomes the dispersion of the surface profile, or R_q^2.

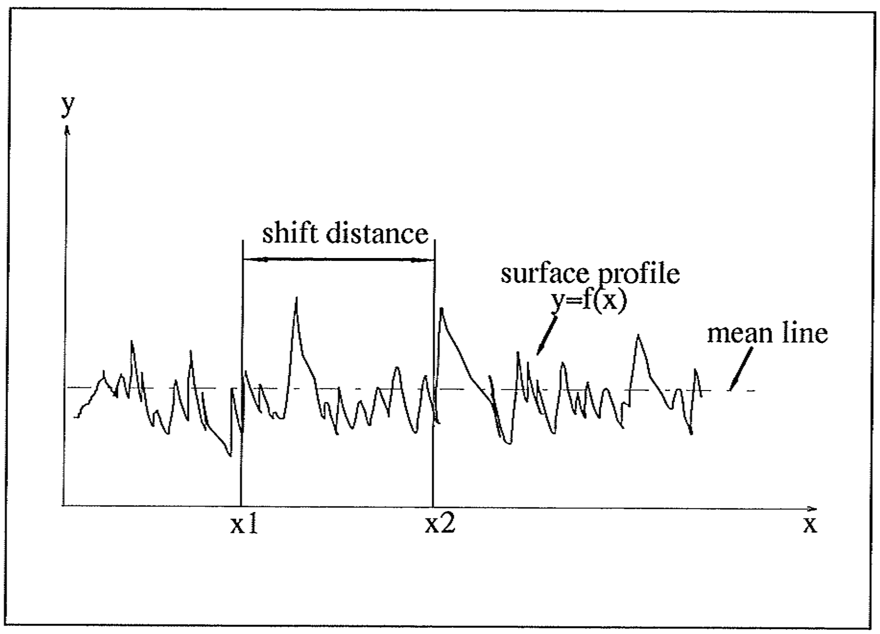

Fig.A6 A random surface profile.

A surface generated in a machining operation will lie between two extreme basic profile descriptions of a deterministic function without random distortion and a wide band random noise. The profiles considered in the first group are the line (Fig.A7(a)) and sine wave (Fig.A7(c)) whose autocorrelation functions are shown in Figs. A7(b) and (d), respectively. It is well known that such theoretical surfaces never exist in practice. However, the autocorrelation functions represent important elements included in the actual surfaces. The autocorrelation function of the second group of surface profiles (Fig.A7(e)) is approximated by an exponential function, as shown in Fig.A7(f). A real surface profile can be distinguished by defining the sum or product of the autocorrelation functions of the above two limiting profiles. Figs.A7(g) and (h), for example, show a mixed surface profile and its autocorrelation function, respectively.

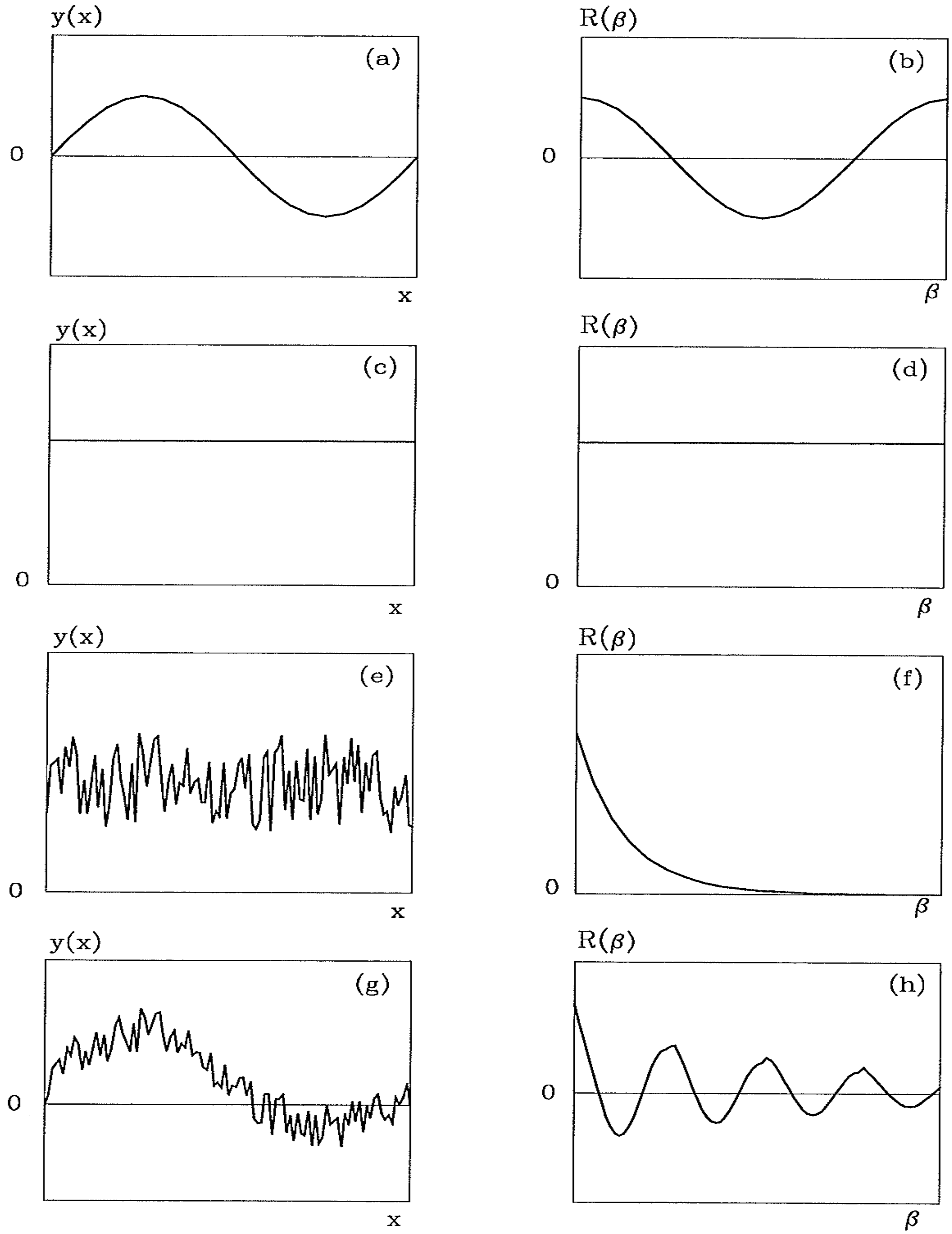

Fig.A7 Different surface profiles, *f(x)*, and their autocorrelation functions, R(β). (a), (b): sine wave; (c), (d): line; (e), (f): wide-band random noise; (g), (h): mixed profile.

5. Power spectral density function

The Weiner-Khinchin theorem provides the Fourier cosine transform that relates the autocorrelation function, $R(\beta)$, to the power spectral density function, $S(\omega)$, as

$$S(\omega) = \frac{2}{\pi}\int_0^{\infty} R(\beta)\cos\omega\beta \, d\beta. \tag{A7}$$

The power spectral density function is used to characterize both the asperity amplitude and spacings of the surface. The inverse of each spatial frequency, ω, is the spatial wavelength of that component. The power spectral density function for a ground surface , for example, demonstrates a fairly smooth distribution of the surface wavelengths due to random nature of the surface.

The relation between the power spectrum, $S(\omega)$,and variance, D_x of a stationary surface profile, y(x), is given by [B84]

$$D_x = \int_{-\infty}^{\infty} S(\omega)d\omega = R(0). \tag{A8}$$

APPENDIX B: Program Code

Program 1.	Snapshot.c
Program 2.	INVERSE.FOR
Program 3.	Neural.c
Program 4.	New_flan.c
Program 5.	Est_Rmax.c

Program 1:

```
/*************************************************************
                    Snapshot.c

            Last revised : 02/06/92
The code is used to sample ASLP data on-line.The program now includes
NEW software which addresses 318H and 319H. These addresses are decoded
with the LS32(B) OR gate using base address 318H (select 6) and the buffered A0 line.
A second D flipflop (FF1)(1/2 LS74HC74) is utilized to control the
CLEAR on the other D FF2 in the same chip. The reason for this is to
provide the clock pulses for the 4040's during the READing of the
SRAM. If the CLEAR(1) is not maintained LOW during the SRAM
READ the 4040 counters WILL NOT be incremented and erroneous data
will be obtained. To trigger FF1, a READ is made on address 30CH
connected to CLOCK1 which puts (Q bar) LOW.
To reset FF1 a READ is made at address 318H connected to the
CLEAR1. This effectively sets (Q bar) HIGH allowing a READ of
address 319H to reset FF2.

            28/05/92

It appears that the d1 delay factor can be made a constant value of 30.
This was arrived at after considerable experimentation
Also only 1 snap shot needs to be taken when d1 is set to 30
(the corresponding delay time is about 3.75 µs). These
factors are related directedly to the DISCHARGE rate of the CCD array
and the timing circuitry. There may be a small variation in the
CCD response due to the counting loop variations in the software.

Revised 23/10/92: averaging on three snapshots cancelled.
Total of 36 single snapshots are written into 4 data files or 18 single
snapshots written in two data files.
*************************************************************/
#include <stdio.h>
#include <alloc.h>
#include <math.h>
#include <io.h>
#include <time.h>
#include <string.h>
#include <conio.h>
#include <dos.h>
```

```
#include <stdlib.h>

/* Set addresses on CUSTOM interface board*/

#define    START_CONVERT     0x300
#define    STATIC_RAM       0x304
#define    RESET_4040S      0x310    /* 784dec    */
#define    CCD_FIRE         0x314    /* 788dec    */

#define    CLEAR_DFF1       0x318    /*WRITE to   */
#define    CLEAR_DFF2       0x319    /* SRAM      */

#define    TRIGGER_READ     0x30c    /*READing from */
                                     /* SRAM      */

#define    FINISHED         0x31c    /*dataline d0 */
#define    LATCH_ENCODER    0x31c    /*dataline d7 */

/*variables*/

unsigned char far *ram8=NULL;
unsigned char far *ram9=NULL;

struct date today;
struct time now;

int j,snap,k,i,sample_no,xch,pixel_count;
int x,y,d1,no_reads,top;
int not_needed,max_snaps;

unsigned num_pixels=3489;
unsigned m1,m2,m3,m4,m5,m6,m7,m8,m9;
unsigned m10,m11,m12,m13,m14,m15,m16,m17,m18;
unsigned ct,count,kk,readings;

unsigned long maxsize=63000L;

unsigned char start,resett,fire_ccd;
unsigned char trigger,clear_d;

unsigned long val;

char wp[3],s[15],code[10];
```

```
char name1[15],name2[15],name3[15],name4[15];

clock_t start_t,end_t;

float distance,spindle,feedrate;
float one_rev,no_of_rev,tot_time;



main()
/*int argc;
char *argv[];*/

{

FILE *data_out1,*data_out2, *data_out3,*data_out4;


/**************************START PROGRAM*********************/
/* speed, feedrate can be input on command line. */
/*if(argc == 6)
  {*/
  printf("enter a three-character workpiece code!\n");
  scanf("%s",code);
  printf("enter speed (rps) and feedrate (mm/rev)!\n");
  scanf("%f%f", &spindle, &feedrate);
/*  spindle = atof(argv[1]);
  feedrate = atof(argv[2]);*/

  /*d1 = atoi(argv[ ]);  set at a constant 30... 01/06/92      */
  d1 = 30;
/*  no_reads = atoi(argv[3]);
  distance = atof(argv[4]);*/
  /*snap = atoi(argv[ ]); set at a constant 1 ...28/05/92      */
  printf("enter number of readings!\n");
  scanf("%d", &no_reads);
  distance = 144.0/no_reads ;
  snap = 1;
  if( no_reads % 9 )
    {
    clrscr();
    printf("\n\n\nSample number must be a number of 9X (e.g. 9, 18 or 36)\n\n");
    printf("\nThe time for %3.1fmm, based on speed & feedrate\
```

```
                    is %6.3fsec\n\n",distance,tot_time);
      exit(1);
      }
   /* }
  else
   {
   clrscr();
   printf("File snapshot.exe... Last revision 23/10/92\n");

   printf("\n1st line argument...\n");
   printf("Enter the spindle speed in RPS...(float)\n");

   printf("\n2nd line argument...\n");
   printf("Enter the feed rate (mm/rev)...(float)\n");

   printf("\n3th line argument...\n");
   printf("Enter a value for number of samples to be taken..(int)\n");
   printf("Number of samples should be divisible by 9!\n");

   printf("\n4th line argument...\n");
   printf("Enter the LENGTH in 'mm' between samples..(float)\n");

   printf("\n5th line argument...\n");
   printf("Enter a 3 letter workpiece code..(char)\n");

   printf("\nExample...'newslave 19.26 .0457 18 1.7 12b'\n");
   exit(1);
   }  */

  clrscr();

  /*Allocate memory for data   */

  /*printf("\n%lu",farcoreleft());*/
  ram8 = (unsigned char far *)(farmalloc(63000L));
  /*printf("\n%lu",farcoreleft());*/

  if (ram8)
    puts("\nram8 allocation ok.");
  else
    {
    puts("\nProblems with ram8 memory allocation!!");
    exit(1);
```

```
  }

 ram9 = (unsigned char far *)(farmalloc(63000L));
 /*printf("\n%lu",farcoreleft());*/

 if (ram9)
   puts("\nram9 allocation ok.");
 else
   {
   puts("\nProblems with ram9 memory allocation!!");
   exit(1);
   }

 /*Set up the filenames & extensions    */

 strcpy(name1,"snap1.");
 strcpy(name2,"snap2.");
 strcpy(name3,"snap3.");
 strcpy(name4,"snap4.");
 strcat(name1,code);
 strcat(name2,code);
 strcat(name3,code);
 strcat(name4,code);

 /*Determine approx. time for feed movement of X.0 or 'distance' mm */
 one_rev = 1.0/spindle;          /*time for 1 revolution      */
 no_of_rev = distance/feedrate;  /*num of revs in X.0mm       */
 tot_time = no_of_rev*one_rev;   /*total time for X.0mm       */

 sample_no = 1;    /*first sample*/

 printf("\n\nProgram to capture images from the CCD array....\n\n");
 printf("Calculated time for %3.1fmm based on speed & feedrate %6.3fsec",
   distance,tot_time);

 printf("\n\nPress any key to START sampling................");
 getch();

 /***********************************************************/

 /* CLEAR FF1 for WRITE to SRAM */          /* Q1 bar set high */
 start = inportb(CLEAR_DFF1);
```

```
/* CLEAR FF2 */                    /* Q2 set low     */
start = inportb(CLEAR_DFF2);

/* RESET 4040s    */
resett = inportb(RESET_4040S);

/*********************************************************************/

clrscr();

ct = 0;        /*set counters*/
top = 1;        /*use ram8 for first 18 samples & ram9 for remainder*/
pixel_count = 0;

val = (unsigned long)(no_reads)*(unsigned long)(num_pixels);

printf("\n\n");
printf("\nWriting and transferring %d samples from SRAM...\n",no_reads);
printf("representing %lu individual pixel reads...\n",val);


x = 30;
y = 10;
gotoxy(x,y);

printf("COMPLETED SAMPLE #");

x = wherex();
y = wherey();

/*First sample will be taken 6.0 mm from start of cutting*/

start_t = clock();
end_t = clock();
while( ((end_t-start_t)/CLK_TCK) <= (6.0/(feedrate * spindle)) )
        end_t = clock();

while( kbhit() )
 getch();                        /*clear keyboard buffer */

while( sample_no <= no_reads )
 {
/*wait for encoder pulse, latch and READ              */
```

```
 while( !(inportb(LATCH_ENCODER) & 0x80) );   /* read bit 7*/

/*for(i=0; i<20000; i++) simulated encoder wait state
k=i;                                          */

 /*send pulse to FIRE ccd  (encoder trigger) */
    printf("\n");
/*   printf("Start sampling the first ASLP!\n");   */

    start = inportb(START_CONVERT);
    fire_ccd = inportb(CCD_FIRE);

    while(inportb(FINISHED) & 01);  /*Finished 3489 pixels???*/
                                      /*NOT STORED in 128K SRAM*/
/*   printf("First ASLP finished but discarded!\n");   */
    printf("\n\n");  /*required for running the program.  */

   for(j=0; j<snap; j++)
   {
    /*add a small delay (d1)....SETS EXPOSURE TIME..........
                 entered as the third COMMAND line argument          */
                for(i=0; i<d1; i++)
                        k=i;

                /*reset DFF2 ready for next encoder pulse capture        */
                clear_d = inportb(CLEAR_DFF2);

    /* Now fire again (reduced EXPOSURE trigger)              */
/*   printf("Start sampling the second ASLP!\n");   */

        start = inportb(START_CONVERT);
        fire_ccd = inportb(CCD_FIRE);

    while(inportb(FINISHED) & 01);  /*Finished 3489 pixels???*/
                                      /*STORED in 128K SRAM    */
/*   printf("Second ASLP finished and stored!\n");   */

   } /*end for loop*/

/*Start the timing for the next 'distance' mm length */
start_t = clock();

                /*THEN READ the 8K RAM right away    */
```

```
if(top)
 {
 /*     if(sample_no == 1)
        printf("first, measure the ambient light!\n");
        else
        if(sample_no == 2)
        {
        printf("now, open the laser cover to start SAMPLING!\n");
        printf("press any key if done.\n");
        getch();
         } */

  pixel_count = 0;
  do {
    *(ram8+ct) = inportb(STATIC_RAM);
    ct++;
    pixel_count++;
    }while(pixel_count<3489);

  if( ct > 62780L )       /*make sure #19 READ is NOT stored */
    {                /*else 64K boundary will be wrapped*/
     top = 0;
     puts(" .. 2nd file sets..");
     ct = 0;
     }
  }
else
 {
 pixel_count = 0;

 do {
   *(ram9+ct) = inportb(STATIC_RAM);  /*store #19-36 reads */
   ct++;
   pixel_count++;
   }while(pixel_count<3489);
  }

  /*put in a reset for the continuous cycling mode */
 /*if(ct >= (val-36)  ) */  /*125567 max */
  /*  ct = 0L; */

 gotoxy(x,y);
 printf("%d",sample_no);
```

```
sample_no++;

/*Now determine the delay if required so that the 36 readings
 are distributed along the 150mm workpiece... ie approx.
 every 4.0 mm. This depends on spindle speed and feedrate is
 determined by the command line arguments before each cutting
 session begins (tot_time) for 4.0 mm.              */

end_t = clock();
while( ((end_t-start_t)/CLK_TCK) <= tot_time )
end_t = clock();

/*Now wait for the NEXT encoder pulse and REPEAT process */

} /*END WHILE STATEMENT */

top = 1;
sample_no--;

clrscr();
printf("Number of pixels read....%u\n",val);

printf("\n\n\n%d samples completed...\n\n",sample_no);
/* printf("AVERAGING of 3 sets of data taking place....\n\n");*/

/*Averaging 3 sets of data..........  */
/* ie 0,1,2  3,4,5  6,7 8  etc      */

/*ct = 0;
kk = 0;
count = 0;
readings = no_reads - 3;

do {
  do {
        m1 = (num_pixels*count) + ct;
        m2 = (num_pixels*(count+1)) + ct;
        m3 = (num_pixels*(count+2)) + ct;
        *(ram9+kk) = ( (*(ram9+m1)) + (*(ram9+m2)) + (*(ram9+m3)) )/3;
        ct++;
        kk++;
        }while( ct < num_pixels);
  count+=3;
```

```
    ct = 0;
          }while( count<= readings && count <= 15 );

          if( no_reads > 18 )
            {
            ct = 0;
            kk = 0;
            count = 0;
            readings = no_reads - 21; */    /*increments of 3 only*/

/*          do {
                    do {
                              m1 = (num_pixels*count) + ct;
                              m2 = (num_pixels*(count+1)) + ct;
                              m3 = (num_pixels*(count+2)) + ct;
                              *(ram9+kk) = ( (*(ram9+m1)) + (*(ram9+m2)) + (*(ram9+m3))
)/3;
                              ct++;
                              kk++;
                              }while( ct < num_pixels);
        count+=3;
        ct = 0;
                    }while( count<= readings );
          }

printf("FILING of averaged data sets taking place....\n");*/

/*Now open file(s) as required and print data..... */
getdate(&today);
gettime(&now);

if( no_reads >18 )      /* 4 files required */
 {
        data_out1 = fopen(name1,"w");
  if(data_out1==NULL)
   {
   puts("\nProblems with opening first file");
   exit(1);
   }
        data_out2 = fopen(name2,"w");
         if(data_out2==NULL)
                {
                puts("\nProblems with opening second file");
```

```
            exit(1);
            }
      data_out3 = fopen(name3,"w");
       if(data_out3==NULL)
            {
            puts("\nProblems with opening first file");
            exit(1);
            }
      data_out4 = fopen(name4,"w");
       if(data_out4==NULL)
            {
   puts("\nProblems with opening second file");
   exit(1);
   }

fprintf(data_out1,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
no_reads,code,today.da_mon,today.da_day,today.da_year,
now.ti_hour,now.ti_min,now.ti_sec);

fprintf(data_out2,"%d wp#%s, samples 10-18, %d/%d/%d,time %02d:%02d:%02d\n",
no_reads,code,today.da_mon,today.da_day,today.da_year,
now.ti_hour,now.ti_min,now.ti_sec);

fprintf(data_out3,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
no_reads,code,today.da_mon,today.da_day,today.da_year,
now.ti_hour,now.ti_min,now.ti_sec);

fprintf(data_out4,"%d wp#%s, samples 10-18, %d/%d/%d,time %02d:%02d:%02d\n",
no_reads,code,today.da_mon,today.da_day,today.da_year,
now.ti_hour,now.ti_min,now.ti_sec);

printf("First set of files(1-18) are being processed!.......");
printf("\n\n\n            PLEASE WAIT...... ");

ct = 0;
do {
  m1 = ct;
  m2 = num_pixels + ct;
  m3 = (num_pixels*2) + ct;
  m4 = (num_pixels*3) + ct;
  m5 = (num_pixels*4) + ct;
  m6 = (num_pixels*5) + ct;
  m7 = (num_pixels*6) + ct;
```

```
    m8 = (num_pixels*7) + ct;
    m9 = (num_pixels*8) + ct;
    m10 = (num_pixels*9) + ct;
    m11 = (num_pixels*10) + ct;
    m12 = (num_pixels*11) + ct;
    m13 = (num_pixels*12) + ct;
    m14 = (num_pixels*13) + ct;
    m15 = (num_pixels*14) + ct;
    m16 = (num_pixels*15) + ct;
    m17 = (num_pixels*16) + ct;
    m18 = (num_pixels*17) + ct;

    fprintf(data_out1,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m1)),
            (*(ram8+m2)),(*(ram8+m3)),(*(ram8+m4)),(*(ram8+m5)),
            (*(ram8+m6)),(*(ram8+m7)),(*(ram8+m8)),(*(ram8+m9)));

    fprintf(data_out2,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m10)),
            (*(ram8+m11)),(*(ram8+m12)),(*(ram8+m13)),(*(ram8+m14)),
            (*(ram8+m15)),(*(ram8+m16)),(*(ram8+m17)),(*(ram8+m18)));
    ct++;
    }while(ct < num_pixels);

  farfree(ram8);
  fclose(data_out1);
  fclose(data_out2);
  printf("\n\nBetween files check...ct = %u\n",ct);

  printf("Second set of files(19-36) are being processed!.......");
  printf("\n\n\n             PLEASE WAIT...... ");

  ct = 0;
  do {
    m1 = ct;
    m2 = num_pixels + ct;
    m3 = (num_pixels*2) + ct;
    m4 = (num_pixels*3) + ct;
    m5 = (num_pixels*4) + ct;
    m6 = (num_pixels*5) + ct;
    m7 = (num_pixels*6) + ct;
    m8 = (num_pixels*7) + ct;
    m9 = (num_pixels*8) + ct;
    m10 = (num_pixels*9) + ct;
    m11 = (num_pixels*10) + ct;
```

```
    m12 = (num_pixels*11) + ct;
    m13 = (num_pixels*12) + ct;
    m14 = (num_pixels*13) + ct;
    m15 = (num_pixels*14) + ct;
    m16 = (num_pixels*15) + ct;
    m17 = (num_pixels*16) + ct;
    m18 = (num_pixels*17) + ct;

    fprintf(data_out3,"%d %d %d %d %d %d %d %d %d\n",(*(ram9+m1)),
            (*(ram9+m2)),(*(ram9+m3)),(*(ram9+m4)),(*(ram9+m5)),
            (*(ram9+m6)),(*(ram9+m7)),(*(ram9+m8)),(*(ram9+m9)));

    fprintf(data_out4,"%d %d %d %d %d %d %d %d %d\n",(*(ram9+m10)),
            (*(ram9+m11)),(*(ram9+m12)),(*(ram9+m13)),(*(ram9+m14)),
            (*(ram9+m15)),(*(ram9+m16)),(*(ram9+m17)),(*(ram9+m18)));

    ct++;
    }while(ct < num_pixels);

  farfree(ram9);
  fclose(data_out3);
  fclose(data_out4);
  printf("\n%lu\n",farcoreleft());
  printf("\nFiles %s %s %s & %s have been completed!\n",
       name1,name2,name3,name4);
  }
  else  if (no_reads > 9 )        /*only 2 file required*/
   {
   data_out1 = fopen(name1,"w");
   if(data_out1==NULL)
     {
     puts("\nProblems with opening file");
     exit(1);
     }

   data_out2 = fopen(name2,"w");
   if(data_out2==NULL)
     {
    puts("\nProblems with opening file");
    exit(1);
     }

 fprintf(data_out1,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
```

```
        no_reads,code,today.da_mon,today.da_day,today.da_year,
        now.ti_hour,now.ti_min,now.ti_sec);
 fprintf(data_out2,"%d wp#%s, samples 10-18, %d/%d/%d,time %02d:%02d:%02d\n",
        no_reads,code,today.da_mon,today.da_day,today.da_year,
        now.ti_hour,now.ti_min,now.ti_sec);

 printf("Files(1-18) are being processed!.......");
 printf("\n\n\n            PLEASE WAIT...... ");

 ct = 0;
 do {
   m1 = ct;
   m2 = num_pixels + ct;
   m3 = (num_pixels*2) + ct;
   m4 = (num_pixels*3) + ct;
   m5 = (num_pixels*4) + ct;
   m6 = (num_pixels*5) + ct;
   m7 = (num_pixels*6) + ct;
   m8 = (num_pixels*7) + ct;
   m9 = (num_pixels*8) + ct;
   m10 = (num_pixels*9) + ct;
   m11 = (num_pixels*10) + ct;
   m12 = (num_pixels*11) + ct;
   m13 = (num_pixels*12) + ct;
   m14 = (num_pixels*13) + ct;
   m15 = (num_pixels*14) + ct;
   m16 = (num_pixels*15) + ct;
   m17 = (num_pixels*16) + ct;
   m18 = (num_pixels*17) + ct;

  fprintf(data_out1,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m1)),
            (*(ram8+m2)),(*(ram8+m3)),(*(ram8+m4)),(*(ram8+m5)),
            (*(ram8+m6)),(*(ram8+m7)),(*(ram8+m8)),(*(ram8+m9)));

  fprintf(data_out2,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m10)),
            (*(ram8+m11)),(*(ram8+m12)),(*(ram8+m13)),(*(ram8+m14)),
            (*(ram8+m15)),(*(ram8+m16)),(*(ram8+m17)),(*(ram8+m18)));

  ct++;
  }while(ct < num_pixels);

farfree(ram8);
fclose(data_out1);
```

```
fclose(data_out2);
printf("\n%lu\n",farcoreleft());
printf("\nFiles %s %s has been completed!\n",name1,name2);
  }

else    /* one file required  */
  {
  data_out1 = fopen(name1,"w");
  if(data_out1==NULL)
   {
   puts("\nProblems with opening file");
   exit(1);
   }

fprintf(data_out1,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
        no_reads,code,today.da_mon,today.da_day,today.da_year,
        now.ti_hour,now.ti_min,now.ti_sec);

printf("Files(1-9) are being processed!.......");
         printf("\n\n\n              PLEASE WAIT...... ");

ct = 0;
do {
   m1 = ct;
   m2 = num_pixels + ct;
   m3 = (num_pixels*2) + ct;
   m4 = (num_pixels*3) + ct;
   m5 = (num_pixels*4) + ct;
   m6 = (num_pixels*5) + ct;
   m7 = (num_pixels*6) + ct;
   m8 = (num_pixels*7) + ct;
   m9 = (num_pixels*8) + ct;

   fprintf(data_out1,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m1)),
          (*(ram8+m2)),(*(ram8+m3)),(*(ram8+m4)),(*(ram8+m5)),
          (*(ram8+m6)),(*(ram8+m7)),(*(ram8+m8)),(*(ram8+m9)));

   ct++;
   }while(ct < num_pixels);

farfree(ram8);
fclose(data_out1);
```

```
printf("\n%lu\n",farcoreleft());
printf("\nFiles %s has been completed!\n",name1);

}
printf("\n\nCCD sampling and data writing completed!!!\n\n\n");
printf("\n\nPlease SHUT THE LASER COVER before the new sampling!!!\n\n\n");

}
```

Program 2:

```
cccc***********************************************************
cccc                INVERSE.FOR
cccc  This program is used to solve the inverse problem of light
cccc  scattring . Angular Scattered Light Pattern (ASLP) is used
cccc  as input and a Rmax value of the surface will be output.
cccc                April 25, 1992
cccc***********************************************************
C$NOEXT
      PROGRAM INVERSE
      DIMENSION AMBL(3489),RRMAX(9)
      DIMENSION AMB(3454),CENTER(9)
      DIMENSION YS(3489,9),Y(3454,1),Y0(3454,1)
      DIMENSION X(3454,1), X1(3454,1)
      DIMENSION B(3454),THETA(3454),DELTA(3454)
      CHARACTER*12 FIN(50),NIN(50)
      CHARACTER*72 TITLE
      REAL SCALEA,SCALEB,SCALEC,SCALED
      REAL SCALEA0,SCALEB0,SCALEC0,SCALED0
      REAL SCALE,RMAX,ROOT,FACTOR,RTC
      REAL RMAXC,RMAXD,FACTORC,FACTORD,ROOTC,ROOTD
      REAL LENGTH, DIA,THETA1,LAMDA,RHO
      REAL L,FL,FC,PEAK,REFF
      REAL SPACE,STEP1,STEP2,INTER
      REAL WEIGHT(9), SUM1,RMAXM1,RMAXM2
      INTEGER FILENO,N,COUNT
CCCC
      WRITE(*,*) 'ENTER THE NUMBER OF ASLP DATA TO BE PROCESSED!'
      READ(*,*) FILENO
CCCC
CCCC  WRITE(*,*)'NUMBER OF DATA POINTS?'
CCCC  PROBABLY, #3473 (3455+18) PIXEL IS BROKEN (ALWAY SATURATED),
CCCC  SO, THE LAST TWO PIXELS CANNOT BE USED AND THE NUMBER
CCCC  CHANGES FROM 3456 TO 3454.
CCCC  READ(*,*) N
      N=3454
      OPEN(1, FILE='INVERSE.DAT', STATUS='UNKNOWN')
CCCC  THE RANGE OF COREECTION FACTOR:
CCCC  (4.0--40.0)*0.00001
CCCC  This range needs re-calibrated each time the setting of the
```

```
CCCC   measuring system is changed.

CCCC  READ*, SCALEA0, SCALEB0
      SCALEA0=4.0
      SCALEB0=40.0
cccc  Initial two points within the search range:
      SCALEC0=SCALEA0+0.382*(SCALEB0-SCALEA0)
      SCALED0=SCALEA0+0.618*(SCALEB0-SCALEA0)
cccc
CCCC  Nominal nose radius(two carbide tool insert):
cccc  ACT (American Carbide Tool Company) BR-6 370: 0.50 mm
cccc  Super Tool (Michigan, U.S.) BR-6 C-6: 0.38-0.40 mm
CCCC
CCCC  WRITE(*,*) ' ENTER THE VALUE OF SEARCH STEP SIZE!'
CCCC  READ(*,*) STEP1,STEP2
      STEP1=5.0
      STEP2=2.5
CCCC
CCCC  SET REFERENCE LEVEL:
      REFF=38.0
CCCC This reference level should be adjusted each time the setting of the
CCCC measuring system is changed.
      OPEN(2, FILE='NEWSHOT.FIL', STATUS='OLD')
      DO 501 KKKK=1,FILENO
CCCC  AMBIENT DATA FILE
      READ(2, '(A12)') NIN(KKKK)
CCCC  ASLP DATA FILE
      READ(2, '(A12)') FIN(KKKK)
CCCC  TOOL'S NOSE RADIUS
      READ(2, *) RHO
CCCC
      OPEN(3,FILE=NIN(KKKK), STATUS='UNKNOWN')
      OPEN(4,FILE=FIN(KKKK), STATUS='UNKNOWN')
      WRITE(*,502) FIN(KKKK),NIN(KKKK)
502   FORMAT(' NOW, ANALYZING FILE:',2(1X,A12))
      WRITE(*,'(F4.2)') RHO
CCCC
      WRITE(1,217) FIN(KKKK),NIN(KKKK)
217   FORMAT(' ANALYZING FILE:',2(1X,A12))
      WRITE(1,2170) RHO
2170  FORMAT('Nose radius Rho =',1X,F4.2)
      WRITE(1,*)
CCCC
```

```
CCCC exponential function: Y=K1*X*EXP(-K2*X)
CCCC KK=(PHO-Rmax)/Rmax
CCCC K1=EXP(CE1)
CCCC K2=-CE2
CCCC CE1,CE2 --- COEFFICIENTS TO BE ESTIMATED BY
CCCC          LEAST-SQUARES CURVE FITTING
CCCC
CCCC READ IN AMBIENT LIGHT DATA (SINGLE COLUMN)
      DO 909 I=1,3489
        READ(3,*) AMBL(I)
909   CONTINUE
      CLOSE(3)
CCCC REMOVE THE FIRST 18 PIXEL DATA
      DO 808 I=1,N
        AMB(I)=AMBL(I+18)
808   CONTINUE
CCCC
      DO 807 I=1,N
        AMB(I)=AMB(I)-REFF
807   CONTINUE
      READ(4,'(A72)') TITLE
CCCC READ IN THE SAMPLED ASLP DATA (9 COLUMNS)
      DO 401 I=1,3489
       READ(4,*)YS(I,1),YS(I,2),YS(I,3),YS(I,4),YS(I,5),
     *         YS(I,6),YS(I,7),YS(I,8),YS(I,9)
401   CONTINUE
      CLOSE (4)
CCCC
CCCC
      SUM1=0.
      RMAXM2=0.
      COUNT=0
CCCC
      DO 4403 J=1,9
        DO 403 I=1,N
          Y(I,1)=YS((I+18),J)
          Y(I,1)=Y(I,1)-AMB(I)
          IF(Y(I,1).LE.0.) THEN
            Y(I,1)=AMB(I)
          ENDIF
          Y0(I,1)=Y(I,1)
403     CONTINUE
        PEAK=10.0
```

```
      DO 409 I=1,N
        IF(Y(I,1).GT.PEAK) THEN
         PEAK=Y(I,1)
         CENTER(J)=I
        ENDIF
409     CONTINUE
CCCC  L ---DISTANCE FROM THE LENS TO THE SURFACE (MM)
    L=237.8
CCCC  L=127.8 (FOR COLLECTION ANGLE (-5 TO +5 DEGREE, TOTAL 10
DEGREE)
CCCC  FC ---DISTANCE FROM THE FOCAL POINT OF THE LENS TO CCD
CCCC  ARRAY (MM)
    FC=11.8
CCCC  FC=62.73
CCCC  FOCAL LENGTH OF THE LENS (MM)
    FL=68.0
CCCC  WAVELENGTH OF LASER LIGHT (MM)
    LAMDA=0.0006348
CCCC  WRITE(*,*) 'ENTER WORKPIECE'S DIAMETER (MM).'
     READ(*,*) DIA
CCCC   DIA=40.
CCCC  DIAMETER OF THE LASER SPOT ON THE SURFACE (MM)
    LENGTH=0.75
CCCC  DETERMINE THE VALUE OF Bi AND THETA22
CCCC  ANGLE OF INCIDENCE
    THETA1=15.0*3.141593/180.
    DO 40 I=1,CENTER(J)
      B(I)=(CENTER(J)-I)*0.0106
      DELTA(I)=ATAN((FL*B(I))/(L*FC))
      THETA(I)=THETA1+DELTA(I)
40   CONTINUE
CCC  DETERMINE THE VALUE OF THE OTHER SIDE FOR SYMMETRY
    DO 50 I=CENTER(J)+1,N
      B(I)=(I-CENTER(J))*0.0106
      DELTA(I)=ATAN((FL*B(I))/(L*FC))
      THETA(I)=THETA1-DELTA(I)
50   CONTINUE
CCCC  DETERMINE THE TWO VARIABLES: X AND X1
    DO 60 I=1,N
        X(I,1)=((SIN(THETA1)-SIN(THETA(I))+(2*LENGTH*(COS(THETA1)+
   *     COS(THETA(I))))/DIA)/(2*(COS(THETA1)+COS(THETA(I)))))
   *       **2
```

```
     X1(I,1)=(LAMDA*(1+COS(THETA1+THETA(I)))**2)/(2.0*3.141593*
    *       (COS(THETA1))**2*(COS(THETA1)+COS(THETA(I)))**3)
 60  CONTINUE
     SCALEA=SCALEA0
     SCALEB=SCALEB0
     SCALEC=SCALEC0
     SCALED=SCALED0
     WRITE(*,*)'Please wait!'
     WRITE(*,1205) J
 1205 FORMAT('SNAPSHOT #',1X, I1)
     WRITE(*,*) 'THIS IS ITERATION 1 !'
     CALL INVER(n,y,Y0,x1,x,RHO,scalec,FACTORC,RMAXC,ROOTC,RTC)
     WRITE(*,205) SCALEC, RMAXC
 205  FORMAT('Kc =',F8.3,2X,'Rmaxc=', 2X, F6.1,1X,'Micron')
     CALL INVER(n,y,Y0,x1,x,RHO,scaled,FACTORD,RMAXD,ROOTD,RTC)
     WRITE(*,2055) SCALED, RMAXD
 2055 FORMAT('Kd =',F8.3,2X,'Rmaxd=', 2X, F6.1,1X,'Micron')
     WRITE(*,*) SCALEC,SCALED
     SPACE=ABS(SCALED-SCALEC)
     WRITE(*,*) FACTORC,FACTORD
     JJJ=2
 99  WRITE(*,*)
     WRITE(*,91)JJJ
 91  FORMAT(1X,'THIS IS ITERATION',1X, I3)
CCCC WRITE(*,*)'PLEASE WAIT!'
     IF(FACTORC.LT.FACTORD) THEN
     SCALEB=SCALED
     SCALED=SCALEC
     FACTORD=FACTORC
     RMAXD=RMAXC
     ROOTD=ROOTC
     SCALEC=SCALEA+0.382*(SCALEB-SCALEA)
CCCC WRITE(*,93) FACTORD,RMAXD
CCCC FORMAT('Epsilond=',E12.6,1X,'Rmaxd=',F6.1)
     CALL INVER(n,y,Y0,x1,x,RHO,scalec,FACTORC,RMAXC,ROOTC,RTC)
     ELSE
     SCALEA=SCALEC
     SCALEC=SCALED
     FACTORC=FACTORD
     RMAXC=RMAXD
     ROOTC=ROOTD
     SCALED=SCALEA+0.618*(SCALEB-SCALEA)
CCCC WRITE(*,94) FACTORC,RMAXC
```

```
CCCC FORMAT('Epsilonc=',E12.6,1X,'Rmaxc=',F6.1)
     CALL INVER(n,y,Y0,x1,x,RHO,scaled,FACTORD,RMAXD,ROOTD,RTC)
     ENDIF
     WRITE(*,*)'new Kc... new Kd...'
     WRITE(*,*) SCALEC,SCALED
     WRITE(*,*)'new Epsilonc...new Epsilond...'
     WRITE(*,*) FACTORC,FACTORD
     WRITE(*,*)'new Rmaxc...new Rmaxd...'
     WRITE(*,*) RMAXC,RMAXD
     INTER=(RMAXC+RMAXD)/2.0
     SPACE=ABS(SCALED-SCALEC)
CCCC
     IF(INTER.LE.10.0) THEN
       IF(SPACE.LE.STEP1) GOTO 509
     ELSE
       IF(SPACE.LE.STEP2) GOTO 509
     ENDIF
     JJJ=JJJ+1
     GOTO 99
CCCC
CCCC
509   FACTOR=AMIN1(FACTORC,FACTORD)
      IF(FACTORC.EQ.FACTOR) THEN
       SCALE=SCALEC
       ROOT=ROOTC
       RMAX=RMAXC
      ELSE
        SCALE=SCALED
        ROOT=ROOTD
        RMAX=RMAXD
       ENDIF
cccc    WRITE(*,*)
cccc    WRITE(1,*)
cccc    GOTO 4403
cccc   ELSE
      IF(RMAX.GT.35.0) THEN
        COUNT=COUNT+1
      ENDIF
CCCC
      IF(RMAX.LT.3.0) THEN
        COUNT=COUNT+1
      ENDIF
CCCC
```

```
      RRMAX(J)=RMAX
      SUM1=SUM1+RRMAX(J)
      WRITE(*,609) J,SCALE,RMAX,ROOT,RTC
 609    FORMAT('#',1X,I1,1X,'K =',F7.2,1X,'Rmax=',F6.1,1X,
     *     'Sigmam=',F8.3,1X,'Sigmac=',F8.3)
      WRITE(1,709) J,SCALE,RMAX,ROOT,RTC
 709    FORMAT('#',1X,I1,1X,'K =',F7.2,1X,'Rmax=',F6.1,1X,
     *     'Sigmam=',F8.3,1X,'Sigmac=',F8.3)
      WRITE(*,*)
cccc  ENDIF
4403  CONTINUE
     DO 1105 J=1,9
       WEIGHT(J)=RRMAX(J)/SUM1
       RMAXM2=RMAXM2+WEIGHT(J)*RRMAX(J)
cccc    write(*,*) j, weight(j)
1105  CONTINUE
     RMAXM1=SUM1/9
     WRITE(1,1106) FIN(KKKK), RMAXM1, RMAXM2
1106  FORMAT(A12,1X,'AVERAGED Rmax=',F6.1,1X,
     *     'WEIGHTED MEAN Rmax=',F6.1)
     write(1,*)
     WRITE(*,1107) FIN(KKKK), RMAXM1, RMAXM2
1107  FORMAT(A12,1X,'AVERAGED Rmax=',F6.1,1X,
     *     'WEIGHTED MEAN Rmax=',F6.1)
     IF(COUNT.GE.1) THEN
      WRITE(1,1609) COUNT
1609   FORMAT(I1, 1X, 'times beyond the measuring range!')
     ENDIF
501  CONTINUE
CCCC
     CLOSE(2)
     STOP
     END
cccc
     SUBROUTINE
INVER(N,Y,Y0,X1,X,RHO,scale,FACTOR,RMAX,ROOTM,ROOTC)
CCCC  THIS PROGRAM IS USED TO DETERMINE THE VALUE OF Rmax BY
CCC   USING LEAST-SQUARES CURVE FITTING BASED ON THE SCATTERED
CCC   PATTERN DATA
     INTEGER M,MP1,N
     DIMENSION F(3454,6), FT(6,3454),YY(3454,1),Y(N,1)
     DIMENSION X1(N,1),C(6,7),D(6,6),CE(6,6),X(N,1),YC(3454,1)
     DIMENSION Y0(N,1)
```

```
      REAL KK,K1,K2,RHO,RMAX,LENGTH,SCALE
      REAL ROOTM,ROOTC,FACTOR
CCCC  LINEARIZED BY LOGRITHM CONVERSION
      M=2
      LENGTH=0.75
9001    DO 404 I=1,N
         Y(I,1)=Y0(I,1)
         YY(I,1)=(0.00001*SCALE*Y(I,1))/X1(I,1)
         YY(I,1)=LOG(YY(I,1))
404    CONTINUE
C    GENERATE F-MATRIX
      DO 10 I=1,N
        DO 11 J=1,M
          F(I,J)=X(I,1)**(J-1)
11      CONTINUE
10    CONTINUE
C    GENERATE THE TRANSPOSE OF THE F-MATRIX
      DO 20 I=1,N
        DO 21 J=1,M
          FT(J,I)=F(I,J)
21      CONTINUE
20    CONTINUE
C    DETERMINE MATRIX [C]=[FT]*[F] OF SIMULTANEOUS EQUATION
SYSTEM
CCCC  WRITE(3,*)' THIS IS THE (FT*F) MATRIX!'
      DO 100 I=1,M
        DO 100 J=1,M+1
          C(I,J)=0.
100   CONTINUE
      DO 99 I=1,M
        DO 99 J=1,M
          D(I,J)=0.
99    CONTINUE
      DO 104 I=1,M
        DO 104 J=1,M
          DO 104 K=1,N
            C(I,J)=C(I,J)+FT(I,K)*F(K,J)
104   CONTINUE
      DO 105 I=1,M
        DO 105 J=1,M
CCCC     WRITE(3,106) I,J,C(I,J)
CCCC     FORMAT(2X,'RESULTANT MATRIX: C(', I2,I2,')=',2X,E14.7)
105   CONTINUE
```

```
CCCC  WRITE(3,*)
CCCC  WRITE(3,*)'THIS IS THE ARRAY {FT*Y}!'
      DO 101 I=1,M
        DO 101 J=1,1
          DO 101 K=1,N
            D(I,J)=D(I,J)+FT(I,K)*YY(K,1)
101   CONTINUE
      DO 102 I=1,M
        DO 102 J=1,1
CCCC     WRITE(3,103) I,J,D(I,J)
CCCC     FORMAT(2X,'RESULTANT ARRAY: D(', I2,I2,')=',2X,E14.7)
102   CONTINUE
CCCC  WRITE(3,*)
C
      DO 30 I=1,M
        C(I,M+1)= D(I,1)
30    CONTINUE
C     DETERMINE THE VALUES OF {C} BY SOLVING SIMULTANEOUS
EQUATIONS
C     USING CHOLESKY METHOD
      MP1=M+1
      CALL CHLSKY(C,M,MP1,CE)
CCCC  WRITE(3,*) 'THIS IS THE REQUIRED COEFFICIENTS!'
CCCC  WRITE(3,201) (I,CE(I,1),I=1,M)
CCCC  FORMAT(3X,'CE(',i1,')=',E14.7)
CCCC  WRITE(3,*)
CCCC  DETERMINE K1=SQRT(2.0*PI)*KK/L ,WHERE,KK=(RHO-Rmax)/Rmax
      K1=2.7182**CE(1,1)
      K2=-CE(2,1)
      KK=(LENGTH*K1)/SQRT(2.0*3.1415)
      RMAX=1000.*RHO/(1+KK)
CCCC  WRITE(3,206) SCALE,RMAX
CCCC  FORMAT(1X,F8.3,2X,F6.1)
      CALL DIRECT(N,KK,X,X1,YC)
      CALL CCDSTAT(N,Y0,ROOTM)
      CALL CCDSTAT(N,YC,ROOTC)
CCCC  WRITE(*,*)' THE SIGMA VALUE IS ...'
CCCC  WRITE(*,109) ROOTM
CCCC  FORMAT(4X,'MEASURED',1X,'SIGMA=',E12.6)
CCCC  WRITE(*,119) ROOTC
CCCC  FORMAT(4X,'CALCULATED',1X,'SIGMA=',E12.6)
CCCC  WRITE(3,110) JJJ,ROOTM,ROOTC
CCCC  FORMAT(I3,1X, 'MEASURED',1X,E12.6,1X,'CALCULATED',
```

```
CCCC *1X, E12.6)
     FACTOR=ABS(ROOTM-ROOTC)/ROOTM
CCCC  WRITE(*,901) FACTOR
CCCC  FORMAT(1X,'FACTOR='1X, E12.6)
CCCC  WRITE(3,801) FACTOR
CCCC  FORMAT(1X,'FACTOR='1X, E12.6)
     RETURN
     END
CCCC
CCCC
     SUBROUTINE CHLSKY(A,N,M,X)
     DIMENSION A(6,7), X(6,6)
CCC  CALCULATE THE FIRST ROW OF UPPER UNIT TRIANGULAR MATRIX
     DO 200 J=2,M
       A(1,J)=A(1,J)/A(1,1)
200  CONTINUE
C    CALCULATE OTHER ELEMENTS OF MATRICES [U] AND [L]
     DO 8 I=2,N
       J=I
       DO 5 II=J,N
         SUM=0.
         JM1=J-1
         DO 4 K=1,JM1
           SUM=SUM+A(II,K)*A(K,J)
4          CONTINUE
         A(II,J)=A(II,J)-SUM
5        CONTINUE
       IP1=I+1
       DO 7 JJ=IP1,M
         SUM=0.
         IM1=I-1
         DO 6 K=1,IM1
           SUM=SUM+A(I,K)*A(K,JJ)
6          CONTINUE
       A(I,JJ) = (A(I,JJ)-SUM)/A(I,I)
7      CONTINUE
8    CONTINUE
CCCC
CCCC  SOLVE FOR {X(I)} BY BACK SUBSTITUTION
     X(N,1)=A(N,N+1)
     L=N-1
     DO 300 NN=1,L
       SUM=0.
```

```
      I=N-NN
      IP1=I+1
      DO 301 J=IP1,N
        SUM=SUM+A(I,J)*X(J,1)
301     CONTINUE
      X(I,1)=A(I,M)-SUM
300   CONTINUE
      RETURN
      END
CCCC
CCCC
      SUBROUTINE DIRECT(N,KK,X,X1,YC)
C     THIS PROGRAM IS USED TO DETERMINE THE SCATTERED PATTERN
BASED
CCC   ON AN ESTIMATED SURFACE ROUGHNESS AND CORRECTION FACTOR
      INTEGER N
      DIMENSION X(N,1),X1(N,1)
      DIMENSION YC(N,1)
      REAL LENGTH,KK
      REAL PI
CCCC
      PI=3.1415
      LENGTH=0.75
CCCC  INPUT THE ROUGHNESS DATA
      DO 70 I=1,N
        YC(I,1)=SQRT(2.0*PI*KK)*X1(I,1)*2.7215**(-KK*X(I,1))/LENGTH
70    CONTINUE
      RETURN
      END
CCCC
CCCC  FIND STATITICAL DATA OF ASLP
      SUBROUTINE CCDSTAT(N,Y,ROOT)
      REAL Y(N,1),PROB(3454)
      REAL TOTALA,SPSIGMA,ROOT,MEANINT
      REAL BARI
CCCC
cccc  OPEN(5,FILE='STAT.DAT',STATUS='UNKNOWN')
      TOTALA=0.
      ROOT=0.
      SPSIGMA=0.
      MEANINT=0.
      BARI=0.
CCCC
```

```
CCCC
      DO 421 I=1,N
        PROB(I)=0.
421     CONTINUE
      DO 402 I=1,N
      TOTALA=TOTALA+Y(I,1)
402     CONTINUE
CCCC    WRITE(*,*) 'TOTAL INTENSITY'
CCCC    WRITE(*,*) TOTALA
      DO 1417 I=1,N
      PROB(I)=Y(I,1)/TOTALA
      BARI=BARI+I*PROB(I)
1417    CONTINUE
CCCC    WRITE(*,*) 'AVERAGE PIXEL NUMBER IS ...'
CCCC    WRITE(*,*) BARI
      DO 418 I=1,N
        SPSIGMA=SPSIGMA+PROB(I)*(I-BARI)**2
        MEANINT=MEANINT+PROB(I)*ABS(I-BARI)
418     CONTINUE
CCCC    WRITE(*,*) ' THE VALUE OF SP SIGMA ...'
CCCC    WRITE(*,*) SPSIGMA
      ROOT=0.1*SQRT(SPSIGMA)
CCCC
CCCC  WRITE(*,*) 'STATISTICAL DATA OF ASLP.....'
CCCC  WRITE(*,*)' TOTALA... MEAN ... VARIANCE...SIGMA...'
CCCC  WRITE(*,*) TOTALA, MEANINT, SPSIGMA,ROOT
CCCC  WRITE(*,*)
cccc  WRITE(5,*) 'SCATTERED PATTERN DOMAIN DATA.....'
cccc  WRITE(5,*)' TOTALA... MEAN ... VARIANCE...SIGMA...'
cccc  WRITE(5,808) TOTALA, MEANINT, SPSIGMA,ROOT
cccc  FORMAT(1X,4(E12.6,1X))
cccc  WRITE(5,*)
   RETURN
   END
```

Program 3:

```
/*****************************************************************
                    Neural.c
This program  combines the  snapshot.c and Back-propagation  program from [MR88]
into one package to predict Rmax in  real-time.

                September 3, 1993
*****************************************************************/

#include <stdio.h>
#include <alloc.h>
#include <math.h>
#include <io.h>
#include <time.h>
#include <string.h>
#include <conio.h>
#include <dos.h>
#include <stdlib.h>

#include "general.h"
#include "command.h"

 /* Set addresses on CUSTOM interface board*/
#define     START_CONVERT       0x300
#define     STATIC_RAM        0x304
#define     RESET_4040S       0x310     /* 784dec      */
#define     CCD_FIRE          0x314     /* 788dec      */

#define     CLEAR_DFF1        0x318     /*WRITE to     */
#define     CLEAR_DFF2        0x319     /* SRAM        */

#define     TRIGGER_READ       0x30c     /*READing from */

#define     FINISHED          0x31c     /*dataline d0  */
#define     LATCH_ENCODER      0x31c     /*dataline d7  */

/*variables*/
unsigned char far *ram8=NULL;
unsigned char far *ram9=NULL;

struct date today;
struct time now;
```

```
int j,snap,k,i,sample_no,xch,pixel_count;
int x,y,d1,no_reads,top;
int not_needed,max_snaps;

unsigned num_pixels=3489;
unsigned m1,m2,m3,m4,m5,m6,m7,m8,m9;
unsigned m10,m11,m12,m13,m14,m15,m16,m17,m18;
unsigned ct,count,kk,readings;
unsigned long maxsize=63000L;
unsigned char start,resett,fire_ccd;
unsigned char trigger,clear_d;
unsigned long val;
unsigned long aa;

char wp[3],s[15],code[10];
char name1[15],name2[15],name3[15],name4[15];
clock_t start_t,end_t;

float distance,spindle,feedrate;
float one_rev,no_of_rev,tot_time;

extern char **pname;
extern float **ipattern, **tpattern;

main(argc,argv)  /*int argc; char *argv[];*/
{FILE *data_out1, *data_out2, *data_out3,*data_out4;
 int quit=0;

 printf("select: 1 for training; 2 for testing. Please input 1/2?:");
 scanf("%i", &no_reads);
 if (no_reads==1)
  {genet(argc,argv);
        while(quit!=END){quit=do_command(Prompt,(int*) BASEMENU);}
  }
 else if ((no_reads!=1)&&(no_reads!=2)) exit(1);
 else
{printf("enter a three-character workpiece code!\n");
 scanf("%s",code);
 printf("enter speed (rps) and feedrate(mm/rev)!\n");
 scanf("%f%f", &spindle, &feedrate);
 printf("number of readings?\n");
 scanf("%d", &no_reads);
```

```
d1=30;
distance = 3.5;
snap = 1;

 /* if( no_reads % 9 )
 {clrscr();
  printf("\n\n\nSample number must be 9,18 or 36!\n\n");
   Determine approx. time for feed movement of 'X'mm ...
  one_rev = 1.0/spindle;          time for 1 revolution
  no_of_rev = distance/feedrate;   num of revs in X.0mm
  tot_time = no_of_rev*one_rev;    total time for X.0mm
  printf("\nThe time for %3.1fmm, based on speed & feedrate\
          is %6.3fsec\n\n",distance,tot_time);
  exit(1);
 }
 */

clrscr();
ram8=(unsigned char far *)(farmalloc(63000L));  /*Allocate memory for data*/

if (ram8) puts("\nram8 allocation ok.");
else {puts("\nProblems with ram8 memory allocation!!"); exit(1); }

ram9 = (unsigned char far *)(farmalloc(63000L));
if (ram9) puts("\nram9 allocation ok.");
else {puts("\nProblems with ram9 memory allocation!!"); exit(1); }

strcpy(name1,"snap1.");     /*Set up the filenames & extensions */
strcpy(name2,"snap2.");
strcpy(name3,"snap3.");
strcpy(name4,"snap4.");
strcat(name1,code);
strcat(name2,code);
strcat(name3,code);
strcat(name4,code);

        /*Determine approx. time for feed movement of 3.5 mm ... */
one_rev = 1.0/spindle;          /*time for 1 revolution      */
no_of_rev = distance/feedrate;  /*num of revs in X.0mm       */
tot_time = no_of_rev*one_rev;   /*total time for X.0mm       */
sample_no = 1;                  /*first sample*/
```

```
printf("\n\nProgram to capture images from the CCD array....\n\n");
printf("Calculated time for %3.1fmm based on speed & feedrate %6.3fsec",
        distance,tot_time);
printf("\n\nPress any key to START sampling.................");
getch();
genet(argc,argv); /*initialize network and set screen */
        /*****************************************************/
start=inportb(CLEAR_DFF1);/*CLEAR FF1 for WRITE to SRAM*//*Q1 bar set high*/
start=inportb(CLEAR_DFF2);    /* CLEAR FF2 */ /* Q2 set low */
resett=inportb(RESET_4040S);  /* RESET 4040s     */
        /*****************************************************/

/* clrscr(); */
ct = 0;         /*set counters*/
top = 1;        /*use ram8 for first 18 samples & ram9 for remainder*/
pixel_count = 0;
val = (unsigned long)(no_reads)*(unsigned long)(num_pixels);

/*printf("\n\n");                                   */
/*printf("\nWriting and transferring %d samples from SRAM...\n",no_reads);*/
/*printf("representing %lu individual pixel reads...\n",val);           */

x = 30;
y = 10;
 /* gotoxy(x,y);                */
 /* printf("COMPLETED SAMPLE #"); */
x = wherex();
y = wherey();

   /*First sample will be taken 'X'mm from start of cutting*/
start_t = clock();
end_t = clock();
while( ((end_t-start_t)/CLK_TCK) <= tot_time )     end_t = clock();
while( kbhit() ) getch(); /*clear keyboard buffer */

while( sample_no <= no_reads ) /*wait for encoder pulse, latch and READ */
 {while( !(inportb(LATCH_ENCODER) & 0x80) );    /* read bit 7*/
   /*for(i=0; i<20000; i++) k=i; simulated encoder wait state */
   /*send pulse to FIRE ccd  (encoder trigger) */
  start = inportb(START_CONVERT);
  fire_ccd = inportb(CCD_FIRE);
  while(inportb(FINISHED)&01); /*Finished 3489 pixels?NOT STORED in 128K
SRAM*/
```

```
for(j=0; j<snap; j++) /*add a small delay (d1)...SETS EXPOSURE TIME...
                        entered as the third COMMAND line argument */
 {for(i=0;i<d1;i++)k=i;/*reset DFF2 ready for next encoder pulse capture*/
  clear_d = inportb(CLEAR_DFF2);
  start=inportb(START_CONVERT);/*Now fire again (reduced EXPOSURE trigger)*/
  fire_ccd = inportb(CCD_FIRE);
  while(inportb(FINISHED)&01);/*Finished 3489 pixels???STORED in 128K SRAM*/
 } /*end for loop*/

start_t = clock();  /*Start the timing for the 3.5 mm length */
if(top)           /*THEN READ the 8K RAM right away    */
 {pixel_count = 0;
  do
   {*(ram8+ct) = inportb(STATIC_RAM);
    ct++;
    pixel_count++;
   } while(pixel_count<3489);
  if( ct > 62780L )        /*make sure #19 READ is NOT stored */
   {top = 0;               /*else 64K boundary will be wrapped*/
       /* puts("  .. 2nd file sets.."); */
    ct = 0;
   }
 }
else
 {pixel_count = 0;
  do
   {*(ram9+ct) = inportb(STATIC_RAM);  /*store #19-36 reads */
    ct++;
    pixel_count++;
   } while (pixel_count<3489);
 }

pat();
test_pattern();

 /*put in a reset for the continuous cycling mode */
 /*gotoxy(x,y);            */
 /*printf("%d",sample_no); */
sample_no++;

/*Now determine the delay if required so that the 36 readings
  are distributed along the 150mm workpiece... ie approx.
  every 3.5 mm. This depends on spindle speed and feedrate is
```

```
   determined by the command line arguments before each cutting
   session begins (tot_time) for 3.5 mm.
  */

 end_t = clock();
 while( ((end_t-start_t)/CLK_TCK) <= tot_time ) end_t = clock();
  /*Now wait for the NEXT encoder pulse and REPEAT process */
 } /*END WHILE STATEMENT */


top = 1;
sample_no--;
/*clrscr();                              */
/*printf("Number of pixels read....%u\n",val);           */
/*printf("\n\n\n%d samples completed...\n\n",sample_no); */

   /*Now open file(s) as required and print data..... */
getdate(&today);
gettime(&now);

if( no_reads >18 )      /* 4 files required */
 {data_out1 = fopen(name1,"w");
  if(data_out1==NULL)
   {puts("\nProblems with opening first file");
    exit(1);
   }
  data_out2 = fopen(name2,"w");
  if(data_out2==NULL)
   {puts("\nProblems with opening second file");
    exit(1);
   }
  data_out3 = fopen(name3,"w");
  if(data_out3==NULL)
   {puts("\nProblems with opening first file");
    exit(1);
   }
  data_out4 = fopen(name4,"w");
  if(data_out4==NULL)
   {puts("\nProblems with opening second file");
    exit(1);
   }
  fprintf(data_out1,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
        no_reads,code,today.da_mon,today.da_day,today.da_year,
```

```
        now.ti_hour,now.ti_min,now.ti_sec);
  fprintf(data_out2,"%d wp#%s, samples 10-18, %d/%d/%d,time %02d:%02d:%02d\n",
        no_reads,code,today.da_mon,today.da_day,today.da_year,
        now.ti_hour,now.ti_min,now.ti_sec);
  fprintf(data_out3,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
        no_reads,code,today.da_mon,today.da_day,today.da_year,
        now.ti_hour,now.ti_min,now.ti_sec);
  fprintf(data_out4,"%d wp#%s, samples 10-18, %d/%d/%d,time %02d:%02d:%02d\n",
        no_reads,code,today.da_mon,today.da_day,today.da_year,
        now.ti_hour,now.ti_min,now.ti_sec);
  printf("\n\n\n First set of files(1-18) are being processed!...");
  printf("\n\n            PLEASE WAIT... ");
  ct = 0;
  do
   {m1 = ct;
   m2 = num_pixels + ct;
   m3 = (num_pixels*2) + ct;
   m4 = (num_pixels*3) + ct;
   m5 = (num_pixels*4) + ct;
   m6 = (num_pixels*5) + ct;
   m7 = (num_pixels*6) + ct;
   m8 = (num_pixels*7) + ct;
   m9 = (num_pixels*8) + ct;
   m10 = (num_pixels*9) + ct;
   m11 = (num_pixels*10) + ct;
   m12 = (num_pixels*11) + ct;
   m13 = (num_pixels*12) + ct;
   m14 = (num_pixels*13) + ct;
   m15 = (num_pixels*14) + ct;
   m16 = (num_pixels*15) + ct;
   m17 = (num_pixels*16) + ct;
   m18 = (num_pixels*17) + ct;
   fprintf(data_out1,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m1)),
        (*(ram8+m2)),(*(ram8+m3)),(*(ram8+m4)),(*(ram8+m5)),
        (*(ram8+m6)),(*(ram8+m7)),(*(ram8+m8)),(*(ram8+m9)));
   fprintf(data_out2,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m10)),
        (*(ram8+m11)),(*(ram8+m12)),(*(ram8+m13)),(*(ram8+m14)),
        (*(ram8+m15)),(*(ram8+m16)),(*(ram8+m17)),(*(ram8+m18)));
   ct++;
   } while (ct < num_pixels);
 farfree(ram8);
 fclose(data_out1);
 fclose(data_out2);
```

```
printf("\n\nBetween files check...ct = %u\n",ct);
printf("Second set of files(19-36) are being processed!.......");
printf("\n\n              PLEASE WAIT...... ");
ct = 0;
do
 {m1 = ct;
 m2 = num_pixels + ct;
 m3 = (num_pixels*2) + ct;
 m4 = (num_pixels*3) + ct;
 m5 = (num_pixels*4) + ct;
 m6 = (num_pixels*5) + ct;
 m7 = (num_pixels*6) + ct;
 m8 = (num_pixels*7) + ct;
 m9 = (num_pixels*8) + ct;
 m10 = (num_pixels*9) + ct;
 m11 = (num_pixels*10) + ct;
 m12 = (num_pixels*11) + ct;
 m13 = (num_pixels*12) + ct;
 m14 = (num_pixels*13) + ct;
 m15 = (num_pixels*14) + ct;
 m16 = (num_pixels*15) + ct;
 m17 = (num_pixels*16) + ct;
 m18 = (num_pixels*17) + ct;
 fprintf(data_out3,"%d %d %d %d %d %d %d %d %d\n",(*(ram9+m1)),
     (*(ram9+m2)),(*(ram9+m3)),(*(ram9+m4)),(*(ram9+m5)),
     (*(ram9+m6)),(*(ram9+m7)),(*(ram9+m8)),(*(ram9+m9)));
 fprintf(data_out4,"%d %d %d %d %d %d %d %d %d\n",(*(ram9+m10)),
     (*(ram9+m11)),(*(ram9+m12)),(*(ram9+m13)),(*(ram9+m14)),
     (*(ram9+m15)),(*(ram9+m16)),(*(ram9+m17)),(*(ram9+m18)));
 ct++;
} while (ct < num_pixels);
farfree(ram9);
fclose(data_out3);
fclose(data_out4);
printf("\n%lu\n",farcoreleft());
printf("\nFiles%s%s%s&%s have been completed!\n",name1,name2,name3,name4);
}

else if (no_reads > 9 )          /*only 2 file required*/
 {data_out1 = fopen(name1,"w");
 if(data_out1==NULL)
  {puts("\nProblems with opening file");
  exit(1);
```

```
 }
data_out2 = fopen(name2,"w");
if(data_out2==NULL)
 {puts("\nProblems with opening file");
  exit(1);
 }
fprintf(data_out1,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
   no_reads,code,today.da_mon,today.da_day,today.da_year,
   now.ti_hour,now.ti_min,now.ti_sec);
fprintf(data_out2,"%d wp#%s, samples 10-18, %d/%d/%d,time %02d:%02d:%02d\n",
   no_reads,code,today.da_mon,today.da_day,today.da_year,
   now.ti_hour,now.ti_min,now.ti_sec);
 printf("\n\n\nFiles(1-18) are being processed!...");
 printf("\n\n              PLEASE WAIT... ");
ct = 0;
do
 {m1 = ct;
 m2 = num_pixels + ct;
 m3 = (num_pixels*2) + ct;
 m4 = (num_pixels*3) + ct;
 m5 = (num_pixels*4) + ct;
 m6 = (num_pixels*5) + ct;
 m7 = (num_pixels*6) + ct;
 m8 = (num_pixels*7) + ct;
 m9 = (num_pixels*8) + ct;
 m10 = (num_pixels*9) + ct;
 m11 = (num_pixels*10) + ct;
 m12 = (num_pixels*11) + ct;
 m13 = (num_pixels*12) + ct;
 m14 = (num_pixels*13) + ct;
 m15 = (num_pixels*14) + ct;
 m16 = (num_pixels*15) + ct;
 m17 = (num_pixels*16) + ct;
 m18 = (num_pixels*17) + ct;
 fprintf(data_out1,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m1)),
      (*(ram8+m2)),(*(ram8+m3)),(*(ram8+m4)),(*(ram8+m5)),
      (*(ram8+m6)),(*(ram8+m7)),(*(ram8+m8)),(*(ram8+m9)));
 fprintf(data_out2,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m10)),
      (*(ram8+m11)),(*(ram8+m12)),(*(ram8+m13)),(*(ram8+m14)),
      (*(ram8+m15)),(*(ram8+m16)),(*(ram8+m17)),(*(ram8+m18)));
 ct++;
 } while (ct < num_pixels);
farfree(ram8);
```

```
 fclose(data_out1);
 fclose(data_out2);
 printf("\n%lu\n",farcoreleft());
 printf("\nFiles %s %s has been completed!\n",name1,name2);
 }

else        /* one file required */
 {data_out1 = fopen(name1,"w");
 if(data_out1==NULL)
  {puts("\nProblems with opening file");
  exit(1);
  }
 fprintf(data_out1,"%d wp#%s, samples 1-9, %d/%d/%d,time %02d:%02d:%02d\n",
  no_reads,code,today.da_mon,today.da_day,today.da_year,
  now.ti_hour,now.ti_min,now.ti_sec);
 printf("\n\n\nFiles(1-9) are being processed!...");
 printf("\n\n              PLEASE WAIT... ");
 ct = 0;
 do
  {m1 = ct;
  m2 = num_pixels + ct;
  m3 = (num_pixels*2) + ct;
  m4 = (num_pixels*3) + ct;
  m5 = (num_pixels*4) + ct;
  m6 = (num_pixels*5) + ct;
  m7 = (num_pixels*6) + ct;
  m8 = (num_pixels*7) + ct;
  m9 = (num_pixels*8) + ct;
  fprintf(data_out1,"%d %d %d %d %d %d %d %d %d\n",(*(ram8+m1)),
       (*(ram8+m2)),(*(ram8+m3)),(*(ram8+m4)),(*(ram8+m5)),
       (*(ram8+m6)),(*(ram8+m7)),(*(ram8+m8)),(*(ram8+m9)));
  ct++;
 } while (ct < num_pixels);

 farfree(ram8);
 fclose(data_out1);
 printf("\n%lu\n",farcoreleft());
 printf("\nFiles %s has been completed!\n",name1);
 }

printf("\n\nCCD sampling and data writing completed!!!\n");
}
}
```

```
pat ()     /* transform piexl data into network input pattern */
{FILE *out;
 int k0,k1,k2=50,k3=20,ix;
 float ix0[174],ix1;
 char sam[3];
 register int i,j;

 if (sample_no==1)
 if ((out=fopen("temp.pat","w"))==NULL)
  {printf("cannot open output pattern file");
  exit(1);
  }

  ix1=0.0;
  for (j=0;j<174;j++)
   {ix0[j]=0.0;
   for (i=0;i<k3;i++)
    {m1=k3*j+i+num_pixels*((sample_no%18)-1);
    if(!(sample_no%18)) m1=k3*j+i+num_pixels*17;
    if (sample_no<=18) ix=(*(ram8+m1));
    else      ix=(*(ram9+m1));
    ix0[j]=ix0[j]+(float) ix;
    }
   if(fabs(ix0[j])>fabs(ix1)) {ix1=ix0[j]; k0=j; }
   }
  itoa(sample_no, sam, 10);
  strcpy(pname[0],sam);
  fprintf(out,"%s%i\n",name1,sample_no);
  k1=k0-k2/2;
  if (k1<0) k1=0;
  if (k1>(173-k2)) k1=173-k2;
  for (j=k1;j<k1+k2;j++)
   {ix0[j]=((ix0[j]/(float)k3)-20.0)/10.0;
   if (((j-k1)!=0)&&(((j-k1)%15)==0)) fprintf(out,"\n");
   fprintf(out,"%5.1f",ix0[j]);
   ipattern[0][j-k1]=ix0[j];
   }
  fprintf(out,"%5.1f",0.0);
  fprintf(out,"\n");
  tpattern[0][0]=0.0;
if (sample_no==no_reads) fclose(out);)
```

Program 4:

```
/*********************************************************************
                    New_flan.c
     This program is used to estimate flank wear from the sampled spindle speed
change data .
                    Feb.16, 1993
               Revised March 5,1993
                    Revised June 5,1993 (updated Kv)

*********************************************************************/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
#include <alloc.h>
#include <string.h>
#include <ctype.h>

#define POINT 8

FILE *data_out, *spn_dat;

void main(int argc, char *argv[])
{
char name1[20], batch[20];

float *deltahf = NULL, *ns = NULL, *hf = NULL;
float *deltan = NULL, *ave_spn = NULL;

float deltat, kk = 0., k_p = 0., k_tr = 0, k_v = 0., k_j = 0;
float hf0, f, moment, d, diam, cp, rpm0, rpm1, rpm2;
float belta1, belta2, belta3, pi;

int i = 0, INI = 0, pole = 0, code = 0;
int ii = 0, cut = 0;
unsigned aa1;

FILE *data_in1, *data_in2;

/* function declaration   */
void spnmove(FILE *data_in2, int INI, float *deltan,
```

```
                    float *ave_spn, float rpm0);

 float init_spn(FILE *data_in2);
 /*10/05/93...Command line agruments....
  First.... cut_(a,b,g etc).dat
  Second... batch file with speed data...cut_(a,b,g etc).fil
  Third.... 2.. indicating that the first 2 data points are used
             as the reference point
  */
 if ( argc == 4) {
         strcpy(name1,argv[1]);
         strcpy(batch,argv[2]);
         INI = atoi(argv[3]);
         clrscr();
         printf("\n\n");
         }
 else {
         clrscr();
         printf("THREE command line auguments:\n");
         printf("cut file, output file, batch file and INI !\n");
         exit(1);
         }

 /* open input and output files  */
 if ((data_in1 = fopen (name1, "r")) == NULL) {
         printf("\nERROR - Cannot open the designated R1 file\n");
         exit(1);
          }

 data_out = fopen("flank.dat", "w");
 spn_dat = fopen("spn.dat","w");

 /* read in the basic cuting condition data  */

 fscanf(data_in1, "%d%d%d%f%f%f%f", &pole, &code, &cut,&f, &d, &diam, &deltat);
         /* motor pole, material code, cut number, feedrate, depth of cut,
                 workpiece's diameter, time interval between two points  */
 fscanf(data_in1, "%f", &hf0); /* initial flank wear  */
 fscanf(data_in1, "%f%f%f", &belta1, &belta2, &belta3); /* index  */
 fclose(data_in1);

 /*printf("enter cut number.\n");
 scanf("%d", &cut); */
```

```
pi = 3.14156;
moment = 0.048;

/* Cp value */
if(code)  /* hard material  *//*aver=5535.5*/
  cp = 5535.5; /* range: 5402.3-5668.8 *//*+15%=6365.8 -15%=4705.2*/
else      /* soft material  */
  cp = 7920.3;  /* range: 7734.7-8105.9   */

/* Transmission ratio chart */
/*  2-pole motor   4-pole motor    belt ratio     gear ratio

 n: 2320 rpm       1156 rpm        32/83          45/27
 n:  660 rpm       330 rpm         32/83          23/49
 n:  410 rpm       205 rpm         32/83          17/55
 n:  110 rpm        55 rpm         32/83    23/49, 27/45, 17/55 */

 /* open batch file  */
 if ((data_in2 = fopen (batch, "r")) == NULL) {
        printf("\nERROR - Cannot open the designated batch file\n");
        exit(1);
   }
/* loop for processing a number of cuts' speed data  */

for(ii = 0; ii < cut; ++ii) {

 rpm1 = init_spn(data_in2);
 rpm2 = init_spn(data_in2);
 rpm0 = (rpm1 + rpm2) / 2.0;

 printf("\nThis is Cut #%d:\n", ii+1);
 aa1 = coreleft();
 printf("core memory avilable after initial loop is: %u\n", aa1);

 /* memory allocation   */
hf = (float *) malloc((POINT+1) * sizeof(float));
deltahf = (float *) malloc(POINT * sizeof(float));
ns = (float *) malloc(POINT * sizeof(float));
deltan = (float *) malloc(POINT * sizeof(float));
ave_spn = (float *) malloc(POINT * sizeof(float));

if(hf == NULL) {
```

```
  printf(" No memory allocation for hf\n");
  exit(1);
  }
 if(deltahf == NULL) {
  printf(" No memory allocation for deltahf\n");
  exit(1);
        }
 if(ns == NULL) {
  printf(" No memory allocation for ns\n");
        exit(1);
  }
 if(deltan == NULL) {
  printf(" No memory allocation for deltan\n");
        exit(1);
        }
 if(ave_spn == NULL) {
  printf(" No memory allocation for ave_spn\n");
  exit(1);
        }

 *(hf+0) = hf0;

 spnmove(data_in2, INI, deltan, ave_spn, rpm0);
 for(i = 0; i < POINT; ++i)
        {
        fprintf(spn_dat, "%.2f\n", *(deltan+i));
        }
 /* various correction factors for transmission ratio  */

/* correction factor: kk = k_tr * k_p * k_v
        k_tr:  transmission factor, used to consider the effect of
                transmission ratio on deltan, calculated transmission chain value;
        k_p:   motor pole factor, since 2-pole motor is used as reference,
                k_p = 1.0 for a 2-pole motor, and k_p = 2.0 for a 4-pole motor;
        k_v:   speed factor,  k_v =(rpm_s-rpm0)/(rpm_s-rpm_max), used to
               consider the effect of the speed change on deltan. In eq., rpm_max is the
               maximum rpm speed for a motor (e.g. 2-pole motor, rpm_max = 2320,
               rpm0 is the nominal speed used, rpm_s is the synchronized speed.
        k_j:  inertia factor, used to consider the increment of moment od
                inertia caused by spindle assembly.  k_j = 1.30  in our case */

        k_j = 1.30;
```

```
if(pole)   /* 4 pole motor  */
        {
        k_p = 2.0;
        k_v = (1800.0 - rpm0)/ (1800.0 - 1156.0);

        if(rpm0 > 500.0) /* for n = 1156 rpm   */
                k_tr = (1156.0/rpm0) * (83.0/32.0) * (27.0/45.0);
        else
                k_tr = (330.0/rpm0) * (83.0/32.0) * ( 49.0/23.0);  /* for n = 330 rpm  */

        kk = k_j * k_p * k_tr * k_v;
        }
else   /* 2 pole motor  */
        {
        k_p = 1.0;
        k_v = (3600.0 - rpm0)/(3600.0 - 2320.0);

        if (rpm0 > 1000.0)  /* for n = 2320 rpm   */
                k_tr = (2320.0/rpm0) * (83.0/32.0) * (27.0/45.0);
        else
                k_tr = (660.0/rpm0) * (83.0/32.0) * ( 49.0/23.0);   /*for n = 660 rpm */

        kk = k_j * k_p * k_tr * k_v;
        }

/* estimate the flank wear value  */
for(i = 0; i < POINT; ++i)
         *(ns+i) = pi *(*(ave_spn+i)) * diam/1000.0;

        if( *(deltan+0) < 0.) *(deltan+0) = 0.;
                                        /* negative speed drop not allowed  */

        *(deltahf+0) = (4000 * pi * moment * kk * (*(deltan+0))) /
                (60 * diam * cp * pow( (*(ns+0)), belta1) *
                pow( f, belta2) * pow( d, belta3) * deltat);
                                        /* flank wear increment   */
         *(hf+0) = hf0 + (*(deltahf+0));
         printf("  1   %.4f mm\n", *(hf+0));
/*  fprintf(data_out, "  1    %.4f     %.2f    %.2f\n",
                *(hf+0), *(deltan+0), kk);   */
         fprintf(data_out, "%.4f\n", *(hf+0));
        for(i = 1; i < POINT; ++i)
         {
```

```
    if( *(deltan+i) < 0.) *(deltan+i) = 0.;
                                    /* negative speed drop not allowed */

        *(deltahf+i) = (4000 * pi * moment * kk * (*(deltan+i))) /
              (60 * diam * cp * pow( (*(ns+i)), belta1) *
              pow( f, belta2) * pow( d, belta3) * deltat);
                                    /* flank wear increment */
    *(hf+i) = *(hf+i-1) + (*(deltahf +i));

    printf("%3d   %.4f mm\n", i+1, *(hf+i));
 /*  fprintf(data_out, "%3d   %.4f     %.2f    %.2f\n",
        i+1, *(hf+i), *(deltan+i), kk); */
    fprintf(data_out, "%.4f\n", *(hf+i));
    }

  hf0 = *(hf+POINT-1);

  /*printf("press any key to continue.\n");
  getch(); */
   }

 fprintf(data_out, "\n");
 fclose(data_out);
 printf("\nCalculation completed!\n");

 }

 float init_spn(FILE *data_in2)
 {
  char tit[100], name[20];
  int i = 0, a, val1, number = 0, count = 0;
  unsigned  sp, nt_used;

  unsigned *spp = NULL;

  float val2f, val2f_sum, ave_val2f, rpm, rps;
  float sspeed,eproms,c,dd,e,ff,g,h,s;

  FILE *indata;

 /* read the no_cut filename  */

 fscanf( data_in2, "%s", name);
```

```
 if ((indata = fopen(name, "r")) == NULL) {
        printf("\nERROR - Cannot open the designated no_cut file\n");
        exit(1);
          }

 /* read in the headings of second no_cut file          */

 for (i = 0; i < 11; ++i)
    fgets(tit, 80, indata);

 fscanf(indata, "%d%f%f%f%f%f%f%d%f%f%f",
                  &a,&sspeed,&eproms,&c,&dd,&e,&ff,&number,&g,&h,&s);

/*memory allocation  */
 if(spp)
   free(spp);

 spp =(unsigned  *) malloc(number * sizeof(unsigned));

 if(spp == NULL)  {
  printf(" No memory allocation for spp.\n");
  exit(1);
        }

/* read in the sampled speed data   */
 while(!feof(indata)) {
         fscanf(indata,"%u %u",&sp,&nt_used);
         *(spp+count) = sp;
         count++;
          }
 fclose(indata);

 count--;

 val2f_sum = 0.;

 for(i = 0; i < count; ++i) {
          val1 = 50000 -(*(spp+i));
          val2f = .1*val1;
          val2f_sum = val2f_sum + val2f;
          }
       ave_val2f = val2f_sum / count;
 rps = 1.0/((1024.0*ave_val2f*4.0)/(1000000.0*eproms));
```

```
  rpm = rps*60.0;

  free(spp);
  count = 0;

  return(rpm);
 }

void spnmove(FILE *data_in2, int INI,float *delt,
          float *spn_ave, float ref0)
{
 char name0[20], title[80];

 int   i, j, k, val1, tot_number = 0, a, RANGE, ct = 0;
 unsigned sp, nt_used, aa1;

 unsigned *ssp = NULL;

 float rpm1,rps1,sspeed,eproms,c,dd,e,ff,g,h,s;
 float val2f, ref = 0.;
 float spn_sum, nominal;

 FILE *data_in5;

/* read the main speed data filename  */
 fscanf( data_in2, "%s", name0);
 printf("Reading the speed data file:  %s\n", name0);
 if ((data_in5 = fopen (name0, "r")) == NULL) {
        printf("\nERROR - Cannot open the designated main data file\n");
        exit(1);
          }

 /* read in the headings of speed data file          */

 fgets(title, 80, data_in5);
 /*fputs(title, data_out);  */
 puts(title);

 fscanf(data_in5, "%d%f%f%f%f%f%f%d%f%f%f",
              &a,&sspeed,&eproms,&c,&dd,&e,&ff,&tot_number,&g,&h,&s);

 /* memory allocation    */
 if(ssp)
```

```
    free(ssp);

  ssp =(unsigned *) malloc(3 * tot_number * sizeof(unsigned));

  if(ssp == NULL)  {
   printf(" No memory allocation for ssp.\n");
   exit(1);
   }

        /* read in the sampled speed data   */
  while(!feof(data_in5)) {
          fscanf(data_in5,"%u %u",&sp,&nt_used);
          *(ssp+ct) = sp;
          ct++;
                  }
  fclose(data_in5);

  ct--;

/* sampled data converted to RPM value    */
for(i = 0; i < ct; ++i) {
                  val1 = 50000 -(*(ssp+i));
          val2f = .1*val1;
          rps1 = 1.0/((1024.0*val2f*4.0)/(1000000.0*eproms));
          rpm1 = rps1*60.0;
          *(ssp+i) = rpm1;
          }

for(i = 0; i < INI; ++i)
    ref = ref + *(ssp+i);

 ref = ref / INI;

 RANGE = floor((ct-INI)/POINT);

/* for loop to find POINT points which are moving-averaged result within
   every RANGE points.  */

for (j = 0; j < POINT; ++j){
         *(spn_ave+j) = 0;
         *(delt+j) = 0;
          }
```

```
for (j = 0; j < POINT; ++j) {
        spn_sum = 0.;
        k = j * RANGE;

        for (i = 0; i < RANGE; ++i)
                spn_sum = spn_sum + *(ssp+k+1+i);

        *(spn_ave+j) = spn_sum/RANGE;
        }
nominal = ref0 - ref;
printf("Speed drop due to nominal cutting torque is: %.2f RPM\n", nominal);
/* fprintf(data_out, " nominal speed drop:  %.2f RPM\n", nominal);  */

*(delt+0) = ref - (*(spn_ave+0));

for(i = 1; i < POINT; ++i)
        *(delt+i) = (*(spn_ave+i-1)) - (*(spn_ave+i));

free(ssp);
aa1 = coreleft();
printf("core memory available: %u\n", aa1);
ct = 0;
return; }
```

Program 5:

```
/*****************************************************************
                        Est_Rmax.c
  This program is used to predict Rmax of a turned surface by using the nominal
 cutting variables and tool geometry. It consists of three components: theoretical
component depending
  on feed rate and nose radius, tool wear component and vibrarion component.
  It outputs eight estimated Rmax values for each cut. An average flank wear for
  each cut is used.
                        May 25, 1993
*****************************************************************/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <malloc.h>
#include <math.h>
#include <string.h>

#define RANGE 8   /* point number within each cut  */
#define TRUE 1
#define pi 3.1415
#define INIT 0.0 /* initial displacement, m.  */
#define INIT_V 0.0 /* initial velocity, m/sec.   */

FILE *data_in1, *data_out1, *forcey;
/* data_in1: basic cutting data file  */

void main(int argc, char *argv[])
{
 char name1[20], name2[20], outfile[20];

 float *Rmax = NULL, *rmax = NULL;
 float vel, feed, depth, diam, Ks;
 float nose, k0, alpha, gamma, phai, phai1, flank, diff;
 float tmpp = 0., tmpa = 0., feed1 = 0.;
 float tmp1 = 0., tmp2 = 0., tmp2_1 = 0., tmp2_2 = 0., tmp3 = 0.;

 int  j = 0, cut = 0;

 FILE *data_out;

 /* function declaration   */
```

```
float sur_vib(int j, float vel, float feed, float depth, float diam,
        float flank, float alpha, float phi, float Ks);

 if ( argc == 3 ) {
        strcpy(name1,argv[1]);
        strcpy(outfile,argv[2]);
        clrscr();
        printf("\n\n");
        }
 else {
        clrscr();
        printf("Two command line auguments:\n");
        printf("cut_data file and output file!\n");
        exit(1);
        }

 /* open input and output files  */
if ((data_in1 = fopen (name1, "r")) == NULL) {
        printf("\nERROR - Cannot open the designated cut_data file\n");
        exit(1);
         }

data_out = fopen(outfile, "w");
forcey = fopen("force.dat", "w");
data_out1 = fopen("zt.dat", "w");

/* read in the basic cutting data  */
fscanf(data_in1, "%d", &cut);  /* cut number  */
fscanf(data_in1, "%f%f", &vel, &depth);
 /* cutting conditions: speed, depth of cut  */

fscanf(data_in1, "%f", &diam);
 /* workpiece data: cutting length, diameter  */

fscanf(data_in1, "%f%f%f%f%f", &nose, &alpha, &gamma, &phai, &phai1);
 /* tool data: nose radius, relief angle, rake angle, cutting edge angle */

printf("enter Ks, feedrate and flank wear:\n");
scanf("%f%f%f", &Ks, &feed, &flank);

phai = phai * pi/180.0;
phai1 = phai1 * pi/180.0;
```

```
alpha = alpha * pi/180.0;
gamma = gamma * pi/180.0;

/* memory allocation */
 Rmax = (float *) malloc(cut * sizeof(float));
 rmax = (float *) malloc(cut * sizeof(float));

 if(Rmax == NULL) {
        printf(" No memory allocation for Rmax\n");
        exit(1);
        }

 if(rmax == NULL) {
        printf(" No memory allocation for rmax\n");
        exit(1);
        }

/* loop to calculate Rmax for each cut.  */

for(j = 0; j < cut; ++j)
 {
/* determine Rmax due to vibration */
 tmp3 = sur_vib(j, vel, feed, depth, diam, flank, alpha, phai, Ks);
 tmp3 = 1000000.0 * tmp3 ;  /* convert m. to micron  */

/* loop to calculate Rmax value (number = POINT) within each cut  */
/* concentrated flank wear = 1.3333 * flank wear (Solaja)
        difference between concentrated flank wear and flank wear */
                diff = 0.3333 * flank ;

/* nose change on principal cutting edge  */
         tmpp = flank * tan(alpha)/(1 - tan(alpha) * tan(gamma));

/* feed rate change due to concentrated flank wear  */
         feed1 = feed * sin(phai) - 0.001 * tmp3 * cos(phai);

/* nose change on auxiliary cutting edge  */
         k0 = 1.0;
         tmpa = tmpp * k0 * feed1 * sin(phai)/(2.0 * depth);
        /* k0 = tan(alphaa) * (1.0 - tan(alphap) * tan(gammap))/
                        (tan(alphap) * (1.0 - tan(alphaa) * tan(gammaa)));
                assume k0 = 1.0 , p - principal, a - auxiliary */
  /* from Rosenberg,(1956): hfa/hfp = Va/Vp = feed*sin(phai)/2*depth
```

```
    Va and Vp - chip velocity on auxiliary and principal flank face  */

/* theoretical Rmax value from tool geometry  */
/*         tmp1 = 1000.0 * pow(feed, 2)/(8.0 * (nose-tmpp)); */

           tmp1 = 1000.0 * pow(feed1, 2)/(8.0 *(nose-tmpp));

/* Rmax due to flank wear  */
/* 1. Yan: */
           tmp2 = 1000.0 * cos(phai) * tmpp;
/* 2. Parag:*/
           tmp2_1 = 1000.0 * (tmpp * sin(phai1) + tmpa * sin(phai));
/* 3. Solaja */
           tmp2_2 = 1000.0 * diff * tan(alpha)/(1 - tan(alpha) * tan(gamma));

        *(Rmax+j) = tmp1 + tmp2 + tmp3;

        *(rmax+j) = tmp1 + sqrt(pow(tmp2,2) + pow(tmp3,2));

 printf("Cut #%2d  Flank: %.4f\n", j+1, flank);
 printf("Basic:  %.2f  Tool wear:  %.2f   Vibration:   %.2f\n",
                        tmp1, tmp2, tmp3);
 printf("Rmax:  %.2f\n", *(Rmax+j));
 printf("\n");

 fprintf(data_out, "%.4f  %.2f  %.2f  %.2f  %.2f  %.2f\n",
                flank, tmp1, tmp2, tmp3, *(Rmax+j), *(rmax+j));
 }

 fclose(data_in1);
 fclose(data_out);
 fclose(data_out1);
 fclose(forcey);

 printf("Program executed!\n");
 fprintf(data_out, "End of calculation!\n");
}


float sur_vib(int j, float vel,float feed,float depth, float diam,
           float flank, float alpha, float phi, float Ks)
{
static float mw, crw, ksiw, kw;
```

```
static float mt, crt, ksit, kt;
static float h;
float cw, ct;
float a, T, t;
float *zt, *dot_zt;
float zzt, zzt1, dot_zzt, r_vib;

float xw[2], xt[2], uuw[2], uut[2];

int n, i, ne, count;

/* function rk4 declaration */

void rk4(float t, float dh, int ne, float uu[2], float x[2], float vel, float m,
        float c, float k, float feed, float depth, float diam, float alpha,
        float phi, float flank, float zzt, float zzt1, float dot_zzt, float Ks);

if(j == 0)  /* reading data once  */
{
 /* input required time step for numerical integration. */
fscanf(data_in1, "%f", &h);

 /*input m, c, k data for workpiece and tool  */
fscanf(data_in1, "%f%f", &ksiw, &kw);
fscanf(data_in1, "%f%f%f", &mt, &ksit, &kt);
}
/* chuck mass: 22.6 pound or 10.27 kg */
 mw = 10.27 + 0.25 * pi * 300.0 * pow(diam, 2) * 0.000000001 * 7800.0;
 crw = 2.0 * sqrt(mw * kw);
 cw = ksiw * crw;

/* tool+transducer mass: 2.82 kg  */
 crt = 2.0 * sqrt(mt * kt);
 ct = ksit * crt;

 a = 0.0;
 T = 0.10;
 n = floor((T - a)/h);
/* h = 6.2832/n; interval data for test*/

 /* memory allocation  */
zt = (float *) malloc( n * sizeof(float));
dot_zt = (float *) malloc( n * sizeof(float));
```

```
 t = a;
 ne = 2;
 vel = vel / 60.0;   /* converted to m/s */

/*printf("time        xw          xt          z(t)\n"); */
fprintf(data_out1,"Cut #%2d\n", j+1);
fprintf(data_out1,
 "time       xw        xt           z(t)\n");

/* data for test */
/* n = 40;
 h = 6.2832/n;
 mw = 1.0;
 cw = 0.2;
 kw = 1.0;

 mt = 1.0;
 ct = 0.2;
 kt = 1.0;  */

/* initial conditions   */
xw[0] = INIT;
xw[1] = INIT_V;
xt[0] = INIT;
xt[1] = INIT_V;

/* initial relative displacement   */
*(zt+0) = INIT;
*(dot_zt+0) = INIT_V;

zzt = *(zt+0);
dot_zzt = *(dot_zt+0);

zzt1 = INIT; /* relative displacement during previous cut */

r_vib = 0.;
count = 1;
/* loop to find the vibration response
        using fourth-order Runge-Kutta method */

for(i = 1; i <= n; ++i)
 {
```

```
 /* call runge-kutta function   */
 rk4(t, h, ne, uuw, xw, vel,mw, cw, kw, feed, depth, diam, alpha, phi,
          flank, zzt, zzt1, dot_zzt, Ks);
 rk4(t, h, ne, uut, xt,vel, mt, ct, kt, feed, depth, diam, alpha, phi,
          flank, zzt, zzt1,dot_zzt, Ks);

/* Now, the relative motion   */
 zzt1 = zzt; /* keep the present value as the previous one  */

 /* if(xw[0] < 0. &&  xt[0] < 0)
          {
     *(zt+i) = fabs(xw[0] + xt[0]);
    }
  else if(xw[0] < 0)
    {
     *(zt+i) = fabs(xw[0]);
    }
  else if(xt[0] < 0)
    {
     *(zt+i) = fabs(xt[0]);
    }
  else
    {
   *(zt+i) = 0.;
          }  */

 *(zt+i) = xw[0] + xt[0];
 zzt = *(zt + i);

 *(dot_zt+i) = xw[1] + xt[1]; /* new value for relative velocity  */
 dot_zzt = *(dot_zt + i);

 r_vib = r_vib + zzt;
 count++;

 t = t + h;

/*printf("%6.4f %12.3e %12.3e %12.3e\n",  t, xw[0], xt[0], *(zt+i)); */
 fprintf(data_out1, "%6.4f %12.3e %12.3e %12.3e\n", t, xw[0], xt[0], *(zt+i));

 }
 r_vib = r_vib/count;
```

```
/* printf("vibration:  %12.3e\n", r_vib); */
 fprintf(data_out1, "average u(t):  %14.8e\n", r_vib);
 fprintf(data_out1, "\n");

 return(r_vib);
 }


/* function rk4 definition  */

void rk4(float t, float dh, int ne, float uu[2], float x[2], float vel,float m,
          float c, float k, float feed, float depth, float diam, float alpha,
          float phi, float flank, float zzt, float zzt1, float dot_zzt, float Ks)
{
 float a[2], K1[2], K2[2], K3[2], K4[2];
 float tn;
 int i;

 /* function declaration  */
void fun1(float t, float a[2], float x[2], float vel,float m,
          float c, float k, float alpha, float phi, float flank,
          float zzt, float zzt1, float dot_zzt, float feed,
          float depth, float dia, float Ks);

 fun1(t, a, x, vel,m, c, k, alpha, phi, flank, zzt, zzt1, dot_zzt, feed, depth, diam, Ks);
 for(i = 0; i < ne; ++i)
  {
        K1[i] = dh * a[i];
        uu[i] = x[i] + K1[i]/2.0;
  }
  tn = t + dh/2.0;

 fun1(tn, a, uu, vel,m, c, k, alpha, phi, flank, zzt, zzt1, dot_zzt, feed, depth, diam, Ks);
 for(i = 0; i < ne; ++i)
  {
        K2[i] = a[i] * dh;
        uu[i] = x[i] + K2[i]/2.0;
  }

 fun1(tn, a, uu, vel,m, c, k, alpha, phi, flank, zzt, zzt1, dot_zzt, feed, depth, diam, Ks);
 for(i = 0; i < ne; ++i)
  {
        K3[i] = a[i] * dh;
```

```
        uu[i] = x[i] + K3[i];
  }
  tn = t + dh;

 fun1(tn, a, uu, vel, m, c, k, alpha, phi, flank, zzt,zzt1, dot_zzt,feed,depth, diam, Ks);
 for(i = 0; i < ne; ++i)
  {
        K4[i] = a[i] * dh;
        x[i] = x[i] + (K1[i] + 2.0*K2[i] + 2.0*K3[i] + K4[i])/6.0;
  }
 t = t + dh;

return;

}

void fun1(float t, float a[2], float x[2], float vel,float m, float c, float k,
          float alpha, float phi, float flank, float zzt, float zzt1, float dot_zzt,
          float feed, float depth, float dia, float Ks)
{
 float olap = 0.8;
 float force, f1, fy, f2, f3;
 float Khf = 150.0;
 float cd = 0.; /* damping coefficient */
 float factor = 0.55; /* ratio between radial and tangential */

 float  mu = 0.05, K = 0.25, sigmay = 45.0; /* yield strength */

/* determine dynamic cutting force first. */
/* f1 --- principal component of dynamic cutting force  */

  f1 = Ks * factor * ((depth - (flank*tan(alpha)*cos(phi) + zzt))/sin(phi)) *
                  ((feed - olap * (zzt-zzt1) / tan(phi))*sin(phi));
        /* zzt-zzt1 -- the effect from previous cut */

/* f2 --- component due to dampling effect */
 cd = mu * (depth - (flank*tan(alpha)*cos(phi) + zzt)) *
     K * (1.57 - 2.0 * alpha) * sigmay/(vel * pow(alpha,2));
        /* based on  T.R. Sission, 1969 */

 f2 = cd * dot_zzt;

/* f3 --- component due to tool wear  */
```

```
 f3 = Khf * (depth - (flank*tan(alpha)*cos(phi) + zzt)) * flank;

force = f1 - f2 + f3;
fy = f1/factor - f2 + f3/factor;

fprintf(forcey, "%6.3f %6.3f %6.3f %6.3f\n", f1/factor, f2, f3, fy);

 /* test force  */
/* force = 1.0 - sin(0.5 * t); */

a[0] = x[1];

a[1] = (force - c * x[1] - k * x[0])/ m;

return;
}
```

APPENDIX C: Relationship between the RMS slope and R_{max} [S87]

A relationship between the RMS slope of the surface profile and R_{max} will be derived in this appendix.

A geometrical relation can be found from Fig.C1 as:

$$x^2 + (y - \rho_o)^2 = \rho_o^2 \tag{C1}$$

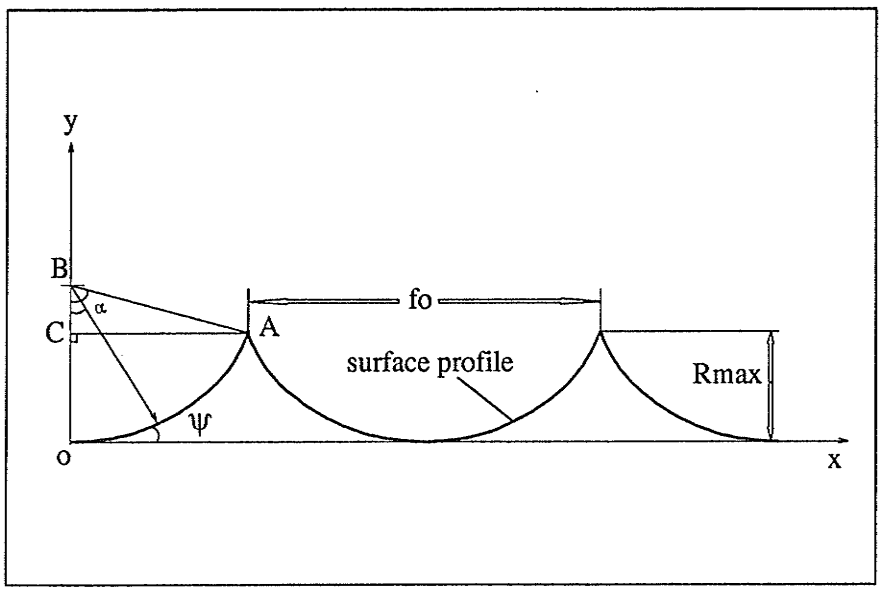

Fig.C1 Geometrical details needed to determine the RMS slope of a surface profile.

The slope of the surface profile, OA, shown in Fig.C1 is

$$\frac{dy}{dx} = \frac{x}{\rho_o - y} = \tan\psi \tag{C2}$$

where ψ is the angle between two tangents of the profile with one at O and the other at the point concerned. The mean angle of the slope can be written as:

$$mean(\frac{dy}{dx}) = \frac{1}{\alpha}\int_0^{\alpha} \tan\psi d\psi \qquad \text{(C3)}$$

where α is the angle between BO and BA. Similarly, the RMS slope, u, is:

$$u = \sqrt{\frac{1}{\alpha}\int_0^{\alpha} (\tan\psi)^2 d\psi} \qquad \text{(C4)}$$

and, it is straightforward to obtain

$$u^2 = \frac{1}{\alpha}\tan\alpha - 1. \qquad \text{(C5)}$$

On the other hand, the following relations hold from Fig.C1:

$$AB = R_{\max}, BC = \rho_o - R_{\max}, AC = \sqrt{R_{\max}(2\rho_o - R_{\max})} \qquad \text{(C6)}$$

$$\alpha = \sin^{-1}(\frac{AC}{AB}) = \sin^{-1}(\frac{\sqrt{R_{\max}(2\rho_o - R_{\max})}}{\rho_o - R_{\max}}) \qquad \text{(C7)}$$

and

$$\tan\alpha = \frac{AC}{BC} = \frac{\sqrt{R_{\max}(2\rho_o - R_{\max})}}{\rho_o - R_{\max}}. \qquad \text{(C8)}$$

The ρ_o is usually less than 1 mm and $R_{\max}$ is less than 30 μm in finish turning, so that equation (C7) can be written as:

$$\alpha \approx \frac{\sqrt{R_{\max}(2\rho_o - R_{\max})}}{\rho_o}. \qquad \text{(C9)}$$

Substituting equations (C8) and (C9) into equation (C5) yields:

$$u \approx \sqrt{\frac{R_{\max}}{\rho_o - R_{\max}}}. \qquad \text{(C10)}$$

APPENDIX D: PC401 CCD evaluation board+

The PC401 is designed to facilitate the evaluation of the Texas Instruments' range of linear CCD imagers. It operates the following

TC103 (2048X1 pixel organization)

TC104 (3456X1 pixel organization)

TC106 (2592X1 pixel organization)

The board is intended to be used as a construction aid for experimental systems using the above CCD linear imagers. It contains the basic operating electronics to drive the CCD imagers and requires only the input of power supplies.

The board contains an internal clock generator that can provide data transfer rates from 200 KHz to 2 MHz. If required, this internal clock can be overridden by an input from an external clock. The board also provides an internal provision for exposure times from 1 ms to 16 ms, which interface the logic CCD levels.

The low level Transport, Transfer and Reset clocks are controlled directly from one of the external supplies (V_{IL}). The high level Transport, Transfer and Reset clock levels are supplied from the board and can be adjusted (V_{CH}(adj)).

The output is buffered to drive a high capacitive load.

The electronics are all located on the opposite side of the board to the imager socket to give maximum flexibility in optics, and to facilitate experimental modifications.

Procedure:

1) Adjust power supplies as follows:

+5 volts, current limit 200 mA.

+ Excerpted from *PC401/PC402 evaluation board for linear CCD imagers*, Texas Instruments, 1992.

+16 volts, current limit 100 mA.

-16 volts, current limit 50 mA.

2) Connect power supplies to the evaluation board as shown in Fig.D1.

3) Adjust the oscilloscope with the main sweep set to 2 ms per division and vertical sensitivity set to 5 volts per division. Remove jumper (J2) first for white reference, connect the external sync of the oscilloscope to the white reference clock (WRCK) and test point shown in Fig.D1 to synchronize the oscilloscope with the exposure time rate.

4) The following clock voltage amplitudes can be checked at the test points shown in Fig.D1.

	Clock	Low level	High level
WRCK	white reference clock	6 to 8 v	15 to 16 v
XCK	Transfer clock*	-16 to -13 v	1 to 5 v
TCK	Transport clock*	-16 to -13 v	1 to 5 v
RCK	Reset clock*	-16 to -13 v	1 to 5 v

* Set per CCD specification in data sheet for the device being evaluated.

The high level voltage for XCK, TCK, and RCK are adjustable by using the V_{CH} (adj) potentiometer shown in Fig.D1. The low voltage is set by adjusting the -16 volt power supply (V_{IL}). The high level clock voltage recommended for the TC103, TC104, and TC106 is nominally +4V, generally +3.5V is necessary for the high level of the Transport clock ($V_{IH}(T)$). If a high enough Transport clock voltage is not provided, then the full data sheet dynamic range and transfer efficiency may not be realized.

5) When the clock levels have been checked, plug the CCD imager to be evaluated into the socket and connect the vertical oscilloscope probe to the output-signal test point.(See Fig.D1.) Set the vertical amplifier to "ac coupled" and set the sensitivity to 0.2V/division.

Note: Recheck clock levels after inserting the CCD into the socket and readjust clock levels if necessary.

6) Begin with a low level light on the imager and increase the light level until an output level of approximately 0.4 volts is obtained. Note that if a shadow is cast across the imager, the amplitude of the output signal is affected in the region of the shadow.

7) Check the effects of exposure time by adjusting the exposure time control. (See Fig.D1.) Observe the time between the white reference output signals. (See Fig.D2.) Set the exposure time to 10 ms. Instructions for the external exposure time control are given in the next section.

Equipment required for typical setup

1) Oscilloscope (Tektronix Model 765 or equivalent).
2) Current limited power supplies (Two HP 6216A, one HP6214A).
3) Edge connector (TRW/Cinch 50-44A-30, 22 way 0.156" pitch).
4) CCD imager under evaluation.

Power supply requirements

1) +5 volts, 160 mA. 2) +16 volts, 60 mA. 3) -16 volts, 40 mA.

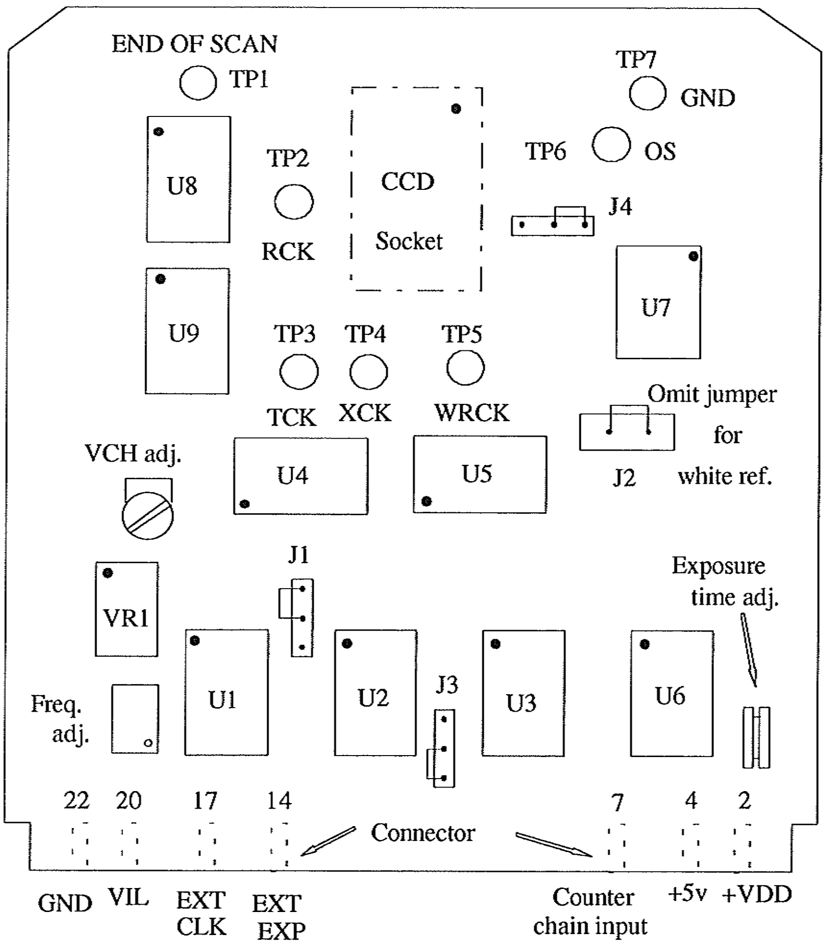

Fig.D1 Evaluation board for linear CCD imagers (J1= Internal/External clock select, J3 = Internal/External exposure time select, J4 = Internal/External reference voltage (V_{REF}), select J1 and J3 shown in internal select position.)

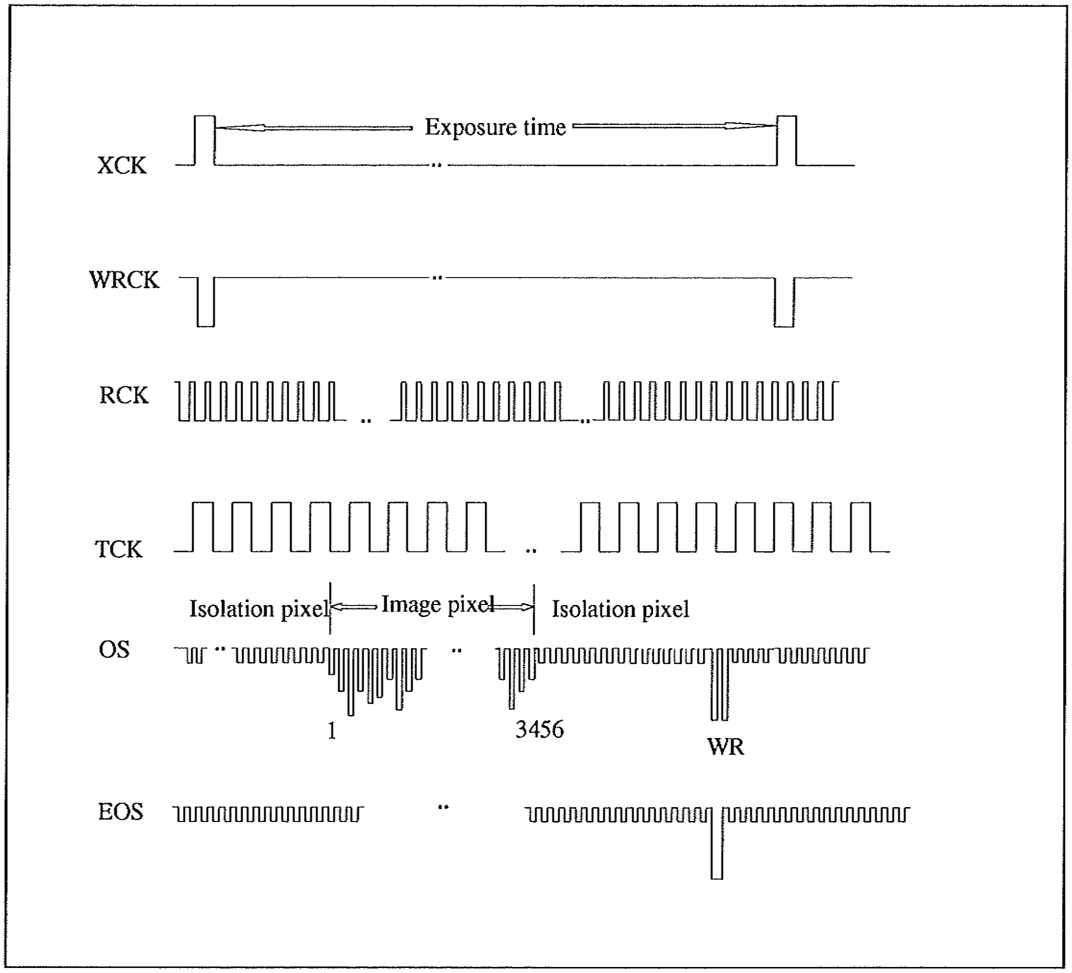

Fig.D2 Input and output voltage waveforms for the operation of the CCD evaluation board.

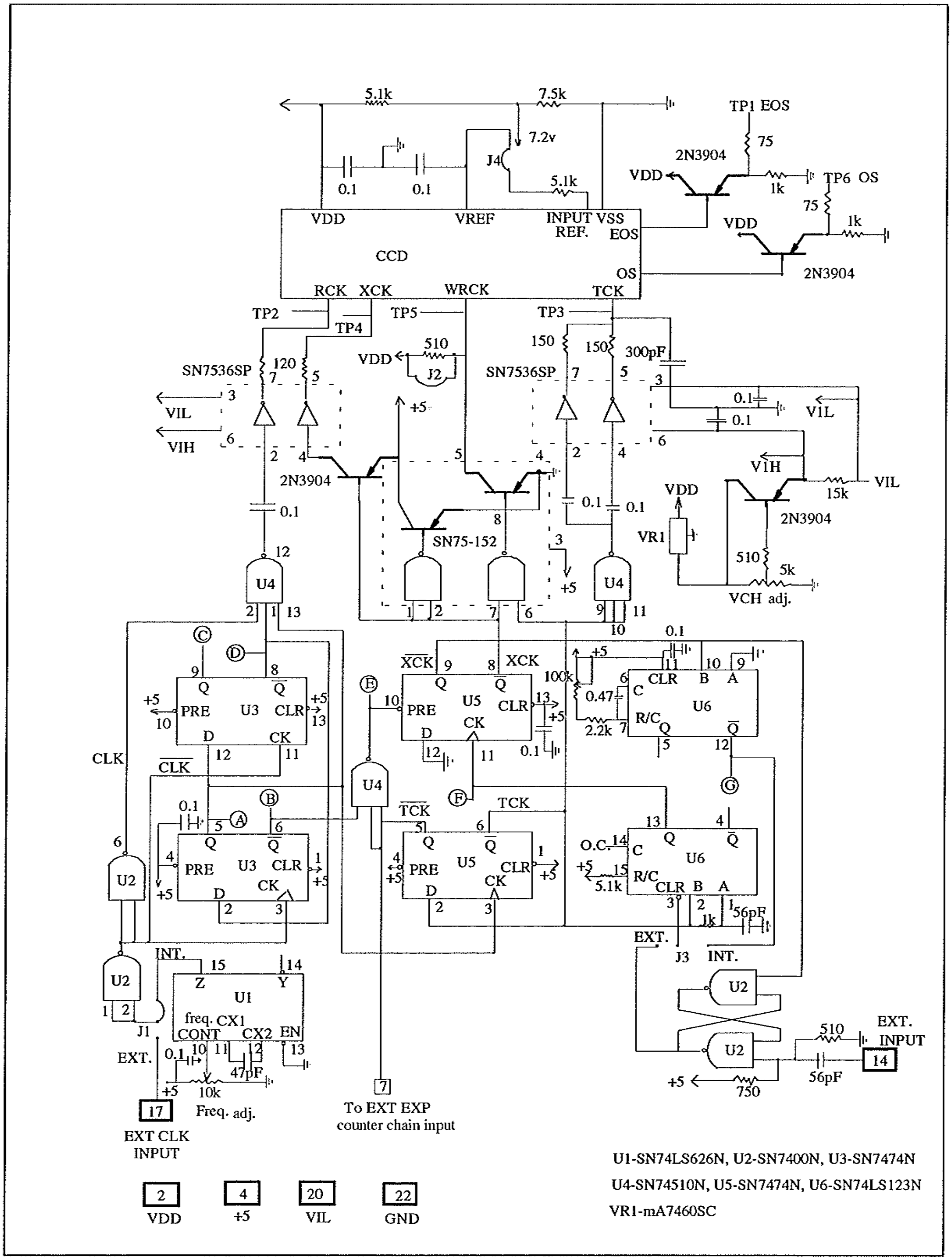

Fig.D3 Schematic of the CCD evaluation board.

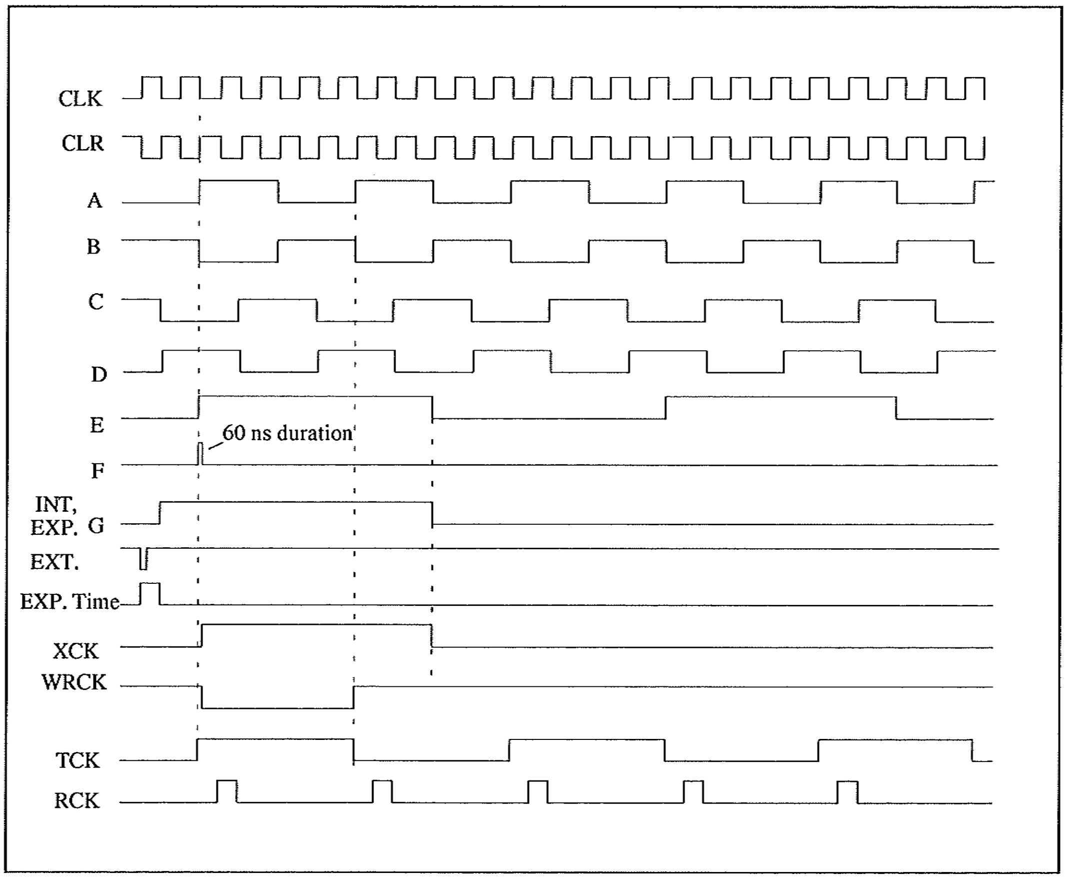

Fig.D4 Logic timing waveforms for the operation of the CCD evaluation board.

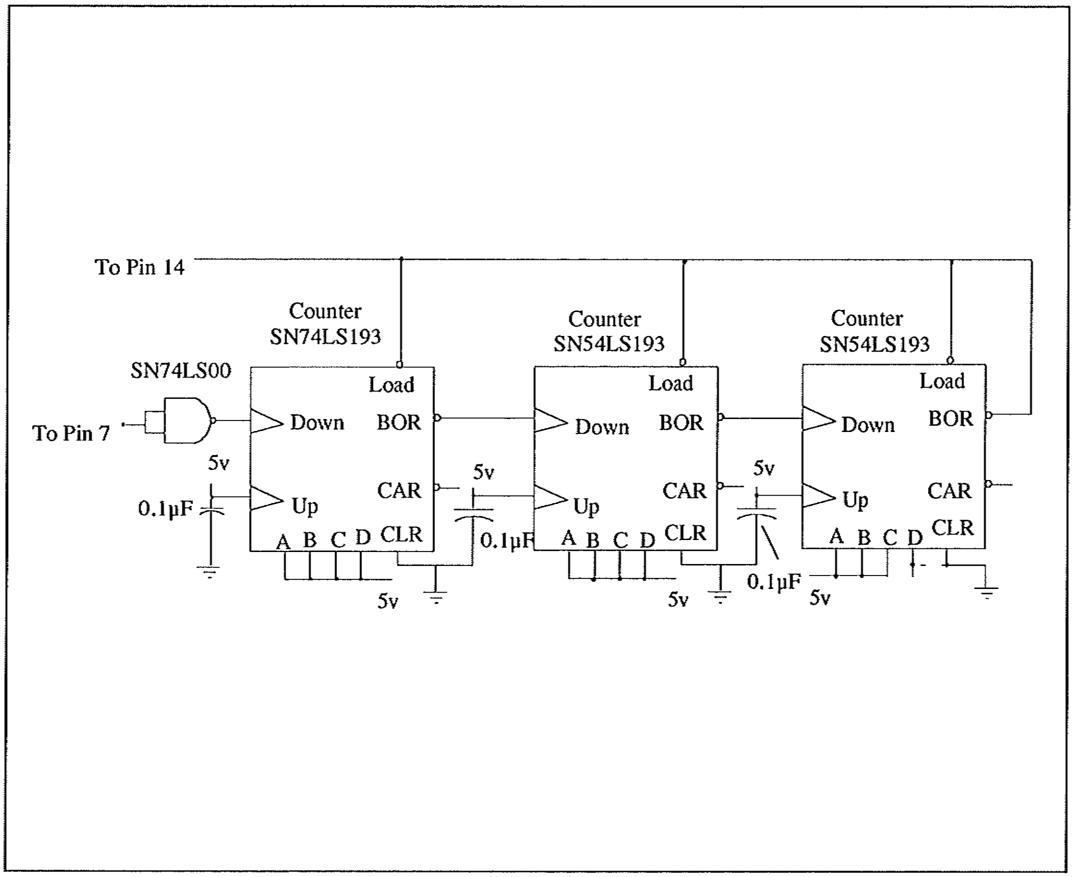

Fig.D5 External time exposure circuit of the CCD evaluation board.

APPENDIX E: Performance of dynamometer

A dynamometer, developed as part of a previous study [YJPB90], was used to measure the dynamic cutting forces, on-line, in three mutually perpendicular directions. It has a double-decked structure made of two octagonal, elastic rings having vertical and horizontal grooves. The dynamometer also serves as the tool post during turning. When a cutting force is applied, there will be a deflection in the elastic rings which are picked up by 24 strain gauges connected in the form of six Wheatstone bridges. The ideal locations for bonding the strain gauges on the dynamometer were determined from a sensitivity and crosstalk analysis. The commercial ANSYS package was used to do a finite element analysis of a model having 109 isoparametric solid elements (Type STIF 45) involving 276 nodes.The performance of the dynamometer is summarized below.

1. Load capacity (N): 1,000
2. Crosstalk:

Table E1 Crosstalk between the Measured Cutting Force Components.

Direction of applied load	Response in X (%)	Response in Y (%)	Response in Z (%)
Radial (X)		10.0	0.1
Tangential (Y)	40.0		0.8
Feed (Z)	15.0	9.0	

3. Minimum sensitivity (μstrain/N): 0.1

4. Maximum deflection at tool tip when a load of 900 N is applied (μm):

Radial-direction (X)	48.9
Tangential-direction (Y)	24.6
Feed-direction (Z)	11.9

5.

Table E2 Natural Frequencies (Hz).

Mode	ω_{ni}
1	1,203
2	1,273
3	1,881

6. Calibration curve:

- Radial direction (X)

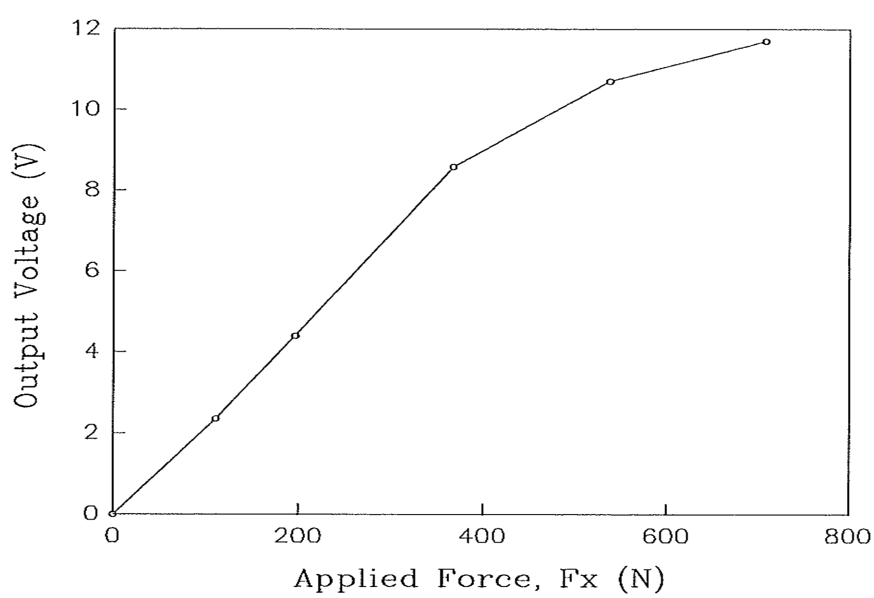

Fig.E1 Calibration curve of the dynamometer in the radial direction.

- Tangential direction (Y)

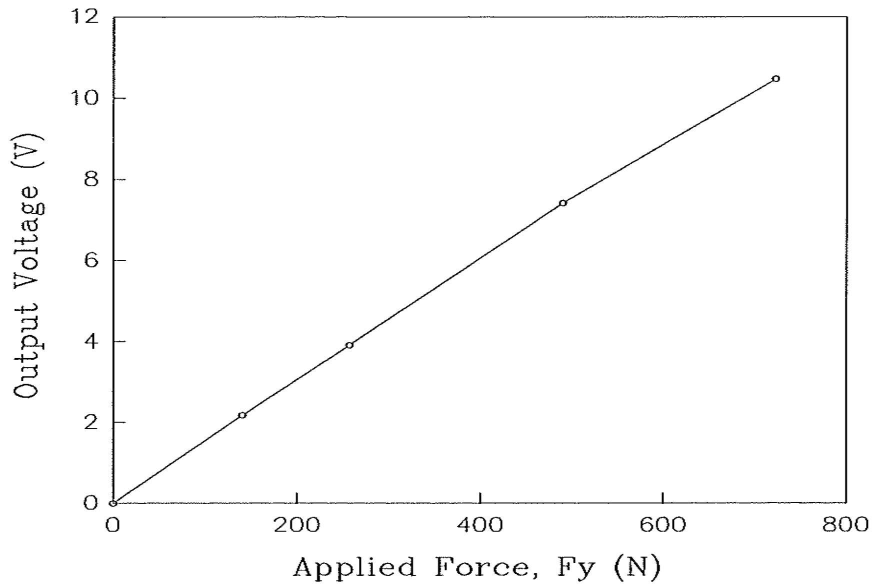

Fig.E2 Calibration curve of the dynamometer in the tangential direction.

- Feed direction (Z)

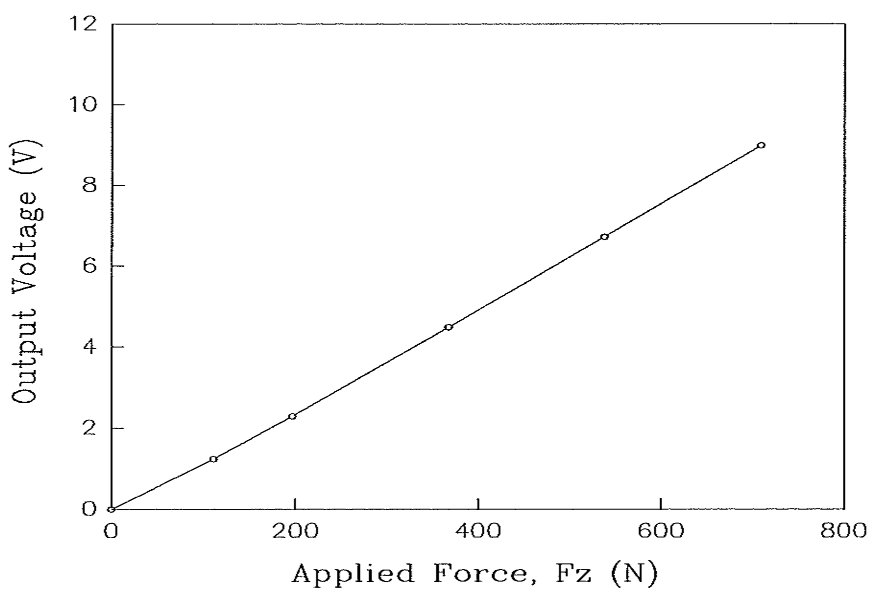

Fig.E3 Calibration curve of the dynamometer in the feed direction.

BIBLIOGRAPHY

[A62] ASA standard B46.1-1962, *American Standard on Surface Texture*, The American Standards Association, 1962.

[A89] ANSYS 4.2 package, Swanson Analysis Systems, Inc.,1989.

[B79] Bonac, T., "Wood roughness volume and depth estimated from pneumatic surface measurement", *Wood Science*, **11**(4), pp.227-232, 1979.

[B84] Bhattacharyya, A., *Metal Cutting Theory and Practice*, Central Book Publishers, Madras, India, 1984.

[BF93] Burden, R.L., and Faires, J.D., *Numerical analysis*, pp.254-260, 5th edition, PWS-Kent Publishing Co., Boston, MA., 1993.

[BGT85] Brodmann, R., Gerstorfer, O., and Thurn, G.,"Optical roughness measuring instrument for fine-machined surfaces", *Optical Engineering*, **24**(3), pp.408-413, 1985.

[BM89] Bennett, J.M., and Mattsson, L., *Introduction to Surface Roughness and Scattering*, The Optical Society of America, Washington D.C., 1989

[BPW79] Beightler, C.S., Phillips, D.T., and Wilde, D.J., *Foundations of optimization*, pp.186-189, Prentice -Hall Inc., Eaglewood, N.J., 1979

[BS63] Beckmann, P., and Spizzichino, A., *The Scattering of Electromagnetic Waves from Rough Surfaces*, pp.9-98, Pergamon Press, London, 1963.

[BSI61] BSI 1134-1961, *Central-Line-Average Height Method for the Assessment of Surface Texture*, British Standards, 1961.

[BSI88] Blum, T., Suzuki, I., and Inasaki, I., "Development of a condition monitoring system for cutting tools using an acoustic emission sensor", *Bulletin of the Japanese Society of Precision Engineering*, **22**, pp.301-308, 1988.

[C70] Crane, R.B., "Use of a laser-produced speckle patterns to determine surface roughness", *Journal of the Optical Society of America*, **60**(12), pp.1658-1663, 1970.

[CB87] Constantinides, N., and Bennett, S.,"Investigation of methods for the on-line estimation of tool wear", *International Journal of Machine Tools and Manufacture*, **27**(2), pp.225-237, 1987.

[CH92] Cuthbert, L., and Huynh, V.M.,"Statistical analysis of optical fourier transform patterns for surface texture assessment", *Measurement Science and Technology*, **3**, pp.740-745, 1992.

[CSA62] CSA standard B95-1962, *Surface Texture (Roughness, Waviness and Lay)*, the Canadian Standards Association, Ottawa, July 1962.

[CVGL91] Cao, L.X., Vorburger, T.V., Lieberman, A.G., and Lettieri, T.R.,"Light-scattering measurement of the rms slopes of rough surfaces", *Applied Optics*, **30**(22), pp.3221-3227, 1991.

[D85] Davis, C.M., "Fiber optic sensors: an overview", *Optical Engineering*, **24**(2), pp.347-351, 1985.

[DEL92] Du, R.X, Elbestawi, M.A., and Li, S., "Tool condition monitoring in turning using fuzzy set theory", *International Journal of Machine Tools and Manufacture*, **32**(6), pp.781-796, 1992.

[FA77] Fujii, H., and Asakura, T., "Roughness measurements of metal surfaces using laser speckle", *Journal of the Optical Society of America*, **67**(9), pp.1171-11176, 1977.

[FAS76] Fujii, H., Asakura, T., and Shindo, Y., "Measurement of surface roughness properties by using image speckle contrast", *Journal of the Optical Society of America,* **66**(11), pp.1217-1221, 1976.

[FH92] Fan, Y., and Huynh, V.M, "Investigation of light scattering from rough periodic surfaces - Numerical solutions", *Optics and Laser Technology,* **24**(3), pp.145-150, 1992

[FS91] Freeman J.A., and Skapura, D.M., *Neural Networks: Algorithms, Applications, and Programming Techniques,* pp.89-110, Addison-Wesley Publishing Company Inc., Reading, MA., 1991.

[FUA76] Fujii, H., Uozumi, J., and Asakura, T., "Computer simulation study of the image speckle patterns with relation to object surface profile", *Journal of the Optical Society of America,* **66**(11), pp.1222-1236, 1976.

[G75] Goodman, J.W., "Statistical properties of laser speckle patterns", from *Laser Speckle and Related Phenomena*", edited by J.C. Dainty, Springer, Berlin, 1975.

[G76] George, N., "Speckle from rough, moving objects", *Journal of the Optical Society of America,* **66**(11), pp.1182-1203, 1976.

[GKJR92] Garbini, J.l., Koh, S., Jorgensen, J.E., and Ramula, M., "Surface profile measurement during turning using fringe-field capacitive profilometry", *ASME Journal of Engineering for Industry,* **114**, pp.234-243, June 1992.

[GKY91] Gupta, P., Kaye, J.E., and Yan, D., "Predictability of tool wear and surface roughness", *Proceedings of the Canadian Conference of Applied Mechanics,* pp.694-695, Winnipeg, 1991.

[GRL88] Gyimesi, M., Richter, P., and Lorincz, E., "Derivation of surface macroroughness parameters of a random rough surface from an optical interferometric measurement", *Applied Optics*, **27**(5), pp.983-986, 1988.

[GST87] Giusti, F., Santochi, M., and Tantussi, G., "On-line sensing of flank and crater wear of cutting tools", *Annals of the CIRP*, **36**(1), pp.41-44, 1987.

[H93a] Hammerstrom, D., "Neural networks at work", *IEEE Spectrum Magazine*, pp.26-32, June 1993.

[H93b] Hammerstrom, D., "Working with neural networks", *IEEE Spectrum Magazine*, pp. 46-53, July 1993.

[HF92] Huynh, V.M., and Fan, Y.,"Surface-texture measurement and characterisation with applications to machine-tool monitoring", *International Journal of Advanced Manufacturing Technology*, **7**(1), pp.2-10, 1992.

[HH93] Hush, D.R., and Horne, B.G., "Progress in supervised neural networks", *IEEE Signal Processing Magazine*, **10**(1), pp.8-39, January 1993.

[HSL76] Hasegawa, M., Seireg, A., and Lindberg, R.A., "Surface roughness model for turning.", *Tribology International*, pp.285-289, Dec. 1976.

[IEDU93] Ismail, F., Elbestawi, M.A., Du, R., and Urbasik, K.,"Generation of milled surfaces including tool dynamics and wear", *ASME Journal of Engineering for Industry*, **115**, pp.245-252, Aug. 1993.

[J85] M. Johnson, "Fiber displacement sensors for metrology and control", *Optical Engineering*, **24**(6), pp.961-965, 1985.

[JIS76] JIS standard B0651-1976, *Instruments for the Measurement of Surface Roughness by the Stylus Method*, Japanese Industrial Standards, Tokyo, 1976.

[JRB84] Jansson, D.G., Rourke, J.M., and Bell, A.C., "High-speed surface roughness measurement", *ASME Journal of Engineering for Industry*, **106**, pp.34-39, Feb. 1984.

[JS89] Jang, D.Y., and Seireg, A.,"Dynamic simulation for predicting surface roughness in turning", *Proceedings of the Canadian Conference of Applied Mechanics*, pp.31-36, Montreal, 1989.

[K67] Kronenberg, M., *Machining science and applications*, Pergamon Press, London, 1967.

[K78] Koren, Y., "Flank wear models of cutting tools using control theory", *ASME Journal of Engineering for Industry*, **100**, pp.103-109, Feb. 1978.

[K84] Kluft, W., "In-process monitoring of cutting tool forces", *2nd Biennial International Machine-tool Conference, the National Machine-tool Builders' Association, Germany, 1984.*

[K90] Koelsch, J.R.,"Sensors-the missing link", *Manufacturing Engineering*, pp.53-56, Oct. 1990.

[K91] Koren, Y., "Flank wear estimation under varying cutting conditions", *ASME Journal of Dynamic Systems, Measurement and Control,* **113**, pp.300-307, June 1991.

[KBPM90] Kaye, J.E., Balakrishnan, S., Popplewell, N., and Menzies, R., "High resolution adaptive tachometer for spindle motor drives", *International Journal of Electronics*, **69**(6), pp.835-848, 1990.

[KYPB93] Kaye, J.E., Yan, D., Popplewell, N., and Balakrishnan, S., "Predicting tool flank wear by using spindle speed measurement", *International Journal of Machine Tools & Manufacture.* Submitted for publication June 1993.

[KYPBT93]Kaye, J.E., Yan, D., Popplewell, N., Balakrishnan, S., and Thomsom, D.J., "Electronic System for Surface Roughness Measurements in Turning", *International Journal of Electronics*, **69**(6), pp.835-848, 1993.

[L78] Lonardo, P.M., "Testing a new optical sensor for in-process detection of surface roughness", *CIRP Annals*, **27**, p.531, 1978.

[L87] Lippmann, R.P., "An introduction to computing with neural nets", *IEEE Acoustics, Speech and Signal Processing Magazine*, **4**(2), pp.4-22, April 1987.

[LHN89] Luk, F., Huynh, V.M., and Norht, W., "Measurement of surface roughness by machine vision system", *Journal of Physics E: Scientific Instrument*, **22**, pp.977-980, 1989.

[LLG89] Lee, L.G., Lee, K.S., and Gan, C.S.,"On the correlation between dynamic cutting force and tool wear", *International Journal of Machine Tools and Manufacture*, **29**(3), pp.295-303, 1989.

[LM90] Li, D., and Mathew, J., "Tool wear and failure monitoring techniques for turning-a review", *International Journal of Machine Tools and Manufacture*, **30**(4), pp.579-598, 1990.

[LMP75] Leger, D., Mathieu, E., and Perrin, J.C., "Optical surface roughness determination using speckle correlation technique", *Applied Optics*, **14**(4), pp.872-877, 1975.

[LMSV91] Lettieri, T.R., Marx, E., Song, J.F., and Vorburger, T.V., "Light scattering from glossy coatings on paper", *Applied Optics*, **30**(30), pp4439-4447, 1991.

[LP76] Leger, D., and Perrin, J.C., "Real-time measurement of surface roughness by correlation of speckle patterns", *Journal of the Optical Society of America*, **66**, pp.1210-1217, 1976.

[MFY87] Maeda, Y., Fujino, D., and Yamamoto, A.,"Convenient estimation of machined surface profiles in cylindrical turning", *Bulletin of the Japanese Society of Precision Engineering*, **21**(1), pp.57-59, 1987.

[MKST85] Mori, M., Kumehara, H., Suda, H., and Takahashi,K., "Variation of machined surface roughness with progressing tool wear", *Bulletin of the Japanese Society of Mechanical Engineering*, **19**(3), pp.216-217, 1985.

[MM88] Mital, A., and Mehta, M., "Surface finish prediction models for fine turning", *International Journal of Production Research*, **26**(12), pp.1861-1876, 1988.

[MR88] McClelland, J.L., and Rumelhart, D.E., *Explorations in Parallel Distributed Processing*, pp.130-142, The MIT Press, Cambridge, MA., 1988.

[MR92] Moon, Y.B., and Roy, U., "Learning group-technology part families from solid models by parallel distributed processing", *International Journal of Advanced Manufacturing Technology*, **7**(2), pp.109-118, 1992.

[MV90] Marx, E.,and Vorburger, T.V.,"Direct and inverse problems for light scattered by rough surfaces", *Applied Optics*, **29**(25), pp.3613-3626, 1990.

[O68] Olsen, K.V., "Surface roughness on turned steel components and relevant mathematical analysis",*The Production Engineer*, pp.593-606, December, 1968.

[OY91] Osakada, K., and Yang, B., "Application of neural networks to an expert system for cold forging", *International Journal of Machine Tools and Manufacture*, **31**(4), pp.577-587, 1990.

[P64] Pesante, M., "Determination of surface roughness typology by means of amplitude density curves", *CIRP Annalen*, **12**(2), pp.61-68, 1964.

[P74] Petropoulos, P.G., "Statistical basis for surface roughness assessment in oblique finish turning of steel components", *International Journal of Production Research*, **12**(3), pp.345-360, 1974.

[P90] Pedersen, K.B., "Wear measurement of cutting tools by computer vision", *International Journal of Machine Tools and Manufacture*, **30**(1), pp.131-139, 1990.

[PK67] Peklenik, J., "Investigation of the surface typology", *Annals of the CIRP*, **15**, pp.381-385, 1967.

[PK68] Peklenik, J., and Kubo, M., "A basic study of a three-dimensional assessment of the surface generated in a manufacturing process", *Annals of the CIRP*, **16**, pp.257-265, 1968.

[QTM88] Quercioli, F., Tiribilli, B., and Molesini, G.,"Optical surface profile transducer", *Optical Engineering*, **27**(2), pp.135-142, 1988.

[ROS73] Rakhit, A.K., Osman, M.O.M., and Sankar, T.S., "Machine tool vibrations: its effect on manufactured surfaces", *Proceedings of Canadian Congress of Applied Mechanics*, pp.463-464, Montreal, 1973.

[RSO76] Rakhit, A.K., Sankar, T.S., and Osman, M.O.M., "The influence of metal cutting forces on the formation of surface texture in turning", *International Journal of Machine Tools, Design and Research*, pp.281-292, **16**(4), 1976.

[S58] Solaja, V., "Wear of carbide tools and surface finish generated in finish turning of steel", *Wear*, **2**, pp.40-58, 1958.

[S63] Sata, T., " Surface finish in metal cutting", *Annals of the CIRP*, **12**(4), pp.190-197, 1963.

[S75] Selvam, M.S., "Tool vibration and its influence on surface roughness in turning", *Wear*, **35**, pp.149-157, 1975.

[S81] Shiraishi, M., "In-process measurement of surface roughness in turning by laser beams", *ASME Journal of Engineering for Industry*, **103**, pp.203-209, May 1981.

[S87] Shiraishi, M., "A consideration of surface roughness measurement by optical method", *ASME Journal of Engineering for Industry*, **109**, pp.100-105, May 1987.

[S89] Shaw, M.C., *Metal Cutting Principles*, pp.487-543, Oxford University Press, New York, N.Y., 1989

[SK69] Sisson, T.R., and Kegg, R.L., "An explanation of low speed chatter effects", *ASME Journal of Engineering for Industry,* **91**, pp.951-955, Nov. 1969.

[SL81] Sundaram, R.M., and Lambert, B.K., "Mathematical models to predict surface finish in fine turning of steel. Part 1", *International Journal of Production Research*, **19**(5), pp. 547-556, 1981.

[SO85] Sasaki, O., and Okazaki, H., "Detection of time-varying intensity distribution with CCD image sensors", *Applied Optics*, **24**(14), pp.2124-2126, 1985.

[SO87] Sato, H., and O-hori, M., "Surface roughness measurement using scanning electron microscope with digital processing", *ASME Journal of Engineering for Industry*, **109**, pp.106-111, May 1987.

[SR73] Selvam, M.S., and Radhakrishnan, V., "Characteristics of a surface machined with a single point tool", *Tribology*, pp.93-96, June 1973.

[SS90] Shiraishi, M., and Sato, S., "Dimensional and surface roughness controls in a turning operation", *ASME Journal of Engineering for Industry*, **112**, pp.78-83, Feb. 1990.

[T65] Tobias, S.A., *Machine tool vibration* , pp.201-221, J.Wiley and Sons, Inc., New York, 1965.

[TI83] Type TC104-1 CCD linear image sensor, *Product Data D2687*, Texas Instruments, Dec. 1983.

[TN86] Tsukada, T., and Noguchi, S., "An assessment of surface roughness by specific intensity distribution of reflected light", *Bulletin of the Japanese Society of Precision Engineering*, **20**(3), pp.203-204, 1986

[TSY74] Taraman, K., Swando, R., and Yamauchi, W., "Relationships between tool forces and flank wear", *SME technical paper MR74-704*, 1974.

[TWT91] Tansel, L.N., Wagiman, A., and Tziranis, A., "Recognition of chatter with neural networks", *International Journal of Machine Tools and Manufacture*, **31**(4), pp.539-552, 1991.

[UFA77] Uozumi, J., Fujii, H., and Asakura, T., "Furhter computer simulation study of the image speckle patterns with relation to object surface profile", *Journal of the Optical Society of America,* **67**(6), pp.808-815, 1977.

[VN92] Van Ooyen, A., and Nienhuis, B., " Improving the convergence of the back-propagation algorithm", *Neural Networks*, **5**, pp.465-471, 1992.

[VT80] Vorburger, T.V., and Teague, E.C., "Optical techniques for on-line measurement of surface topography", *Precision Engineering*, **3**(2), pp.61-83, 1980.

[W78] Whitehouse, D.J., "Surface-a link between manufacture and function", *Proceedings of Institute of Mechanical Engineers*, **192**, pp.179-188, 1978.

[WP78] Whitehouse, D.J., and Phillips, M.J., "Discrete properties of random surfaces", *Phil. trans. Royal Society,* **290 A**, pp.267-298, 1978.

[YCPB93] Yan, D.,Cheng,M.,Popplewell, N., and Balakrishnan, S.,"Real-time roughness measurements of a finish-turned surface", *International Journal of Production Research,* Submitted for publication Oct. 1993

[YDL90] Yoo, S.M., Dornfeld, D.A., and Lemaster, R.L., "Analysis and modeling of laser measurement system performance for wood surface", *ASME Journal of Engineering for Industry*, **112**, pp.69-77, Feb. 1990.

[YJPB90] Yan, D., Jiang, Z., Popplewell, N., and Balakrishnan, S., "A three-dimensional dynamometer for turning", *Research Report*, Department of Mechanical and Industrial Engineering,University of Manitoba, 1990.

[YK90] Yim, D.Y., and Kim, S.W., "Optical roughness measurement of ground surfaces by light scattering", *International Journal of Machine Tools and Manufacture*, **30**(2), pp.283-289, 1990.

[YKBP93] Yan, D., Kaye, J.E., Balakrishnan, S., and Popplewell, N., "Surface roughness measurement in finish turning", *International Journal of Advanced Manufacturing Technology*, Accepted for publication Sept.1, 1993

[YKPB93] Yan, D., Kaye, J.E., Popplewell, N., and Balakrishnan, S., "On-line prediction of surface roughness in finish turning", *ASME Journal of Engineering for Industry*, Submitted for publication Sept. 1993

[YN87] Yanagi, K., and Nakamura, M., "An indirect optical measuring method of surface roughness on internal ground surface", *Bulletin of the Japanese Society of Precision Engineering*, **21**(3), pp.217-218, 1987

[Z92] Zurada, J.M., *Introduction to Artificial Neural Systems*, West Publishing Company, St. Paul, MN, 1992.

[ZK91a] Zhang, G.M., and Kapoor, S.G., "Dynamic generation of machined surfaces, Part 1: Description of a random excitation system", *ASME Journal of Engineering for Industry*, **113**, pp.137-144, May 1991.

[ZK91b] Zhang, G.M., and Kapoor, S.G., "Dynamic generation of machined surfaces, Part 2: Construction of surface topography", *ASME Journal of Engineering for Industry*, **113**, pp.145-153, May 1991.

[ZYL91] Zhang, G.M., Yerramareddy, Y., Lee, S.M., and Lu, S.C-Y, "Simulation of surface topography formed during the intermittent turning process", *ASME Journal of Dynamic Systems, Measurement, and Control*, **113**, pp. 273-279, June 1991.