THE MODELLING AND CORRECTION OF BALL-SCREW GEOMETRIC, THERMAL AND LOAD ERRORS ON CNC MACHINE TOOLS

GEOFFREY HOLROYD

A Thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for a degree of Doctor of Philosophy

The University of Huddersfield
2007

ABSTRACT

In the modern global economy, there is a demand for high precision in manufacture as competitive pressures drive businesses to seek greater productivity. This results in a demand for a reduction in the errors associated with CNC machine tools. To this end, it is useful to develop a greater understanding of the mechanisms which give rise to errors in machine tool drives.

This programme of research covers the geometric, thermal and load errors commonly encountered on CNC machine tools. Several mathematical models have been developed or extended which enable a deeper understanding of the interaction between these errors, various details of ballscrew design and the dynamic behaviour of ballscrew driven systems.

Some useful models based on the discrete matter or "lumped mass" approach have been devised. One extends the classical eigenvalue method for finding the natural frequencies and other dynamic characteristics of ballscrew systems to include viscous damping effects using a generalised eigenvalue approach. This gives the damping coefficient of each predicted vibration mode along with the estimates of the natural frequencies, enabling many of the natural frequencies predicted by standard undamped natural frequency analyses to be dismissed as being of little consequence to the vibratory behaviour of the system.

A development of this modelling method gives the sensitivity of the system to changes in stiffness and damping characteristics, which is helpful at the preliminary design stage of a ballscrew system, and is an aid in deciding the most convenient remedy to vibration problems which may occur in service.

The second set of lumped-mass models is specially developed to take account of the changes in the configuration of the system with time as the nut moves along the screw while taking into account the non-linear phenomena of backlash and Coulomb friction. These can deal with the axial, torsional and transverse degrees of freedom of the system and predict many aspects of the dynamic behaviour of a ballscrew system which have an effect on the errors arising from such systems. They also include features which calculate the energy converted to heat by all the energy dissipative mechanisms in the model which can be used in conjunction with models already developed at the University of Huddersfield to predict thermal errors.

Further, a strategy for compensation of some of these errors has been devised.

ACKNOWLEDGEMENTS

I would like to acknowledge my dept of gratitude for the support I have received during the course of this Research Project.

I am grateful for the funding provided by the ROBCON - EPSRC project '**Rob**ust On-line Parameter Identification and Modelling Applied to a Precision 3-axis Machine Tool for **Con**trol and Condition Monitoring Purposes', EPSRC Grant GR/S07827/01.

I would also like to thank the people who have acted as Director of Studies and Supervisors throughout the Project. First Professor Derek Ford, now Emeritus Professor at the University of Huddersfield, for his overall leadership and many helpful and useful comments and suggestions; Dr Mike Freeman for his help and encouragement with the mathematical analysis; Dr Simon Fletcher for his oversight of much of the practical work and his valuable input to the thermal aspects of the work; Dr Crinela Pislaru for her help and advice on various aspects of modelling, especially hybrid modelling, for her collaboration over the aspects relating to measurements taken on a CNC machine then in use at Huddersfield, and her help and advice in preparing this Thesis; Mr Steve Millwood for his help in the early days of programming in C; Dr Veimar Castaneda for his insights into the detailed working of the Heidenhain controller used on the Linear Guide Rig, and the technical support staff for their support in making parts specially required for the experimental work.

Finally, I would like to thank my wife Nora and my children Chloe and Christian for their support and patience over the years.

Geoff Holroyd
Lane Ings,
Marsden

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Chapter 3

c, d	semi-axes of elliptical area of contact zone	m
d_b	diameter of balls	m
d_i	inner race diameter of bearing, equation (3.1)	m
d_i	inner effective diameter (§3.5.2)	m
d_o	outer race diameter of bearing, equation (3.1)	m
d_o	outer effective diameter of screw (§3.5.2)	m
d_p	pitch diameter of ball action	m
f_{bp}	ball passing frequency	Hz
$f_{s,n}$	conformity ratio - screw, nut	-
f_0	index for bearing friction	N $sec^{2/3}/mm^{10/3}$
f_1	index for bearing friction	-
fl_b	nut body axial flexibility	m/N
fl_c	contact flexibility	m/N
$fl_{cb\,x}$	single ball contact flexibility – axial direction	m/N
fl_s	axial flexibility component of nut flange	m/N
fl_w	axial flexibility component of nut wall and screw (§3.5.2)	m/N
k_c	contact stiffness	N/m
$k_{c\,x}$	axial stiffness of balls in contact	N/m
l_n	axial length of nut	m
n	screw speed	rpm
p	screw pitch	m
r	effective radius of screw (§3.4)	m
x	axial movement of nut	m
y	movement between “remote” points in bodies 1 and 2 (§3.5)	m
z_1	number of balls in contact	-
C_E	material factor for contact stress	m^2/N
D_{bp}	ball pitch circle diameter of ballscrew	m
D_i	inner effective diameter of nut (§3.5.2)	m
D_o	outer effective diameter of nut (§3.5.2)	m
D_{PCD}	pitch circle diameter of nut mounting flange holes	m
E	Young’s modulus	N/m^2

E_i	Young's modulus	N/m²
F	axial force delivered by screw to nut	N
F_a	sum of contact forces resolved in axial direction (§3.4)	N
F_a	axial load (§3.5.3)	N
F_n	sum of contact forces of normal to screw helix	N
F_r	radial load (§3.5.3)	N
F_t	sum of contact forces acting tangential to axis of screw	N
G_n	shear modulus of the nut material	N/m²
K_D	geometry factor for contact stress	m
M_f	total friction moment	N m
M_{f0}	load independent friction moment	N m
M_{f1}	load dependent component	N m
N_b	number of balls	-
N_s.	number of helices = number of starts on screw	-
N_t	number of active turns in nut	-
P	contact force (§3.5)	N
P_0, P_1	load factors (§3.5.3)	N
R	screw ratio	m/rad
R_n	nut ball track section radii (§3.5)	m
R_s	screw ball track section radii (§3.5)	m
R_1, R_1'	radii of curvature of body 1 at point of Hertzian contact	m
R_2, R_2'	radii of curvature of body 2 at point of Hertzian contact	m
X_0	radial factor (§3.5.3)	-
Y_0	thrust factor (§3.5.3)	-
α_c	contact angle	rad
α, β, γ	factors in determining Hertzian stress and deflection (Table 3.1)	-
λ	ballscrew screw helix angle or lead angle	rad
θ	angle of rotation of screw relative to nut (§3.4)	rad
θ	angle used in Hertzian theory (§3.5)	rad
μ	coefficient of friction between screw and nut	N
ν	operating kinematic viscosity (§3.5.3)	mm²/sec = cS
ν_i	Poisson's ratio	-

ξ, η	local axes (§3.5)	m
σ_c	contact pressure	N/m²
$\sigma_{c\,max}$	maximum contact pressure	N/m²
ϕ	angle between plane containing $1/R_1$ in body 1 and that containing $1/R_2$ in body 2	rad
ω_s	angular speed of the shaft	rad/sec
Γ	torque applied to screw	N m

Chapter 4

c	wave velocity (§4.1)	m/sec
c	damping coefficient (§4.3)	N sec/m, N m sec/rad
c_{brg_bs}	torsional damping coefficient of ballscrew bearings	N m sec/rad
c_{brg_ms}	torsional damping coefficient of motor bearings	N m sec/rad
$c_{d\,ij}$	damping coefficient of damper connecting m_i to jth node	N sec/m, N sec, N m sec/rad
c_v	damping coefficient of viscous damper	N sec/m
$f(x,t)$	forcing term	N, N m
$f_i(t)$	forcing term (§4.2)	N, N m
$\mathbf{f}(t)$	forcing term vector	N, N m
iL	ballscrew node at which nut is connected in "lumped mass" model	-
k	stiffness (§4.3)	N/m, N m/rad
k_{ax}	axial stiffness of ballscrew	N/m
k_{bl}	torsional stiffness of drive belt	N m/rad
k_{ms}	torsional stiffness of motor shaft	N m/rad
k_{nut}	axial stiffness of ballscrew nut	N/m
$k_{s\,ij}$	stiffness of spring connecting m_i to jth node	N/m, N, N m/rad
l	length of shaft (§4.1)	m
l	element length (§4.3)	m
m	mass	kg
m_{bm}	mass of ballscrew	kg
m_i	mass of ith element	kg
m_{tab}	mass of the table and saddle	kg
t	time	sec

u	motor drive ratio	-
u	axial motion of (§4.1)	m
u_b	amplitude of backward travelling wave	m
u_f	amplitude of forward travelling wave	m
v	volume	m³
x	axial coordinate	m
x_i	displacement of ith node (§4.2)	m, rad
$\mathbf{x}$	displacement vector	m, rad
$\mathbf{x}^{\prime}$	modified displacement vector	√kg m
$\mathbf{x_1}$	part of solution vector (§4.3.2)	m/sec, rad/sec
$\dot{x}_i$	velocity of ith node (§4.2)	m/sec, rad/sec
y	transverse displacement	m
A	cross sectional area of shaft	m²
$\mathbf{B_1}$	submatrix (§4.3.2)	1/(kg sec), 1/(kg m sec), 1/(kg m² sec)
$\mathbf{B_2}$	submatrix (§4.3.2)	1/(kg sec²), 1/(kg m sec²), 1/(kg m² sec²)
C_{ij}	elements of damping matrix $\mathbf{C}$	N sec/m, N sec, N m sec/rad
$\mathbf{C}$	damping matrix	N sec/m, N sec, N m sec/rad
E	Young's modulus	N/m²
E_t	sum of energy in system and dissipated energy	J
F	force generated by spring / damper (§4.3)	N, N m
F_e	external force	N
G	shear modulus	N/m²
I	second moment of area of beam	m^4
$\mathbf{I}$	is the unity matrix	-
J	tilt inertia (§4.3)	kg m²
J_{bs}	rotational inertia of ballscrew	kg m²
J_m	rotational inertia of drive motor	kg m²
J_{p1}	rotational inertia of driving (motor) pulley	kg m²
J_{p2}	rotational inertia of driven pulley	kg m²

K_{ij}	elements of stiffness matrix **K**	N/m, N, N m/rad
K	stiffness matrix	N/m, N, N m/rad
$\mathbf{K}_{mod}$	modified stiffness matrix	1/(kg sec²), 1/(kg m sec²), 1/(kg m² sec²)
M	inertia matrix	kg, kg m²
$\mathbf{M}^e$	consistent mass matrix	kg, kg m, kg m²
M_{ii}	elements of inertia matrix	kg, kg m²
N	shape matrix	-
N	number of ballscrew elements in "lumped mass" model	-
N_n	number of nodes in system	-
R	ballscrew ratio	m/rad
T	axial tension	N
θ	tilt angle (§4.3)	rad
ξ_b, ξ_f	dummy variables	m
ρ	density	kg/m³
ω	natural frequency	rad/sec
Δt	time interval	sec

Chapter 5

$\mathbf{a}_k$	coefficients of time in displacement time series	m/sec^k, rad/sec^k
$\mathbf{a}_0$	initial displacement vector	m, rad
$\mathbf{a}_1$	initial velocity vector	m/sec, rad/sec
a_1, a_2	axial stiffness of ballscrew bearings	N/m
$\mathbf{b}_k$	coefficients of displacement time series	m, rad
b_l	backlash	m
c	damping coefficient of damper, equations (5.1) and (5.3)	N sec/m
c	coefficient of damping of single element of ballscrew	N sec/m
$\mathbf{c}_k$	coefficients of velocity time series	m/sec, rad/sec
c_n	damping between nut and screw	N sec/m
c_Θ	coefficient of torsional damping	N m sec/rad
d	displacement vector	m, rad
$\mathbf{d}_0$	initial displacement vector	m, rad
{**d**, **v**}	state of system	m, rad, m/sec, rad/sec
{$\mathbf{d}_0$, $\mathbf{v}_0$}	initial state of system	m, rad, m/sec, rad/sec

f	fraction of distance of contact point along element length	-
f_1, f_2	stiffness / damping distribution factors	-
i	node identifier	-
i, j, k	array/matrix element subscripts	-
j	subscript designating massless node	-
^{k}	index in time series	-
k	axial stiffness of a single element of ballscrew	N/m
k	stiffness of spring, equations (5.1) - (5.4) and (5.6)	N/m
k_n	axial stiffness of nut	N/m
l	length of screw	m
m	mass	kg
m_i	mass at node i	kg, kg m²
m_n	mass of nut	kg
p	pitch of screw	m
p	stiffness submatrix representing partition of matrix which includes nut, load and nodes of ballscrew at either side of connecting point	N/m, N, N m/rad
$\mathbf{p}_{\mathbf{red}}$	reduced stiffness matrix	N/m, N, N m/rad
q, **q′**	submatrices which connect **p** and **r**	N/m, N, N m/rad
r	stiffness submatrix representing massless node	N/m, N, N m/rad
$\mathbf{r}^{-1}$	flexibility submatrix representing massless node	m/N, 1/N, rad/(N m)
s, **s′**	submatrices which connect **p** and **t**	N/m, N, N m/rad
t	time	sec
t	stiffness submatrix representing part of system where external forces normally act	N/m, N, N m/rad
v	velocity vector	m/sec, rad/sec
$\mathbf{v}_0$	initial velocity vector	m/sec, rad/sec
x	displacement of mass	m
x	displacement array	m, rad
$\mathbf{x}'$	transpose of vector **x**	m, rad
x_{bs}	axial position of part of screw in same plane as nut centre	m
x_i	displacement of node i	m, rad
x_j	displacement of massless node	m, rad
x_n	displacement of nut	m
x_p	axial displacement of point of contact on screw	m

x_{pn}	displacement of nut at point of contact	m
x_{st}	steady state displacement response	m
z	distance of ballscrew axis below the centre of mass of load	m
A	screw cross-sectional area	m²
B_1, B_2	nodes where journal bearings are attached	-
B_{lf1}, B_{lf2}	backlash factors	-
C	damping coefficient matrix of ballscrew system	N sec/m, N sec, N m sec/rad
D'	modified determinant	-
E	Young's modulus	N/m²
E_d	energy dissipated	J
E_{in}	energy input	J
F	forcing term, equation (5.1)	N
F	external force	N
F	force array	N, N m
F_i	forces	N, N m
$F_i(t)$	external force applied at ith node	N
$F_{n\,i}$	contact force acting on sliding face	N
F_0	amplitude of harmonic forcing term	N
G	shear modulus of screw material	N/m²
J	inertia of load about Y axis through centre of mass	kg m²
J_i	rotational inertia of ith node	kg m²
K	torsional constant of screw	m^4
K	stiffness matrix of ballscrew system	N/m, N, N m/rad
$\mathbf{K_{bs}}$	submatrix representing the ballscrew	N/m, N, N m/rad
$\mathbf{K_{bs\,n}}$	submatrix representing the cross terms between the screw and the nut	N/m, N, N m/rad
$\mathbf{K_{dr}}$	submatrix representing the controller, drive motor and mechanical coupling device	N/m, N, N m/rad
$\mathbf{K_{dr\,bs}}$	submatrix representing the cross terms between the driving mechanism and the screw	N/m, N, N m/rad
K_{ij}	term of stiffness matrix	N/m, N, N m/rad
$\mathbf{K_n}$	submatrix representing the nut and load	N/m, N, N m/rad
KE	kinetic energy	J

M	mass of saddle or table	kg
$\mathbf{M}$	mass matrix of ballscrew system	kg, kg m²
N_f	number of degrees of freedom at each node of ballscrew	-
N_s	number of starts	-
PE	potential energy	J
R	ballscrew ratio	m/rad
S	number of spring elements in ballscrew	-
T	time interval or time step	sec
X_i	x coordinates of nodes	m
X_n	position of ballscrew nut	m
$\boldsymbol{\alpha}$, $\boldsymbol{\beta}$	special matrices used to develop time series	-
$\boldsymbol{\delta}$	displacements of degrees of freedom of submatrix **p**	m, rad
$\boldsymbol{\varepsilon}$	displacements of massless node	m, rad
ζ	fraction of critical damping	-
θ	phase lag	rad
θ_{bs}	angular position of part of screw in same plane as nut centre	rad
θ_p	rotation of the screw at point of contact of nut on screw	rad
μ_i	coefficient of friction	-
ξ_i	stiffness/damping distribution factors	-
$\boldsymbol{\varphi}$	displacements of degrees of freedom where external forces act	m, rad
ϕ_n	tilt displacement of saddle/table	rad
ω	phase velocity	rad/sec
ω_n	undamped natural frequency	rad/sec
$\Gamma(t)$	system applied torque	N m
$\Gamma_i(t)$	torque on ith node	N m
Θ	torsional stiffness of a single element of ballscrew	N m/rad
Φ	torsional stiffness of slideways	N m/rad

Chapter 6

a_n	amplitude of acceleration vector	m/sec²
a_{circ}	"circular" component of acceleration	m/sec²
f	fraction of distance of contact point along element length	-

fl(i,j)	flexibility matrix	m/N, 1/N, rad/(N m)
"g"	acceleration due to gravity	m/sec²
g	gravitational field vector	m/sec²
i	node number	-
ib	bearing number = 1 or 2	-
$j_{str\,y}, j_{str\,z}$	effective tilt inertia of support structure close to bearing	kg m²
$k_{ax\,ib}$	axial or thrust stiffness of ib^{th} bearing	N/m
k_n	radial stiffness of nut	N/m
$k_{slide\,y}$	support stiffness of saddle/table slideways in Y direction	N/m
$k_{slide\,z}$	support stiffness of saddle/table slideways in Z direction	N/m
$k_{slide\,\theta y}$	tilt stiffness of saddle/table slideways in Y direction	N/m
$k_{slide\,\theta z}$	tilt stiffness of saddle/table slideways in Z direction	N/m
$k_{tilt\,\theta y\,ib}$	tilt or rocking stiffness of ib^{th} bearing about Y axis	N m/rad
$k_{tilt\,\theta z\,ib}$	tilt or rocking stiffness of ib^{th} bearing about Z axis	N m/rad
$k_{tr\,y\,ib}$	transverse or radial stiffness of ib^{th} bearing in Y direction	N/m
$k_{tr\,z\,ib}$	transverse or radial stiffness of ib^{th} bearing in Z direction	N/m
l	beam or element length	m
m	mass	kg
m_i	mass at node i	kg
$m_{str\,x}, m_{str\,y}, m_{str\,z}$	effective mass of support structure close to bearing	kg
$\mathbf{p_{red}}$	reduced stiffness matrix	N/m, N, N m/rad
r_i	radial distance of centre of mass of i^{th} node from X axis	m
r_{circ}	radius of node orbit	m
t	time	sec
v_n	amplitude of velocity vector	m/sec
$\mathbf{x}$	global displacement vector	m, rad
$\mathbf{x_r}$	displacements involving Y and Θ_z degrees of freedom	m, rad
y	transverse displacement	m
y_i, z_i	components of displacement r_i in local X and Y directions	m
$\{z_i\}_i$	deflected shape at i^{th} of iteration of pre-tension transverse deflection calculation	m
A	cross-sectional area	m²

B_i	out-of-balance at node *i*	kg m
D	modified determinant	-
E	Young's modulus	N/m²
$\mathbf{F}$	force vector	N, N m
$F_{a\,i}$	external axial force	N
$\mathbf{F}_{gyr}$	gyroscopic torque vector	N m
$\mathbf{F}_i$	force generated by out-of-balance mass, equation (6.38)	N
G	shear modulus	N/m²
I	second moment of area	m^4
J_{tilt}	tilt inertia	kg m²
J_{tors}	torsional moment of inertia	kg m²
J_x	polar moment inertia	kg m²
J_{xx}	moment of inertia about X axis	kg m²
J_0	moment of inertia about X direction through centre of mass	kg m²
K	torsional constant	m^4
$\mathbf{K}$	stiffness matrix	N/m, N, N m/rad
$\mathbf{K}_{+ib}$	modified bearing stiffness matrix	N/m, N m/rad
$\mathbf{K}_{+slide}$	modified stiffness matrix, slideways degrees of freedom	N/m, N m/rad
$\mathbf{K_{ang}}$	modified bearing stiffness submatrix, angular degrees of freedom	N m/rad
$\mathbf{K_{lin}}$	modified bearing stiffness submatrix, linear degrees of freedom	N/m
$\mathbf{K_r}$	stiffness submatrix involving Y and Θ_z degrees of freedom	N/m, N, N m/rad
$\mathbf{K}_{str\,i}$	stiffness matrix of support structure close to bearing	N/m, N m/rad
$\mathbf{K_z}$	stiffness submatrix involving the Y, Z, Θ_y and Θ_z degrees of freedom	N/m, N, N m/rad
$\mathbf{K_1}$, $\mathbf{K_2}$ and $\mathbf{K_3}$	– stiffness submatrices, Figure 6.3	N/m, N, N m/rad
M_i	angular momentum of node *i*	kg m²/sec
$\mathbf{M}$	mass matrix	kg, kg m²
$\mathbf{M_r}$	mass submatrix involving Y and Θ_z degrees of freedom	kg, kg m²
$\mathbf{M}_{str\,i}$	effective mass matrix of support structure close to bearing	kg, kg m²
$M_{y\,i}$	titling moment of i^{th} node of ballscrew	N m
$\mathbf{M_z}$	mass submatrix involving the Y, Z, Θ_y and Θ_z degrees of freedom	kg, kg m²
$S_{dz\,2}$	measure of the difference between two deflection shapes in iteration	m²

S_{z2}	measure of the "size" of a deflection shape	m²
T	pre-tension	N
T_+	tension in element "upstream" of node	N
T_-	tension in element "downstream" of node	N
δM	small change of angular momentum vector	kg m²/sec
δt	time interval	sec
θ	tilt angle	rad
$\theta_{x\,i}$	torsional position	rad
$\theta_{x\,i}$, $\theta_{y\,i}$, $\theta_{z\,i}$	angular position of i^{th} node	rad
$\theta_{x\,ib}$	angular position of ballscrew in bearing	rad
ϕ_a	direction of acceleration vector	rad
ϕ_n	tilt stiffness of nut	N/m
ϕ_v	direction of velocity vector	rad
ω	angular speed of screw	rad/sec
Γ	torque	N m
$\Gamma_{e\,i}$	external torque	N m
Γ_y, Γ_z	tilt torques about Y and Z axes	N m

Chapter 7

d	diameter of tube	m
h	heat transfer coefficient	W/(m² ºK)
h_r	heat transfer coefficient	W/(m^2 K)
$itime$	row number of time history array	-
jx	node number	-
jQ	node at which heat is applied	-
k	thermal conductivity	W/(m ºK)
l	length of coolant hole	m
m_{jx}	node mass	kg
q	nett heat flux from surface (eq 7.1)	W/m²
t	time	sec
t_h	time of initial heating	sec
t_{stop}	time at end of cooling phase	sec
tc_{jx}	node thermal capacity	W/ºK
u_∞	convective flow rate of bulk of fluid	m/sec

v_{per}	peripheral speed	m/sec
v_w	average speed of cooling water through hole	m/sec
x	axial coordinate	m
A_c	cross-sectional area of conductivity elements	m²
As_i	cylindrical surface area of ith element	m²
Asi_{jx},	surface area of inner cylindrical surface of node	m²
Aso_{jx}	surface area of outer cylindrical surface of node	m²
C_p	specific heat at constant pressure	J/(kg ºK)
H	heat production rate density (equ 7.6 and eq 7.7)	W/kg
H	heat transfer rate for the whole ballscrew (eq 7.23)	W/ºK
L	characteristic length of surface (eq 7.5)	m
L	length of bar	m
Nu	Nusselt number	-
Q	rate of heat supply at node jQ	W
Q_c	heat conducted along element	J
N	number of nodes or “thermal masses”	-
Re	Reynolds number	-
T	absolute temperature of surface	ºK
T_a	representative temperature of surroundings (radiation)	ºK
T_a	ambient temperature	ºK
T_b	temperature of ballscrew	ºK
T_w	cooling water temperature	ºK
T_{wi}	cooling water inlet temperature	ºK
Tav_{jx}	average temperature distribution	ºK
$Th_{itime\ ix}$	temperature time history array	ºK
α	absorptivity of surface	-
δq	heat gain or loss from cooling water	J
δt	time interval (smaller)	sec
δx	element length	m
δT	temperature difference	ºK
ε	emissivity of surface	-
κ	thermal diffusivity	m²/sec
ν	kinematic viscosity	m²/sec

ρ	material density	kg/m³
ρ_w	density of water	kg/m³
σ	Stefann-Boltzmann constant = 5.672×10^{-8}	W/(m^2 K^4)
Δt	time interval (larger)	sec

Chapter8

d_i	inside diameter of ballscrew	m
d_o	outside diameter of ballscrew	m
$d_{o\,eq}$	equivalent outside diameter of ballscrew	mm
g	acceleration due to gravity	m/sec²
i, j	element or member identifiers	-
k	roots of characteristic equation of equation (8.A3.4)	1/m
l	length of the ballscrew between bearing pack centres	m
n	a non-zero integer	-
p	pitch of ballscrew groove	m
r_g	radius of ballscrew groove	mm
s	screw section number	-
w	weight per unit length of beam	N/m
x_j	axial position on ballscrew	m
y	transverse deflections of ballscrew centre	m
$y(x)$	deflected shape of ballscrew	m
y_g	distance from outer cylinder of screw to groove's centre of area	mm
y_m	measured values of transverse movement	m
y_{max}	maximum transverse deflection	m
A	amplitude of solution of equation (8.A3.4)	m
A	dimension in Figure 8.3	mm
A_g	cross sectional area of ballscrew groove	mm²
A_i	constants of integration	(*units dependent on equation*)
$\mathbf{A_j}$	subvectors of terms used to define deflected shape of ballscrew	(*various*)
B	dimension in Figure 8.3	mm
B_i	constants of integration	(*units dependent on equation*)
E	modulus of elasticity	N/m²
EI	flexural rigidity of beam	N m²

$\mathbf{H_{ij}}$	submatrices of quantities determined by geometry of ballscrew	(*various*)
I	second moment of area	m^4
K_i	radial stiffness of bearing	N/m
L	length of beam along X axis	m
M	bending moment	N m
M_i	applied moment at intermediate points on ballscrew	N m
M_L	applied moment at non-drive end of ballscrew	N m
M_0	applied moment at drive end of ballscrew	N m
N_m	number of groups of measurements in a set of readings	-
Q	shear force	N
Q_i	shear load at intermediate points on ballscrew	N
Q_L	shear load at non-drive end of ballscrew	N
Q_0	shear load at drive end of ballscrew	N
$\mathbf{V_i}$	subvectors of the forces vector	N, N m
W	total weight of ballscrew	N
T	pre-tension in ballscrew	N
V_p	volume of material in a length of screw one pitch long	mm^3
X_i	axial coordinates	m
α	roots of characteristic equation equation (8.A3.4)	1/m
δx	length of element of beam	m
ε	radial errors	m
θ	angle used in derivation of equivalent outside diameter of screw	rad
λ	wave number, equations (8.A2.7) to (8.A2.52)	1/m
λ	non-dimensional factor used in Levenburg-Marquardt method	-
μ	mass per unit length or linear density of beam	kg/m
ξ	roots of characteristic equation (8.A2.5)	1/m
ρ	density	kg/m^3
χ^2	"chi squared" measure of error	m^2
ω	rotational speed of ballscrew	rad/sec
Φ_i	tilt stiffness of bearing	N m/rad

Chapter 9

d_i	inner diameter of ballscrew	m
d_o	outer diameter of ballscrew	m

fl_{bs}	axial flexibility of the ballscrew	m/N
fl_{sys}	total system flexibility referred to the motor shaft between the motor and the ballscrew nut	rad/(N m)
l_{bs}	length of ballscrew between its supporting bearings	m
$k_{ax\,B1}$, $k_{ax\,B2}$	axial stiffness of the ballscrew support bearings	N/m
k_{dr}	stiffness of drive coupling	N m/rad
k_{ms}	stiffness of motor shaft	N m/rad
$k\theta_{bs}$	torsional stiffness of ballscrew	N m/rad
m_{load}	mass of the nut and saddle and/or table	kg
p	screw pitch	m
x	linear position of nut	m
A_{bs}	cross sectional area of ballscrew	m²
C_E	material factor for contact stress	m²/N
E	modulus of elasticity	N/m²
E_i	Young's modulus	N/m²
F_{nut}	axial force delivered by the ballscrew to the nut	N
G	shear modulus	N/m²
J_m	motor inertia	kg m²
J_{tot}	total referred inertia of the ballscrew drive system	kg m²
N_s	number of helices = number of starts on screw	-
R	screw ratio	m/rad
T_m	torque applied to the motor rotor by its electrical fields	N m
$T_{m\,net}$	motor output torque	N m
δx_{bs}	axial deflection at the centre of the nut	m
δx_{comp}	total deflection to be compensated for	m
θ	angular position of ballscrew	rad
θ_{bs}	torsional deflection of ballscrew at centre of nut	rad
ν_i	Poisson's ratio	-

Chapter 1 - INTRODUCTION

In the modern global economy, there is a demand for high precision in manufacture as competitive pressures drive businesses to seek greater productivity. The resulting high-volume production has meant that machines have replaced manual labour in many instances. Following the development of reliable digital electronic technology it became possible to produce computer numerically controlled (CNC) machine tools. At the present time a great variety of products or product parts are made on such machines. The annual world wide demand for machine tools is of the order of £500 million [1]. Japan is one of the world's leading suppliers and some 84% of her total production comprise numerically controlled machines [2].

Most manufactured products are an assembly of many parts. It is highly desirable that these parts are made to the highest standards of accuracy [3]. Workpiece accuracy is influenced by the environment in which a machine tool operates, the way it is used and errors associated with the machine itself. In order to achieve the desired levels of accuracy the errors associated with machine tools must be reduced to the lowest practicable level. Errors in machine tools can be grouped under three main categories [4] – geometric, thermal [5] and load [6-9].

In the operation of a machine tool, one or more **machine tool drives** are needed to move the tool relative to the workpiece. If the required motion is linear, the motion is generated by a screw mechanism in which the rotational motion of a screw is transferred to linear motion of the nut. For machines with above 3 m travel, **rack and pinion** drives are often employed. **Linear motors** are sometimes employed for smaller machines, but screws remain the most widely used mechanisms. In most modern machine tools, the control of the motion of its various parts is by means of motor-driven mechanisms governed by an electronic **controller**.

The controller receives signals from transducers placed at suitable locations on the machine which enable the position and speed of the different parts to be determined. It uses this information, together with a set of instructions known as a **part program**, to determine how the machine should operate to produce the desired shape in the workpiece. A typical machine tool drive therefore consists of an electronic controller, an electric motor, a drive mechanism and transducers or encoders which provide feedback signals to the controller [10]. Modern control systems use digital technology and so errors in the drive system can arise due to the sampling and quantisation involved in converting analogue signals into digital form. Errors such as interpolation and servo up-date errors can also be generated while converting the digital output to the analogue signal used to drive the motor.

Traditionally the screw mechanisms consisted of a steel helical screw of a "square" profile with a matching nut lubricated by oil or grease. Such mechanisms involved a great deal of sliding **friction**. In order to reduce problems associated with friction, the **ballscrew** was developed where a set of rolling elements are introduced between the screw and the nut. The basic elements of a ballscrew are

- a **screw** with one or more helical grooves on its outer surface,
- a sleeve known as a **nut** which has corresponding grooves on its inner surface, and
- a set of **balls** which provide rolling contact between the screw and the nut.

The main considerations taken into account when designing a ballscrew are that

- it should be **strong** enough to carry the **loads** imposed
- it should be **durable** enough in terms of **fatigue life** and **wear characteristics** of the materials involved to last for the **design life**
- it should be **rigid** enough to meet the required precision
- it should be dynamically **stable** throughout its operating envelope. In practice this means that the system should not operate at or near sensitive natural frequencies for long enough for troublesome vibrations to occur.

In order that the nut runs freely on the screw, there must be a small amount of clearance between the balls and the grooves they run in. This gives rise to **backlash** between the nut and the screw. In most instances the screw, nut and balls are made of an appropriate grade of steel and can be made to a high level of accuracy, especially where ground screws are employed. Cooling of the parts of the ballscrew can be achieved by passing a water based coolant through hollow passage ways in the parts where the cooling is desired. Ballscrew drives are used in a wide variety of machines with drive travels of up to about three metres.

Ballscrews are the first choice for many machine tool drives. The error associated with ballscrews which has a direct effect on the machine performance is the pitch error associated with the screw grooves. Others include thermal growth of the screw and elastic deformation due to the dynamic loads imposed on the screw. An understanding of these errors is important to improving the accuracy of machine tool drives.

Two strategies can be deployed to improve the accuracy of a machine, viz:

- error avoidance – this approach minimises errors through better machine design, construction and control of the machine's environment [**11**].

- error compensation – this approach seeks to compensate for errors which cannot be eliminated. The assumption is made that many errors are repeatable and predictable. A model of the machine's behaviour can therefore be made which describes the machine tool and which can be validated experimentally [**12**]. The information received from the modelled simulations can be used along with data from the machine's transducers by the machine's controller to improve the accuracy of the manufactured parts.

Error avoidance in ballscrews is normally achieved by making the components from suitably tough grades of steel and maximising dimensional accuracy by using grinding to finish the surfaces most relevant to positional accuracy. Backlash can be removed by using one of the several ways of applying **preload**, (see §3.1, p 22).

Commercially it is desirable to be able to achieve the desired accuracy by compensating in the most cost-effective way. In this context it is best, where possible, to avoid using expensive special environments such as temperature controlled rooms. Also it is better to use pre-calibrated measurements for known errors in machine tools rather than rely on additional measuring equipment which may prove vulnerable to coolant, swarf and other hostile elements in the machine tool environment. Well formulated mathematical models can take a useful role in understanding the effects which generate errors so that improvements in the manufacturing process can be made. Advances have been made recently in the modelling of CNC machine tool feed drives [**13**, **14**], particularly hybrid modelling [**15**], though these do not deal with details of the behaviour which might arise from movement of the nut. These have been used as a starting point for the development of a new approach to ballscrew modelling.

The programme of research described in this investigation will seek to develop means of understanding the static, dynamic and thermal behaviour of a ballscrew system and to incorporate that understanding in a set of mathematical models. The prime purpose of a ballscrew is to move one part of a machine relative to another. This movement causes some of the characteristics of the machine, for example, the effective stiffness of the ballscrew to change. Therefore, a case of special concern in ballscrew systems is that of the dynamics of a system where the characteristics change with time. This can be been addressed under two broad headings – first, axial and torsional, and second transverse.

Any torsional or axial deflection of the screw, or axial movement within the support bearings, or axial movement in the nut has a direct impact on errors generated by the ballscrew system. The modelling method is based on the discrete matter approach and

involves setting up a set of matrices to represent the mass, damping and stiffness of the components of the ballscrew system. The changes in the characteristics of the system are modelled by updating these matrices as the solution proceeds. The solution is obtained by a series method which avoids the repeated inversion of large matrices. Comparisons are made with "classical" solutions based on elastic beam theory.

The transverse case has to do mainly with the vibration of the screw. Since the bearings and nut give radial restraints to the screw, it is to be expected that the natural frequencies of a ballscrew system will change with the position of the nut along the screw. Whether or not any particular natural frequency is of detriment to the performance of the machine of which the ballscrew is part will depend on the severity of the vibration which occurs. This depends in part on the damping involved, the magnitude of the excitation forces and the time that such excitation is effective at frequencies close to that natural frequency. All of these points can be addressed using the method developed in this programme of research. Problems with the method have been identified and overcome. Non-linear phenomena such as Coulomb friction and backlash can be accommodated.

The dynamic model has built into it a means of determining the heat generated by the dissipative mechanisms active in a ballscrew system. The thermal model takes account of the heat generating processes, the means of distributing the heat throughout the ballscrew system and the means of losing heat to its surroundings. It can therefore generate predictions of the temperature distribution throughout the ballscrew system and the resulting thermally induced movements. Consideration is also given to the various means of cooling available for different parts of the ballscrew system.

Use has been made of a special rig developed during the investigation to undertake various tests of these models. Consideration has also been given to the aspects of ballscrew design which influence error generation and what steps may be taken to minimise errors. Finally a means of applying compensations for some of the effects identified is put forward. Taken together, the models contribute to knowledge by enabling a deeper understanding to be gained into

- the interaction between the various details of ballscrew design,
- the dynamic behaviour of a ballscrew drive and
- the thermal behaviour of the systems into which the ballscrew is to be built.

The **aims** of this research therefore are:-

- to prepare a mathematical model of a ballscrew which takes into account the geometric, thermal and load errors commonly encountered on CNC machine tools and
- to devise strategies and algorithms for the correction of these errors.

The **objectives** of the research can be listed as follows:-

- Starting from a basic understanding of the behaviour of a ballscrew, to outline the relevant static elastic theory, dynamic elastic theory and the theory of thermal modelling.
- To develop further the dynamic theory by examining the continuous matter approach including wave solutions and the discrete matter (or "lumped mass") approach with its matrix solutions.
- To extend the classical eigenvalue method for finding the natural frequencies and other dynamic characteristics of ballscrew systems and to include viscous damping effects using a generalised eigenvalue approach. This will give estimates of not only the natural frequencies, but also the damping coefficient of each predicted vibration mode [**16**]. The sensitivity of the system to changes in stiffness and damping characteristics will also be derived.
- To devise a method of modelling a ballscrew driven mechanical system with a moving nut whose configuration changes with time.
- To develop a means of compiling a mathematical model which includes non-linear phenomena such as backlash and Coulomb friction as well as time dependent stiffness and damping characteristics.
- To apply the new approach to the axial and torsional behaviour of the screw with a view to predicting the position error of the ballscrew nut.
- To apply the new approach to the transverse behaviour with a view to predicting the vibration characteristics of the screw.
- To develop the mechanical model in such a way that the energy dissipative mechanisms give a prediction of heat inputs for a model of the thermal behaviour.
- To develop a model to predict the thermal behaviour of a ballscrew system
- To consider the feasibility of cooling parts of the ballscrew system with a view to reducing thermal errors.
- To compare the simulated results of the models with data measured from the test rig and other relevant experimental data that is in the public domain or within the University internal reports.

- To consider ways of reducing errors by changes in the design of a ballscrew.
- To devise a means of correcting ballscrew errors which will improve the accuracy of a machine tool.

The investigation has added a novel method of analysing machine tool drives which takes into account the characteristics of the ballscrew system which change as the nut moves along the screw. This can be especially helpful in analysing a system's susceptibility to high levels of vibration due to the excitation of its natural frequencies. It also includes a useful extension to eigenvalue analysis applied to machine tool drives.

The remainder of the thesis is divided into the following chapters:-

- **Chapter 2** is a critical review of the literature in the public domain. The results of the review lead to the topics covered by this investigation.
- In **Chapter 3** the theoretical background is set out. This includes the basic behaviour of a ballscrew, errors in machine tool drives and static elastic theory as applied to ballscrew action. The applicable dynamic elastic theory and the approach to thermal modelling are also introduced.
- The approach to dynamic modelling is developed in **Chapter 4**. Continuous matter methods including wave solutions and lumped mass methods with their matrix solutions are discussed leading on to a generalised eigenvalue model of a machine tool drive. The chapter finishes by outlining a method of undertaking an analysis of the sensitivity of such a drive to changes in its system parameters.
- A method for analysing the dynamic behaviour of a system whose dynamic characteristics change with time is developed in **Chapter 5**. The case of the axial and torsional degrees of freedom of a machine tool drive including a ballscrew is solved and comparisons with classical elastic theory are made.
- In **Chapter 6** the solution is extended to include the transverse degrees of freedom of the ballscrew. Consideration is made of the static case of gravitational sag and the effects of pre-tension in the screw. A comparison with a conventional consideration of a dynamic case is made.
- Thermal modelling is discussed in **Chapter 7**. The thermal characteristics of various components of a machine tool drive are described and the one dimensional approach to modelling the screw is justified. Various aspects of cooling are also covered.
- **Chapter 8** covers the experimental verification of various aspects of the models.

- In **Chapter 9** error reduction is discussed including reduction through design and reduction through compensation.
- The conclusions to the investigation are drawn in **Chapter 10** and recommendations for further research are made.

The first part of the investigation is the review of literature in public domain and follows in Chapter 2.

Chapter 2 - LITERATURE SURVEY

Screw based mechanisms have been in use since early times. The name of Archimedes (287 – 212 or 211 BC) [17, 18] is associated with a helical device used for moving water. Similar devices are used extensively today for moving bulk solids [19]. The use of screws for transmitting force and motion also dates back to the middle ages. In the *Mittelalterliche Hausbuch* of about 1480 a lathe with a screw device to control the motion of the cutting tool was shown [20].

2.1 Basic modelling of screw mechanisms

The special properties of screws have given rise to a branch of mathematics called screw theory [21]. The theory of screws was largely developed by Sir Robert Stawell Ball [22] over 100 years ago to investigate general problems in rigid-body mechanics. This has been refined and extended to include the modern formulations of dual numbers, Plücker coordinates and Lie algebra. Lipkin and Duffy have presented an overview of these methodologies along with a perspective of the re-emergence of screw theory as an important tool in robot mechanics, mechanical design, computational geometry and multi-body dynamics.

In machine tools, the ACME thread screw has been used to move and position different parts of machines for many years. The ballscrew was introduced in a bid to reduce friction and backlash. Here a set of balls is constrained to roll between two helical tracks. Knowledge of the kinematic behaviour of bodies in contact is therefore useful in order to gain a thorough understanding of the dynamic behaviour of ball-contact mechanisms. Pfister [23] has developed a point contact kinematic model, using a non-holonomic parameterisation first introduced by Neumann [24] and Richter [25]. The model is built with two auxiliary pairs of imaginary bodies and each body carries a co-ordinate reference frame. By this, one single point pair becomes a serial composite of three pairs: two Levi-Civita and one revolute pair. The approach is general and applicable to any pair of regular, parametric surfaces. The algebra is simplified by the discovery of an intimate relationship between point-contact kinematics and parallel transport of vector fields in the sense of Levi-Civita. A new special motion, Levi-Civita motion, is defined and discussed. The following first- and second-order kinematic properties of Levi-Civita motion are presented: motion screw, motion pitch, Hamilton cylindroid, Sturm theorem, parallelism, Schieldrop-Johnsen vector of non-holonomic deviation, analogy to a rigid body with a fixed point and a kinematic proof of the Gauss-Bonnet theorem of differential geometry.

These methods can be applied to a large variety of phenomena and Pfister obtained convenient, profound and computationally attractive algebraic expressions which can be applied

to a variety of bodies in contact. However, the behaviour of the components of a ballscrew is only a sub-set of the phenomena covered. Balls are the simplest possible form of rolling element, having but one positive curvature, in theory at least. The raceways are most commonly of "gothic" profile. So their section consists of two circles which overlap in such a way that there is a cusp at the lowest point of the profile. This simplifies the pattern of contact to "two point" or "four point" contact. The detailed contact behaviour of the balls and raceways has an effect on the heat generated by the operation of a ballscrew, and on the wear of these parts. However, it is the axial deviations of the raceways, especially that of the screw (screw pitch errors), which contribute most to the geometric errors generated by ballscrew drives. Calculation methods based on a simpler set of assumptions are widely used by ballscrew manufacturers to estimate the static behaviour of their products [**26**].

2.2 **Error compensation in CNC machine tools**

The Ultra Precision Engineering Group at Huddersfield University has carried out extensive work on geometric and thermal errors of machine tools and their drives. They also have developed considerable expertise in modelling machine tool drives.

The geometric errors which occur on a three-axis machine tool can be expressed in terms of 21 components [**27**]. There are six errors for each of the three Cartesian axes: a linear positional error, a straightness error in the two directions perpendicular to the axis, an angular error about the axis itself ("roll") and an angular error about the two perpendicular axes ("pitch" and "yaw"). In addition there are three squareness errors between the axes. It is possible to measure the positional, straightness, pitch and yaw angular errors by means of optical methods which involve counting interference fringes generated by laser beams passing through an appropriate set of lenses and mirrors. The roll angular errors can be measured using devices such as the Talyvel electronic level. Squareness can be measured using laser techniques, but mechanical methods can also be used, the latter entails the use contact probes in conjunction with a special high precision artefact like a granite set square.

Postlethwaite et al [**28**] described a novel technique for the fast, accurate and detailed geometric calibration of CNC machine tools. This technique uses a standard Renishaw laser interferometer system to extract geometric error data from measurements taken dynamically. Tests carried out on a typical machine tool enable the accuracy and potential of this calibration technique to be assessed. Further work in this field [**29**] has led to the development of software which computes the machine volumetric accuracy from the individual geometric error components, and then calculates the effect of machine angular errors. The software is based on

the Windows operating system and incorporates a user-friendly graphical interface. This approach is considerably faster than standard techniques for measuring static errors. It also enables patterns of errors to be identified that normal methods would miss.

The analysis of errors arising from component variations in geometry and temperature of a non-Cartesian machine is far more complex than in the case of the Cartesian machine. Freeman et al [**30**] described techniques leading to the development of general software which is capable of analysing the geometric and thermal errors in any structure consisting of measurable struts. Three dimensional plots of errors in any plane can be produced. The limitation that the machine should be made of struts is not so restrictive as it might appear, since solid components can usually be modelled as interconnected struts if large temperature gradients do not cause thermal deformation of the machine's components. The approach is to divide a machine into sub-assemblies and to analyse each of these individually. This method enables the errors arising from a source of heat, (such as a spindle motor), to be evaluated, and also the effect of random errors in strut lengths. It is very useful when dealing with non-conventional machine formats.

Ford et al [**31**] have developed a unique algorithm, based on an indirect identification pre-calibrated technique, which allows the system to compensate for geometric error components of the normal orthogonal machine tool configuration. The method takes into account both rigid body effects and the behaviour of some machine specific non-rigid elements. It introduces a novel technique for reducing workpiece errors caused by the thermal distortion of a CNC machine tool. The universal rigid body compensation model which works for machines with up to three axes includes machine specific non-rigid body and thermal effects. It can achieve accuracy improvements of the order of 10:1 up to 30:1. A universal non-rigid body compensation was not possible. Models specific to each machine need to be developed and incorporated into the machine's compensation system. A similar approach is needed for thermal distortion.

White et al discussed [**32**] the relative accuracy of ballscrews and linear encoders over a broad range of application configurations and usage conditions. Frank and Ruech [**33**] made similar comparisons. It was shown that ballscrew expansion is the largest thermal error source within a broad range of CNC machine tool configurations and running conditions when a rotary encoder fitted at the end of the ballscrew is used to measure the position of the axis. Such an arrangement exhibits offset, scale and reversal errors that change quickly according to the thermal state of the ball-screw. It was found that linear encoder measurement systems exhibit the same error categories, but generally with smaller maximum values, and slower rates of change. The thermal errors exhibited by ball-screws are primarily due to an inability to de-

couple the measuring system from the heat in the ball-screw and mounting system that sets the position of the tool relative to the work-piece.

Linear expansion and distortion of the structural elements also cause unwanted movement between the tool and workpiece. Heat inputs that cause temperature elevation and gradients come from many sources, both internal and external to the machine tool. Internal sources include drive motors, ballscrews and the cutting action of the tool. External sources include sunlight and draughts from convective heating systems. This makes thermal errors difficult to control without some form of compensation. Many thermal error modelling and compensation systems have been proposed which use neural networks [**34**], multi-regression analysis [**35**], heat modelling [**36**] or probing techniques [**37**]. However, each method suffers from one or more major drawbacks that limits its effectiveness when used in a practical machining environment. One feature of all the thermal error compensation techniques is the lack of flexibility that makes them difficult to apply to more than one machine type in a timely and cost-effective way.

White et al [**38**] described a combined thermal and geometric error compensation system with a flexible structure that is general purpose in its application to any machine tool. The system can accept input from any number of temperature sensors. Using data from a single test on the machine, a novel Matlab model can be programmed in such a way that it estimates its thermal movement. The program then directs error values to a number of outputs which are used by the machine controller to effect compensation by axis position modification. The entire compensation system can be applied either in a stand-alone computer that accepts a wide range of feedback signal types, or integrated into an open architecture machine controller. The system allows the management of temporary or permanent input failures and displays every thermal error component as an aid to fault diagnosis. Both position independent and position dependent thermal errors can be reduced through compensation. The system has been applied to several machine tools, and has shown itself capable of reducing thermal movements between the tool and the workpiece by over 6 times when using a quick heating and cooling test for calibration.

The connectivity of the structural elements determines the effect of the expansion and distortion on the relative positions of the tool and workpiece, leading to thermal errors on the workpiece. Measurement on a wide range of machine tools has confirmed that temperature gradients are significant in their effect on machining accuracy and that they move and change shape during the machining process. White et al [**39**] demonstrated a bending model that estimates the effects of thermal distortion using knowledge of the position of the temperature gradient which is derived. The performance of the bending model was compared with a finite element model and a model that has no knowledge of the position of temperature gradients.

Results obtained from a vertical machining centre show that knowledge of the position and magnitude of temperature gradients is an essential part of predicting thermal distortion accurately.

Freeman et al [**40**] developed a system which minimises the number of temperature sensors used as inputs to a thermal simulation model of the ball-screw which is used to estimate on-line the thermal errors of the ball-screw. Assuming that position measurements of the nut are available from the rotary encoder, it is possible to calculate the speed of the screw. Assuming a knowledge of the frictional and heat transfer characteristics, and using the speed and the measured temperatures of the nut and bearings, the heat generated in the nut, bearings and screw can be estimated. The thermal model constitutes a one-dimensional finite analysis of the whole length of the screw. The output of the model is a temperature distribution along the screw and an estimate of the thermal errors along the screw.

They also described an experimental test rig, which provided facilities to test the model on-line. The position measured with a laser interferometer was compared with the estimated position derived from the model. The results were displayed graphically and saved for future use in optimisation software to determine the parameters of the thermal model which best fit the experimental data. Improvements of better than 90% in the thermal error have been obtained. The software is capable of making estimates of the thermal behaviour of a user-specified ball-screw in an off-line simulation mode. This can be used to aid design and help understand the effect of thermal errors on machine accuracy.

Postlethwaite et al [**41**] described a novel compensation technique, based on indirect measurement of thermal error, that overcomes the main difficulties of applying thermal compensation, making it practical and generally applicable. The technique makes extensive use of thermal imaging for rapid assessment of machine tool thermal behaviour and off-line development of the compensation models. It was applied to the head-slide of a vertical machining centre.

It is difficult to develop a practical single compensation strategy which can deal with different aspects of machine tool thermal behaviour. Systems which attempt to do this are usually complex and time consuming to implement. Postlethwaite et al [**42**] devised a range of relatively simple error tracking techniques, and a formalised philosophy for determining which technique is appropriate in each case. The different techniques were illustrated by four case studies. The approach is to split the problem of finding the best error compensation strategy into two phases:-

1. to undertake an assessment of the thermal behaviour of the machine tool. At this stage an understanding of the type and magnitude of likely thermal errors is gained.

2. The selection and implementation of a suitable compensation method.

Improvements in accuracy of 2.5:1 to 6:1 were achieved in the case studies used.

Machine tool thermal distortion can account for 75% of the total machining error [43]. The approaches taken to achieve thermal error reduction can be divided into two broad categories:-

(a) *reducing structural temperature change* and

(b) *reducing the effect of structural temperature change*.

Postlethwaite et al [43] discussed a number of representative examples of applied thermal error reduction techniques that utilise machine design and compensation. White et al [44] described tests that have been performed on ten different CNC machine tools of widely differing configurations. These tests aimed to identify thermal errors and methods by which these errors may be eliminated from components produced by these machines. The general testing methodology used, the data recorded and analysed from the tests, and a list of actions which may be performed in order to reduce the thermal errors in the machines tested were explained. The general testing methodology has evolved from both the BSI 3800 part 3 standard and the need to identify the factors affecting thermal errors. Thermal imaging, non-contact probes, level sensing equipment, material constructed from invar and appropriate exercise of machine functions have all been used to identify thermal errors. The data recorded has identified thermal errors that can be associated with particular mechanism types. Mechanisms investigated include ball-screws, hydro-screws, linear scales, hydrostatic bearings, structures, and cooling systems within structures. Reduction in the thermal errors can be achieved by design changes, improved chiller settings, application of linear scales and revised probing systems. It has become clear that a fundamental understanding of the mechanical arrangement of a particular machine is essential in order to reduce thermal errors in an economic and effective manner. The paper takes a broad overview of the mechanisms which give rise to thermal errors in CNC machine tools. It draws particular attention to the significance of ballscrew thermal errors and the sensitivity of ballscrew pre-tension to changes in temperature.

Working in a similar field Pahk and Lee [45] measured and modelled thermal error caused by heat generated by the spindle. They have used linear regression, neural network, and system identification methods. The general conclusion was drawn that the System Identification model performed better than those based on linear regression or neural network techniques.

Providing a comprehensive compensation system able to cope with the variety of axis combinations and configurations available in the modern machine tool market is a potentially complex matter. This problem is compounded by the demand from industry for precision 5 axis machining. Postlethwaite et al [46] described a practical geometric error compensation system

that can be applied to any machine tool that has up to 5 axes. The system can correct for the effects of all the geometric errors associated with a machine's linear axes, together with the geometric errors associated with a fork type servo head. The system applies compensation dynamically and in real time as the machine moves throughout its working volume.

Machine calibration has become an important tool for assessing and maintaining machine accuracy, and for providing a measure of production quality. The calibration techniques supported by the calibration standards and the available metrology technology use static measurement cycles. Static calibration cycles can be time consuming to perform, and with coarse step sizes cannot give a complete picture of the machine performance.

Recent advances in the laser interferometer technology used for machine calibration allow data to be captured dynamically. Dynamic data capture technology provides the potential for dynamic machine calibration. Dynamic calibration overcomes the inherent problems of static calibration, being quick to perform and providing detailed information on machine performance. Postlethwaite et al [**47**] described the concept of dynamic machine calibration. In particular a novel dynamic calibration technique, utilising the Renishaw laser system, was devised. Capture rates between 5 kHz and 10 kHz are possible and up to 75,000 data points can be processed.

Ford et al [**48**] described an investigation into a laser feedback transducer for high performance CNC machine tool applications. A commercial system available at the time operated at 1 m/s but it was anticipated that further development using the latest state of the art technology would miniaturise the transducer and lower its cost. The interface between the transducer and the controller incorporated a compensation card through which compensation for machine tool geometric, load and thermal errors can be applied in real time.

2.3 **Feed drives modelling**

In the field of precision machining, vibration problems may occur which cannot be evaluated using compensation methods based on standard algorithms. Weck and Hilbing [**49**] have produced an overview of the requirements of non-linear algorithms for use in the field of precision engineering. Examples are given for the analysis of machine vibrations and an active compensation of the dynamic displacement between workpiece and tool in a turning process is demonstrated. The implementation of non-linear algorithms in real-time applications using a standard PC or DSP is discussed.

Pislaru et al [**50, 51**] described a 3D model proposed for evaluating the performance of CNC machine tool axis drives. A modular approach is used to overcome the shortcomings of traditional methods by allowing for the easy exchange of elements without the need to alter the

whole model. It represents the basis for future incorporation of geometric, non-rigid and thermal models of machine tool behaviour. A two-axis model for CNC machine tool axis drives shows how geometric errors affect the machining accuracy across the working volume. A comparison was made between the simulated errors and those measured directly from the machine by a ball bar. The method described represents an advance on simple lumped-parameter methods and compared well with results from classical control theory. Hybrid models which incorporate distributed loads have also been developed [**15**]. These models take account of the electrical characteristics of the controller, sensors and feed-back loops and the electrical and mechanical behaviour of the drive motor and drive system components using lumped parameter or distributed parameter techniques as appropriate.

Various methods have been used to determine the dynamic parameters, as well as the friction characteristics of machine tool drives. For example Erkorkmaz and Altintas [**52**] have used a method based on jogging the axes at various speeds under closed loop control while observing the disturbance torque through a Kalman filter. The parameters of the model were derived from observations of the dynamic response by a least squares method. As verification of the identified friction model, contouring test results without and with friction compensation are also presented. Working in a similar field, Ro et al [**53**] found that the performance of ball-screw-driven slide systems using standard proportional-integral-derivative (PID) control algorithms is unsatisfactory in sub-micrometre motion control because of non-linear friction effects. They have developed controllers based on a bristle-type nonlinear contact model which have been implemented for sub-micrometre motion. A proportional-derivative (PD) control scheme with a non-linear friction estimate algorithm was developed, and its performance was compared with that of a PID controller. For tracking performance, a disturbance observer was added to reject external disturbances and to improve robustness. The experimental results indicated that the proposed controller has consistent performance in positioning with under 1.5% of steady-state error in the sub-micrometre range. The proposed controller showed good and robust behaviour with respect to parameter variation.

An important limitation in the high performance controlled motion systems is due to the interaction between a flexible mechanism and its feedback controller. Dequidt et al [**54**] have sought to include this interaction in the mechanical design. They have generated some design rules with a generic model which includes this interaction. Afterwards, with these new design rules it is possible to determine the bandwidth and the inertial ratio of the mechanism to meet the controlled motion requirements (e.g., speed and precision). With these rules and some other classical rules, an attempt to raise the selection of design solutions has been set within the

context of the ball screw drive system for high speed machine tools (e.g., kinematics, components and sensor location choices).

Braasch [55], (see Figure 2.1), considered whether linear encoders or rotary encoders and ballscrews represent the best solution for measurement on NC machine tools. The action of the balls - wavy motion, changes in speed of spin going in and out of the recirculation system and the frictional behaviour involved – was discussed without giving any modelling method. He concluded that the primary problem involved with position measurement using a rotary encoder attached to the drive motor and a ballscrew is the thermal expansion of the ballscrew and that a linear encoder would be better as requirements for machine tool accuracy and speed increase.

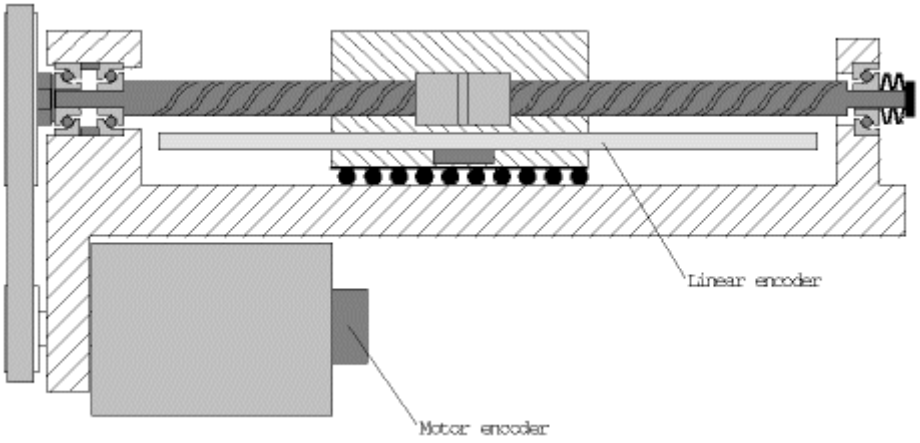


Figure 2.1 – *The essential elements of a typical machine tool drive* [55]

Huang [56] used a multiple regression method to generate a model based on experimental measurements. The front bearing, ballscrew nut and back bearing are the key points of heat sources and were used as independent variables of the analysis model. The method succeeded in predicting the thermal deformation of the ballscrew under a variety of speed conditions. Yun et al [57] have also analysed a ballscrew using a modified lump capacitance model and a fuzzy logic like approach to fitting the model to measured data. Their method gave predictions which compared well with data from a laser interferometer. Only linear positioning error at the tool tip was attributed to the ballscrew. The authors recommended taking into account the characteristics of the guideway as well as those of the ballscrew since some of the errors they measured arose from tilt errors in the guideways.

Kim and Cho [58] tried to model the thermal behaviour of a ballscrew using a finite element method. This worked well for a steady state, but they found that a modified lumped parameter method gave better results for real-time estimation of the temperature distribution. They claim to

be able to estimate the temperature of the ballscrew shaft to within 5% deviation in several milliseconds.

Lin et al [**59**] have developed a method for evaluating the efficiency of a ballscrew nut based on both quasi-static and dynamic considerations and have derived a design procedure from this. An exact theory based on the simultaneous solution of both the Newton-Euler equations of motion and the relevant kinematic equations was employed to determine mechanism efficiency. The steady-state motion of all components within the ballscrew could also be calculated. However, the development of design methods based on this exact theory was difficult due to the extensive computation necessary. Therefore an approximate closed-form representation, that still accounts for the ballscrew dynamics, was derived. The validity of this closed-form solution has been proven. It was then used as a basis for developing an optimum design methodology for the ballscrew mechanism based on efficiency. Additionally, the self braking condition was examined, where the internal friction prevents torsional movement of a screw mechanism in the case where axial loads are applied to the nut. Load capacity considerations were also taken into account.

Greater accuracy in axial positioning a ballscrew nut can be obtained by pre-loading the assembly, but too much pre-load can shorten life. Markhauser [**60**] has given a method for calculating the stiffness and optimum pre-load of a ballscrew to provide maximum accuracy for the required life expectancy.

Thermal stress can have an adverse effect on the accuracy of measurements taken by indirect means. Therefore it is vital to improve our understanding of the thermal behaviour of the drive systems used in modern machine tools in order to improve the drive's performance. Schmitt [**61**] has used a variety of analytical methods to derive a model of the heat transfer process in the mechanical structure of a CNC controlled feed system. The model includes the bearings and belt as well as the ballscrew. Friction characteristics, heat flow, and radiation and convection of heat from the screw are taken into account. A real-time error compensation method was derived. Using this method the CNC controller gave a substantially improved performance in positioning accuracy of the feed axes for a variety of operating conditions. He concluded that an inexpensive improvement in the positioning accuracy of the form feed axes can be obtained using indirect measurements but control of the heat sources falls into the scope of an overall compensation package.

2.4 **Analysis of vibration damping for feed drives**

The amplitude of vibration response to various stimuli is determined to a large extent by the level of damping. Damping can take several forms (viscous, hysteresis, friction etc.) [**62**], but only viscous damping is a linear phenomenon and therefore more amenable to mathematical analysis. The response of damped systems can be studied by a "forced damped" approach [**63**] in which various stimuli (for example out-of-balance forces) can be applied to a system. A picture can be built up for the vibration characteristics of the system by applying such forces at a variety of frequencies.

Holroyd [**64**] modelled the friction and damping in a typical ballscrew machine drive. The machine used was a CNC milling machine which has a vertical spindle mounted on a fixed column. The drives for the horizontal axes consisted of a DC electric motor driving a ballscrew via a belt drive, the ballscrew then driving the table (X drive) or saddle (Y drive). The axis drives were simulated using the MATLAB/Simulink package. The principal natural frequency and its damping factor were measured for both of these drives. A set of mathematically stable models were produced which modelled the main vibration characteristics of the system.

The natural frequencies can be determined by an eigenvalue method without applying stimuli over a range of frequencies if no damping is involved. Holroyd et al [**65**] presented a theoretical analysis of a CNC machine tool drive and used it as a basis on which to develop a method of studying the effects of the damping elements in the drive. The torsional behaviour of the drive's mechanical components was considered. The mechanical elements of the drive were modelled as point inertias coupled by springs and dampers using MATLAB/Simulink. The undamped natural frequencies and their likely mode shapes were determined using an eigenvalue approach in order to act as a guide to the numbers expected. The effect of the distributed mass in the ballscrew was also investigated. Comparison with measurements gave values of the dynamic stiffness and damping factor governing the lowest of the observed natural frequencies. This approach can be used as a basis for investigations into other aspects of the dynamic behaviour of machine tool drives, (for example those arising from Coulomb friction, viscous damping and hydrodynamic drag, and the backlash in the ball-screw). The method is limited to coupled axial and torsional behaviour and some of the parameters are difficult to measure. The models are capable of predicting many of the natural frequencies which have been observed. The models are also limited to the nut being set in a particular position on the screw. This position can be changed from analysis to analysis, but the effects of the continuous change in nut position which in fact takes place in a ballscrew driven system are not included.

The sensitivity of eigensystems has been analysed by the use of derivatives. These methods involve the use of sophisticated mathematical techniques which are necessary when dealing with computationally large problems involving large order matrices.

Adhikari [**66**] derived a method to calculate derivatives of eigenvectors of damped discrete linear dynamic systems with respect to the system parameter. The eigenvectors and their derivatives become complex due to the non-proportional nature of the damping. The derivatives were calculated using a small damping assumption, and the method avoids using the state-space approach. The results were obtained in terms of complex modes and frequencies of the second-order system, which in turn are related to the eigensolutions of the undamped system using a perturbation method. Based on the derivatives, an expression for total change of the complex eigenvectors was obtained for a more general case when all the elements of mass, stiffness and damping matrices are varying.

Nelson [**67**] presented a simplified procedure for the determination of the derivatives of eigenvectors of n^{th} order algebraic eigensystems. The method is applicable to symmetric or nonsymmetric systems, and requires knowledge of only one eigenvalue and its associated right and left eigenvectors. The matrix of the original eigensystem of rank (n-1) is modified to convert it to a matrix of rank n. This can be solved directly for a vector which, together with the eigenvector, gives the eigenvector derivative to within an arbitrary constant. The norm of the eigenvector is used to determine this constant and complete the calculation.

In its early stages, sensitivity analysis found its predominant use in assessing the effect of varying parameters in mathematical models of control systems. Adelman and Haftka [**68**] have surveyed methods for calculating sensitivity derivatives for discrete structural systems. They describe methods for calculating derivatives of static displacements and stresses, eigenvalues and eigenvectors, transient structural response, and derivatives of optimum structural designs with respect to the parameters of the systems involved.

Murthy and Haftka [**69**] have concentrated on reviewing methods for sensitivity analysis of the algebraic eigenvalue problem for non-Hermitian matrices. They put forward a modification of one method based on a better normalising condition. Methods are classified as Direct or Adjoint and are evaluated for efficiency. Operation counts are presented in terms of matrix size, number of design variables and number of eigenvalues and eigenvectors of interest. The effect of the sparsity of the matrix and its derivatives has also been considered, and typical solution times are given. General guidelines are established for the selection of the most efficient method.

Zeng [70] has derived a highly accurate modal superposition method for computing complex eigenvector derivatives in viscous damping systems. The conventional modal superposition method cannot give an accurate solution when higher modes are truncated, and the errors may become significant. In Zeng's method, calculating the derivatives is regarded equivalent to calculating the structural response to harmonic excitation. Using multiple modal accelerations and shifted-poles, highly accurate results would be obtained when only few modes are used. Numerical examples show that it achieves better calculation efficiency than other available modal methods and Nelson's method when more than one eigenvector derivative is of interest. Moreover, the presented method can be used to improve response calculations and substructure syntheses.

Perturbation methods can also be used. For example, Cronin, [71] has derived a perturbation method for the eigenanalysis of non-classically damped dynamic systems. The method appears to be suitable for the rapid determination to any accuracy of one or all of a system's complex eigenvalue-eigenvector pairs. It is not limited to the case where the system damping matrix is symmetric. Being capable of handling non-symmetric matrices, the method is suitable for the eigenanalysis of gyroscopic and other interesting systems. The derivation of the method involves a partial diagonalisation of the homogeneous equations of motion by the eigenvectors of the undamped system. A perturbation quantity based on the off-diagonal terms of the partially diagonalised damping matrix is defined. The eigenvalues and eigenvectors for the damped system are described in terms of power series in the perturbation quantity. Equations have been developed for the general coefficient in each power series. The potential value of the method is illustrated by the eigenanalysis of a set of example systems. For the majority of the systems analysed, the method produced results in less time than the standard Foss approach to curve fitting [72].

Woodhouse [73] used both a dissipation matrix and a general linear model to deal with the case of light damping. Linear damping models for structural vibration were examined: first the familiar dissipation-matrix model, then the general linear model. In both cases, an approximation of light damping was used to obtain simple expressions for damped natural frequencies, complex mode shapes, and transfer functions. Results for transfer functions can be expressed in the form of simple extensions of the expression for the undamped case. This allowed a detailed discussion of the implications of the various models of damping for the interpretation of measured transfer functions, especially in the context of experimental modal analysis. In the case of a dissipation-matrix model, it would be possible in principle to determine all the model parameters from measurements. In the case of the general model, however, there is

a fundamental ambiguity which prevents full determination of the model from measurements on a single structure.

2.5 **Summary**

Modelling of screw mechanisms has developed from the pioneering work of Ball to the more recent sophisticated work of Pfister. A body of expertise on machine tool errors and means of compensating for them has been built up by Ford, Postlethwaite and White. Freeman and White have done some useful research on thermal errors associated with ballscrews. Schmidt and Pislaru have done detailed work on modelling of ballscrew drive systems. Together their work has led to the development of successful models of ballscrew drives which take into account the known characteristics, both mechanical and thermal, of the components of a ballscrew drive and predict some of the dynamic behaviour in such a way that a means of compensation can be applied to the drive system's controller. **However, the way that the dynamic characteristics change as the nut moves along the screw is especially relevant to machine tool applications. An enhancement to modelling methods which deals with this aspect of ballscrew behaviour would therefore be a useful addition to the available methods.**

An approach to this might be to use **finite element analysis** (FEA), a computer simulation technique widely used in the analysis of a variety of engineering and scientific systems. The range of problems which can be covered includes, but is not limited to, analysis of linear and non-linear static systems, problems involving elastic instability (buckling), dynamic analysis, modal analysis, frequency analysis, thermal analysis and fluid flow problems, and various combinations of some of these. It uses a numerical technique called the finite element method (FEM).

Commonly this method is divided into three phases:-

- Pre-processing – defining the finite element model and its environmental factors
- Solution of the finite element model
- Post-processing of the results

Pre-processing pre-processing entails the construction of a finite element model of the structure to be analysed. The usually involves inputting a topological description of the structure's geometric features. In mechanical sytems this can be in either 1D, 2D, or 3D form, modeled by line, shape, or surface representation, respectively. The main purpose of the model is to represent the important features of the real system in the best possible way. This is usually done by representing a complex system by a large number of smaller sub-systems connected together, the smaller sub-systems having well understood mathematical representations.

Analysis (computation of solution) involves carrying out a series of computational procedures involving applied forces, and the properties of the elements which produced the system model. The procedures are commonly based on an energy principle such as the virtual work principle or the minimum total potential energy principle. In a mechanical system such an analysis allows the determination of deformations, strains, and stresses which are caused by the applied structural loads. In a thermal analysis, temperatures and heat flows are computed.

Post-processing. The results can then be presented in a variety of ways. This often involves the use of visualisation tools within the FEA environment to view the results and to identify the implications of the analysis.

Because the method can be applied to very complex systems which cannot be analysed mathematically by any other means, a lot of effort has been put into the development of finite element software packages which are capable of solving such problems. The use of such packages has enabled the speedier development of complex structures and the reduction of the amount of prototype testing required.

The mathematical aspects of finite element analysis were first developed in 1943 by Richard Courant, who used a numerical method to obtain approximate solutions to vibration systems [**130**]. Much of the early structural finite element analysis was developed in British aerospace research. By late 1950s, the key concepts of stiffness matrix and element assembly existed essentially in the form used today and the American space research agency NASA started the development of the finite element software NASTRAN in 1965.

Nowadays there are many finite element packages available, both free and proprietary. Proprietary ones include ALGOR, ANSYS, ABAQUS, LUSAS, LS-DYNA, NASTRAN, SAMCEF, STAMPACK and STRAND7.

ALGOR offers a series of packages based on their FEMPRO finite element modeling, results evaluation and presentation interface, the most comprehensive of which, Professional Multiphysics, includes, mechanical event simulation and static stress analysis with linear and nonlinear material models, linear dynamic analysis, steady-state and transient heat transfer analysis, steady and unsteady fluid flow analysis and electrostatic analysis [**131**].

ANSYS also offers a multiphysics package which combines structural, thermal, computational fluid dynamics (CFD), acoustic and electromagnetic simulation capabilities, together with a series of packages designed to handle particular aspects, e.g. fatigue, structural, drop test etc. [**132**].

ABAQUS supplies a “standard” version which can carry out a wide range of linear and nonlinear engineering simulations efficiently, accurately, and reliably, making it an effective tool for many engineering analyses. They also supply an “explicit” version which is particularly well suited to simulate brief transient dynamic events such as consumer electronics drop testing, automotive crashworthiness, and ballistic impact. A common analysis requirement is the treatment of distinct solution regimes where the characteristic time scale within each regime is different. From any point within an Abaqus/Explicit run, the analysis can be "imported" as the starting conditions for continuation in Abaqus/Standard. Similarly, an analysis that starts in Abaqus/Standard can be imported as the starting conditions for continuation in Abaqus/Explicit. The flexibility provided by this integration allows Abaqus/Explicit to be applied to those portions of the analysis where high-speed, nonlinear, transient response dominates the solution; while Abaqus/Standard may be applied to those portions of the analysis that are well-suited to an implicit solution technique, such as static, low-speed dynamic, or steady-state transport analyses [**133**].

Similarly **LUSAS** sells a “base” version of their software which is suitable for linear static and linear dynamic analysis with optional add-ons which cover non-linear and thermal analyses. They also have some special versions designed to deal with civil engineering applications [**134**].

LS-DYNA is an explicit finite element program for the analysis of the non-linear dynamic response of three dimensional structures. It includes nearly 100 models to simulate a whole range of engineering materials from steels to composites and soft foams to concrete. Coupled thermal/structural problems can also be handled [**135**].

NASTRAN is a finite element analysis program that was originally developed for NASA in the late 1960s. Nowadays NASTRAN source code is used in a number of commercial software packages. It is written primarily in FORTRAN and is compatible with a large variety of computers and operating systems. It has been designed to consist of several modules each of which is a collection of subroutines designed to perform a specific task, e.g. processing model geometry, assembling matrices, applying constraints, solving matrix problems, calculating output quantities, conversing with the database, printing the solution, and so on. The modules are controlled by an internal language called the Direct Matrix Abstraction Program (DMAP). The capabilites include linear and non-linear static, buckling, modal, eigenvalue, frequency response, transient response, non-linear static and non-linear transient with heat transfer, and design optimization and sensitivity analysis. Commercially available versions include MSC Nastran and NEi Nastran [**136**, **137**].

SAMCEF is another suite of general-purpose analysis software modules which uses finite element methods [**138**].

STRAND7 is a finite element analysis package designed and built for Windows®. It includes a variety of one- two- and three-dimensional elements and range of constraints. It can handle a wide range of materials and do linear, buckling and non-linear static analyses, natural frequency analyses, harmonic response and spectral response analyses, transients, and thermal analyses [**139**].

There are also a variety of finite element programs available as free-ware, often in conjunction with the LINUX free-ware operating system.

Although the capabilities of the commercial systems are indeed impressive, several problems would occur using them to investigate the dynamics of ballscrew systems in the case where the main concern is to study the errors which may arise. The first is that modelling a CNC system inevitably includes modelling the controller which makes the problem "multiphysics" rather than purely mechanical. This means that one of the more expensive packages with costs of the order of tens of thousands of pounds would need to be deployed. Secondly, a detailed finite element model would require details of the ballscrew geometry which are not normally available in the literature from ballscrew manufacturers. Finally, running a detailed model for a sufficient number of configurations to build up a picture of what happens dynamically as the nut runs along the screw could involve significantly long execution times. The use of free-ware obviates the cost problem, but for understandable reasons, such packages are only available on an "as is" basis, and the results would need to be checked against known cases. Often the use of free-ware includes the understanding that developments based on them should also be free. This could lead to intellectual property problems.

For these reasons it was decided to develop a method which included the control aspects with the mechanical aspects kept to as simple a form as possible consistent with the objectives of the investigation.

The eigenvalue method is extended in an attempt to determine the damped natural frequencies with their associated modes using a generalised method (see Chapter 4). This is done in order to avoid having to analyse the system with a variety of stimuli over a possibly extended frequency range. The sensitivity of the various modes to changes in the stiffness and damping characteristics of the machine drives are derived. These values can be used as an aid to design or fault correction.

Based on the modelling techniques developed by Pislaru et al [**15**], the problem of predicting how the dynamic behaviour of a ballscrew changes as the nut moves along the screw will be investigated in detail in Chapters 5 and 6. In view of the variety of mechanical phenomena which may be involved, solution methods which can be used conveniently for low or medium order systems, and which are not limited to the linear behaviour, have been developed. Some consideration of a continuous matter approach is given in Chapter 4 before a finite element approach is developed which handles the energy dissipative processes in such a way as to give heat sources to a thermal model.

Chapter 3 sets out the theoretical bases of the various models developed in this investigation.

Chapter 3 - THEORETICAL BACKGROUND

The function of a ballscrew is to move one part of a machine relative to another by transferring the rotational motion of the screw to linear motion of the nut. This interaction between linear and rotary motion is achieved by other screw mechanisms such as screw jacks and various forms of screw fixings.

3.1 **Basic characteristics of a ballscrew**

In order to reduce problems associated with friction and backlash, the **ballscrew** was developed where a set of rolling elements are introduced between the screw and nut. A ballscrew consists of two main components, a **screw** and a **nut** with the rolling elements being **balls**, see Figures 3.1 to 3.3.

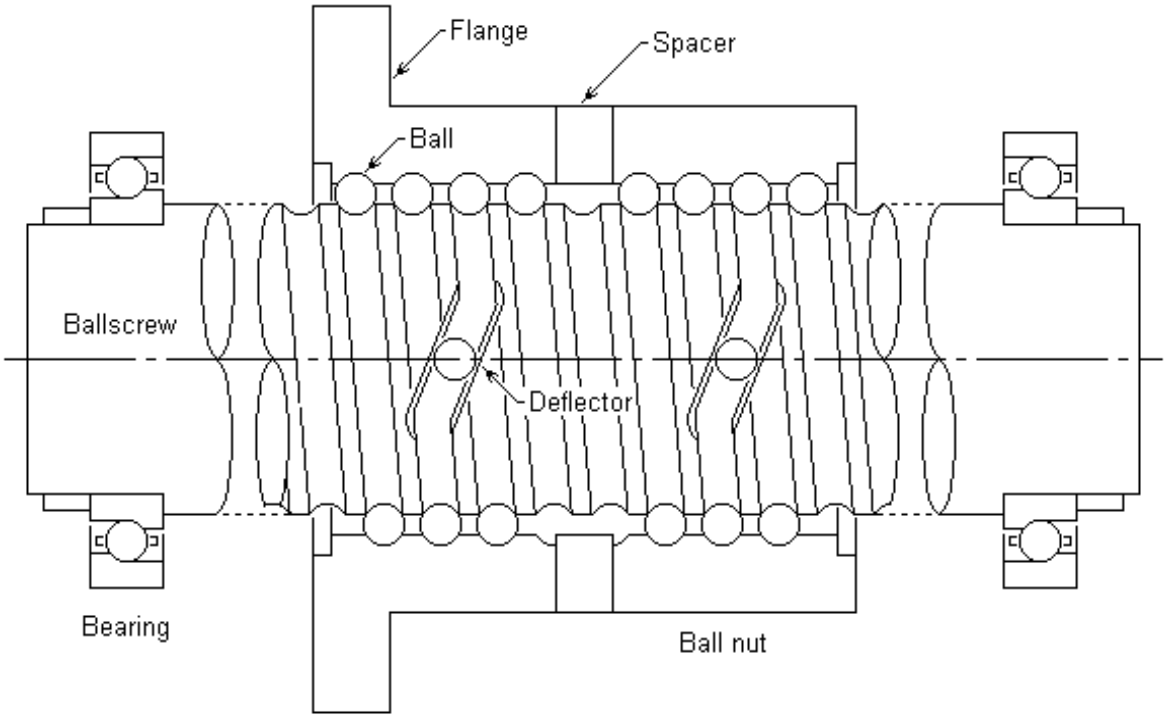


Figure 3.1 - *A double nut ballscrew*

The **screw** is a relatively long cylindrical solid or hollow rod with one or more helical grooves on its outer cylindrical surface. These grooves are close to being a part-circular section, so that the balls can roll in them. The detailed geometry of these grooves and their effect on the ballscrew stiffness and its heat generation processes is included in the scope of this project.

Relative axial movement of the parts of the ballscrew can be achieved by rotating the screw and holding the nut, or by holding the screw and rotating the nut. (Logically, axial movement can be also achieved by rotating both the screw and the nut, though this arrangement is not used in machine tool drives.) In the case where the screw is held, it needs

to be clamped at one or both ends. In the more common case where the screw is rotated it has provision for bearings, at one or both ends, by means of which it is held in the machine. At least one of these bearings needs to be a thrust bearing to provide axial constraint. The screw also has provision at one end for mounting the means of driving the screw, (a flexible coupling, gear wheel, belt drive pulley or chain sprocket etc.).

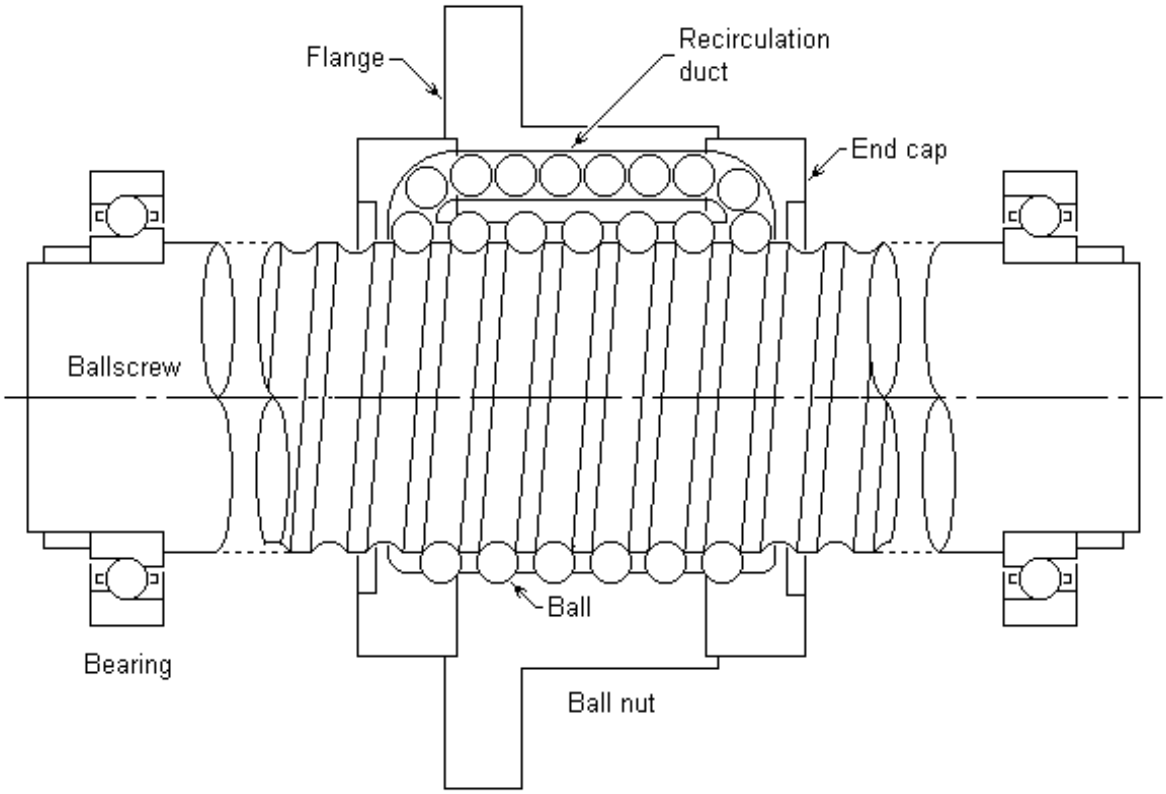


Figure 3.2 - *An end cap ballscrew*

The inner surface of the **nut** is a relatively short hollow cylindrical tube with a diameter a little bigger than the outer diameter of the screw. The outer surface is defined by the requirements of ballscrew application. A very common arrangement is a cylinder with a pair of parallel flat surfaces with a mounting flange at one end. The inner surface of the nut has helical grooves which correspond to those on the screw. The space between the nut and screw is filled with balls which fit into the grooves in the screw and the nut and by this means locate one relative to the other. If the screw is turned relative to the nut, the nut will move axially along the screw.

The nut has to have means of **re-circulating** its balls in order to enable continuous operation. The balls may be re-circulated at each **turn** of the screw or after several. There can therefore be several different sets of balls circulating at once which are known as **circuits**.

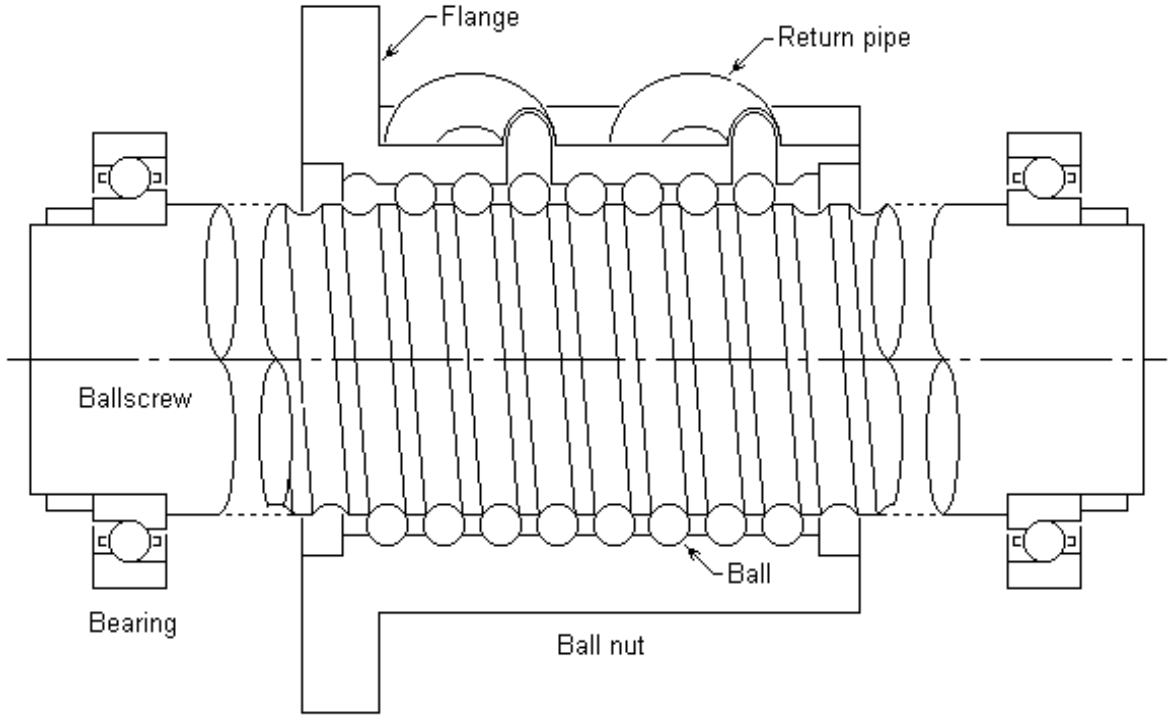


Figure 3.3 - *A return pipe ballscrew*

Re-circulation can be achieved in a variety of ways.

- One makes use of one or more **return tubes**, (see Figure 3.3), inserted into the nut. As the screw rotates, they collect the balls from one end of the groove holding the circuit and guide them back to the other, thus ensuring that the nut is always kept full of balls.
- Another method involves the use of a **deflector** which is placed inside the nut and re-positions the balls to the adjacent groove as the nut rotates, (see Figure 3.1). In this arrangement the balls only make just short of one turn of the screw before they are re-circulated.
- A third way is to use **end caps** which capture the balls and send them back to the other end of the nut through one or more drillings in the nut, (see Figure 3.2).

There should be a small amount of clearance between the balls and the grooves in which they run so that the nut runs freely on the screw. This gives rise to **backlash** between the nut and the screw. In applications where high positional accuracy is important, as is the case with machine tools, it is essential that the backlash should be minimised.

This can be achieved by effectively splitting the nut into two and moving one half axially relative to the other. This traps the balls and puts them under compressive stress thus locating the nut on the screw with a high axial stiffness. This compressive stress is known as **pre-load**.

Commonly, pre-load is introduced in one of five ways:-

1. by having two nuts separated by a **spacer washer**, (see Figure 3.1).
2. to increase or decrease the pitch of the turn in the middle of the nut. This is known as **pitch shift**.
3. Another form of pitch shift is to have the spirals of a double start nut slightly off-set from one another.
4. to arrange the nut in two halves and make the portion between them in the form of a **spring**. This spring can be set up to exert a steady axial load on each half.
5. to use balls which are slightly oversized for the screw and nut groove profiles.

The nut, screw and balls are lubricated by either oil or a charge of grease and a seal is fitted at each end to contain the lubricant and keep foreign matter, (such as swarf and coolant), away from the bearing surfaces.

3.2 Errors in machine tool drives

Ballscrew errors

Based on the previous experience at the University of Huddersfield [27, 28, 30, 31, 38, 40, 50, 51], and incorporating as appropriate the information in the public domain [29, 33, 53, 74], the project seeks to develop relevant models of the mechanisms giving rise to the geometric, thermal and load errors seen on the ballscrews commonly used in machine tools.

These various effects need to be incorporated into the ballscrew models.

The aim of this work is to model the geometric, thermal and load errors associated with a ballscrew as applied in a CNC machine tool. Error associated with a ballscrew can be defined as the difference between the actual position of the ballscrew nut and the position which is expected or required.

Errors can be grouped under three headings – **geometric**, **thermal** and **load**.

Geometric errors arise from manufacturing errors in the ballscrew and its mounting arrangement. In principle six types of error can occur – axial, lateral (in two mutually perpendicular directions), tilting (in two mutually perpendicular directions), and torsional. However, since a ballscrew allows rotation of the screw relative to the nut, the idea of a torsional error is meaningless in this case.

A ballscrew is included in a machine to effect linear movement of one part relative to another along the axis of the ballscrew, for example, the movement of a saddle along a machine bed. The location of these parts in the plane at right angles to the ballscrew axis is looked after by components other than the ballscrew, typically linear guides or slideways.

Therefore, any run-out in the screw is likely to be restrained by the rest of the machine structure and give rise to fluctuations in the lateral forces within the ballscrew nut. Much the same can be said of run-out of the cylindrical surfaces provided for fitting the bearings and swash in the abutments provided for their axial location. The same applies to run-out and swash of the feature provided on the screw for applying the torque and of the mounting flange on the nut. All these geometrical "imperfections" will give rise to additional forces in the screw and nut which will affect the dynamic behaviour, but will only have a second order effect on the axial position of the nut.

The geometrical generating errors in axial positioning are pitch errors in the screw and nut and clearance between the screw, the balls and the nut in the case of ballscrews which are not pre-loaded.

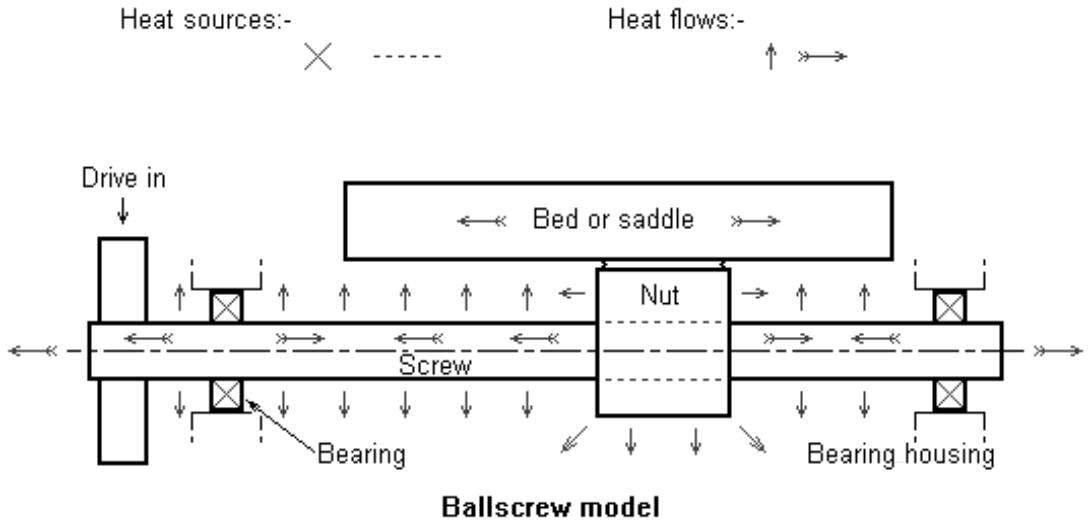


Figure 3.4 - *Thermal model of ballscrew*

Thermal errors arise when the temperature of various parts of the ballscrew differ from ambient temperature and the ballscrew nut moves to a position different from where it would be if all the parts of the ballscrew were to be at ambient temperature. The major sources of thermal error are heat generated by the ballscrew system during running and heat entering or leaving the ballscrew to or from the torque input device, the bearing mountings and the part to which the nut is mounted. The heat sources within the ballscrew system are the nut and the bearings supporting the screw. Several heat generation mechanisms can be identified within the nut:

- friction associated with rolling contact between the balls, the screw and the nut
- friction between the balls and parts of the re-circulation system as the balls change angular velocity as they enter and leave the ballscrew grooves
- friction arising from the opposing velocities of adjacent balls

- churning of the lubricant
- seal friction

Heat is transmitted to, from and through the system by conduction and dissipated from it by convection and radiation, (see Figure 3.4).

Load errors arise from relative movement of the parts of a ballscrew which occur due to the loads imposed on the ballscrew system. Loads can occur due to torques applied, acceleration of parts, pre-load of the screw and self weight. All these loads cause various forms of deformation, such as torsional and axial deformation of the ballscrew, with the possibility of introducing unwanted movement in the axial direction. The magnitude of some of these errors depends on the axial position of the nut.

Also there is an interaction between the various forms of error. For example, run-out of the screw which is a geometric error may induce extra loads in the nut and bearings through out of balance forces. The additional friction in the bearings can then cause extra heating giving rise to an additional thermal error.

3.3 **Factors affecting thermal, geometric and load performance**

The factors contributing to the various errors arise as a result of the action of the different parts of the ballscrew. These are now considered on a component by component basis.

3.3.1 **Bearings**

The bearing arrangement which is used to mount a ballscrew within a machine tool structure can be divided into four types:-

1. an arrangement whereby the screw is held at one end only by a set of bearings capable of carrying radial and bi-directional axial forces and tilt moments. This is called a **"fixed – free" arrangement**, (see Figure 3.5). Such an arrangement is commonly used in a vertical orientation with the free end at the bottom.

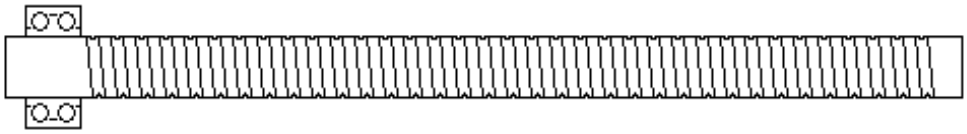


Figure 3.5 – *The "fixed-free" bearing arrangement*

2. The **"fixed - supported"** arrangement (see Figure 3.6) is where axial and radial loads are resisted at one end and only lateral forces are carried by the other bearing,

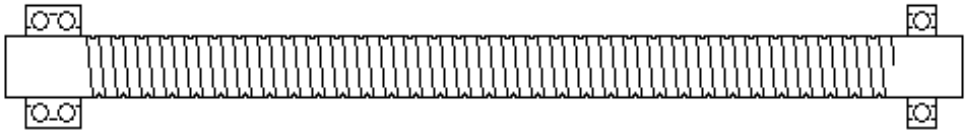


Figure 3.6 – *The "fixed-supported" bearing arrangement*

3. The **"fixed – fixed"** arrangement (see Figure 3.7) uses a bearing or set of bearings at both ends which is able to carry both thrust and radial loads.

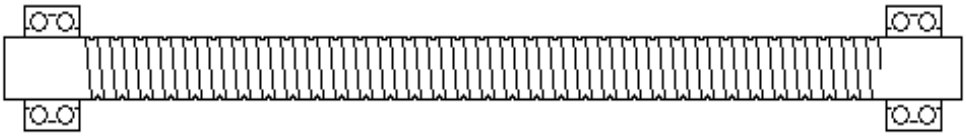


Figure 3.7 – *The "fixed-fixed" bearing arrangement*

4. When the nut is to be rotated rather than the screw, the screw may be clamped at both ends, (see Figure 3.8). This is also a **"fixed – fixed"** arrangement [75].

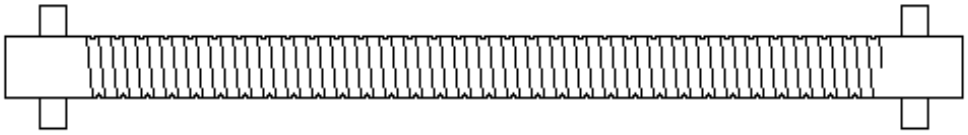


Figure 3.8 – *The "fixed-fixed" screw clamping arrangement*

The bearings affect the behaviour of a ballscrew system by inducing geometric errors, generating and transmitting heat, and providing support to the ballscrew.

Factors which may affect the contribution of bearings to ballscrew errors include:-

(a) *Bearing type*

There is a great variety of types of bearing [76] such as deep groove ball bearings, self-aligning ball bearings, angular contact ball bearings, cylindrical roller bearings, needle roller bearings, spherical roller bearings, taper roller bearings, thrust ball bearings, angular contact thrust ball bearings, cylindrical roller thrust bearings, needle roller thrust bearings, spherical roller thrust bearings and taper roller thrust bearings etc.

For a ballscrew in a machine tool where high running accuracy is required, the types best able to give high accuracy are deep groove ball bearings, back-to-back angular contact bearings and double row cylindrical roller bearings.

(b) *Quality factors*

The number, diameter, surface finish and sphericity of the balls or rollers, etc and the surface finish of the bearing races and their circularity can influence geometric errors to a small degree. The surface finish and variations in size affect friction and hence have thermal consequences. They also induce load fluctuations and hence cause vibration. For example, a

bearing with an outer race diameter of d_o and an inner race diameter of d_i with N_b balls has a ball passing frequency given by:-

$$f_{bp} = \frac{d_i}{d_o + d_i} \frac{\omega_s}{2\pi} \tag{3.1}$$

where ω_s is the angular speed of the shaft. Defects such as pits or scratches in the balls or raceways produce high frequency vibration. The possible occurrence of such vibration is used as a condition monitoring technique in process machinery to detect potential problems with bearings.

(c) ***Effect of caging***

The design and influence of the cages which separate the rolling elements of the bearings can have an effect on friction and heat generation. Phosphor bronze cages are common and cages made of PTFE (polytetrafluoroethylene) can reduce friction. "Full complement" bearings, which do not have cages, increase stiffness but at the expense of greater friction.

(d) ***Lubrication and sealing***

Lubrication can be of two types, oil or grease.

- Oil lubrication, where oil is fed to each bearing and drained away to a separate oil tank, has several advantages – the temperature of the oil can be controlled and the flow can be directed in such a way as to act as a coolant. The lubricant can be changed without stripping down the machine. Its quality can be maintained by appropriate filtration and topping up of depleted additives as needed. Compared to grease, however, sealing can be more difficult. There is also the cost of installing and maintaining the lubrication system with its pumps, pipes, tanks, valves and filters.
- Grease lubrication, where a charge of grease is put in each bearing, can be of the "sealed for life" variety, which needs no further maintenance, or can have grease nipples whereby extra grease can be put in from time to time. Automatic greasing systems are fitted on many modern machine tools. Grease systems have the advantage of being simple and cheaper to install. Much of the heat generated in the bearings in such cases must be transmitted to the shaft or bearing housing and is difficult to predict or control.

Seals can be of two types - contact or non-contact. Non contact seals, (e.g. labyrinth seals), tend to leak especially at low speeds. Contact seals, (e.g. lip seals of various kinds), grip the shaft and the associated friction produces heat.

(e) ***Pre-load***

Pre-load in the bearing system increases the rate of heat generation through friction by imposing internal loads in addition to the other loads carried by the bearing. It also increases stiffness and reduces geometric errors by taking out backlash. Pre-load also reduces the fatigue life of the balls and the bearing race surfaces.

(f) ***Misalignment***

Like pre-load, misalignment can induce additional loads with their consequent additional heating. Increases in out-of-balance caused by radial misalignment cause vibratory excitation forces at the rotational speed of the shaft. Misalignment generally causes vibration to occur at twice the shaft speed and its harmonics.

3.3.2 Ball nut

The ball nut contributes to the behaviour of a ballscrew system by generating geometric errors, by generating and transmitting heat and by providing axial location and force to the machine part to which the nut is attached.

Factors which may affect the contribution of the ball nut to ballscrew errors include:-

(a) ***Nut type***

There is a variety of types of ball nut [77] such as single nuts without pre-load (Figure 3.2), and nuts with pre-load. These include single nuts with slightly oversized balls, double nuts with spacers (Figure 3.1), nuts with a pitch shift arrangement in the middle (Figure 3.3), and double nuts with a spring arrangement in the middle. This last arrangement is meant to give a near-constant pre-load force. Ball nuts also vary as to the means used to re-circulate the balls and the sealing methods.

For a ballscrew in a machine tool, where high running accuracy is required, some pre-load is desirable.

(b) ***Quality factors***

The number, diameter, surface finish and sphericity of the balls and the surface finish of the screw and nut grooves and their geometric accuracy can influence geometric errors to a small degree. The surface finish and variations in size of the balls affect friction and hence have significant thermal consequences. They also induce load fluctuations which cause vibration.

(c) ***Effect of ball re-circulation method***

As the balls circulate along the screw, they spin at a high speed determined by the geometry of the ballscrew. When they leave the screw to be re-circulated, they stop spinning

and are pushed back to where they re-enter the screw. They then start spinning again. This process entails a loss of mechanical energy and hence the generation of heat each time the ball is stopped. Forces are involved in the impacts of the balls with the walls of the re-circulation system and with other balls, which occur as the balls are picked up for re-circulation. The ability of the means of re-circulation to stand these forces sometimes sets a limit on the ballscrew's performance.

(d) ***Lubrication and sealing***

Lubrication can be of two types, oil or grease, and the comments in Section 3.3.2d also apply to ball nuts.

(e) ***Pre-load***

As in the case of bearings, pre-load in the ball nut gives rise to internal loads which act on the rolling elements in addition to the externally imposed loads carried by the nut. These increase the rate of heat generation which is generated by friction. It also increases the stiffness of the nut and reduces geometric errors by taking out backlash. Pre-load also reduces the fatigue life of the balls, and that of the screw and nut groove surfaces.

(f) ***Misalignment***

Like pre-load, misalignment can induce additional loads with their consequent additional heating.

3.3.3 **Pre-tension**

The axial stiffness of a ballscrew system can be increased by putting the screw in a state of tension. This is commonly achieved by fixing the outer races of the support bearings at both ends of the screw, and locating the screw axially at the driven end by butting the bearing inner race against a shoulder on the screw. Pre-tension is then achieved by pulling the screw through the inner race at the non-drive end using a ring nut which is locked in position once the desired tension has been achieved.

Pre-tension also increases the friction losses in the bearings and tends to reduce bearing life. If the screw runs at a higher temperature than the part of the machine in which the ballscrew is mounted, the pre-tension is reduced as the relative temperature increases.

3.4 **Basic behaviour of screw**

Essentially a **screw mechanism** consists of a cylindrical rod with one or more helical grooves cut in the outer surface. If there is more than one helix, these are arranged so that the adjacent grooves are equally spaced axially along the screw. The axial distance between corresponding points on the grooves is known as the **pitch** p, (see Figure 3.9). The number of

helices is known as the **number of starts** N_s. The **nut** includes a cylindrical hole with grooves compatible with those on the screw formed in it. If the screw is rotated by some angle θ relative to the nut, the nut moves axially along the screw by an axial distance x given by:-

$$x = \frac{N_s p}{2\pi}\theta = R\theta \tag{3.2}$$

where θ is expressed in radians and x and p in metres. R can be called the **ratio** of the screw, its units are metres/radian. In the simplest case, the outside of the screw and the inside of the nut are ideal frictionless surfaces. Then the axial force F delivered by the screw to the nut can be derived from the torque Γ applied to the screw by:-

$$F = \frac{2\pi}{N_s p}\Gamma = \frac{\Gamma}{R} \tag{3.3}$$

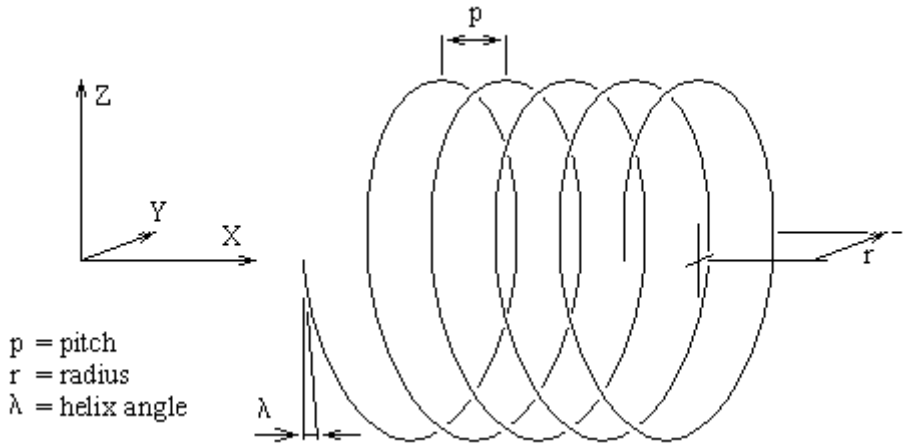


Figure 3.9 – *A simple screw*

Friction plays a very important role in screw mechanisms because without it screws would be quite useless for one of their most common applications, namely holding things together. Consider, for simplicity, the case of the rectangular section screw thread. Such threads were widely used on machine tools drives in situations where a ballscrew would be used today. In this case contact between the screw and the nut takes place along a radial helix. Three distinct modes of operation occur. To illustrate this, consider a lead screw system where the screw rotates and the nut is mounted on a component that does not. A ballscrew without pre-load acts in a similar manner to a lead screw but with a significantly lower coefficient of friction, typically 0.003 – 0.01 [**140**]. It can therefore be considered using the same mathematical formulae. Pre-load removes the backlash and brings extra forces into consideration.

1. The first case is where the contact force in the nut is acting on the nut in the same axial direction as the direction in which the nut is travelling, (see Figure 3.10). In

this case the screw can be said to be “driving” the nut and the force of friction opposes the torque applied to the screw.

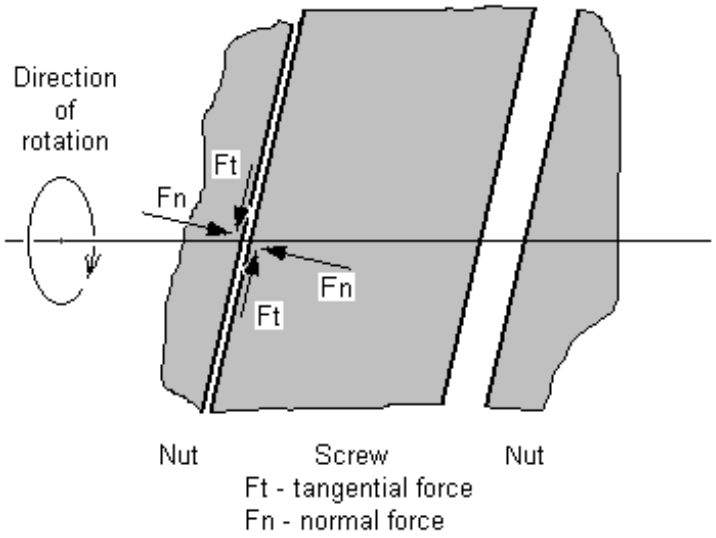


Figure 3.10 – *“Ballscrew driving” mode of operation*

2. The second case is where the contact force in the nut is acting on the nut in the opposite axial direction to that in which the nut is travelling, (see Figure 3.11). In this case the screw can be said to be “driven by” the nut and the friction opposes the force applied to the nut.
3. The third case is where there is backlash between the screw and the nut. In this case contact can be lost and for a short time the nut and screw act independently.

Where the screw is fixed and the nut rotates as might be the case for screw greater than 3 m in length, the same applies but contact takes place on the opposite flanks of the nut and screw profiles. In machine tool drive applications, normally the screw drives the nut,

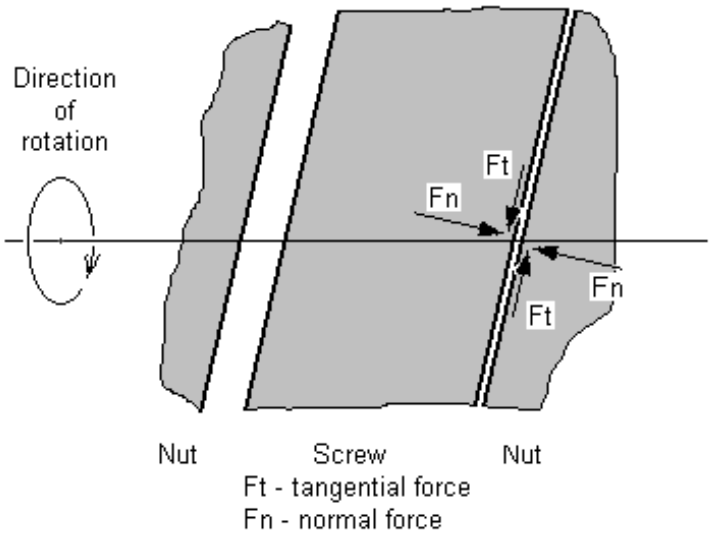


Figure 3.11 – *“Ballscrew driven” mode of operation*

however, if the drive is slowing down the part of the machine which is being driven by the screw, it is possible that the nut will push the screw.

3.4.1 **Ballscrew driving**

For the case where the screw is driving the nut, if the sum of the forces of contact normal to the helix is F_n and the coefficient of friction is μ, then the sum of the forces acting in the axial direction F_a is given by:-

$$F_a = F_n \cos\lambda - \mu F_n \sin\lambda, \tag{3.4}$$

where λ is the helix angle. The sum of the forces acting tangential to the axis of the screw F_t is given by:-

$$F_t = F_n \sin\lambda + \mu F_n \cos\lambda, \tag{3.5}$$

Considering a cylinder of radius r which intersects the helices at the middle of the zone of contact,

$$\cos\lambda = \frac{2\pi r}{\sqrt{(N_s p)^2 + (2\pi r)^2}} = \frac{r}{\sqrt{R^2 + r^2}} \tag{3.6a}$$

$$\sin\lambda = \frac{N_s p}{\sqrt{(N_s p)^2 + (2\pi r)^2}} = \frac{R}{\sqrt{R^2 + r^2}} \tag{3.6b}$$

Substituting into equations (3.4) and (3.5) now gives:-

$$F_a = \frac{F_n}{\sqrt{R^2 + r^2}} \times (r - \mu R) \tag{3.7}$$

$$F_t = \frac{F_n}{\sqrt{R^2 + r^2}} \times (R + \mu r) \tag{3.8}$$

Eliminating F_n and deriving the torque from F_t gives

$$\Gamma = \frac{R + \mu r}{r - \mu R} r F_a \tag{3.9}$$

Assuming that the friction follows Coulomb's law these equations represent a running screw when both the following conditions apply:-

$$-\mu F_n \sin\lambda \le F_a - F_n \cos\lambda \le \mu F_n \sin\lambda \tag{3.10a}$$

$$-\mu F_n \cos\lambda \le F_t - F_n \sin\lambda \le \mu F_n \cos\lambda \tag{3.10b}$$

3.4.2 **Ballscrew driven**

Considering next the screw being driven by the nut, equations (3.4), (3.5), (3.7) and (3.8) become:-

$$F_a = F_n \cos\lambda + \mu F_n \sin\lambda \tag{3.11},$$

$$F_t = F_n \sin\lambda - \mu F_n \cos\lambda \quad (3.12),$$

$$F_a = \frac{F_n}{\sqrt{R^2 + r^2}} \times (r + \mu R) \quad (3.13),$$

$$F_t = \frac{F_n}{\sqrt{R^2 + r^2}} \times (R - \mu r) \quad (3.14).$$

Eliminating F_n and deriving the torque from F_t now gives

$$\Gamma = \frac{R - \mu r}{r + \mu R} r F_a \quad (3.15).$$

A mathematical model of a screw drive which seeks to take into account the behaviour of a screw drive should include these two possible types of operation.

A third mode of operation is of course when the screw moves across the backlash. In this case the dynamic behaviour of the sub-system upstream of the screw/nut connection and that of the sub-system downstream need to be considered independently. This is achieved by making zero the stiffness of the spring which represents the ball action in the models. This aspect is studied in Chapter 5.

3.5 **Static elastic theory**

3.5.1 **General Hertzian theory**

Contact between the screw and its nut is mediated by a set of balls. In order to gain insight into the way a ballscrew responds to the loads imposed upon it, it is necessary to understand how two elastic bodies behave when in contact [**78**].

This problem was solved by Hertz [**79**] in 1882. Consider two elastic bodies 1 and 2 and let them touch at some point. At this point of contact let the curvature of body 1 be described by radii R_1 and R_1' and the curvature of body 2 be described by radii R_2 and R_2'. Let the angle between the plane containing $1/R_1$ in body 1 and that containing $1/R_2$ in body 2 be ϕ.

The material properties of the bodies are Young's modulus E_i and Poisson's ratio ν_i.

The following variables are defined:

a material factor, $C_E = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2}$, (3.16)

a geometry factor $K_D = \frac{1.5}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_1'} + \frac{1}{R_2'}}$ (3.17)

and

$$\cos\theta = \frac{K_D}{1.5}\sqrt{\left(\frac{1}{R_1}-\frac{1}{R_1'}\right)^2+\left(\frac{1}{R_2}-\frac{1}{R_2'}\right)^2+2\left(\frac{1}{R_1}-\frac{1}{R_1'}\right)\left(\frac{1}{R_2}-\frac{1}{R_2'}\right)\cos 2\phi} \qquad (3.18)$$

Let P be the contact force. The bodies contact over an elliptical area of semi-axes c and d. Provided that c and d are small compared to the radii R_1, R'_1, R_2 and R'_2 and that there is no friction between the contacting bodies, c and d are given by:

$$c = \alpha\sqrt[3]{P\,K_D C_E} \qquad (3.19a)$$

$$d = \beta\sqrt[3]{P\,K_D C_E} \qquad (3.19b)$$

The movement between a point in body 1 remote from the contact zone and a similar point in body 2 is given by:

$$y = \gamma\sqrt[3]{\frac{P^2 C_E^2}{K_D}} \qquad (3.20)$$

The dependence of α, β and γ on $\cos\theta$ is as tabulated here in Table 3.1:

Table 3.1 – *Factors used in determining Hertzian stress and deflection* [78]

$\cos\theta$	α	β	γ	$\cos\theta$	α	β	γ
0.00	1.000	1.000	0.750	0.80	2.292	0.544	0.594
0.10	1.070	0.936	0.748	0.85	2.600	0.507	0.559
0.20	1.150	0.878	0.743	0.90	3.093	0.461	0.510
0.30	1.242	0.822	0.734	0.92	3.396	0.438	0.484
0.40	1.351	0.769	0.721	0.94	3.824	0.412	0.452
0.50	1.486	0.717	0.703	0.96	4.508	0.378	0.410
0.60	1.661	0.664	0.678	0.98	5.937	0.328	0.345
0.70	1.905	0.608	0.644	0.99	7.774	0.287	0.288
0.75	2.072	0.578	0.622				

A system of local axes is set up in the contact zone such that the ξ axis is along the c semi-axis and the η axis is along the d semi-axis. The contact pressure σ_c (N/m^2) is spread across the contact zone in a parabolic manner according to the formula:-

$$\sigma_c = \sigma_{c\max} \times \sqrt{1 - \left(\frac{\xi}{c}\right)^2 - \left(\frac{\eta}{d}\right)^2} \tag{3.21}$$

where the maximum pressure $\sigma_{c\,max}$ is given by

$$\sigma_{c\max} = \frac{1.5P}{\pi cd} \tag{3.22}$$

The stiffness is non-linear, because it changes with load. For a given load, the contact stiffness k_c is expressed as follows:-

$$k_c = \frac{dP}{dy} = \frac{3}{2}\sqrt[3]{\frac{K_D}{\gamma^3 C_E^2}}P^{1/3} \tag{3.23a}$$

and the flexibility fl_c as:-

$$fl_c = \frac{dy}{dP} = \frac{2}{3}\sqrt[3]{\frac{\gamma^3 C_E^2}{K_D}} \times \frac{1}{P^{1/3}} \tag{3.23b}$$

Therefore, considering a single contact area between a ball and a raceway, the stiffness of such a contact is zero when the load is zero. For a positive load the stiffness increases with the cube root of the load. Since contact must be established before a set of opposing forces can occur, negative loads cannot arise.

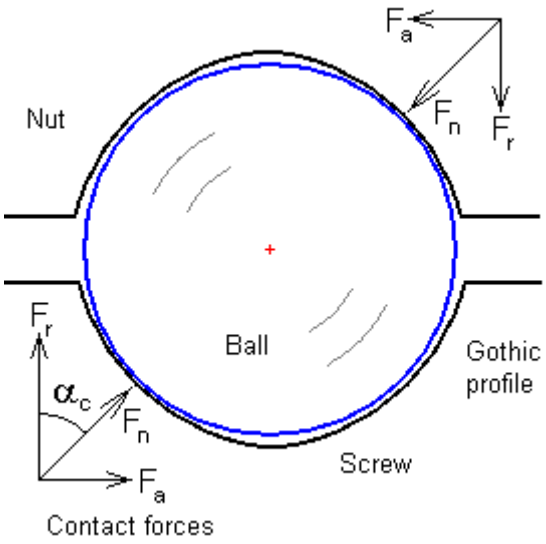


Figure 3.12 – *Contact in ballscrew meshing action*

In the case of a ballscrew, contact between a single ball occurs in two places, between the ball and the screw and between the ball and the nut, (see Figure 3.12). Let the balls have a diameter of d_b, the ball track sections have radii of R_s for the screw and R_n for the nut and the

ballscrew have a ball pitch circle diameter of D_{bp}. Let the contact angle be α_c and the lead angle be λ. The **conformity ratio** $f_{s,n}$ is defined by:-

$$f_s = R_s / d_b, \quad f_n = R_n / d_b \tag{3.24}$$

In terms of the variables used in equations (3.17) and (3.18), for ball / screw contact,

$$\begin{aligned} R_1 &= R_1' = \frac{d_b}{2} \\ R_2 &= -f_s d_b \\ R_2' &= \frac{D_{bp} - d_b \cos\alpha_c}{2\cos\alpha_c} \end{aligned} \tag{3.25a}$$

and for ball / nut contact,

$$\begin{aligned} R_1 &= R_1' = \frac{d_b}{2} \\ R_2 &= -f_n d_b \\ R_2' &= -\frac{D_{bp} + d_b \cos\alpha_c}{2\cos\alpha_c} \end{aligned} \tag{3.25b}$$

In both cases,

$$\phi = 0 \tag{3.25c}$$

Then, for ball / screw contact,

$$K_{Ds} = \frac{1.5}{\frac{4}{d_b} - \frac{1}{f_s d_b} + \frac{2\cos\alpha_c}{D_{bp} - d_b \cos\alpha_c}} \tag{3.26a}$$

$$\cos\theta_s = \left| \frac{-\frac{1}{f_s d_b} - \frac{2\cos\alpha_c}{D_{bp} - d_b \cos\alpha_c}}{\frac{4}{d_b} - \frac{1}{f_s d_b} + \frac{2\cos\alpha_c}{D_{bp} - d_b \cos\alpha_c}} \right| \tag{3.26b}$$

and for ball / nut contact,

$$K_{Dn} = \frac{1.5}{\frac{4}{d_b} - \frac{1}{f_n d_b} - \frac{2\cos\alpha_c}{D_{bp} + d_b \cos\alpha_c}} \tag{3.27a}$$

$$\cos\theta_n = \left| \frac{-\frac{1}{f_n d_b} + \frac{2\cos\alpha_c}{D_{bp} + d_b \cos\alpha_c}}{\frac{4}{d_b} - \frac{1}{f_n d_b} - \frac{2\cos\alpha_c}{D_{bp} + d_b \cos\alpha_c}} \right| \tag{3.27b}$$

For one ball the flexibility is given by

$$fl_{cb} = \frac{2}{3}\left(\sqrt[3]{\frac{\gamma_s^3 C_E^2}{K_{Ds}}} + \sqrt[3]{\frac{\gamma_n^3 C_E^2}{K_{Dn}}}\right) \times \frac{1}{P^{1/3}} \tag{3.28}$$

and in terms of the axial component of force and displacement

$$fl_{cb\,x} = \frac{2}{3} \frac{\sqrt[3]{\frac{\gamma_s^3 C_E^2}{K_{Ds}}} + \sqrt[3]{\frac{\gamma_n^3 C_E^2}{K_{Dn}}}}{(\cos\lambda \sin\alpha_c)^2} \times \frac{1}{P^{1/3}} \tag{3.29}$$

The number of balls in contact z_1 is given by

$$z_1 = N_t\left(\frac{D_{bp}}{\cos\beta}\frac{\pi}{d_b}\right) \tag{3.30}$$

where N_t is the number of active turns in the nut. The axial stiffness of the balls in contact is then:-

$$k_{c\,x} = \frac{3z_1}{2} \frac{(\cos\beta \sin\alpha_t)^2}{\sqrt[3]{\frac{\gamma_s^3 C_E^2}{K_{Ds}}} + \sqrt[3]{\frac{\gamma_n^3 C_E^2}{K_{Dn}}}} \times P^{1/3} \tag{3.31a}$$

$$fl_{c\,x} = \frac{2}{3z_1} \frac{\sqrt[3]{\frac{\gamma_s^3 C_E^2}{K_{Ds}}} + \sqrt[3]{\frac{\gamma_n^3 C_E^2}{K_{Dn}}}}{(\cos\lambda \sin\alpha_c)^2} \times \frac{1}{P^{1/3}} \tag{3.31b}$$

3.5.2 **Other elastic effects**

Besides the Hertzian compression of the balls and raceways, there are other factors which contribute to the overall deflection of the ballscrew nut.

First, the radial forces induced by the action of the balls in contact with the outer surface of the screw and the inner surface of the nut tend to squash the screw and stretch the nut. This effect can be quantified using thick-shell theory [**80**] and gives a flexibility with an axial component given by:-

$$fl_w = \frac{\frac{1}{E_n}\left(\frac{D_o^2 + D_i^2}{D_o^2 - D_i^2} + \nu_n\right) + \frac{1}{E_s}\left(\frac{d_o^2 + d_i^2}{d_o^2 - d_i^2} - \nu_s\right)}{2\pi l_n} \tag{3.32}$$

where d_o and d_i are the outer and inner effective diameters of the screw and D_o and D_i are the outer and inner diameters of the nut. E_n is the Young's modulus of the nut material and ν_n is its Poisson's ratio, E_s and ν_s are the corresponding values for the screw. In the event that one

value of the modulus of elasticity is applicable to the material of both the screw and the nut, which is usually the case, equation (3.32) simplifies to:-

$$fl_w = \frac{\left(\frac{D_o^2 + D_i^2}{D_o^2 - D_i^2} + \frac{d_o^2 + d_i^2}{d_o^2 - d_i^2}\right)}{2\pi E l_n} \qquad (3.33).$$

In a typical ballscrew application D_o and d_o are available from manufacturer's data. Most ballscrews are solid so that $d_i = 0$. D_i is best obtained from the ballscrew manufacturer. In the absence of this information, it is reasonable, in view of the need to maintain as effective contact between the nut and the balls as there is between the balls and the screw, to estimate the nut effective inner diameter as being as far "out" from the ball's pitch circle as the outside diameter of the screw is "in" from it. In this case:-

$$D_i = 2 \times d_p - d_o \qquad (3.33a)$$

where d_p is the pitch circle diameter of the ball action.

Second, the axial forces generated by the screw action usually are applied to the nut at some distance from the flange used to mount it to the saddle, table or whatever part of the machine the ballscrew is meant to drive. A very common arrangement has the mounting flange close to one end of the nut. Taking the centre of action of the forces to be close to the mid-point of the nut, the flexibility associated with the resulting stretch can be estimated as:-

$$fl_b = \frac{l_n/2}{E_n \times \frac{\pi}{4}(D_o^2 - D_i^2)} \qquad (3.34).$$

Third, the flange itself is subject to some deflection. The flexibility associated with this can be estimated from the shear deflection of the material which lies between the mounting screw's pitch circle and the body of the nut as follows:-

$$fl_s = \frac{D_{PCD} - D_o}{G_n \times \pi (D_{PCD} + D_o)} \qquad (3.35).$$

where D_{PCD} is the pitch circle diameter of the mounting flange holes and G_n is the shear modulus of the nut material.

The axial flexibility given by equation (3.31b) should be added to that given by equation (3.32) or equation (3.33) and that given by equations (3.34) and (3.35) to give the total axial flexibility of the ball action.

3.5.3 **Rolling element friction**

The radius of curvature of the contact zone is different from that of the undeformed ball. Under no load, the ball would roll on a circle of diameter equal to that of the ball, and the velocity of contact relative to the ball centre is the angular speed of the ball times its radius. Consider the outer race of a ball bearing. (The same situation also occurs on the inside of a ball nut assuming that the nut is not rotating.) The surface against which the ball is rolling is "stationary". Under load the different parts of the contact surface are at different radii from the axis of rotation of the ball. Since the angular speed of the ball is fixed, the relative velocity of the ball centre and a particular point on the race is dependent on the radius of the part of the ball in contact with the point. In this situation it is not possible for all points on the contact surface to be involved in pure rolling and there must be a certain amount of sliding which gives rise to **rolling friction**. Further, some of the strain energy which is put into the material of the ball and raceway when they come under load is dissipated as heat upon release of the load. There is sliding between adjacent balls and shearing and churning of the lubricant [**81**].

In the simplest terms this can be represented as an equivalent coefficient of friction. Work on rolling element bearings [**82**] suggests a value in the range of 0.0020 – 0.0024 is appropriate for ballscrews. A more sophisticated approach gives the friction moment M_f as the sum of a load independent component M_{f0} and a load dependent component M_{f1} [**83**]:-

$$M_f = M_{f0} + M_{f1} \tag{3.36}$$

in which

$$M_{f0} = \frac{1}{10} f_0 (\nu n)^{2/3} D_{bp}^3 \tag{3.37}$$

and

$$M_{f1} = f_1 P_1 D_{bp} \tag{3.38}$$

where, for an axial load of F_a and a radial load of F_r,

- $f_0 = 3.5 - 6$ for oil lubrication and $f_0 = 1.75 - 3$ for grease.
- $f_1 = 0.001 \left(\frac{P_0}{C_0} \right)^{1/3}$ (3.39a)
- n = screw speed, (rpm)
- $P_0 = X_0 F_r + Y_0 F_a$ (3.39b)
- $P_1 = 1.5 F_a - 0.1 F_r$ (3.39c)

- $X_0 = 1$ and $Y_0 = 0.5$
- ν = operating kinematic viscosity (mm²/sec = cS).

It should be noted that any pre-load built into the nut should be included when estimating the load used in all the relevant equations in Section 3.5.

The mathematical approach can equally be applied to rolling element bearings.

3.6 **Dynamic elastic theory**

The components of a ballscrew drive system have inertial and elastic properties. The laws which govern their behaviour are the laws of elasticity and the laws of motion. There are many ways that these laws can be applied to mechanical systems to derive mathematical models. Four methods have been applied in this investigation:-

1. one using a hybrid modelling technique implemented in SIMULINK,
2. a second based on wave theory
3. the third method was based on generalised eigenvalue theory. The above methods are developed in detail in Chapter 4.
4. The fourth method is a series solution technique which deals with the case where one part of the system moves relative to the rest causing changes in the dynamic characteristics of the system. This novel approach to ballscrew dynamics is covered in Chapters 4 and 5.

The moving mass models consider various energy dissipation mechanisms. The energy flows calculated by these models can be used as energy sources in the model which deals with the thermal behaviour of a ballscrew system.

The next Chapter takes a general overview of the methods which can be used to develop dynamic models of ballscrews. It goes on to demonstrate a general eigenvalue method which can be used to investigate various characteristics of ballscrew driven systems.

Chapter 4 - DYNAMIC MODEL – GENERAL CONSIDERATIONS

A ballscrew drive system is made up of several mechanical components, see Figure 4.1. These are solid bodies made of materials which deform elastically under applied forces. They also have the property of resisting changes in their state of motion which is known as inertia. The system also includes an electric motor governed by a controller. In order to model the mechanical parts it is necessary first to set up equations of motion which are based on the laws of physics to which such mechanical components are subject. These are the laws of elasticity first enunciated by Hooke [**84**] and the laws of motion formulated by Newton [**85**]. There are several ways that these laws can be applied to mechanical systems in order to investigate their dynamic properties.

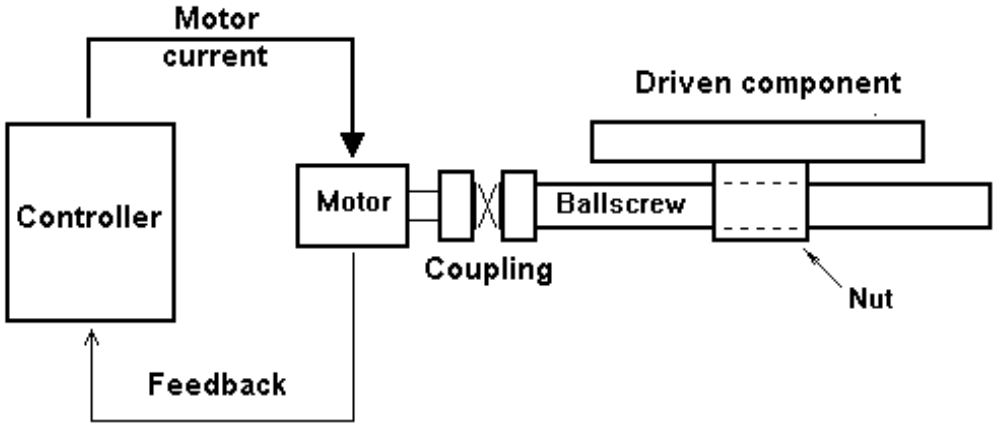


Figure 4.1 - *A feed drive system*

In this investigation work has been done under four broad headings

1. a **hybrid modelling approach**,
2. a **wave theory approach**,
3. a **generalised eigenvalue approach** and
4. the **moving mass models**.

The first three are developed in this Chapter and the moving mass models are dealt with in detail in Chapters 5 and 6.

There are two ways of looking at the bodies of which a mechanical system is made. One is the "continuous matter" approach and the other is the "discrete matter" or "lumped mass" approach.

In the **"continuous matter" approach** the mass is considered to be distributed throughout the volume occupied by the body in a smooth manner, so that at any point in the body a mass per unit volume (density) ρ can be defined. The mass δm of any arbitrarily small volume δv is thus $\rho.\delta v$. The elastic properties of the material are defined by its elastic moduli, which are the stress

required to produce a unit strain. The equation of motion of such an arbitrarily small volume is derived which normally gives rise to a partial differential equation. The solution of such an equation often is a set of waves. This approach is discussed in Section 4.1.

The **"discrete matter" approach** can involve modelling the mechanical system under consideration as a set of "lumped masses" and springs, see Figure 4.2. In other words the mass or inertia in the system has to be considered to act at a set of discrete points in a multi-dimensional space. These point masses are connected by a set of "springs" in which the stiffness of the system is considered to act. The points at which the masses are considered to act are commonly called **nodes** and the springs which connect them **elements**. The discrete matter approach is in essence a finite element method. Therefore the behaviour of the system can be represented by a finite number of differential equations which are derived by considering the equations of motion of each element of the system. The solutions of such sets of differential equations lend themselves naturally to matrix methods. This approach is used in Sections 4.2 and 4.3.

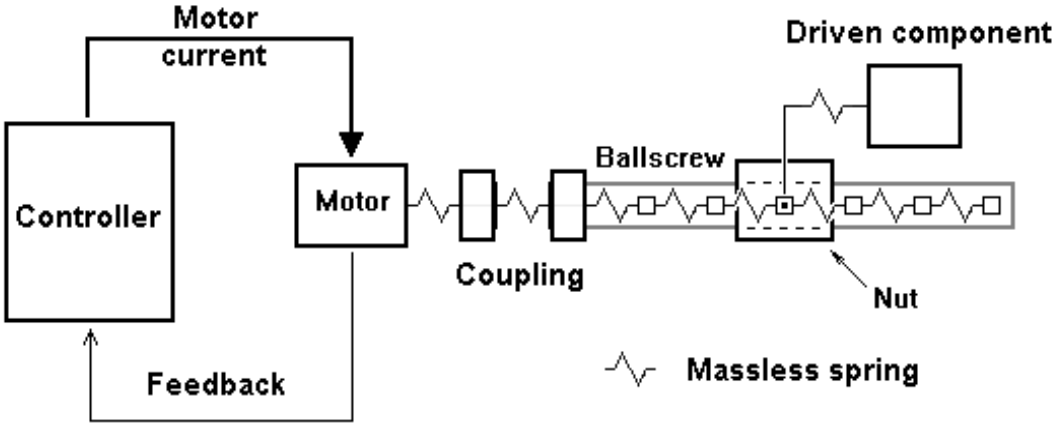


Figure 4.2 - *A ballscrew drive system - discrete matter approach*

4.1 Continuous matter approach – wave solutions

The analysis of dynamic systems using a continuous matter approach gives rise to differential equations. These can be sub-divided into **homogeneous equations**, where some function based on the partial derivatives of the variables involved (one or more spatial dimensions and time) is equal to zero, and **inhomogeneous** or **non-homogenous equations** where the function involving derivatives equals a function of time which in general is non-zero. In a mechanical system, this latter function expresses the external forces on the system and is commonly known as the "forcing term". Solving the homogeneous equation with the appropriate boundary conditions gives an indication of how a system will behave after it has been set in motion and then subject to no further force. This will often show a tendency of the system to vibrate at one or more

"natural frequencies". Solving the non-homogeneous equation gives an indication of how a system will respond to a particular set of forces.

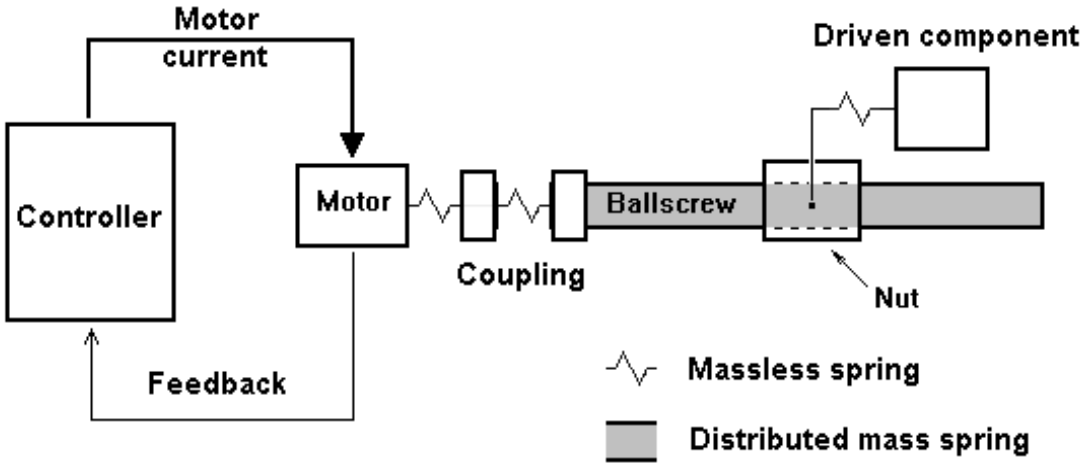


Figure 4.3 - *A ballscrew drive system - continuous matter approach*

Some parts of a ballscrew system can be modelled conveniently using the "lumped mass" approach because the mass, (particularly the rotational or tilt inertia), is concentrated close to the centre of a particular component. A gear or drive belt pulley serves as an example of such a component. (Also the distributed mass behaviour of such components can be very complex and very difficult to analyse in detail.) The component which is least suited to this approach is the ballscrew itself which is often the longest component of a ballscrew system, see Figure 4.3. The distributed mass behaviour of slender bodies can be represented by fairly simple equations [**86**]. It was therefore thought desirable to have an element which would represent closely the behaviour of the screw using a distributed mass approach. The remainder of the system would be modelled using lumped mass elements connected by spring/damper elements.

One method of solving the non-homogeneous equations of motion is the **forced-damped method**, in which transfer matrices are used to calculate the steady-state response to a set of exciting forces. Pestel and Leckie [**63**] constructed the transfer matrices for distributed mass and stiffness elements. This method enables a smaller number of elements to be used than would be needed to get a good model using the lumped mass approach. The forced-damped response method needs the frequency of excitation to be specified because they contain terms which are dependent on frequency. It would be useful to have an element which can respond to any form of excitation. Wave theory was used to derive an element with distributed mass and stiffness which could be used with any input excitation.

The ballscrew has to transmit thrust loads and torsional loads (torque) as the nut is driven along the screw. Motion in the axial direction is the simplest to model mathematically and so it was considered first.

Considering a shaft with cross sectional area A, made from a material with density ρ and Young's modulus E, which lies along the x axis. Let u represent the axial motion of a small element at a point x along the axis. It can be shown that this motion is governed by a wave equation shown by Coulson [87]:-

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2}, \tag{4.1}$$

$$\text{where } c = \sqrt{\frac{E}{\rho}}, \text{ and } t = \text{time} \tag{4.2}$$

The general solution is a superposition of a wave u_f travelling forwards with a velocity c and a wave u_b travelling backwards with the same speed:-

$$u(x,t) = u_f(x - ct) + u_b(x + ct) \tag{4.3}$$

Torsional movement of a shaft with a shear modulus of G is represented by a similar set of equations summarised by De Silva [88] where:-

$$c = \sqrt{\frac{G}{\rho}} \tag{4.4}$$

Considering the axial strain of the shaft at any point along its length, the axial tension T is given by:-

$$T = EA\frac{\partial u}{\partial x} \tag{4.5}.$$

Introducing the dummy variables $\xi_f = x - ct$ and $\xi_b = x + ct$ then

$$\frac{\partial u_f}{\partial x} = \frac{du_f}{d\xi_f}\frac{\partial \xi_f}{\partial x} = \frac{du_f}{d\xi_f} \quad \text{and} \quad \frac{\partial u_b}{\partial x} = \frac{du_b}{d\xi_b}\frac{\partial \xi_b}{\partial x} = \frac{du_b}{d\xi_b} \tag{4.6a), (4.6b)}$$

$$\frac{\partial u_f}{\partial t} = \frac{du_f}{d\xi_f}\frac{\partial \xi_f}{\partial t} = -c\frac{du_f}{d\xi_f} \text{ and } \frac{\partial u_b}{\partial t} = \frac{du_b}{d\xi_b}\frac{\partial \xi_b}{\partial t} = c\frac{du_b}{d\xi_b} \tag{4.7a), (4.7b)}$$

from which it follows that

$$T = \frac{EA}{c}\left(-\frac{\partial u_f}{\partial t} + \frac{\partial u_b}{\partial t}\right) \tag{4.8}.$$

It was therefore decided to design an element based on velocity waves to model the behaviour of the screw.

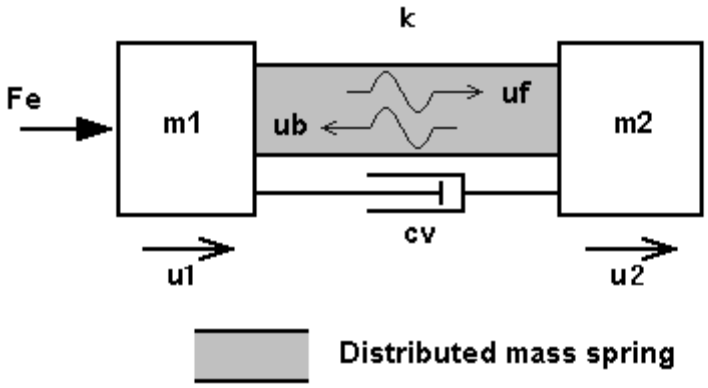


Figure 4.4 - *A test model made of two masses connected by a continuous matter spring*

To test this element, a simple system consisting of two masses m_1 and m_2 connected by such an element was constructed using Simulink, (see Figure 4.4). The first mass could be subject to an external force, F_e. A viscous damper parallel to the spring was included to prevent the system going into a state of perpetual oscillation. The force on the mass was determined by including three terms:-

- the external force
- the force from the forward and backward velocity waves using equation (4.8)
- the force from a viscous damper determined by the difference between the velocity of the two masses multiplied by its damping coefficient.

$$F_e + \frac{EA}{c}\left(-\frac{\partial u_f}{\partial t} + \frac{\partial u_b}{\partial t}\right) + c_v\left(\frac{du_2}{dt} - \frac{du_1}{dt}\right) = m_1 \frac{d^2 u_1}{dt^2} \tag{4.9}$$

where c_v is the damping coefficient of the viscous damper.

The acceleration was calculated by applying the force to the mass. This was integrated to give the velocity of the first mass

$$\frac{du_1}{dt} = \int \frac{d^2 u_1}{dt^2}.dt \tag{4.10}.$$

The magnitude of the forward wave was then given by using the fact that

$$\frac{\partial u}{\partial t} = \frac{\partial u_f}{\partial t} + \frac{\partial u_b}{\partial t} \tag{4.11}$$

and taking into account the backward wave coming from the second mass.

The transmission of the forward wave from the first mass to the second was simulated by using a delay block whereby the wave arrived at the second mass Δt after it left the first. The time delay is determined from the wave velocity and the length l of the shaft:-

$$\Delta t = \frac{l}{c} \tag{4.12}.$$

The motion of the second mass was derived as for the first one, except that an external force was not applied. The magnitude for the backward wave was calculated from the velocity of the second mass taking into account the forward wave coming from the first mass.

$$\frac{EA}{c}\left(-\frac{\partial u_f}{\partial t}+\frac{\partial u_b}{\partial t}\right)-c_v\left(\frac{du_2}{dt}-\frac{du_1}{dt}\right)=m_2\frac{d^2u_2}{dt^2} \tag{4.13}$$

$$\frac{du_2}{dt}=\int\frac{d^2u_2}{dt^2}.dt \tag{4.14}$$

The displacement of each mass was calculated by integrating the velocity of the respective mass.

$$u_{1,2}=\int\frac{du_{1,2}}{dt}.dt \tag{4.15}$$

In addition to calculating the motion of the masses, the energy balance of the system was calculated by summing the following terms:-

- the kinetic energy of the first mass using "½ mass × velocity^2"
- the total energy of the spring by integrating the nett power supplied to the spring by both masses. Since the internal motion of the various parts of the spring were not computed, it was not possible to separate the kinetic energy from the potential energy
- the kinetic energy of the second mass using "½ mass × velocity^2"
- the energy dissipated by integrating the product of the force in the damper and the relative velocity of the masses which it connects.

$$\begin{aligned}E_t=m_1\left(\frac{du_1}{dt}\right)^2+\int\left\{\frac{EA}{c}\left(-\frac{\partial u_f}{\partial t}+\frac{\partial u_b}{\partial t}\right)_1\frac{du_1}{dt}-\frac{EA}{c}\left(-\frac{\partial u_f}{\partial t}+\frac{\partial u_b}{\partial t}\right)_2\frac{du_2}{dt}\right\}.dt\\+m_1\left(\frac{du_2}{dt}\right)^2+\int c_v\left(\frac{du_2}{dt}-\frac{du_1}{dt}\right)^2.dt\end{aligned} \tag{4.16}$$

where E_t is the sum of the energy in the system and the dissipated energy.

The model is illustrated in Figure 4.5. The values of the model parameters were kept to small round numbers to make the results easy to check by hand calculation. An example is given in Appendix 4.1. A force of 1 N for the first second of motion and zero thereafter, which resulted in a pattern of motion whose detailed characteristics are described in the appendix. It was found that the damped natural frequency and the logarithmic decrement predicted by the model were very close to what would be expected. The value for the energy in the spring checked out against the kinetic energy of the spring calculated on a "½ mass × velocity^2" basis. When the system settled down there was a continuous small interchange of energy between the masses and

the spring at a frequency higher than but not obviously related to the natural frequency. More research should be performed in order to determine the causes of this phenomenon.

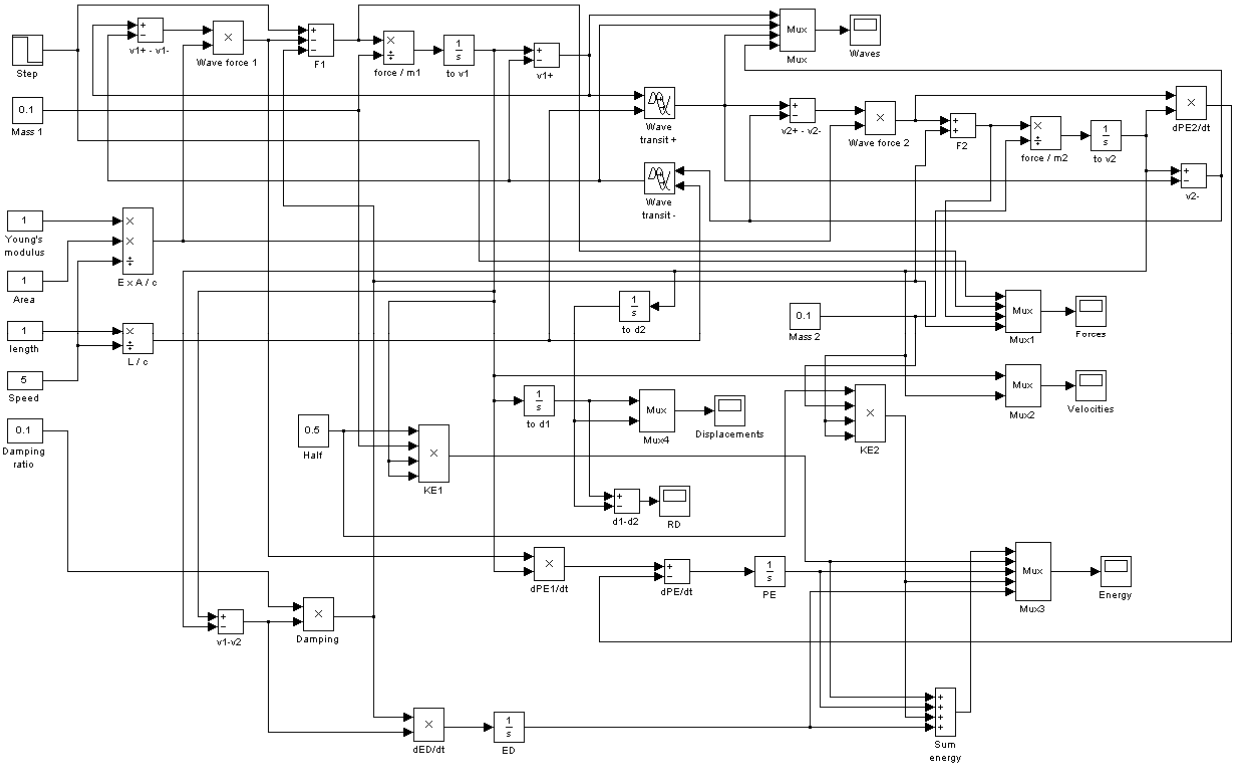


Figure 4.5 – *A model based on wave theory of two masses connected by a spring*

Therefore using velocity, rather than displacement waves, it proved possible to make sensible predictions about simple "mass and spring" models, although some problems remain to be solved.

A more fundamental difficulty arose when considering the lateral degrees of freedom of a beam, when beam flexure occurs. The equation governing flexural vibration of a thin beam whose section has a second moment of area *I* is:-

$$\rho A \frac{\partial^2 y}{\partial t^2} + \frac{\partial^2}{\partial x^2}\left(EI \frac{\partial^2 y}{\partial x^2} \right) = f(x,t) \tag{4.17}$$

where *y* represents movement of the centre of the beam normal to the axis and *f(x,t)* the forcing term [**86**]. For beams of constant section this simplifies to:-

$$\rho A \frac{\partial^2 y}{\partial t^2} + EI \frac{\partial^4 y}{\partial x^4} = f(x,t) \tag{4.18}$$

This equation is fourth order in x and second order in t and so cannot be solved by the simple wave equation. No simple way could be seen around this difficulty and so the line of investigation was abandoned.

4.2 Lumped mass approach – matrix solutions

The "discrete matter" approach entails representing the behaviour of the system at a finite set of nodes by an equation of motion for each node. The mass of such systems can be represented by a consistent mass matrix $\mathbf{M}^e$ defined by:-

$$\mathbf{M^e} = \int \rho \mathbf{N}^T \mathbf{N}.dv \tag{4.19}$$

where ρ is the density, $\mathbf{N}$ is a matrix representing the shape of the system and v is the volume. This can be simplified by using direct lumping, that is modelling the mechanical components as a set of "lumped masses" in which the mass or inertia in the system acts at the nodes. This simplification entails significant computational advantages in calculations which use the inverse of the mass matrix $\mathbf{M}^{-1}$ [**141**] The methods developed in Chapters 5 and 6 make extensive use of the inverse of the mass matrix, so this simplification has proved useful. The approach has been tested against classical methods and found to give satisfactory results, (see Table 4.2 and Figures 4.10 to 4.13). The flexibility inherent in the components is modelled by springs which connect the point masses together. Damping effects can be modelled by including a set of dampers "in parallel" with the springs and a differential equation can be written for each node in the system.

Holroyd et al [**89**] applied this method to modelling the Y axis drive of a Beaver VC35 CNC milling machine, see Figure 4.6.

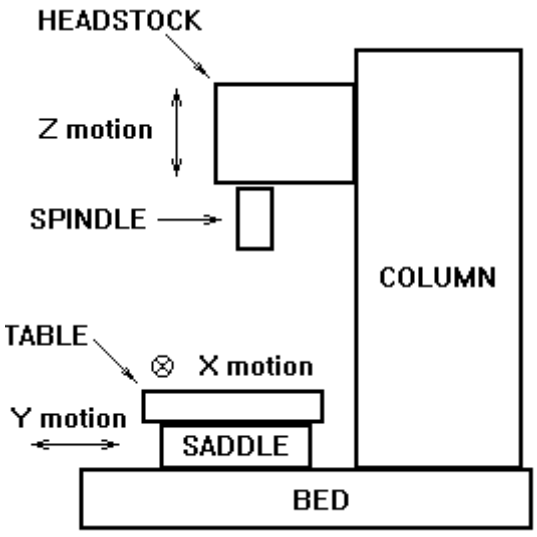


Figure 4.6 - *Beaver VC35 CNC milling machine*

This fixed vertical column CNC milling machine has a saddle which moves towards and away from the column in the Y direction, carrying a table which moves sideways in front of the column in the X direction. Z motion is achieved by moving the spindle up and down on the front face of the column.

The drive for the Y direction consists of a DC electric motor which drives a ballscrew via a belt drive, the ballscrew then drives the saddle, (see Figure 4.7).

The ballscrew is considered as comprising N elements of equal length. The drive is then modelled as follows:-

- The rotational inertia of the drive motor is represented by a single mass (J_m);
- The motor shaft is represented by a torsional spring (k_{ms});
- The rotational inertia of the driving (motor) pulley is depicted by a single mass (J_{p1});
- The flexibility of the belt which transmits the driving torque from the motor to the ballscrew is represented by a torsional spring (k_{bl});
- One element consists of a mass J_{p2} representing the rotational inertia of the driven pulley plus $1/(2N)$ × the rotational inertia of the ball-screw (J_{bs});
- The rest of the ball-screw is modelled as a set of N-1 masses representing $1/N$ of the torsional inertia of the ballscrew and one mass representing $1/(2N)$ of the ball-screw inertia at the free end, connected in series to each other by a set of N torsional springs describing the elastic behaviour of the ballscrew. Each spring has a stiffness of N × the torsional stiffness of the ballscrew itself.
- A mass m_{bm} representing the mass of the ballscrew connected at the iL^{th} node;
- The axial stiffness of the ballscrew k_{ax} is illustrated by a linear spring that connects the ball-screw centre mass to "earth";
- The axial stiffness of the ballscrew nut k_{nut} is depicted by a linear spring linking the ball-screw centre to the saddle, and
- The mass of the table and saddle m_{tab}.

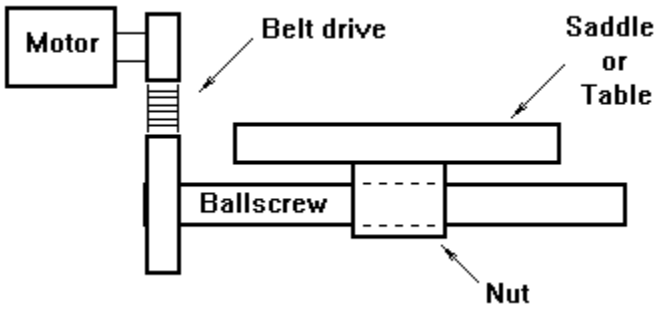


Figure 4.7 - *Beaver Machine – X or Y drive*

The drive ratio which is the number of turns the motor makes while the screw turns once is represented by u, and the ballscrew ratio (see Chapter 3, equation 3.1) is represented by R.

A schematic representation of the model for the mechanical transmission of the CNC machine tool axis drive is presented in Figure 4.8 and the numerical values of parameters are shown in Appendix 4.2.

If there are N_n nodes in the system, in principle it is possible to connect each node with the other N_n-1. Let the masses be represented by m_i, the stiffness of the spring connecting m_i to the jth node be $k_{s\,ij}$ and the damping coefficient of the corresponding damper be $c_{d\,ij}$. If the dissipitave behaviour can be modelled by viscous damping, where the force generated by the damper is proportional to the relative velocity of its two ends, then a linear differential equation for the displacement x_i of each node can be written for the ith node by applying Newton's second law of motion to the mass:-

$$m_i\ddot{x}_i = \sum_{j\neq i}\left(k_{s\,ij}(x_j - x_i) + c_{d\,ij}(\dot{x}_j - \dot{x}_i)\right) + f_i(t) \qquad (4.20),$$

where $f_i(t)$ is the forcing term. Rearranging this equation gives:-

$$m_i\ddot{x}_i + (\sum_{j\neq i} c_{d\,ij})\dot{x}_i - \sum_{j\neq i} c_{d\,ij}\dot{x}_j + (\sum_{j\neq i} k_{s\,ij})x_i - \sum_{j\neq i} k_{s\,ij}x_j = f_i(t) \qquad (4.21).$$

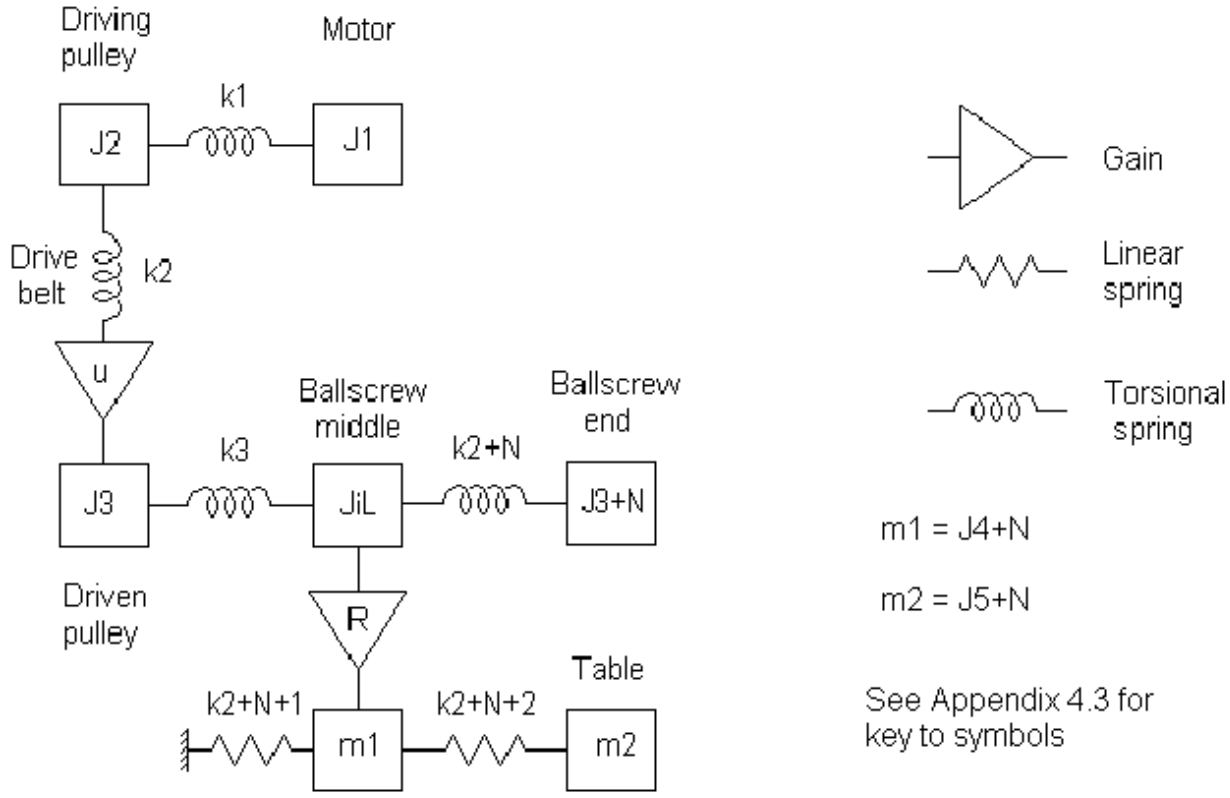


Figure 4.8 - *Schematic representation of the mechanical elements of the drive model*

If the coefficients of x_i are replaced as follows:-

$$K_{ij} = \sum_{j \neq i} k_{s\,ij} \quad \text{for} \quad j = i \quad \text{and} \quad K_{ij} = -k_{s\,ij} \quad \text{for} \quad j \neq i \tag{4.22a}$$

and the coefficients of $\dot{x}_i$ are replaced as follows:-

$$C_{ij} = \sum_{j \neq i} c_{d\,ij} \quad \text{for} \quad j = i \quad \text{and} \quad C_{ij} = -c_{d\,ij} \quad \text{for} \quad j \neq i \tag{4.22b}$$

equation (4.21) can be replaced by:-

$$m_i \ddot{x}_i + \sum C_{ij} \dot{x}_j + \sum K_{ij} x_j = F(t) \tag{4.23}.$$

When the equations for all of the N_n nodes are put together they form the matrix equation:-

$$\mathbf{M}\frac{d^2\mathbf{x}}{dt^2} + \mathbf{C}\frac{d\mathbf{x}}{dt} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \tag{4.24}$$

For details of how each term of the mass and stiffness matrices are derived from the physical parameters of the system see Appendix 4.3. The remainder of this Chapter and Chapters 5 and 6 deal with solving this equation in a variety of ways relevant to the dynamics of ballscrew driven systems.

4.3 **Dynamic model – a generalised eigenvalue approach**

The amplitude of the vibration response of an electro-mechanical system to various stimuli is determined to a large extent by the level of damping. Damping can take several forms (viscous, hysteresis, friction etc.) [**62**], but only viscous damping is a linear phenomenon and therefore amenable to simple mathematical analysis. The response of damped systems can be studied by a "forced damped" approach [**63**] in which various stimuli (for example out-of-balance forces) can be applied to a system. A picture can be built up of the vibration characteristics of the system by applying such forces at a variety of frequencies.

It has been thought useful to extend the eigenvalue method so an attempt has been made to determine the damped natural frequencies with their associated modes using a generalised eigenvalue method. This avoids having to analyse the system with a variety of stimuli over a possibly extended frequency range. The method of approach adopted here is to seek to understand the vibration characteristics of a system in two stages:-

1. to determine the natural frequencies and modes of vibration which would be expected if there were no damping
2. to investigate how damping modifies this behaviour.

Using the MATLAB/Simulink approach it is possible to model "damping" in the form of Coulomb friction, viscous damping and hydrodynamic drag, and also to model the backlash in the ballscrew nut [**90**]. Strictly speaking the method used here can only take into account linear

effects, that is viscous damping. Various researchers have devised means of deriving viscous damping coefficients which are equivalent to various non-linear damping phenomena in a restricted set of circumstances. Tan and Rogers [**91**] have come up with a method of deriving a coefficient of viscous damping which is equivalent to Coulomb friction. Their approach can also be applied to other forms of damping. Careful use of these methods can extend the scope to which this approach is applicable.

4.3.1 **An eigenvalue approach – undamped case**

In order to develop the method, the system without damping has been considered first, then the method has been extended to include viscous damping.

If there is no damping, a system will vibrate at certain frequencies with little stimulus [**92**]. A **natural frequency** of a system is a frequency at which such free vibration takes place.

A vibration **mode** is a characteristic pattern assumed by a system in which the motion of every particle is simple harmonic with the same frequency [**92**]. The **mode shape** can be represented by plotting on a graph the relative amplitude of the different points taking part in the vibration mode. The mode shape gives an idea of where in the system the greatest vibratory activity may be seen.

A common approach to determining the undamped natural frequencies and mode shapes is the eigenvalue method [**93-95**].

To use this method it is necessary to model the mechanical system under consideration as a set of "lumped masses" and springs. In other words the mass or inertia in the system has to be considered to act at a set of discrete points in some sort of space. These point masses are connected to one another by a set of "springs" in which the stiffness of the system is considered to act. The behaviour of such a system can be represented by a finite number of differential equations which are derived by considering the equations of motion of each element of the system. These equations can be solved to give the natural frequencies and the mode shapes as will be shown later.

There are as many natural frequencies as there are point masses in the system although not all of them are necessarily distinct. If the motion is unrestrained, there will be one zero frequency mode for each degree of freedom of motion (e.g. axial movement, lateral movement, angle of twist, angle of tilt etc.) of the points in the system. Also there is a mode shape corresponding to each natural frequency.

Writing an equation of motion for each mass and grouping them in matrix form gives

$$\mathbf{M}\frac{d^2\mathbf{x}}{dt^2} + \mathbf{K}\mathbf{x} = \mathbf{f}(t) \tag{4.25}$$

where **M** is the inertia matrix, a diagonal matrix of the form (M_{ii}), **K** is the stiffness matrix of the form (K_{ij}), **x** is the displacement vector and **f**(t) is the forcing term vector. In this case used here as an example the vectors have N+5 elements and the matrices are (N+5) × (N+5), where N is the number of equal length elements into which the ballscrew is divided. However, the equations apply equally well to a system of any finite number of masses.

In the example analysed here, the elements of the mass matrix are defined as follows:

$$M_{11} = J_m + \frac{J_{ms}}{2}, \quad M_{22} = J_{p1} + \frac{J_{ms}}{2}, \quad M_{33} = J_{p2} + \frac{J_{bs}}{2N}, \tag{4.26a}$$

for i = 4 to i = N+2 $$\ldots . \; M_{ii} = \frac{J_{bs}}{N} \; \ldots , \tag{4.26b}$$

$$M_{N+3\,N+3} = \frac{J_{bs}}{2N}, M_{N+4\,N+4} = m_{bm}, \quad M_{N+5\,N+5} = m_{tab} \tag{4.26c}$$

The upper half of the stiffness matrix is assembled as follows

$$\begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} & .. \\ - & K_{22} & K_{23} & K_{24} & .. \\ - & - & K_{33} & K_{34} & .. \\ - & - & - & K_{44} & .. \\ .. & .. & .. & .. & .. \end{pmatrix} = \begin{pmatrix} k_{ms} & -k_{ms} & 0 & 0 & .. \\ - & k_{ms}+k_{bl} & -u\,k_{bl} & 0 & .. \\ - & - & u^2 k_{bl} + N k_{bs} & -N k_{bs} & .. \\ - & - & - & 2N k_{bs} & .. \\ .. & .. & .. & .. & .. \end{pmatrix} \tag{4.27a}$$

for i = 4 to i = N+2, except i = iL, the ballscrew node where the spring representing the nut is attached,

$$\begin{pmatrix} .. & .. & .. & .. \\ .. & K_{ii} & K_{i\,i+1} & .. \\ .. & .. & .. & .. \end{pmatrix} = \begin{pmatrix} .. & .. & .. & .. \\ .. & 2N k_{bs} & -N k_{bs} & .. \\ .. & .. & .. & .. \end{pmatrix} \tag{4.27b}$$

for i = iL

$$\begin{pmatrix} .. & .. & .. & .. & .. & .. \\ .. & K_{iL\,iL} & K_{iL\,iL+1} & .. & K_{iL\,N+4} & K_{iL\,N+5} \\ .. & .. & .. & .. & .. & .. \end{pmatrix} = \begin{pmatrix} .. & .. & .. & .. & .. & .. \\ .. & 2N k_{bs} + R^2 k_{nut} & -N k_{bs} & .. & R k_{nut} & -R k_{nut} \\ .. & .. & .. & .. & .. & .. \end{pmatrix} \tag{4.27c}$$

and

$$\begin{pmatrix} .. & .. & .. & .. \\ .. & K_{N+3\,N+3} & K_{N+3\,N+4} & K_{N+3\,N+5} \\ .. & - & K_{N+4\,N+4} & K_{N+4\,N+5} \\ .. & - & - & K_{N+5\,N+5} \end{pmatrix} = \begin{pmatrix} .. & .. & .. & .. \\ .. & N k_{bs} & 0 & 0 \\ .. & - & k_{nut}+k_{ax} & -k_{nut} \\ .. & - & - & k_{nut} \end{pmatrix} \tag{4.27d}$$

For all other values $$K_{ij} = 0 \tag{4.28}$$

The full matrix is then completed using $K_{ij} = K_{ji}$ (4.29)

Solutions of the form $\mathbf{x} = (x_i e^{j\omega t})$ when $\mathbf{f}(t) = \mathbf{0}$ are tried to determine the undamped natural frequencies of the system. Equation (4.25) now becomes:

$$(\mathbf{K} - \mathbf{M}\omega^2)\mathbf{x} = \mathbf{0} \tag{4.30}$$

Dividing each equation by the square root of its inertia term M_{ii} and replacing $\mathbf{x}$ by $\mathbf{x}' = (\sqrt{M_{ii}} \times x_i e^{j\omega t})$ gives the following eigenvalue equation:

$$(\mathbf{K}_{\text{mod}} - \mathbf{I}\omega^2)\mathbf{x}' = \mathbf{0} \tag{4.31}$$

where $\mathbf{K}_{\text{mod}}$ is derived from $\mathbf{K}$ by dividing its rows and columns by $\sqrt{M_{ii}}$ and $\mathbf{I}$ is the unity matrix. This is an eigenvalue equation.

This eigenvalue equation has a non-trivial solution for $\mathbf{x}'$ when

$$\left|(\mathbf{K}_{\text{mod}} - \mathbf{I}\omega^2)\right| = \mathbf{0} \tag{4.32}$$

In this case $\mathbf{K}_{\text{mod}}$ is real and symmetric and there are standard solutions to such equations available in many mathematics software packages [96]. The number of solutions for ω^2 is equal to the order of matrix $\mathbf{K}_{\text{mod}}$ although not all of them are bound to be distinct. To each value of ω^2 there is a vector called an eigenvector which gives the mode shape of $\mathbf{x}'$ for that frequency. The mode shape in terms of the original $\mathbf{x}$ is obtained by dividing each term by $\sqrt{M_{ii}}$.

The eigenvectors are arbitrary to one scalar factor; therefore only their relative values are determined and the absolute value can be any desired value. For the purposes of presentation, the eigenvectors are normalised such that the sum of the squares of their absolute values equals one. Also because the ball-screw ratio R is so small (0. 0015915 m/rad) for the studied case, any mode shape selected for plotting is set to equivalent ball-screw rotation before plotting. This drive model was incorporated into the MATLAB software. The resulting program (Appendix 4.3) calculates the eigenvalues and eigenvectors and orders them in order of ascending natural frequency.

The results of the sample undamped natural frequency analysis are presented in Table 4.1. The mode shapes are shown in Figure 4.9. The Y-axis represents the normalised vibration amplitude. The nodes or masses are laid out along the X-axis in the order that they are included in the mass matrix. Thus the point at x = 1 represents the torsional motion of the motor, the point at x = 2 represents the torsional motion of the driving pulley, that at x = 3 the torsional motion of the driven pulley and those at x = 4 to 11 the torsional motion of the points along the ballscrew in increasing distance from the driven pulley. The point at x = 12 represents the axial motion of the point at the ballscrew centre at the position along the ballscrew where the nut is

Table 4.1 The sensitivity of vibration behaviour to model parameters (Y-axis drive) [16]

Mode	1	2	3	4	5
Nat. freq.					
Undamped, Hz	0.0	75.1	111.8	524.7	581.6
Damped, Hz	0.0	75.3	111.6	524.8	574.2
Damping, 1/sec	-0.0	-25.7	-36.7	-21.2	-491.8
δnf / δk					
- k_ms	NaN	0.0605	0.0274	0.0171	0.4002
- k_bl	NaN	0.2653	0.1396	0.0000	0.0978
- k_bs	NaN	0.0107	0.0036	0.4730	0.0145
- k_nut	NaN	0.0773	0.1601	0.0032	0.0000
- k_ax	NaN	0.0781	0.1698	0.0049	0.0001
δζ / δk					
- k_ms	0.0	0.1115	0.3564	0.1323	-0.4346
- k_bl	0.0	-1.3340	0.0868	0.0007	-0.0578
- k_bs	0.0	0.0424	-0.0226	-0.5528	-0.0125
- k_nut	0.0	0.2698	-0.6709	-0.1723	-0.0004
- k_ax	0.0	0.4185	-0.2532	0.2160	-0.0003
δnf / δc					
- c_ms	NaN	-0.0000	-0.0000	-0.0000	-0.0000
- c_bl	NaN	0.0082	-0.0053	-0.0000	-0.0262
- c_bs	NaN	-0.0000	0.0000	-0.0000	-0.0000
- c_nut	NaN	-0.0013	0.0015	0.0004	0.0000
- c_ax	NaN	-0.0003	-0.0001	-0.0001	0.0000
- c_brg_ms	NaN	-0.0000	-0.0000	-0.0000	0.0000
- c_brg_bs	NaN	0.0000	-0.0000	-0.0000	0.0000
δζ / δc					
- c_ms	0.0	0.0001	0.0001	0.0016	0.0019
- c_bl	0.0	0.9020	0.7236	0.0001	1.0033
- c_bs	0.0	0.0003	0.0001	0.7386	0.0011
- c_nut	0.0	0.0791	0.2504	0.1925	0.0001
- c_ax	0.0	0.0119	0.0394	0.0435	0.0000
- c_brg_ms	0.0	0.0000	0.0000	-0.0000	0.0000
- c_brg_bs	0.0	0.0011	0.0007	0.0106	0.0000

NaN = not a number, that is, the variable is not defined in these cases.

considered to be, and the point at x = 13 represents the axial motion of the table to which the ballscrew nut is attached. This information is summarised in the left-hand box on Figure 4.9.

The results of the analysis were compared with measured data taken from the milling machine drive operated by a controller. The controller is set up to ensure that the motion of the motor follows a pre-determined pattern. This has the effect of restraining the vibratory motion of the motor. In order to simulate this restraining effect, the inertia of the motor in the model was increased by a factor of 1000. The first mode is, as expected, the "roll mode" at 0 Hz. The second mode, predicted for the resonance frequency of 75 Hz for the Y-axis drive, involves the driven pulley, ball-screw and load bouncing against the motor and driving pulley. The third mode corresponding to 112 Hz involves mainly axial movement. The fourth and fifth modes are quite high in our frequency range for experimental measurement of 1 - 600 Hz.

The results computed by the eigenvalue method are compared with the measured ones further on in Chapter 8.

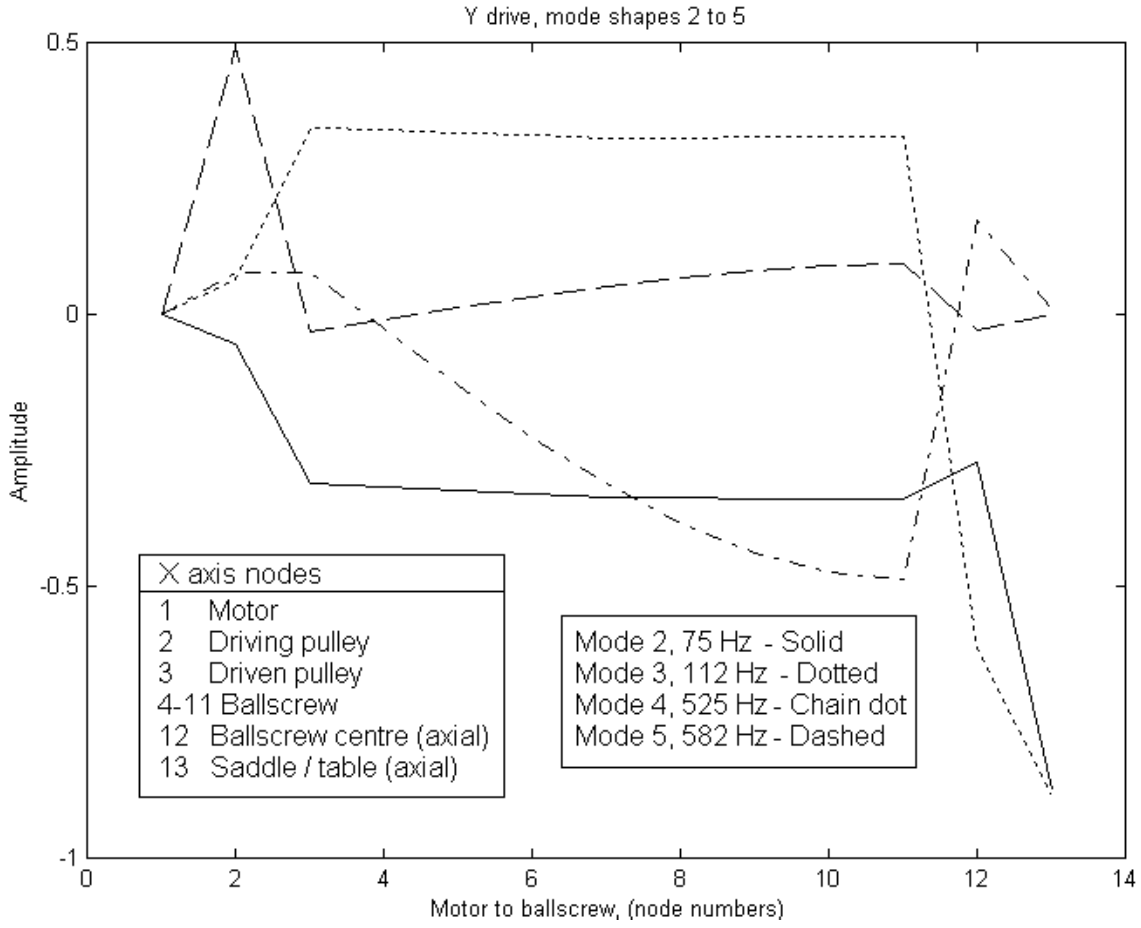


Figure 4.9 - *Undamped mode shapes predicted by the eigenvalue method*

In order to account for transverse vibration, extra degrees of freedom must be accommodated. In its simplest case transverse vibration is a "two dimensional" phenomenon. The displacements

are transverse displacement y and tilt angle θ. The inertia terms are mass and tilt inertia which can be represented by a matrix **M** such as

$$\mathbf{M}=\begin{pmatrix} m & 0 \\ 0 & J \end{pmatrix} \tag{4.33}$$

where m is the mass and J is the tilt inertia.

The stiffness is represented by springs which are 2 × 2 matrices such as

$$\mathbf{K}=\begin{pmatrix} \frac{12EI}{l^3} & -\frac{6EI}{l^2} \\ -\frac{6EI}{l^2} & \frac{4EI}{l} \end{pmatrix} \tag{4.34}$$

where E is Young's modulus, I is the second moment of area and l is the element length.

When assembling a submatrix for a spring into a system stiffness matrix, the submatrix added is typically

$$\begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \mathbf{K} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \frac{12EI}{l^3} & \frac{6EI}{l^2} & -\frac{12EI}{l^3} & \frac{6EI}{l^2} & \cdots \\ \cdots & \frac{6EI}{l^2} & \frac{4EI}{l} & -\frac{6EI}{l^2} & \frac{2EI}{l} & \cdots \\ \cdots & -\frac{12EI}{l^3} & -\frac{6EI}{l^2} & \frac{12EI}{l^3} & -\frac{6EI}{l^2} & \cdots \\ \cdots & \frac{6EI}{l^2} & \frac{2EI}{l} & -\frac{6EI}{l^2} & \frac{4EI}{l} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{4.35}$$

One such submatrix is added in for each spring modelled.

Table 4.2 *Natural frequencies (Hz) predicted by lumped-mass models and by beam theory*

Natural frequency (Hz)		1	2	3	4
Lines on Figures 4.10 to 4.13		solid	dashed	chained	dotted
Free – free	M	62.91	172.9	337.9	556.3
Figure 4.10	C	63.2	174	341	564
Fixed – free	M	9.912	62.03	173.4	338.9
Figure 4.11	C	9.93	62.0	174	341
Supported – supported	M	27.82	111.2	249.8	443.0
Figure 4.12	C	27.8	111	251	446
Fixed – fixed	M	63.06	173.6	339.8	560.4
Figure 4.13	C	63.2	174	341	564

In order to prove the method, MATLAB program "natf_tr1", which uses the eigenvalue approach was produced which gave natural frequencies and mode shapes for the case of free vibration of the ballscrew, (Appendix 4.4 and Figure 4.10). These were compared with the standard results given by Harris [**92**]. It was found that the predicted natural frequencies were very sensitive to the number of elements used and only gave results close to those standard results in the case where a large number of elements was used. The first two modes are, as expected, the "tilt modes" at 0 Hz and are not included on the mode shape plot.

By including an extremely large mass element it is possible to provide a "pinned" restraint and by including a large tilt inertia as well it is possible to model a "built-in" restraint. Using such methods it was possible to model a cantilever beam which also gave results that compared well with the standard results when a large number of elements was used. Comparisons between natural frequencies predicted by the models (M rows) and those predicted by classical methods (C rows) are given in Table 4.2. The mode shapes are illustrated in Figures 4.10 to 4.13.

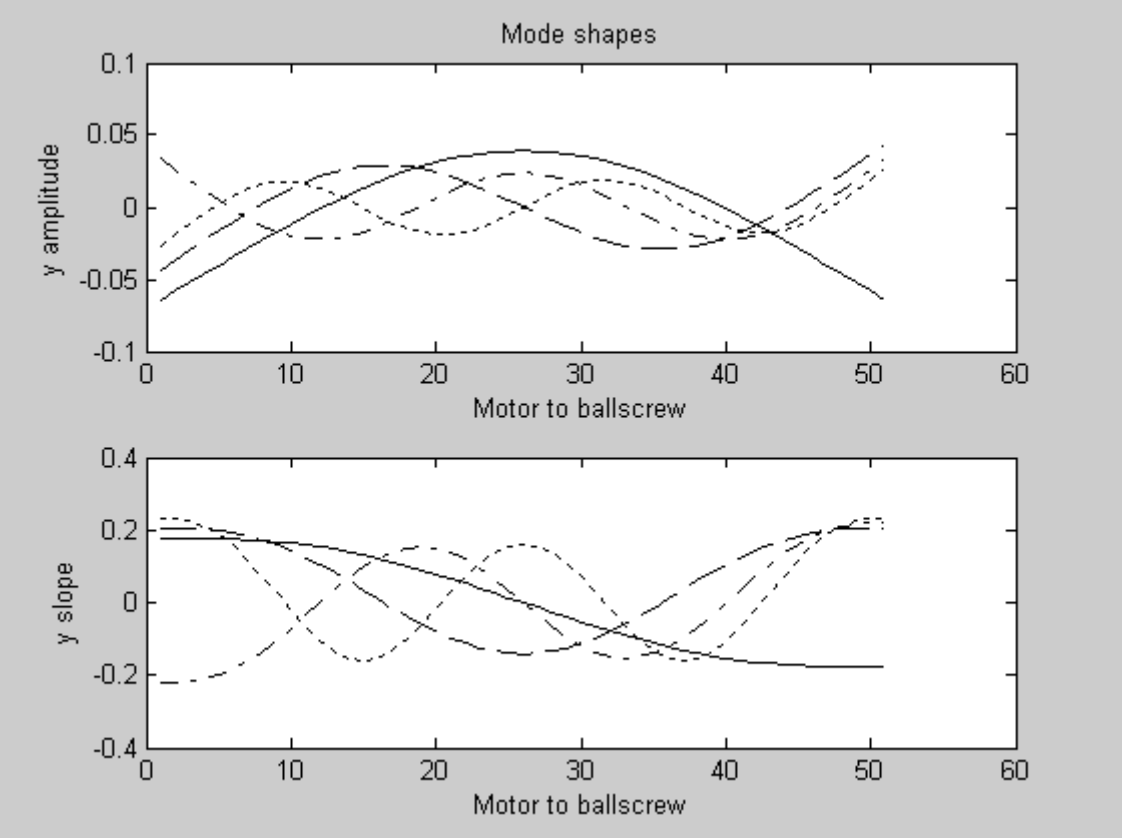


Figure 4.10 - *Natf_tr1.m – Natural frequency and modes shapes – free-free*

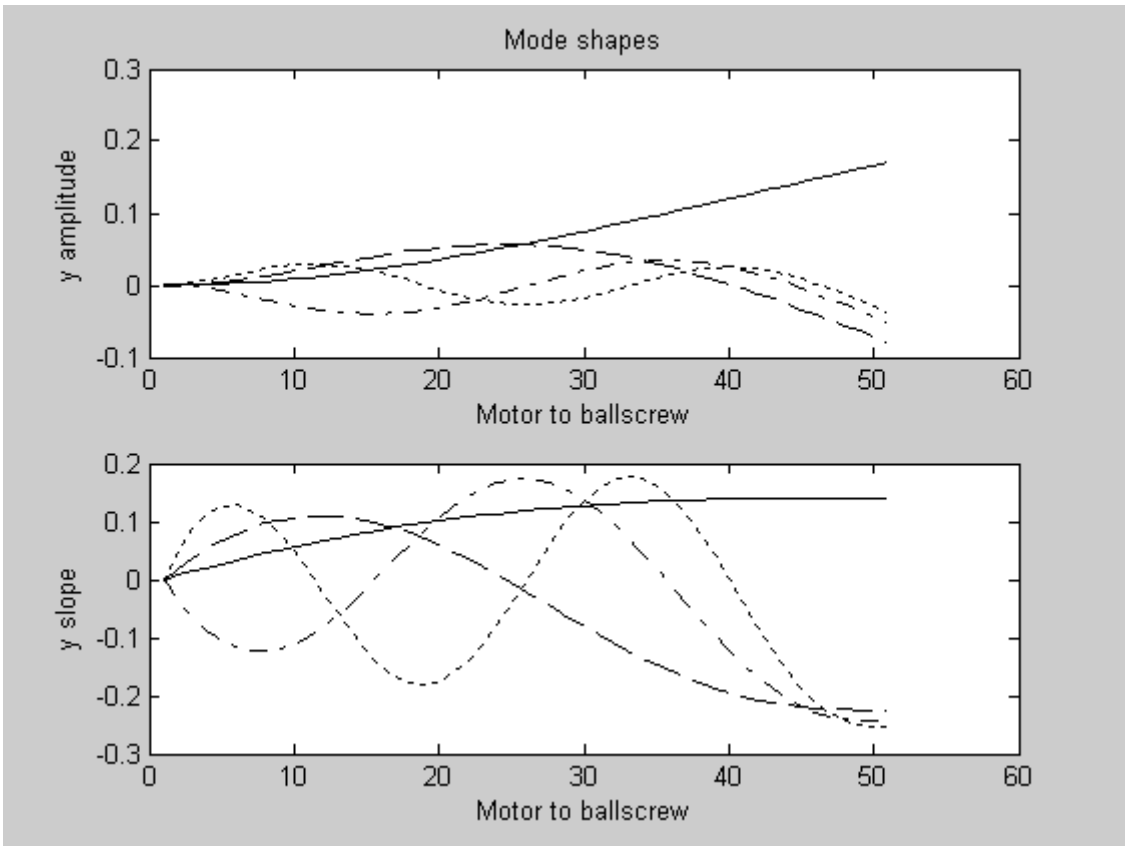


Figure 4.11 - *Natf_tr2.m – Natural frequency and modes shapes – fixed-free*

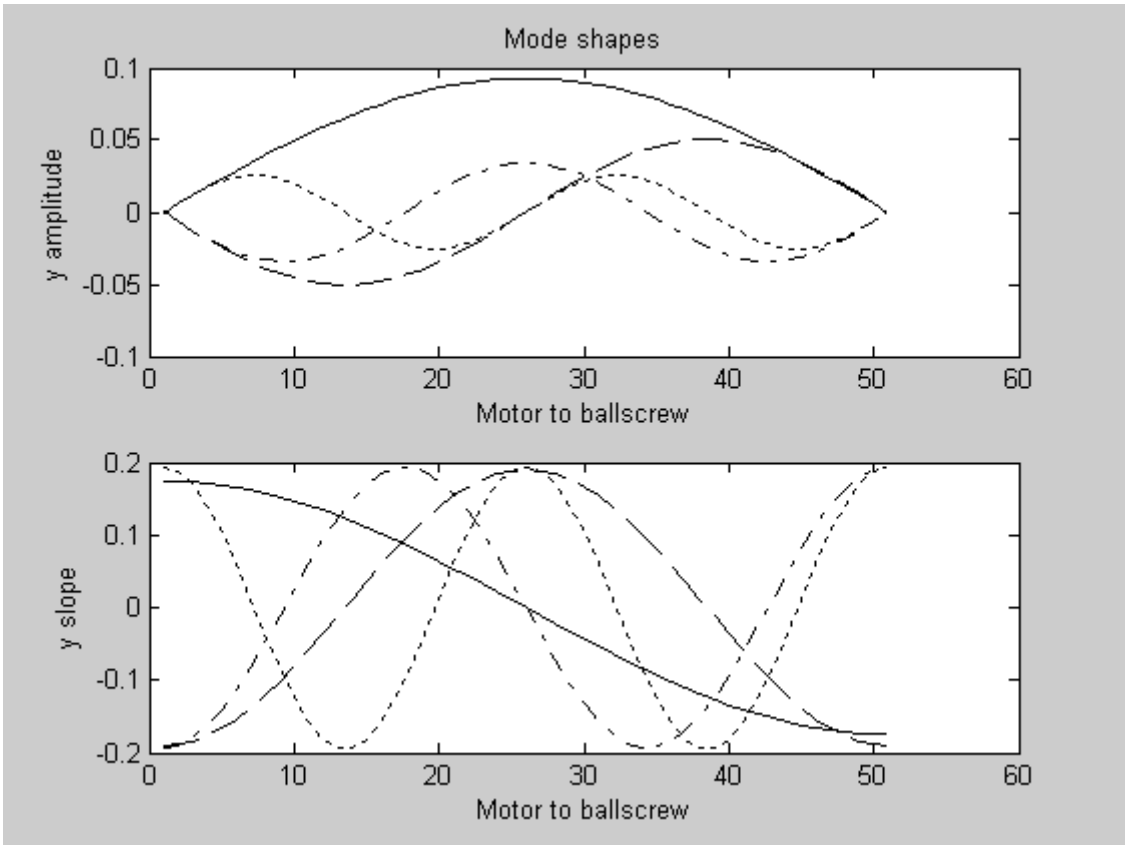


Figure 4.12 - *Natf_tr3.m – Natural frequency and modes shapes – supported-supported*

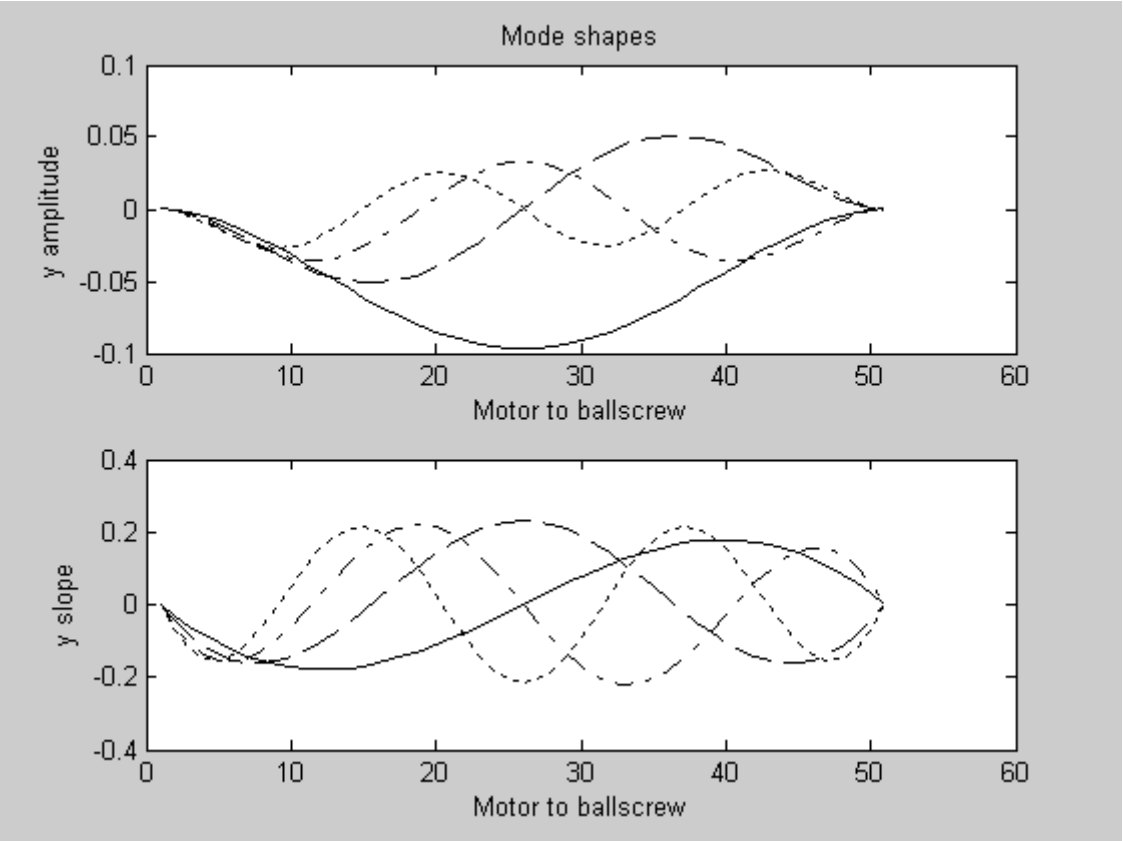


Figure 4.13 - *Natf_tr4.m – Natural frequency and modes shapes – fixed-fixed*

These results validate the method of modelling the screw. The same method of representing the screw is used in Chapter 6 as part of the model representing the transverse vibration behaviour of a ballscrew with a moving mass.

4.3.2 **A generalised eigenvalue approach – damped case**

a) *Theoretical treatment of the linear dissipative effects*

Viscous damping gives rise to a force that is proportional to the relative velocity of two parts of a system. Therefore viscous damping has a linear effect and fits in readily with linear differential equations. It can be introduced deliberately as in the case of a hydraulic damper or "shock absorber" in a vehicle suspension, or it might occur naturally as the result of the dissipative properties of the component's material.

In this investigation, the elements representing shafts are considered as a viscous damper in parallel with a spring as in Figure 4.14.

The force F generated by the spring / damper is given by the relation:

$$F = k \times (x_2 - x_1) + c \times \left(\frac{dx_2}{dt} - \frac{dx_1}{dt}\right) \tag{4.36}$$

where x_1, x_2 = displacements, k = stiffness and c = the damping coefficient.

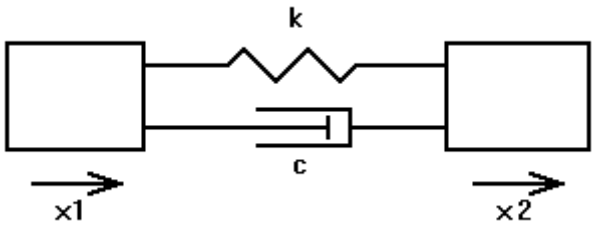


Figure 4.14 - *Representation of viscous damping phenomenon*

b) *Damped natural frequencies and mode shapes*

A **damped natural frequency** of a system can be understood as a frequency of free vibration, that is, it is a frequency at which a system will vibrate after some stimulus with the amplitude of the vibrations dying away with time.

A vibration **mode** is a characteristic pattern assumed by a system in which the motion of every particle is simple harmonic with the same frequency. The **mode shape** of a damped mode can be represented by plotting on a graph or graphs the relative amplitude of the different points taking part in the vibration mode. Phase changes of other than ±180º are expected to occur in the case of a damped mode so the mode shape has to be represented by two sets of numbers. The mode shape shows where in the system the greatest vibratory activity may be seen.

The approach to determining the damped natural frequencies and mode shapes used here is the generalised eigenvalue method [**97**, **98**]. The drives are modelled as detailed in Section 4.3.1 for the undamped case with the additional features as follows:-

- A torsional damper is placed parallel to each torsional spring
- A linear damper is placed parallel to each linear spring
- A torsional damper c_{brg_ms} is placed between the motor inertia and "earth" to represent the viscous drag in the motor bearings
- A torsional damper c_{brg_bs} is placed between the driven pulley inertia and "earth" to represent the viscous drag in the ballscrew pulley end bearing
- A torsional damper is placed between the ballscrew end inertia and "earth" to represent the viscous drag in the ballscrew free end bearing

The model for the mechanical transmission of the CNC machine tool axis drive considering viscous damping is depicted in Figure 4.15.

The generalised eigenvalue method is used for determining the damped natural frequencies and mode shapes. Writing an equation of motion for each mass and grouping them in matrix form gives the same equation as in 4.24. The damping matrix **C**, a square matrix of the form (C_{ij}). Again the vectors have $N+5$ elements and the matrices are $(N+5) \times (N+5)$.

The upper half of the damping matrix is assembled in a manner similar to that used for the stiffness matrix (Equations (4.27a), (4.27b), (4.27c) and (4.27d)) using the damping coefficients c corresponding to the stiffness terms k except that damping coefficients are included to account for the viscous drag in the bearings as follows:

$$\begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & .. \\ - & C_{22} & C_{23} & C_{24} & .. \\ - & - & C_{33} & C_{34} & .. \\ - & - & - & C_{44} & .. \\ .. & .. & .. & .. & .. \end{pmatrix} = \begin{pmatrix} c_{ms} & -c_{ms}+c_{mb} & 0 & 0 & .. \\ - & c_{ms}+c_{bl} & -u\,c_{bl} & 0 & .. \\ - & - & u^2 c_{bl}+Nc_{bs}+c_{brg1} & -Nc_{bs} & .. \\ - & - & - & 2Nc_{bs} & .. \\ .. & .. & .. & .. & .. \end{pmatrix} \quad (4.37a)$$

$$\begin{pmatrix} .. & .. & .. & .. \\ .. & C_{N+3\,N+3} & C_{N+3\,N+4} & C_{N+3\,N+5} \\ .. & - & C_{N+4\,N+4} & C_{N+4\,N+5} \\ .. & - & - & C_{N+5\,N+5} \end{pmatrix} = \begin{pmatrix} .. & .. & .. & .. \\ .. & Nc_{bs}+c_{brg2} & 0 & 0 \\ .. & - & c_{nut}+c_{ax} & -c_{nut} \\ .. & - & - & c_{nut} \end{pmatrix} \quad (4.37b)$$

For all other values $C_{ij} = 0$ (4.38)

The full matrix is then completed using $C_{ij} = C_{ji}$ (4.39).

c) *The generalised eigenvalue method – mathematical details*

The first step in calculating the damped natural frequencies of the system is to try solutions of the form $\mathbf{x} = (x_i\, e^{\omega t})$ when $\mathbf{f}(t) = \mathbf{0}$. In this case the Equation (4.24) yields:

$$(\mathbf{M}\omega^2 + \mathbf{C}\omega + \mathbf{K})\mathbf{x} = \mathbf{0} \quad (4.40)$$

Substituting $\mathbf{x_1} = \omega \mathbf{x}$ in equation (4.40) gives

$$\mathbf{M}\omega\,\mathbf{x_1} + \mathbf{C}\mathbf{x_1} + \mathbf{K}\mathbf{x} = \mathbf{0} \quad (4.41a)$$

or rearranging

$$\omega\,\mathbf{x_1} = -\mathbf{M}^{-1}\mathbf{C}\mathbf{x_1} - \mathbf{M}^{-1}\mathbf{K}\mathbf{x} \quad (4.41b)$$

Substituting $\mathbf{B_1} = -\mathbf{M}^{-1}\mathbf{C}$ and $\mathbf{B_2} = -\mathbf{M}^{-1}\mathbf{K}$ and putting alongside the definition of $\mathbf{x_1}$ yields

$$\begin{aligned} \mathbf{x_1} &= \omega\,\mathbf{x} \\ \mathbf{B_2}\mathbf{x} + \mathbf{B_1}\mathbf{x_1} &= \omega\,\mathbf{x_1} \end{aligned} \quad (4.42a)$$

which in matrix form is

$$\begin{pmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{B_2} & \mathbf{B_1} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{x_1} \end{pmatrix} = \omega \begin{pmatrix} \mathbf{x} \\ \mathbf{x_1} \end{pmatrix} \quad (4.42b)$$

This is an eigenvalue equation. The vectors have 2N+10 elements and the matrices are $(2N+10) \times (2N+10)$.

In this case the matrix is real and not symmetric and there are standard solutions to such equations available in many mathematics software packages [96]. The number of solutions for ω is equal to twice the order of matrices **M**, **C** and **K** although not all of the solutions are bound to be distinct. To each value of ω there is a vector called an eigenvector from which can be extracted the mode shape of **x** for that frequency.

This model was incorporated into MATLAB model "natf_dr3", (Appendix 4.5).

The eigenvalues come in conjugate pairs in the case that the mode is sub-critically damped, (when a sudden application of an external force gives rise to oscillations that die away), therefore a lightly damped model might have up to *N*+5 pairs. The imaginary part is the natural frequency in radians per second and the real part is the coefficient of damping. In order to represent a positive coefficient of damping the real part of the eigenvalue should be negative. If the mode is critically damped or over-critically damped, all the solutions will be real. This means that there is not an oscillatory response to a step input. There are two real "solutions" for each mass element. Since the sub-critical natural frequencies come in conjugate pairs, a sense of mathematical tidiness suggests that the two real solutions also be grouped together. At this stage it is not understood how to pair them up as two solutions of one "mode".

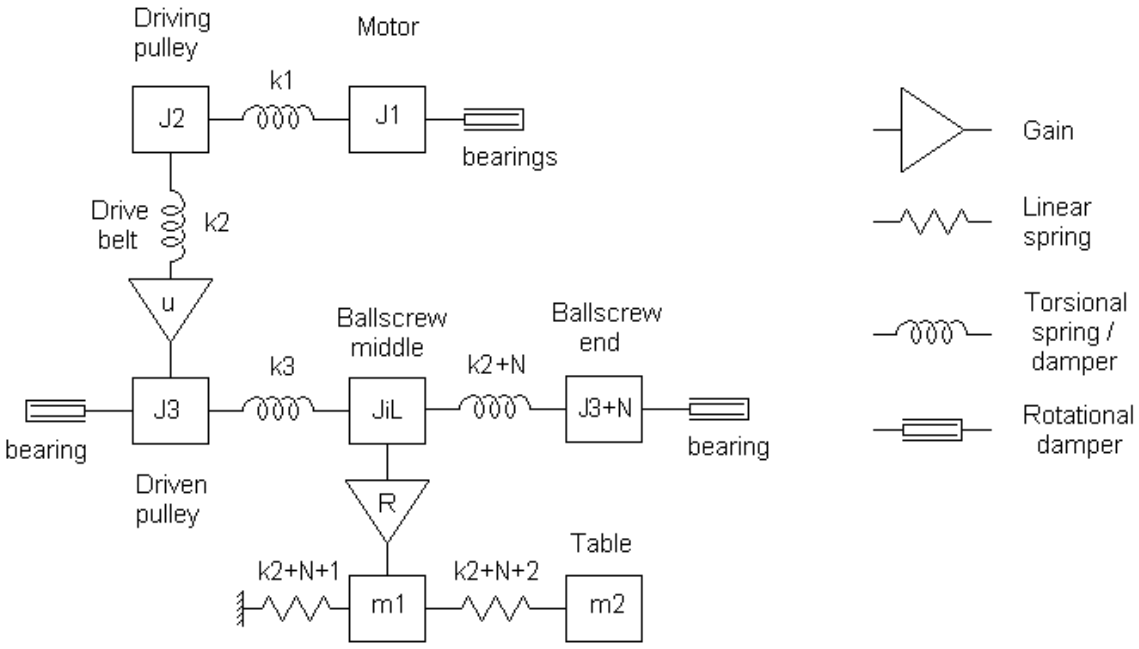


Figure 4.15 - *Model of the mechanical transmission of the CNC machine tool axis drive considering viscous damping*

The eigenvectors are arbitrary to one scalar factor; so only their relative values are determined, the absolute value can be chosen for convenience. For the purposes of presentation the eigenvectors are normalised such that the sum of the squares of their absolute values equals

one. Then any vector selected for plotting is re-normalised such that the term with the largest amplitude has a phase of zero. Finally, because the ballscrew ratio R is so small (0.0015915 m/rad), the mode shape is set to equivalent ballscrew rotation before plotting. The MATLAB program (Appendix 4.5) is similar to the one presented in Section 4.3.1 but incorporating the generalised eigenvalue method. This program also orders the eigenvalues and eigenvectors in order of ascending natural frequency.

The eigenvectors from which the mode shapes are derived are essentially complex numbers. In order to give a visual representation of these, two approaches have been used. In the first case, each mode shape is shown in a Cartesian form with the real component designated "in phase" and the imaginary component "quadrature" (see Figure 4.16). In the other case, the mode shapes are shown in a polar from with the "amplitude" and "phase" being plotted (see Figure 4.17).

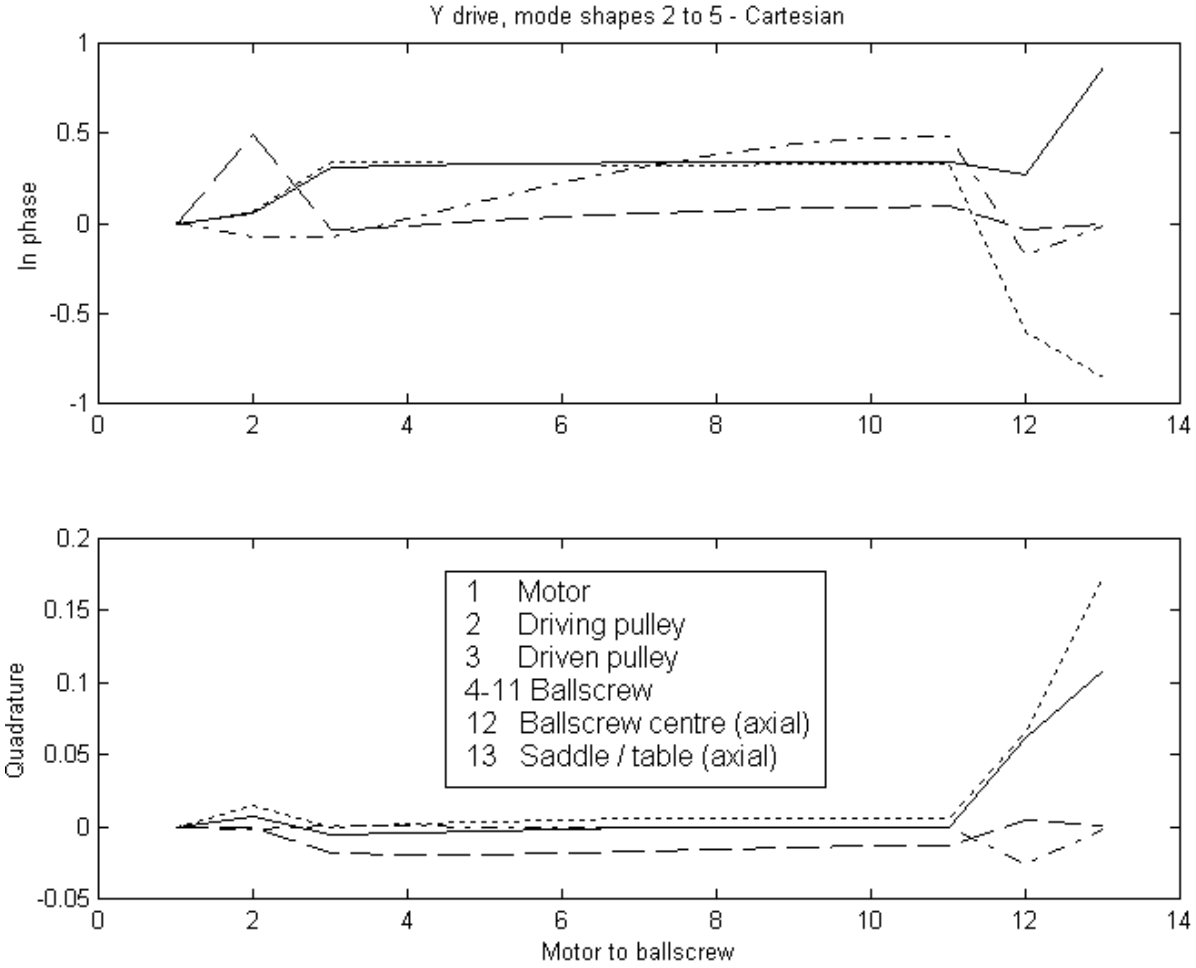


Figure 4.16 - *Damped mode shapes predicted by the generalised eigenvalue method - Cartesian*

It is to be expected that the "in phase" component will compare most directly with the undamped mode shape in the case of light damping. Although in many cases this is so, in some instances the damped mode shapes are close to the undamped mode shape reflected on the

X axis. Then, considering that the eigenvectors on which the mode shapes are based are arbitrary to a scalar factor, it is possible to make them more easily comparable with the undamped mode shapes by multiplying one of the sets by minus one. This difference can arise because the normalisation process used on the complex raw eigenvectors is slightly different from that used on the real eigenvectors which occur in the undamped case.

The polar form (Figure 4.17) is comparable with measurement results. The amplitude of the mode shapes can be compared with vibration readings and the phase can be compared with the phase data generated by many data analysis packages.

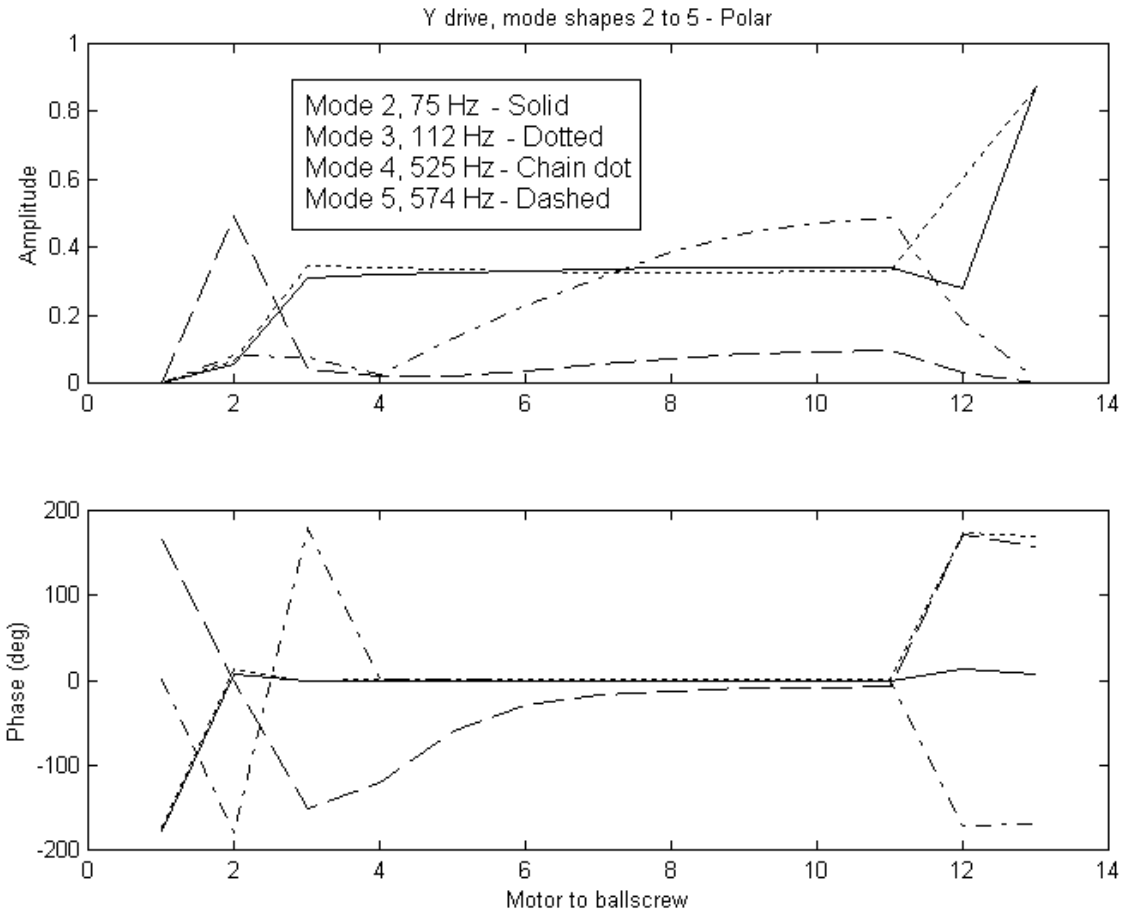


Figure 4.17 - Damped mode shapes predicted by the generalised eigenvalue method – polar

4.3.3 **Sensitivity analysis**

The values for the sensitivity of the various modes to changes in the stiffness and damping characteristics of the machine drives are also derived. This is done by developing programs which give the sensitivity of the system to a 1% change in one of the model's parameters. This enables decisions to be made more easily about, say, where the best place is to increase the damping to reduce the level of a troublesome mode of vibration, or which stiffness should be

changed to move an undesirable natural frequency. Such sensitivity information can be used as an aid to design or fault correction.

These values can also prove useful in adjusting the parameters of a model for a CNC machine tool axis drive, for example that developed by Pislaru [**15**]**,** in order to adjust the simulated results to fit best to the measured ones. An estimate of the amount of change necessary to make the simulation results fit the experimental ones can be easily obtained.

An example of a set of mode shape plots is included in Figure 4.16 and Figure 4.17. A table of sensitivity data is included in Table 4.1.

The MATLAB programs described in this Chapter can be used for the following purposes:

- to determine the natural frequencies and damping factors for the elements of the drive using the generalised eigenvalue method;
- to plot the mode shapes in order of ascending natural frequency;
- to calculate the sensitivity of natural frequency to changes in stiffness, and to changes in damping coefficient;
- to calculate the sensitivity of damping ratio to changes in stiffness, and to changes in damping coefficient.

The next stage of the study is to investigate the effects on the dynamics of the ballscrew drive of the nut moving along the screw. In Chapter 5 the dynamics of the principal degrees of freedom involved in ballscrew action will be dealt with, that is torsional movement of the screw and axial movement of the nut and screw. The effects of non-linear phenomena such as friction and backlash will be included.

Chapter 5 - AXIAL AND TORSIONAL CASE FOR MOVING MASS MODEL

When a ballscrew is being selected for a particular application, the dynamic behaviour is taken into account. One area of concern is whether or not the ballscrew will vibrate excessively because it is running at one of its natural frequencies. One method used commercially for sizing ballscrews to avoid whirling uses a set of criteria which only includes the restraining effects of the ballscrew bearings and takes no account of the nut [77].

In fact the nut has a significant lateral restraining effect, but the effect varies with time as the nut moves. This implies that the natural frequencies of a ballscrew system change with time. This chapter deals with the analysis of ballscrew system dynamics taking into account the changes in stiffness and damping characteristics which occur as the nut travels along the screw. Initially, the analysis focuses mainly on the degrees of freedom involved in the action of the ballscrew, namely the torsional movement of the screw and the axial displacement of the nut. The effects of backlash in the nut are also investigated. Considerations of the transverse degrees of freedom of the screw are analysed in Chapter 6.

This changing of resonant frequency with time can be a dominant factor in the behaviour of a ballscrew system which contains a relatively slender screw. This is because such a screw will have low natural frequencies which might be excited by quite low running speeds.

The classical equations which predict the amplitude of vibrating systems give the "steady state" value [99], that is the level to which the system settles down after any transient response which might arise as the force is applied has died away. For example, consider the simplest such system, a mass *m* mounted to earth by a spring of stiffness *k* with a viscous damper of damping coefficient *c* along side it.

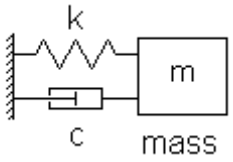


Figure 5.1 - *Simple "mass on a spring" system*

The equation of motion for this system is

$$m\ddot{x} + c\dot{x} + kx = F \tag{5.1}$$

where x is the displacement of m and F is the forcing term. When $c = 0$, the solution of this equation gives rise to harmonic motion with an undamped natural frequency ω_n given by

$$\omega_n = \sqrt{\frac{k}{m}} \tag{5.2}$$

When a small amount of damping is included, the natural frequency is reduced and the amplitude dies away exponentially after the forcing term is removed. The rate of decay of the harmonic motion is proportional to the damping coefficient, thus as *c* gets bigger the motion dies away more rapidly. At the point of **critical damping** the motion tends most quickly to zero. As the damping is increased beyond critical, the motion tends more slowly to zero but with no oscillation. The fraction of critical damping ζ is given by

$$\zeta = \frac{c}{2\sqrt{km}} \tag{5.3}$$

The steady state displacement response x_{st} to a sinusoidal force of $F = F_0 \sin \omega t$ is

$$x_{st} = \frac{\sin(\omega t - \theta)}{\sqrt{(1 - \omega^2 / \omega_n^2)^2 + (\zeta \omega / \omega_n)^2}} \frac{F_0}{k} \tag{5.4}$$

where the phase lag θ is given by

$$\theta = \tan^{-1}\left(\frac{2\zeta \omega / \omega_n}{1 - \omega^2 / \omega_n^2} \right) \tag{5.5}$$

It will be noted that if $\omega = \omega_n$,

$$x_{st} = \frac{F_0}{\zeta k} \sin(\omega t - \theta) \text{ where } \theta = \tan^{-1}(2\zeta) \tag{5.6}$$

which gives a value of infinity for x_{st} if there is no damping. However infinite amplitude implies infinite energy which will take an infinite time to achieve with a finite force pushing it. It is therefore necessary for a system to be excited for a "significant time" at a natural frequency for the vibration amplitude to build up to a level high enough to be detrimental. What a "significant time" is will vary from system to system, and will depend on several factors:-

- how close to the natural frequency is the frequency of the forcing term,
- the amplitude of the forcing term, and
- the level of damping involved.

If the natural frequency changes with time, it may be that the vibration amplitude always remains low enough to be of no detriment to the ballscrew or the machine of which it is part. It was therefore thought worth while to try to model the dynamic behaviour of a ballscrew with a moving nut.

5.1 Dynamics of a ballscrew with a moving nut – the axial case

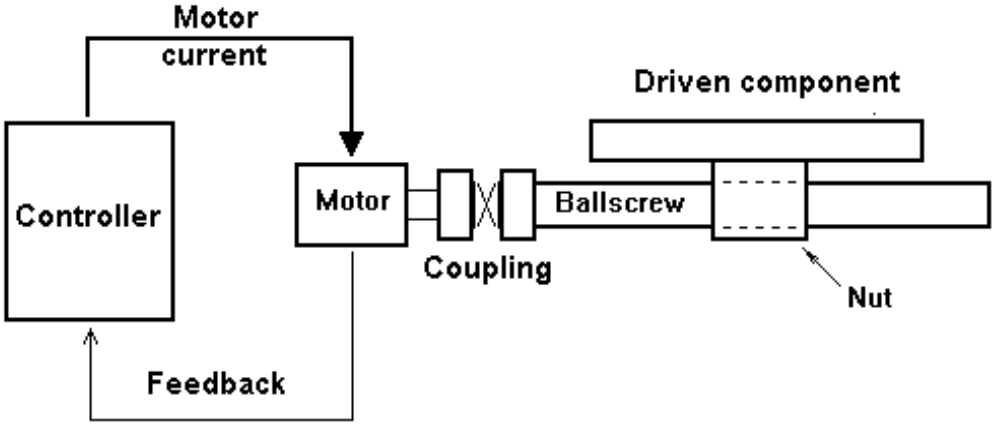


Figure 5.2 - *A ballscrew drive system*

The essential features of a ballscrew in a machine tool drive are shown in Figure 5.2. The drive from the motor is passed to the screw through some mechanism such as a belt drive, a chain drive, gears or a flexible coupling. The first part of the ballscrew proper is therefore an "inertial component", the belt pulley, chain sprocket, driven gear or the driven half of the flexible coupling, see Figure 5.3. The screw is an almost cylindrical component mounted in two bearings and the nut makes contact with the screw at a various positions along the screw. The nut carries an inertial load which represents the part of the machine to which it is attached.

It is common to model such systems as a beam with the bearings giving some defined restraints to the motion of the beam. At first the possibility of an analytical solution in terms of beam theory was considered. However a fundamental problem was that it was natural to split the beam representing the screw into two at the nut. At this point the boundary conditions and continuity are difficult to define since the split point changes with time.

A special method has been devised to overcome this disadvantage. Using this approach, the moving mass is connected to all the other masses in the system, the effect of the moving point of contact being achieved by varying the stiffness between the node representing the nut and the nodes representing the ballscrew. This has the advantage that each degree of freedom included in the matrices always refers to the same physical quantity.

Applying the method in detail, the ballscrew was split into S equal portions thus generating S+1 nodes with x coordinates X_i. The set of these coordinates $\{X_i\}$ form the abscissa for mode shape plots. If the screw is of length l, the first and the S+1th element are considered to be of length $l/2S$ and all the intermediate elements of length l/S and the mass of the screw is considered to act at the nodes. The stiffness is considered to act in the S elements. The bearings are modelled as restraints applied at the appropriate position on the screw.

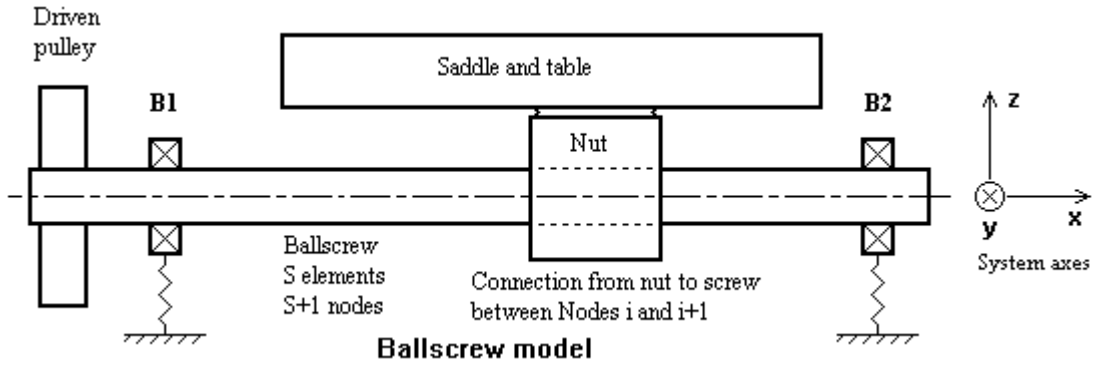


Figure 5.3 - *A typical ballscrew in a machine tool drive*

The case of axial degrees of freedom only is considered first, so that the mathematical method can be developed while keeping the algebraic expressions as simple as possible. In this case the stiffness k of the *i*th element of the ballscrew is given by

$$k = \frac{EA}{X_{i+1} - X_i}, \quad i = 1, 2, ... S+1 \qquad (5.7)$$

where E is Young's modulus for the screw material and A is the screw cross-sectional area.

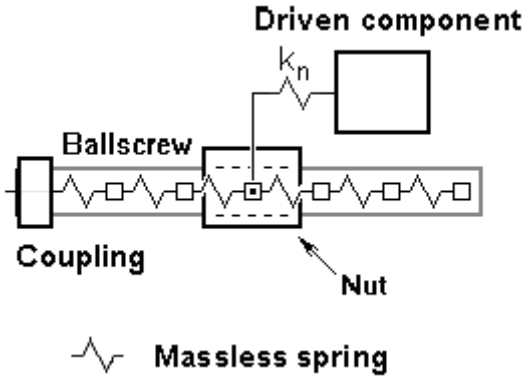


Figure 5.4 - *Ballscrew model*

The nut is modelled as a single node at which the mass is considered to act and the stiffness between the nut and the screw is considered as a single value k_n, (see Figure 5.4). To allow for motion of the nut along the screw, the analysis has been set up so that the nut is considered to be connected to all of the ballscrew nodes. Writing an equation of motion for each node of the ballscrew in turn gives a set of equations as follows.

For node 1: $m_1\ddot{x}_1 - c(\dot{x}_2 - \dot{x}_1) - \xi_1 c_n(\dot{x}_n - \dot{x}_1) - k(x_2 - x_1) - \xi_1 k_n(x_n - x_1) = F_1(t)$ (5.8a)

For node *i*, where $2 \le i \le S$:

$$\begin{aligned} m_i\ddot{x}_i &+ c(\dot{x}_i - \dot{x}_{i-1}) - c(\dot{x}_{i+1} - \dot{x}_i) - \xi_i c_n(\dot{x}_n - \dot{x}_i) \\ &+ k(x_i - x_{i-1}) - k(x_{i+1} - x_i) - \xi_i k_n(x_n - x_i) = F_i(t) \end{aligned} \qquad (5.8b)$$

For node *S*+1:

$$m_{S+1}\ddot{x}_{S+1} + c(\dot{x}_{S+1} - \dot{x}_S) + \xi_{S+1} c_n(\dot{x}_n - \dot{x}_{S+1}) + k(x_{S+1} - x_S) + \xi_{S+1} k_n(x_n - x_{S+1}) = F_{S+1}(t) \qquad (5.8c)$$

Finally for the nut: $m_n\ddot{x}_n + \sum_{i=1}^{S+1} \xi_i c_n(\dot{x}_n - x_i) + \sum_{i=1}^{S+1} \xi_i k_n(x_n - \dot{x}_i) = F_n(t)$ (5.9)

Here k is the stiffness of a single element of the ballscrew, c is the coefficient of damping of the element and ξ_i are factors whose value depends on the position of the nut. Since the nut must be somewhere along the screw, the factors must satisfy the condition

$$\sum_{i=1}^{S+1} \xi_i = 1 \tag{5.10}$$

If the ballscrew nut is at a position X_n between nodes *j-1* and *j*, the effect of the nut is put in the correct place by using values of ξ_{j-1} and ξ_j given by

$$\xi_{j-1} = \frac{X_j - X_n}{l/S}$$
$$\xi_j = \frac{X_n - X_{j-1}}{l/S} \tag{5.11a}$$

For $i \neq j-1$ or $i \neq j$, $$\xi_i = 0 \tag{5.11b}$$

$F_i(t)$ is the external force applied at the ith node.

Put together these equations give rise to a stiffness matrix for the ballscrew system as follows

$$\mathbf{K} = \begin{pmatrix} k+\xi_1 k_n & -k & 0 & \ldots & 0 & 0 & -\xi_1 k_n \\ -k & 2k+\xi_2 k_n & -k & \ldots & 0 & 0 & -\xi_2 k_n \\ 0 & -k & 2k+\xi_3 k_n & \ldots & 0 & 0 & -\xi_3 k_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 2k+\xi_S k_n & -k & -\xi_S k_n \\ 0 & 0 & 0 & \ldots & -k & k+\xi_{S+1} k_n & -\xi_{S+1} k_n \\ -\xi_1 k_n & -\xi_2 k_n & -\xi_3 k_n & \ldots & -\xi_S k_n & -\xi_{S+1} k_n & k_n \end{pmatrix} \tag{5.12}$$

In a similar way a matrix **C** of damping coefficients can be built up.

$$\mathbf{C} = \begin{pmatrix} c+\xi_1 c_n & -c & 0 & \ldots & 0 & 0 & -\xi_1 c_n \\ -c & 2c+\xi_2 c_n & -c & \ldots & 0 & 0 & -\xi_2 c_n \\ 0 & -c & 2c+\xi_3 c_n & \ldots & 0 & 0 & -\xi_3 c_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 2c+\xi_S c_n & -c & -\xi_S c_n \\ 0 & 0 & 0 & \ldots & -c & c+\xi_{S+1} c_n & -\xi_{S+1} c_n \\ -\xi_1 c_n & -\xi_2 c_n & -\xi_3 c_n & \ldots & -\xi_S c_n & -\xi_{S+1} c_n & c_n \end{pmatrix} \tag{5.13}$$

The solution of the moving ballscrew problem therefore reduces to the solution of the equation

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{F}(t) \tag{5.14}$$

where **M** is the mass matrix, **x** is the displacement array and **F** is the force array. The matrices **C** and **K** can change with time.

5.2 Solution of differential equation with time-dependent coefficients

Equation (5.14) is a non-homogeneous differential equation and, of course, the displacement array **x** and the force array **F** are functions of time. Because the nut moves, the ξ_i terms in equations (5.12) and (5.13) can be expected to vary continuously with time in a manner, which in the general case, is not periodic. Therefore the stiffness matrix **K** and the damping matrix **C** also change with time. The solution to equation (5.14) can therefore be expected to be a continuous but non-periodic function of time. It was therefore decided to seek solutions of the form

$$\mathbf{x} = \sum_{i=0}^{\infty} \mathbf{a}_i t^i \tag{5.15}$$

Differentiating with respect to time gives

$$\dot{\mathbf{x}} = \sum_{i=0}^{\infty} (i+1)\,\mathbf{a}_{i+1} t^i \tag{5.16}$$

A further differentiation gives

$$\ddot{\mathbf{x}} = \sum_{i=0}^{\infty} (i+1)(i+2)\,\mathbf{a}_{i+2} t^i \tag{5.17}$$

Initially, the force is regarded as a constant vector for the time interval over which a solution is sought, while **C** and **K** also remain constant. This is reasonable provided that the time interval over which the solution is sought is short compared with the scale over which changes in **f**, **C** and **K** take place. In this case, substituting equations (5.15)-(5.17) into equation (5.14) gives, for the constant terms

$$\mathbf{M} \times (0+1)(0+2) \times \mathbf{a_{0+2}} + \mathbf{C} \times (0+1) \times \mathbf{a_{0+1}} + \mathbf{K} \times \mathbf{a_0} = \mathbf{F}$$

whence

$$\mathbf{a_2} = -\frac{\mathbf{M^{-1}C}}{2}\mathbf{a_1} - \frac{\mathbf{M^{-1}K}}{2}\mathbf{a_0} + \frac{\mathbf{M^{-1}F}}{2} \tag{5.18}$$

Comparing coefficients for t^k gives

$$\mathbf{M} \times (k+1)(k+2) \times \mathbf{a_{0+2}} + \mathbf{C} \times (k+1) \times \mathbf{a_{0+1}} + \mathbf{K} \times \mathbf{a}_k = \mathbf{0}$$

whence

$$\mathbf{a}_{k+2} = -\frac{\mathbf{M^{-1}C}}{k+2}\mathbf{a}_{k+1} - \frac{\mathbf{M^{-1}K}}{(k+1)(k+2)}\mathbf{a}_k \tag{5.19}$$

or

$$\mathbf{a}_k = -\frac{\mathbf{M}^{-1}\mathbf{C}}{k}\mathbf{a}_{k-1} - \frac{\mathbf{M}^{-1}\mathbf{K}}{(k-1)k}\mathbf{a}_{k-2} \tag{5.20}$$

Thus, given the initial displacement vector $\mathbf{a}_0$ and the initial velocity vector $\mathbf{a}_1$, all the other coefficients $\mathbf{a}_k$ are calculable. Furthermore, the higher order coefficients are derived from the lower order ones by dividing by larger and larger numbers. Comparing this with the power series expansions for e^x, *sin x* and *cos x* it seems most probable that the series will converge to a finite solution.

Physical constraints on the system mean that $\mathbf{F}(t)$ has to be finite with, at worst, finite discontinuities. This means that the "roughest ride" for the system is one with finite discontinuities in acceleration. This implies that the velocity will always be continuous and the displacement "smooth". Such mathematically "well behaved" functions are likely to cause relatively little trouble in computation.

A series of programs based on this approach "T1.cpp" - "T3.cpp" have been written. Using values typical of the Beaver VC35 CNC machine tool (see Figure 4.12 and Section 4.2), it was found that $\mathbf{a}_k$ diverged significantly before converging. However, since the time interval considered was small, t^i converged rapidly. To avoid problems with multiplying very large numbers by very small ones, the displacement **d** at the end of a time interval T can be obtained by using a series

$$\mathbf{d} = \mathbf{x} = \sum_{i=0}^{\infty}\mathbf{b}_i \text{ where } \mathbf{b}_i = \mathbf{a}_i T^i \tag{5.21}$$

Letting $\mathbf{d}_0$ be the displacement and $\mathbf{v}_0$ be the velocity at the beginning of the time interval, $\mathbf{b}_0$ and $\mathbf{b}_1$ are calculable from the initial conditions as follows

$$\mathbf{b}_0 = \mathbf{d}_0 \text{ and } \mathbf{b}_1 = \mathbf{v}_0 T \tag{5.22), (5.23}$$

Multiplying equation (5.18) by T^2 gives

$$\begin{aligned}\mathbf{b}_2 = \mathbf{a}_2 T^2 &= -\frac{\mathbf{M}^{-1}\mathbf{C}}{2}\mathbf{a}_1 T^2 - \frac{\mathbf{M}^{-1}\mathbf{K}}{2}\mathbf{a}_0 T^2 + \frac{\mathbf{M}^{-1}\mathbf{f}}{2}T^2 \\ &= -\frac{\mathbf{M}^{-1}\mathbf{C}T}{2}\mathbf{a}_1 T - \frac{\mathbf{M}^{-1}\mathbf{K}T^2}{2}\mathbf{a}_0 T^0 + \frac{\mathbf{M}^{-1}\mathbf{f}}{2}T^2 \\ &= -\frac{\mathbf{M}^{-1}\mathbf{C}T}{2}\mathbf{b}_1 - \frac{\mathbf{M}^{-1}\mathbf{K}T^2}{2}\mathbf{b}_0 + \frac{\mathbf{M}^{-1}\mathbf{f}}{2}T^2\end{aligned} \tag{5.24}$$

and multiplying equation (5.20) by T^k gives

$$\begin{aligned}\mathbf{b}_k &= \mathbf{a}_k T^k = -\frac{\mathbf{M}^{-1}\mathbf{C}}{k}\mathbf{a}_{k-1}\,T^k - \frac{\mathbf{M}^{-1}\mathbf{K}}{(k-1)k}\mathbf{a}_{k-2}T^k \\ &= -\frac{\mathbf{M}^{-1}\mathbf{C}T}{k}\mathbf{a}_{k-1}\,T^{k-1} - \frac{\mathbf{M}^{-1}\mathbf{K}T^2}{(k-1)k}\mathbf{a}_{k-2}T^{k-2} \\ &= -\frac{\mathbf{M}^{-1}\mathbf{C}T}{k}\mathbf{b}_{k-1} - \frac{\mathbf{M}^{-1}\mathbf{K}T^2}{(k-1)k}\mathbf{b}_{k-2}\end{aligned} \tag{5.25}$$

Making the substitutions $\boldsymbol{\alpha} = -\mathbf{M}^{-1}\mathbf{K}T^2$ (5.26) and $\boldsymbol{\beta} = -\mathbf{M}^{-1}\mathbf{C}T$ (5.27) into equations (5.24) and (5.25) gives

$$\mathbf{b}_2 = \frac{\boldsymbol{\beta}}{2}\mathbf{b}_1 + \frac{\boldsymbol{\alpha}}{2}\mathbf{b}_0 + \frac{\mathbf{M}^{-1}\mathbf{f}}{2}T^2 \tag{5.28}$$

$$\mathbf{b}_k = \frac{\boldsymbol{\beta}}{k}\mathbf{b}_{k-1} + \frac{\boldsymbol{\alpha}}{(k-1)k}\mathbf{b}_{k-2} \tag{5.29}$$

Thus, knowing the force, $\mathbf{b}_2$ can be derived from $\mathbf{b}_0$ and $\mathbf{b}_1$, which are set by the initial conditions, and $\mathbf{b}_k$ can be derived from $\mathbf{b}_{k-1}$ and $\mathbf{b}_{k-2}$ for $k \geq 3$. It should be noted that all $\mathbf{b}_k$ have the dimension of the degree freedom being computed. For example, if the degree of freedom represents axial movement all $\mathbf{b}_k$ have units of length.

$\mathbf{b}_k$ are column vectors N long and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are $N \times N$ matrices, (N being the order of **M**, **C** and **K**).

The velocity can also be calculated. From equation (5.21) it is possible to define

$$\mathbf{v} = \dot{\mathbf{x}} = \sum_{i=0}^{\infty}(i+1)\mathbf{a}_{i+1}T^i = \sum_{i=0}^{\infty}\mathbf{c}_i \tag{5.30}$$

Thus

$$\mathbf{c}_i = (i+1)\mathbf{a}_{i+1}T^i \tag{5.31}$$

$$\mathbf{c}_0 = (0+1)\mathbf{a}_{0+1}T^0 = \mathbf{a}_1 = \mathbf{v}_0 \tag{5.32}$$

$$\begin{aligned}\mathbf{c}_1 &= (1+1)\mathbf{a}_{1+1}T^1 = {}^{2\mathbf{a}_2 T^2}\!\!\diagup_{T} \\ &= {}^{2\mathbf{b}_2}\!\!\diagup_{T}\end{aligned} \tag{5.33}$$

A recurrence relation for $\mathbf{c}_k$ can be derived by substituting for $\mathbf{a}_{i+1}$ in equation (5.31) using equation (5.20)

$$
\begin{aligned}
\mathbf{c}_k &= (k+1)\,\mathbf{a}_{k+1}T^k = -\frac{(k+1)\mathbf{M}^{-1}\mathbf{C}}{(k+1)}\mathbf{a}_k T^k - \frac{(k+1)\mathbf{M}^{-1}\mathbf{K}}{k(k+1)}\mathbf{a}_{k-1}T^k \\
&= -\frac{\mathbf{M}^{-1}\mathbf{C}T}{k}k\mathbf{a}_k T^{k-1} - \frac{\mathbf{M}^{-1}\mathbf{K}T^2}{(k-1)k}(k-1)\mathbf{a}_{k-1}T^{k-2} \\
&= \frac{\boldsymbol{\beta}}{k}k\mathbf{a}_k T^{k-1} + \frac{\boldsymbol{\alpha}}{(k-1)k}(k-1)\mathbf{a}_{k-1}T^{k-2} \\
&= \frac{\boldsymbol{\beta}}{k}\mathbf{c}_{k-1} + \frac{\boldsymbol{\alpha}}{(k-1)k}\mathbf{c}_{k-2}
\end{aligned}
\tag{5.34}
$$

$\mathbf{c}_k$ are column vectors *N* long. Again it should be noted that all $\mathbf{c}_k$ have the dimension of the time derivative of the degree freedom being computed. For example, if the degree of freedom represents axial movement all $\mathbf{c}_k$ have units of velocity.

It is now possible to define the state $\{\mathbf{d}, \mathbf{v}\}$ at the end of a time step *T* in terms of its state $\{\mathbf{d}_0, \mathbf{v}_0\}$ at the beginning of the step.

The dynamic behaviour of a ballscrew system with a moving nut is calculated by starting at some initial state $\{\mathbf{d}_0, \mathbf{v}_0\}$ and solving equation (5.14) for a time step of duration *T*. The final state given by equations (5.21) and (5.30) is used as the initial state of the next time step and so on. Progress of the nut is accounted for by redefining **K** and **C** at each step which means that $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in equations (5.26) and (5.27) and recurrence relations (5.29) and (5.34) are also redefined. This process is repeated for as long as necessary to cover the period of interest.

The energy of the system can also be computed. The kinetic energy *KE* is given by:-

$$KE = \frac{1}{2}\dot{\mathbf{x}}'\,\mathbf{M}\,\dot{\mathbf{x}} \tag{5.35}.$$

Because the mass matrix **M** is diagonal the computation of *KE* can be simplified to:-

$$KE = \frac{1}{2}\sum \dot{x}_i M_{ii}\dot{x}_i = \frac{1}{2}\sum M_{ii}\dot{x}_i^2 \tag{5.36}.$$

(Here symbols in bold, e.g. **x**, **M**, represent whole vectors and matrices and symbols in italics, e.g. *KE*, x_i, K_{ij}, represent scalar quantities, vector elements and matrix elements. $\mathbf{x}'$ represents the transpose of vector **x**.)

The potential energy *PE* is given by:-

$$PE = \frac{1}{2}\mathbf{x}'\,\mathbf{K}\,\mathbf{x} \tag{5.37}.$$

Equation (5.37) could be coded as a double matrix multiplication. However, the evaluation of such an expression would entail adding up terms of the form $K_{ii}x_i^2$, which will be positive, and terms of the form $-K_{ij}x_ix_j$, which could be of a similar magnitude but negative. This could lead

to numerical instability. It was therefore decided to compute the potential energy by summing the potential energy of all the springs in the system. Thus:-

$$PE = \frac{1}{2}\sum K_{ij}(x_i - x_j)^2 \tag{5.38}$$

for all non-zero K_{ij} where $i < j$.

The rate at which energy is dissipated by the viscous dampers in the system is given by:-

$$\begin{aligned}\frac{dE_d}{dt} &= \mathbf{F} \times \mathbf{v} = \mathbf{v}' \times \mathbf{F} \\ &= \dot{\mathbf{x}}' \mathbf{C}\, \dot{\mathbf{x}}\end{aligned} \tag{5.39}$$

For the same reason as the case of the potential energy the rate of energy dissipation was computed by:-

$$\frac{dE_d}{dt} = \sum C_{ij}(\dot{x}_i - \dot{x}_j)^2 \tag{5.40}$$

for all non-zero C_{ij} where $i < j$. The rate of energy dissipation for non-linear phenomena such as Coulomb friction is calculated by multiplying the relevant force by the appropriate velocity on an ad hoc basis. For example

$$\frac{dE_d}{dt} = \sum \mu_i F_{ni} \times \dot{x}_i \tag{5.41},$$

where μ_i is the coefficient of friction and F_{ni} is the contact force acting on the sliding face. The total energy dissipated is determined by integrating over time the sum of the rates of dissipation of the mechanisms involved.

The energy input E_{in} is derived by integrating the applied forces over their respective displacement:-

$$E_{in} = \int \mathbf{f} \times d\mathbf{x} \tag{5.42}$$

The sum of the kinetic and potential energies and the energy dissipated less the energy supplied should, of course, be zero. In fact this is not exactly the case because of rounding errors and approximations in the validity of the time series. Its magnitude compared to a measure of the maximum values of the items involved is used as a measure of the stability of the models' performance.

The solution programs were originally coded in the MATLAB language and were used to model the behaviour of a ballscrew test rig. Using a load cycle typical of such a rig it was found that the execution time for the programs was of the order of several hours. Steps were therefore taken to accelerate the solution.

One step was to convert the program which derives the time series solutions to the C language. To avoid writing graphic routines for C, the results of the solution process were written to file on the computer's hard disc and standard routines available in MATLAB were used to produce the results in graphical form. Another step was to note that the **K** and **C** matrices are sparse and so a large number of the calculations would entail multiplying by zero. If this could be avoided a substantial acceleration should occur. However the form of **K** and **C** is not that of a normal banded matrix since the non-zero terms on a typical row are $K_{i-1\ i}$, $K_{i\ i}$, $K_{i\ i+1}$ and $K_{i\ N}$. Taking advantage of the symmetry property of normal stiffness matrices meant that a $N \times N$ matrix could be reduced to a $N \times 6$ in the axial and torsional case. It was necessary to write special routines to manipulate matrices in this form.

At first the programs wrote the results to file at the end of each main time step and the execution time was significant. The program was therefore changed to write to file at the end of a block of 1000 outer time steps. The result is a program with acceptable execution times, having achieved an acceleration from of the order of hours to run a 25 element model of the ballscrew to about 15 minutes. A MATLAB routine reads the information from files generated by the C program and plots them in graphical form.

In order to avoid problems with convergence of the **b** series, the program has been modified to select its own time step. A plot of the energy balance check as a function of time shows an energy balance of less than 1% of the maximum energy level involved. This is a better precision than that to which many of the errors can be measured so this is considered to be satisfactory.

5.3 Extension to include torsional movement

When extending the model to include torsional behaviour an extra degree of freedom is needed at each of the nodes of the screw. The equations of motion for these additional nodes are as follows.

For node 1: $$J_1\ddot{\theta}_1 - c_\Theta(\dot{\theta}_2 - \dot{\theta}_1) - \Theta(\theta_2 - \theta_1) = \Gamma_1(t) \tag{5.43a}$$

For node i, where $2 \le i \le S$:

$$J_i\ddot{\theta}_i + c_\Theta(\dot{\theta}_i - \dot{\theta}_{i-1}) - c_\Theta(\dot{\theta}_{i+1} - \dot{\theta}i_1) + \Theta(\theta_i - \theta_{i-1}) - \Theta(\theta_{i+1} - \theta_i) = \Gamma_i(t) \tag{5.43b}$$

For node S+1:

$$J_{S+1}\ddot{\theta}_{S+1} + c_\Theta(\dot{\theta}_{S+1} - \dot{\theta}_S) + c_\Theta(\theta_{S+1} - \theta_S) = \Gamma_{S+1}(t) \tag{5.43c}$$

where J_i is the rotational inertia of the ith node, c_Θ is the coefficient of torsional damping of the element and the torque is $\Gamma_i(t)$. Here Θ is the torsional stiffness of a single element of the ballscrew given by

$$\Theta = \frac{GK}{X_{i+1} - X_i}, \quad i = 1, 2, ... S+1 \tag{5.44}$$

where G is shear modulus for the screw material and K is the screw's torsional constant.

Therefore the submatrix representing each element of the screw is extended from

$$\begin{pmatrix} \ddots & \dots & \dots & \dots \\ \dots & k & -k & \dots \\ \dots & -k & k & \dots \\ \dots & \dots & \dots & \ddots \end{pmatrix} \quad \text{to} \quad \begin{pmatrix} \ddots & \dots & \dots & \dots & \dots & \dots \\ \dots & k & 0 & -k & 0 & \dots \\ \dots & 0 & \Theta & 0 & -\Theta & \dots \\ \dots & -k & 0 & k & 0 & \dots \\ \dots & 0 & -\Theta & 0 & \Theta & \dots \\ \dots & \dots & \dots & \dots & \dots & \ddots \end{pmatrix} \tag{5.45}$$

where k is the axial stiffness of the ith element as used previously (equation (5.6)).

At the point of contact, the motion of the nut x_{pn} is derived by taking the axial motion of the point of contact of the screw x_p and adding to it the axial movement caused by the rotation of the screw θ_p

$$x_{pn} = x_p + R\theta_p \tag{5.46}$$

where R is the ballscrew ratio. For a screw of pitch p with N_s starts, this is

$$R = \frac{N_s\, p}{2\pi} \tag{5.47}$$

Neglecting friction, the force delivered to the nut is related to the torque reacted on the screw by

$$Nut\ \ force = R \times Screw\ torque \tag{5.48}$$

The connection between the axial and torsional subsystems is illustrated in Figure 5.5 using as an example a case where only two elements are considered and the nut is connected at the second node.

The thrust bearings which restrain the screw are modelled by springs of axial stiffness a_1 and a_2. The saddle or table onto which the nut is mounted has a mass M.

In many systems the load on the nut can rock about the Y axis. The load's inertia about a line parallel to this axis through its centre of mass is represented by J. The distance of the ballscrew axis below the load's centre of mass is z. The rocking motion is represented by ϕ_n and the restraining effect of the load's supporting slideways is modelled by a torsional spring

of stiffness Φ. The damping is assumed to be zero in order to show the mathematical development in as simple a manner as possible.

The system is driven by a torque $\Gamma(t)$ applied to inertia J_1.

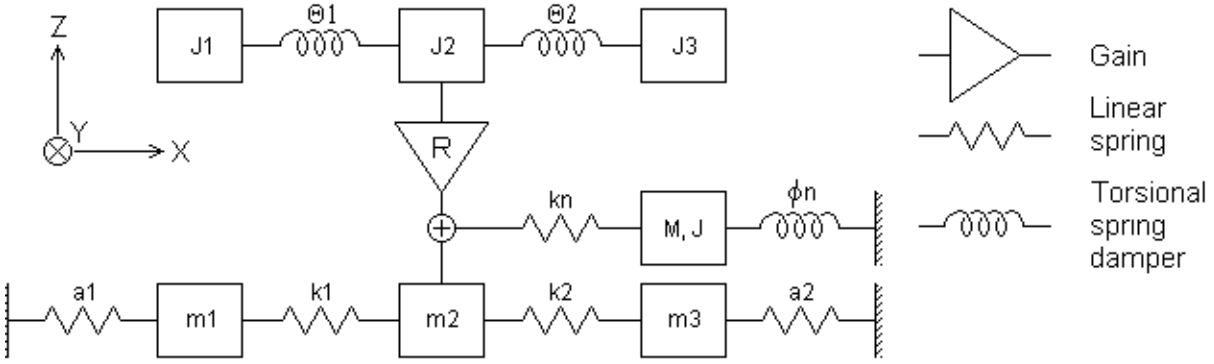


Figure 5.5 - *Ballscrew system model*

The equations of motion for the masses are:-

$$m_1 \ddot{x}_1 = k_1(x_2 - x_1) - a_1 x_1 \tag{5.49a}$$

$$m_2 \ddot{x}_2 = -k_1(x_2 - x_1) + k_2(x_3 - x_2) + k_n(x_n - z\phi_n - x_2 - R\theta_2) \tag{5.49b}$$

$$m_3 \ddot{x}_3 = -k_2(x_3 - x_2) - a_2 x_3 \tag{5.49c}$$

$$M \ddot{x}_n = -k_n(x_n - z\phi_n - x_2 - R\theta_2) \tag{5.49d}$$

and the equations of motion for the inertias are:-

$$J_1 \ddot{\theta}_1 = \Theta_1(\theta_2 - \theta_1) + \Gamma(t) \tag{5.50a}$$

$$J_2 \ddot{\theta}_2 = -\Theta_1(\theta_2 - \theta_1) + \Theta_2(\theta_3 - \theta_2) + Rk_n(x_n - z\phi_n - x_2 - R\theta_2) \tag{5.50b}$$

$$J_3 \ddot{\theta}_3 = -\Theta_2(\theta_3 - \theta_2) \tag{5.50c}$$

$$J \ddot{\phi}_n = zk_n(x_n - z\phi_n - x_2 - R\theta_2) \tag{5.50d}$$

In matrix form the equation of motion is given by

$$\begin{pmatrix} m_1\ddot{x}_1 \\ J_1\ddot{\theta}_1 \\ m_2\ddot{x}_2 \\ J_2\ddot{\theta}_2 \\ m_3\ddot{x}_3 \\ J_3\ddot{\theta}_3 \\ M\ddot{x}_n \\ J\ddot{\phi}_n \end{pmatrix} + \begin{pmatrix} k_1{+}a_1 & 0 & -k_1 & 0 & 0 & 0 & 0 & 0 \\ \dots & \Theta_1 & 0 & -\Theta_1 & 0 & 0 & 0 & 0 \\ \dots & \dots & k_1{+}k_2{+}k_n & Rk_n & -k_2 & 0 & -k_n & zk_n \\ \dots & \dots & \dots & \Theta_1{+}\Theta_2{+}R^2k_n & 0 & -\Theta_2 & -Rk_n & Rzk_n \\ \dots & \dots & \dots & \dots & k_2{+}a_2 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \Theta_2 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & k_n & -zk_n \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & z^2k_n{+}\Phi \end{pmatrix} \begin{pmatrix} x_1 \\ \theta_1 \\ x_2 \\ \theta_2 \\ x_3 \\ \theta_3 \\ x_n \\ \phi_n \end{pmatrix} = \begin{pmatrix} 0 \\ \Gamma \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(5.51)

Here, for compactness of presentation, only the upper half of the stiffness matrix is included because it is symmetric.

Moving on to the general case of S spring/damper elements in the ballscrew and the moving nut is being allowed for, considering the motion of node 1, equation (5.8a) becomes

$$\begin{aligned} m_1\ddot{x}_1 = {} & c(\dot{x}_2 - \dot{x}_1) + \xi_1 c_n(\dot{x}_n - z\dot{\phi}_n - \dot{x}_1 - R\dot{\theta}_1) \\ & + k(x_2 - x_1) + \xi_1 k_n(x_n - z\phi_n - x_1 - R\theta_1) \end{aligned} \tag{5.52a}$$

For node *i*, where $2 \le i \le S$ and $i \ne B_1$ or $i \ne B_2$, where B_1 and B_2 are the nodes at which the journal bearings are attached, equation (5.8b) becomes:

$$\begin{aligned} m_i\ddot{x}_i = {} & -c(\dot{x}_i - \dot{x}_{i-1}) + c(\dot{x}_{i+1} - \dot{x}_i) + \xi_i c_n(\dot{x}_n - z\dot{\phi}_n - \dot{x}_i - R\dot{\theta}_i) \\ & - k(x_i - x_{i-1}) + k(x_{i+1} - x_i) + \xi_i k_n(x_n - z\phi_n - x_i - R\theta_i) \end{aligned} \tag{5.52b}$$

For node S+1, equation (5.8c) becomes:

$$\begin{aligned} m_{S+1}\ddot{x}_{S+1} = {} & -c(\dot{x}_{S+1} - \dot{x}_S) + \xi_{S+1} c_n(\dot{x}_n - z\dot{\phi}_n - \dot{x}_{S+1} - R\dot{\theta}_{S+1}) \\ & - k(x_{S+1} - x_S) + \xi_{S+1} k_n(x_n - z\phi_n - x_{S+1} - R\theta_{S+1}) \end{aligned} \tag{5.52c}$$

Finally for the nut, equation (5.9) becomes:

$$M\ddot{x}_n = -\sum_{i=1}^{S+1} \xi_i c_n(\dot{x}_n - z\dot{\phi}_n - \dot{x}_i - R\dot{\theta}_i) - \sum_{i=1}^{S+1} \xi_i k_n(x_n - z\phi_n - x_i - R\theta_i) \tag{5.52d}$$

The equations of motion for the masses at the bearing nodes are:-

$$\begin{aligned} m_{B1}\ddot{x}_{B1} = {} & -c(\dot{x}_{B1} - \dot{x}_{B1-1}) + c(\dot{x}_{B1+1} - \dot{x}_{B1}) - c_{a1}\dot{x}_{B1} \\ & -k(x_{B1} - x_{B1-1}) + k(x_{B1+1} - x_{B1}) - a_1 x_{B1} \end{aligned} \tag{5.53a}$$

$$\begin{aligned} m_{B2}\ddot{x}_{B2} = {} & -c(\dot{x}_{B2} - \dot{x}_{B2-1}) + c(\dot{x}_{B2+1} - \dot{x}_{B2}) - c_{a2}\dot{x}_{B2} \\ & -k(x_{B2} - x_{B2-1}) + k(x_{B2+1} - x_{B2}) - a_2 x_{B2} \end{aligned} \tag{5.53b}$$

(In the event that $B_1 = 1$, the first and third parentheses in equation (5.53a) become zero and if $B_2 = S + 1$, the second and last parentheses in equation (5.53b) become zero.)

Using the same conditions for the node number *i* as for equation (5.52b), the equations of motion for the inertias are:-

$$\begin{aligned} J_1\ddot{\theta}_1 = {} & c_\theta(\dot{\theta}_2 - \dot{\theta}_1) + \xi_1 R c_{\theta n}(\dot{x}_n - z\dot{\phi}_n - \dot{x}_1 - R\dot{\theta}_1) \\ & + \Theta(\theta_2 - \theta_1) + \xi_1 R k_n(x_n - z\phi_n - x_2 - R\theta_2) + \Gamma(t) \end{aligned} \tag{5.54a}$$

$$\begin{aligned} J_i\ddot{\theta}_i = {} & -c_\theta(\dot{\theta}_i - \dot{\theta}_{i-1}) + c_\theta(\dot{\theta}_{i+1} - \dot{\theta}_i) + \xi_i R c_{\theta n}(\dot{x}_n - z\dot{\phi}_n - \dot{x}_i - R\dot{\theta}_i) \\ & - \Theta(\theta_i - \theta_{i-1}) + \Theta(\theta_{i+1} - \theta_i) + \xi_i R k_n(x_n - z\phi_n - x_i - R\theta_i) \end{aligned} \tag{5.54b}$$

$$\begin{aligned} J_{S+1}\ddot{\theta}_{S+1} = {} & -c_\theta(\dot{\theta}_{S+1} - \dot{\theta}_S) + \xi_{S+1} R c_{\theta n}(\dot{x}_n - z\dot{\phi}_n - \dot{x}_{S+1} - R\dot{\theta}_{S+1}) \\ & - \Theta(\theta_{S+1} - \theta_S) + \xi_{S+1} R k_n(x_n - z\phi_n - x_{S+1} - R\theta_{S+1}) \end{aligned} \tag{5.54c}$$

$$J\ddot{\phi}_n = \sum_{i=1}^{S+1} \xi_i z_n c_n (\dot{x}_n - z\dot{\phi}_n - \dot{x}_i - R\dot{\theta}_i) + \sum_{i=1}^{S+1} \xi_i z_n k_n (x_n - z\phi_n - x_i - R\theta_i) \qquad (5.54d)$$

5.4 Backlash

Backlash is taken into account by allowing b_l free movement between the screw and the nut before contact is made. This is shown in a simplified form in Figure 5.6.

This means that there are three modes of motion:-

- First where the screw is pushing the nut in the negative x direction OR when the nut is "over-riding" the screw in the positive x direction. This condition is designated "negative" and, allowing for rocking of the load, occurs if

$$x_{bs} + R\theta_{bs} < x_n - z\phi_n - b_l / 2 \qquad (5.55a)$$

where x_{bs} is the axial position of the part of the screw which would be in the same plane as the nut centre if the system were to be at rest under no load and θ_{bs} is the angular position of the same point. In this case the backlash indicating factors are set as

$$B_{lf1} = 1, \qquad B_{lf2} = -1 \qquad (5.55b)$$

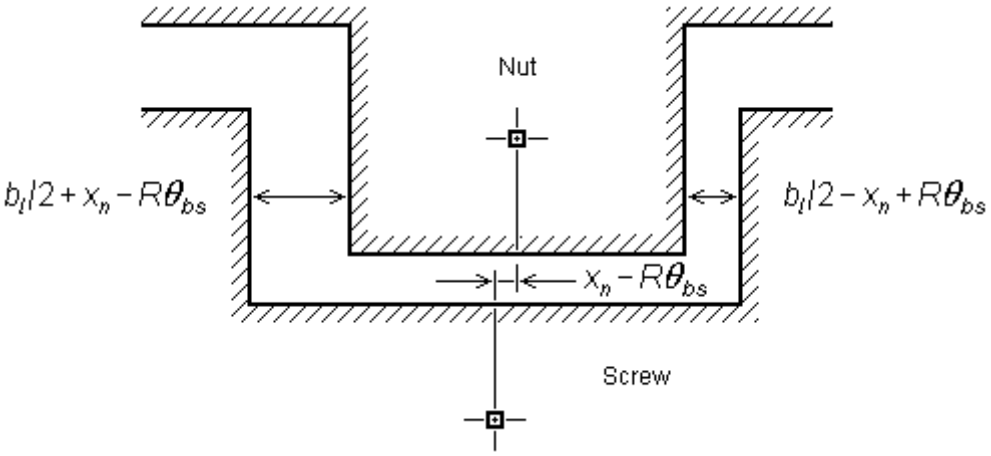


Figure 5.6 - *Ballscrew backlash*

- Second where the screw is pushing the nut in the positive x direction OR when the nut is "over-riding" the screw in the negative x direction. This condition is designated "positive" and, allowing for rocking of the load, occurs if

$$x_n - z\phi_n + b_l / 2 < x_{bs} + R\theta_{bs} \qquad (5.56a)$$

In this case the backlash indicating factors are set as

$$B_{lf1} = 1, \qquad B_{lf2} = 1 \qquad (5.56b)$$

- Third where the screw is riding in the backlash and the screw and its attachments and the load are treated as mechanically independent. This condition is designated "free" and, allowing for rocking of the load, occurs if

$$x_n - z\phi_n - b_l/2 \le x_{bs} + R\theta_{bs} \le x_n - z\phi_n + b_l/2 \tag{5.57a}$$

In this case the backlash indicating factors are set as

$$B_{lf1} = 0, \qquad B_{lf2} = 0 \tag{5.57b}$$

Two of the equations of motion for the masses (5.49b) and (5.49d) need to be modified to:-

$$\begin{aligned} m_2\,\ddot{x}_2 &= -k_1(x_2 - x_1) + k_2(x_3 - x_2) + B_{lf1}k_n(x_n - z\phi_n - x_2 - R\theta_2 - B_{lf2}b_l/2) \\ &= -k_1(x_2 - x_1) + k_2(x_3 - x_2) + B_{lf1}k_n(x_n - z\phi_n - x_2 - R\theta_2) - B_{lf2}k_n b_l/2 \end{aligned} \tag{5.58b}$$

$$\begin{aligned} M\,\ddot{x}_n &= -B_{lf1}k_n(x_n - z\phi_n - x_2 - R\theta_2 - B_{lf2}b_l/2) \\ &= -B_{lf1}k_n(x_n - z\phi_n - x_2 - R\theta_2) + B_{lf2}k_n b_l/2 \end{aligned} \tag{5.58d}$$

and the equations of motion for the inertias (5.50b) and (5.50d) also need to be changed to:-

$$\begin{aligned} J_2\,\ddot{\theta}_2 &= -\Theta_1(\theta_2 - \theta_1) + \Theta_2(\theta_3 - \theta_2) + B_{lf1}Rk_n(x_n - z\phi_n - x_2 - R\theta_2 - B_{lf2}b_l/2) \\ &= -\Theta_1(\theta_2 - \theta_1) + \Theta_2(\theta_3 - \theta_2) + B_{lf1}Rk_n(x_n - z\phi_n - x_2 - R\theta_2) - B_{lf2}k_n b_l/2 \end{aligned} \tag{5.59b}$$

$$\begin{aligned} J\,\ddot{\phi}_n &= B_{lf1}zk_n(x_n - z\phi_n - x_2 - R\theta_2 - B_{lf2}b_l/2) - \Phi\phi_n \\ &= B_{lf1}zk_n(x_n - z\phi_n - x_2 - R\theta_2) - \Phi\phi_n - B_{lf2}k_n b_l/2 \end{aligned} \tag{5.59d}$$

In matrix form the equation of motion has to be changed in two ways. First the stiffness matrix is replaced by

$$\begin{pmatrix} k_1{+}a_1 & 0 & -k_1 & 0 & 0 & 0 & 0 & 0 \\ \ldots & \Theta_1 & 0 & -\Theta_1 & 0 & 0 & 0 & 0 \\ \ldots & \ldots & k_1{+}k_2{+}B_{lf1}k_n & B_{lf1}Rk_n & -k_2 & 0 & -B_{lf1}k_n & B_{lf1}zk_n \\ \ldots & \ldots & \ldots & \Theta_1{+}\Theta_2{+}B_{lf1}R^2k_n & 0 & -\Theta_2 & -B_{lf1}Rk_n & B_{lf1}Rzk_n \\ \ldots & \ldots & \ldots & \ldots & k_2{+}a_2 & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \Theta_2 & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & B_{lf1}k_n & -B_{lf1}zk_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & B_{lf1}z^2k_n{+}\Phi \end{pmatrix} \tag{5.60a}$$

and the force vector by

$$\begin{pmatrix} 0 \\ \Gamma \\ -B_{lf\,2}k_n b_l/2 \\ -B_{lf\,2}Rk_n b_l/2 \\ 0 \\ 0 \\ B_{lf\,2}k_n b_l/2 \\ -B_{lf\,2}zk_n b_l/2 \end{pmatrix} \tag{5.60b}$$

All of these changes to the stiffness matrix and force vector can be accommodated readily by the solution method

5.5 A refinement of the method

The approach outlined in the preceding sections was found to give results consistent with the expected behaviour of a ballscrew system. When the transverse degrees of freedom came to be modelled, it turned out that the displacements exhibited some rather odd behaviour during a trapezoidal velocity demand run. The underlying trend of the displacements showed a series of discontinuities in gradient as the nut crossed the boundaries from one ballscrew element to another. It appeared that the vibration characteristics of the system changed as the nut passed from one spring/damper element to another. This was particularly distressing since the individual elements are mathematical abstractions which do not correspond to a real physical phenomenon of themselves though taken together they model the behaviour of the screw.

Consideration of the cause of this phenomenon has led to a refinement of the method along the following lines.

It will be recalled that the stiffness of a ballscrew system can be represented by a matrix (**K**) which can be partitioned conveniently as follows

$$(\mathbf{K}) = \begin{pmatrix} \mathbf{K_{dr}} & -\mathbf{K_{dr\,bs}} & \mathbf{0} \\ -\mathbf{K'_{dr\,bs}} & \mathbf{K_{bs}} & -\mathbf{K_{bs\,n}} \\ \mathbf{0} & -\mathbf{K'_{bs\,n}} & \mathbf{K_n} \end{pmatrix} \tag{5.61}$$

The submatrix $\mathbf{K_{dr}}$ represents the controller, drive motor and mechanical coupling device, $\mathbf{K_{bs}}$ the ballscrew and $\mathbf{K_n}$ the nut and load. $\mathbf{K_{dr\ bs}}$ represents the cross terms between the driving mechanism and the screw and $\mathbf{K_{bs\ n}}$ the cross terms between the screw and the nut. The submatrix $\mathbf{K_{bs\ n}}$ includes influence coefficients which are zero when the nut is not over the relevant part of the ballscrew and which are between 0 and 1 when the nut is over an element.

Up to now the influence has been shared on a linearly proportional basis with f_1, f_2 being given by

$$f_1 = \frac{x_i - x_n}{x_i - x_{i-1}} \qquad \text{and} \qquad f_2 = \frac{x_n - x_{i-1}}{x_i - x_{i-1}} \qquad \text{(5.62), c.f. (5.11a)}$$

This estimate of distribution gives acceptable results for the axial and torsional cases but it is not strictly correct. The presence of two connections from the nut to the ballscrew at nodes *i-1* and *i* gives in addition to the desired connection to the nut an undesired parallel connection between nodes *i-1* and *i* making the screw stiffer in an undulating way as the nut moves along, (see Figure 5.7).

An estimate of the magnitude of this effect for the ballscrew being used to test the model gives a fluctuation of the axial stiffness between the bearings of the order of 0.15% when 25 elements are used to model the screw. This is presumably why the effect was not noticed earlier.

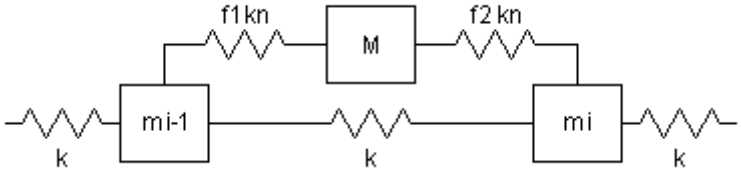


Figure 5.7 - *First method*

(In the transverse case the effect is more pronounced, in part because when the nut is in the middle of a section the two springs between which the stiffness is shared give a moment support not seen when the nut is passing a node.)

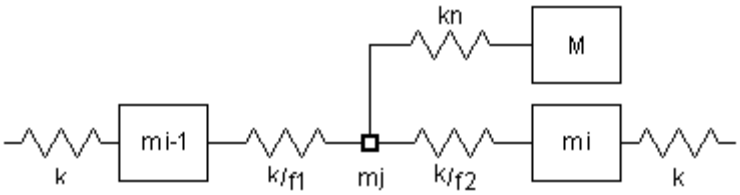


Figure 5.8 - *The refined method*

A more rigorous approach would be to consider the spring representing the nut connected at a massless node between two nodes *i-1* and *i* (see Figure 5.8) and derive the equations of motion as follows

$$\begin{pmatrix} m_{i-1}\ddot{x}_{i-1} \\ m_j\ddot{x}_j \\ m_i\ddot{x}_i \\ m_n\ddot{x}_n \end{pmatrix} + \begin{pmatrix} K_{i-1i-1} & K_{i-1j} & 0 & 0 \\ K_{ji-1} & K_{jj} & K_{ji} & K_{jn} \\ 0 & K_{ij} & K_{ii} & 0 \\ 0 & K_{nj} & 0 & K_{nn} \end{pmatrix} \begin{pmatrix} x_{i-1} \\ x_j \\ x_i \\ x_n \end{pmatrix} = \begin{pmatrix} F_{i-1} \\ 0 \\ F_i \\ F_n \end{pmatrix} \tag{5.63}$$

Here *j* designates the massless node where no external force *F* will be applied. The masses m_i and the stiffnesses K_{ij} will be, in general, $N_f \times N_f$ submatrices, where N_f is the number of degrees of freedom at each node of the ballscrew. The displacements x_i and the forces F_i are $N_f \times 1$ column vectors.

It is necessary to eliminate x_j. This can be done as follows. First re-order and partition the matrix

$$\begin{pmatrix} m_{i-1}\ddot{x}_{i-1} \\ m_i\ddot{x}_i \\ m_n\ddot{x}_n \\ -- \\ m_j\ddot{x}_j \end{pmatrix} + \left(\begin{array}{ccc|c} K_{i-1i-1} & 0 & 0 & K_{i-1j} \\ 0 & K_{ii} & 0 & K_{ij} \\ 0 & 0 & K_{nn} & K_{nj} \\ \hline K_{ji-1} & K_{ji} & K_{jn} & K_{jj} \end{array}\right) \begin{pmatrix} x_{i-1} \\ x_i \\ x_n \\ -- \\ x_j \end{pmatrix} = \begin{pmatrix} F_{i-1} \\ F_i \\ F_n \\ -- \\ 0 \end{pmatrix} \tag{5.64}$$

Grouping the internal and external forces together and bearing in mind that $m_j=0$

$$\begin{pmatrix} \Phi_{i-1} \\ \Phi_i \\ \Phi_n \\ -- \\ 0 \end{pmatrix} = \left(\begin{array}{ccc|c} K_{i-1i-1} & 0 & 0 & K_{i-1j} \\ 0 & K_{ii} & 0 & K_{ij} \\ 0 & 0 & K_{nn} & K_{nj} \\ \hline K_{ji-1} & K_{ji} & K_{jn} & K_{jj} \end{array}\right) \begin{pmatrix} x_{i-1} \\ x_i \\ x_n \\ -- \\ x_j \end{pmatrix} = \begin{pmatrix} F_{i-1} - m_{i-1}\ddot{x}_{i-1} \\ F_i - m_i\ddot{x}_i \\ F_n - m_n\ddot{x}_n \\ -- \\ 0 \end{pmatrix} \tag{5.65a}$$

or in simpler terms

$$\begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{p} & \mathbf{q} \\ \mathbf{q}' & \mathbf{r} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\varepsilon} \end{pmatrix} \tag{5.65b}$$

Expanding

$$\mathbf{f} = \mathbf{p}\boldsymbol{\delta} + \mathbf{q}\boldsymbol{\varepsilon}$$
$$\mathbf{0} = \mathbf{q}'\boldsymbol{\delta} + \mathbf{r}\boldsymbol{\varepsilon}$$

Whence
$$\boldsymbol{\varepsilon} = -\mathbf{r}^{-1}\mathbf{q}'\boldsymbol{\delta}$$

Finally
$$\begin{aligned} \mathbf{f} &= (\mathbf{p} - \mathbf{q}\mathbf{r}^{-1}\mathbf{q}')\boldsymbol{\delta} \\ &= \mathbf{p}_{\mathbf{red}}\,\boldsymbol{\delta} \end{aligned} \tag{5.66}$$

This means that the matrix representing the spring attached to the massless node can be replaced by a reduced matrix of the form $\mathbf{p} - \mathbf{q}\mathbf{r}^{-1}\mathbf{q}'$. In the case of the axial and torsional degrees of freedom, (x, θx), **p** is a 6×6 submatrix, **q** a 6×2 submatrix and $\mathbf{r}^{-1}$ a 2×2 submatrix.

Using f to represent the fraction of the distance of the contact point along the element $\{X_{i-1}, X_i\}$ in the terms of equation (5.62)

$$1-f=f_1 \qquad \text{and} \qquad f=f_2 \tag{5.67}$$

The parts of the stiffness matrix relating to the element in contact with the nut can now be written

$$\begin{pmatrix} \mathbf{p} & \mathbf{q} \\ \mathbf{q}' & \mathbf{r} \end{pmatrix} = \left(\begin{array}{cccccc|cc} \frac{k}{f} & 0 & 0 & 0 & 0 & 0 & -\frac{k}{f} & 0 \\ \ldots & \frac{\Theta}{f} & 0 & 0 & 0 & 0 & 0 & -\frac{\Theta}{f} \\ \ldots & \ldots & \frac{k}{1-f} & 0 & 0 & 0 & -\frac{k}{1-f} & 0 \\ \ldots & \ldots & \ldots & \frac{\Theta}{1-f} & 0 & 0 & 0 & -\frac{\Theta}{1-f} \\ \ldots & \ldots & \ldots & \ldots & k_n & -zk_n & -k_n & -Rk_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & z^2k_n & zk_n & Rzk_n \\ -- & -- & -- & -- & -- & -- & -- & -- \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & (\frac{1}{f}+\frac{1}{1-f})k+k_n & Rk_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & (\frac{1}{f}+\frac{1}{1-f})\Theta+R^2k_n \end{array}\right) \tag{5.68}$$

$\mathbf{r}^{-1}$ is a flexibility matrix and is given by

$$\mathbf{r}^{-1} = \begin{pmatrix} (\frac{1}{f}+\frac{1}{1-f})\Theta+R^2k_n & -Rk_n \\ -Rk_n & (\frac{1}{f}+\frac{1}{1-f})k+k_n \end{pmatrix} / D \tag{5.69a}$$

where the determinant

$$D = \frac{k\Theta + f(1-f)\Theta k_n + f(1-f)R^2 k\, k_n}{f^2(1-f)^2} \tag{5.69b}$$

which, for reasons of convenience in algebraic manipulation, can be rendered as

$$\mathbf{r}^{-1} = \begin{pmatrix} \frac{1}{k}+f(1-f)\frac{R^2k_n}{k\Theta} & -f(1-f)\frac{Rk_n}{k\Theta} \\ -f(1-f)\frac{Rk_n}{k\Theta} & \frac{1}{\Theta}+f(1-f)\frac{k_n}{k\Theta} \end{pmatrix} \times \frac{f(1-f)}{D'} \tag{5.70a}$$

where the modified determinant

$$D' = 1+f(1-f)\frac{k_n}{k}+f(1-f)\frac{R^2k_n}{\Theta} \tag{5.70b}$$

It will be noticed that $\mathbf{r}^{-1}$ vanishes at f=0 and f=1 and that the modified determinant is equal to one at f=0 and f=1.

Deriving the reduced stiffness matrix $\mathbf{p_{red}}$ is made easier by using the expression

$$\begin{aligned} D'qr^{-1}q'_{ij} = {} & f(1-f)q_{i1}\left(\frac{1}{k} + f(1-f)\frac{R^2k_n}{k\Theta}\right)q'_{1j} - f^2(1-f)^2 q_{i1}\frac{Rk_n}{k\Theta}q'_{2j} \\ & - f^2(1-f)^2 q_{i2}\frac{Rk_n}{k\Theta}q'_{1j} + f(1-f)q_{i2}\left(\frac{1}{\Theta} + f(1-f)\frac{k_n}{k\Theta}\right)q'_{2j} \end{aligned} \tag{5.71}$$

To take an example, the first term of the reduced stiffness matrix covering the element connected to the nut load and the nut load freedoms can be derived as follows

$$\begin{aligned} D'p_{red\,11} &= D'p_{11} - D'qr^{-1}q'_{11} \\ &= D'p_{11} - f(1-f)q_{11}\left(\frac{1}{k} + f(1-f)\frac{R^2k_n}{k\Theta}\right)q'_{11} - f^2(1-f)^2 q_{11}\frac{Rk_n}{k\Theta}q'_{21} \\ &\quad - f^2(1-f)^2 q_{12}\frac{Rk_n}{k\Theta}q'_{11} + f(1-f)q_{12}\left(\frac{1}{\Theta} + f(1-f)\frac{k_n}{k\Theta}\right)q'_{21} \\ &= \left(1 + f(1-f)\frac{k_n}{k} + f(1-f)\frac{R^2k_n}{\Theta}\right)\frac{k}{f} - f(1-f)\times -\frac{k}{f}\times\left(\frac{1}{k} + f(1-f)\frac{R^2k_n}{k\Theta}\right)\times -\frac{k}{f} \\ &\quad - f^2(1-f)^2\times -\frac{k}{f}\times\frac{Rk_n}{k\Theta}\times 0 - f^2(1-f)^2\times 0\times\frac{Rk_n}{k\Theta}\times -\frac{k}{f} \\ &\quad + f(1-f)\times 0\times\left(\frac{1}{\Theta} + f(1-f)\frac{k_n}{k\Theta}\right)\times 0 \\ &= \frac{k}{f} + (1-f)k_n + (1-f)\frac{R^2kk_n}{\Theta} - f(1-f)\frac{k}{f} - (1-f)^2\frac{R^2kk_n}{\Theta} \\ &= k + (1-f)k_n + f(1-f)\frac{R^2kk_n}{\Theta} \end{aligned} \tag{5.72}$$

It will be noticed that if $f=0$, $p_{red\,11} = k + k_n$ and if $f=1$, $p_{red\,11} = k$ as one would expect. A similar exercise for the other 20 independent members of matrix $D'\mathbf{p_{red}}$ gives

$$\begin{pmatrix} k+(1-f)k_n & & -k & & & \\ +f(1-f)\dfrac{R^2 k\,k_n}{\Theta} & (1-f)^2 Rk_n & -f(1-f)\dfrac{R^2 k\,k_n}{\Theta} & f(1-f)Rk_n & -(1-f)k_n & (1-f)zk_n \\ & \Theta+(1-f)R^2 k_n & & -\Theta & & \\ \ldots & +f(1-f)\dfrac{k_n\Theta}{k} & f(1-f)Rk_n & -f(1-f)\dfrac{k_n\Theta}{k} & -(1-f)Rk_n & (1-f)Rzk_n \\ & & k+f\,k_n & & & \\ \ldots & \ldots & +f(1-f)\dfrac{R^2 k\,k_n}{\Theta} & f^2 Rk_n & -f\,k_n & f\,zk_n \\ & & & \Theta+fR^2 k_n & & \\ \ldots & \ldots & \ldots & +f(1-f)\dfrac{k_n\Theta}{k} & -fRk_n & fRzk_n \\ \ldots & \ldots & \ldots & \ldots & k_n & -zk_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & z^2 k_n \end{pmatrix} \tag{5.73a}$$

When $f=0$, $D'=1$ and $\mathbf{p}_{\mathbf{red}}$ reduces to

$$\mathbf{p}_{\mathbf{red}} = \begin{pmatrix} k+k_n & Rk_n & -k & 0 & -k_n & zk_n \\ \ldots & \Theta+R^2 k_n & 0 & -\Theta & -Rk_n & Rzk_n \\ \ldots & \ldots & k & 0 & 0 & 0 \\ \ldots & \ldots & \ldots & \Theta & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & k_n & -zk_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & z^2 k_n \end{pmatrix} \tag{5.73b}$$

When $f=1$, $D'=1$ and $\mathbf{p}_{\mathbf{red}}$ reduces to

$$\mathbf{p}_{\mathbf{red}} = \begin{pmatrix} k & 0 & -k & 0 & 0 & 0 \\ \ldots & \Theta & 0 & -\Theta & 0 & 0 \\ \ldots & \ldots & k+k_n & Rk_n & -k_n & zk_n \\ \ldots & \ldots & \ldots & \Theta+R^2 k_n & -Rk_n & Rzk_n \\ \ldots & \ldots & \ldots & \ldots & k_n & -zk_n \\ \ldots & \ldots & \ldots & \ldots & \ldots & z^2 k_n \end{pmatrix} \tag{5.73c}$$

Both of equations (5.73b) and (5.73c) are to be expected by comparison with equation (5.51).

To give a more rigorous approach, equation (5.65b) can be expanded as follows

$$\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} \mathbf{p} & \mathbf{q} & \mathbf{s} \\ \mathbf{q}' & \mathbf{r} & \mathbf{0} \\ \mathbf{s}' & \mathbf{0} & \mathbf{t} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta} \\ \boldsymbol{\varepsilon} \\ \boldsymbol{\varphi} \end{pmatrix} \tag{5.74}$$

Here

- **r** is the matrix representing the massless node at the point where the nut connects to the screw,

- **p** represents the partition of the matrix which includes the nut and its load and the nodes of the ballscrew at either side of the connecting point
- **q**, **q′** represent the submatrices which connect **p** and **r**,
- **t** represents the rest of the system where the external forces normally act,
- **s**, **s′** represent the submatrices which connect **p** and **t** and
- φ represents the displacements of the degrees of freedom where external forces act.

Expanding

$$\mathbf{0} = \mathbf{p}\delta + \mathbf{q}\varepsilon + \mathbf{s}\varphi$$
$$\mathbf{0} = \mathbf{q}'\delta + \mathbf{r}\varepsilon$$
$$\mathbf{f} = \mathbf{s}'\delta + \mathbf{t}\varphi$$

Whence

$$\varepsilon = -\mathbf{r}^{-1}\mathbf{q}'\delta$$

Finally

$$\begin{pmatrix}\mathbf{f}\\ \mathbf{0}\end{pmatrix} = \begin{pmatrix}\mathbf{t} & \mathbf{s}'\\ \mathbf{s} & \mathbf{p}-\mathbf{q}\mathbf{r}^{-1}\mathbf{q}'\end{pmatrix}\begin{pmatrix}\varphi\\ \delta\end{pmatrix} = \begin{pmatrix}\mathbf{t} & \mathbf{s}\\ \mathbf{s} & \mathbf{p}_{\mathrm{red}}\end{pmatrix}\begin{pmatrix}\varphi\\ \delta\end{pmatrix} \tag{5.75}$$

Therefore the relevant parts of the system stiffness matrix can be modified along the lines shown to give a better representation of a ballscrew with a moving nut.

5.6 Comparison with classical theory

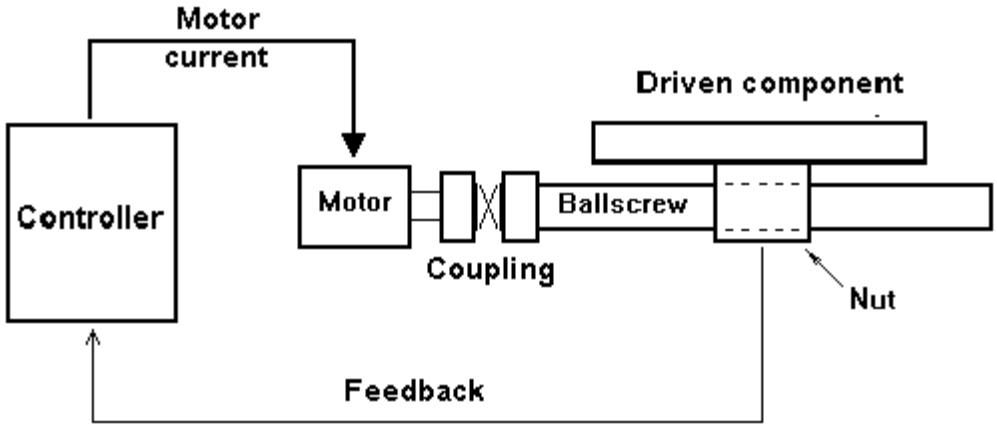


Figure 5.9 - *Comparison with classical theory*

In order to demonstrate the method's capabilities the following test has been undertaken.

A steel ballscrew 38.1 mm in diameter and 1520 mm long was modelled using program T3.cpp. The screw was supported at the drive end with a bearing of 8×10^8 N/m axial stiffness at the tail end with a bearing of 15.6×10^8 N/m axial stiffness. A simple PID controller was coded into the model to enable a trapezoidal velocity profile to be run. This

controller had a proportional constant K_p of 2 N m/rad, an integral constant K_i of 100 N m/(rad sec) and a differential constant K_d of zero. Velocity feedback came from the nut, (see Figure 5.9). The demand signal is the diagonal straight line on Figure 5.13.

At the start of the motion the nut was positioned at 100 mm from the driven end in the centre of 10 μm of backlash. The first part of the motion entailed an acceleration of the nut of 0.2 m/sec^2. In order to avoid the results being swamped by the friction between the table and the slideways, the coefficient of friction between them was assumed to be zero. A load of 359 kg was carried by the nut. The first 50 msec of the ensuing motion is shown on Figures 5.10 to 5.13 and Figures 5.16 and 5.17. The first 100 msec is shown on Figures 5.14 and 5.15.

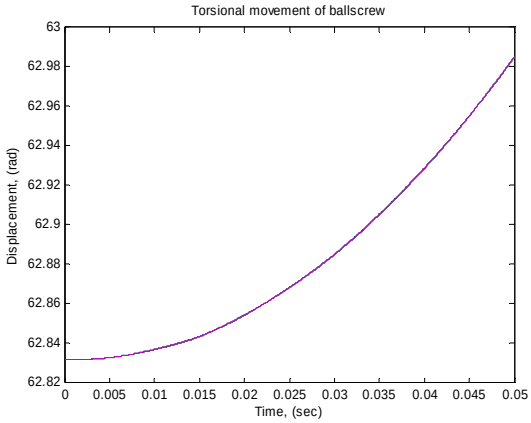


Figure 5.10 – *Ballscrew torsional displacement*

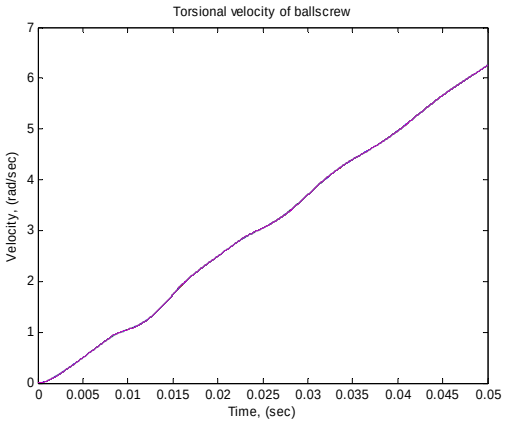


Figure 5.11 – *Ballscrew torsional velocity*

In the first few milliseconds the drive starts the screw (Figures 5.10 and 5.11). The screw runs through the backlash and picks up the nut with a “bump”, (Figure 5.13). The controller reacts to the additional load by increasing significantly the torque to bring the nut to the required position. In doing so the velocity of the screw is increased beyond demand for a short while. This has the effect of flinging the nut into the backlash when the controller

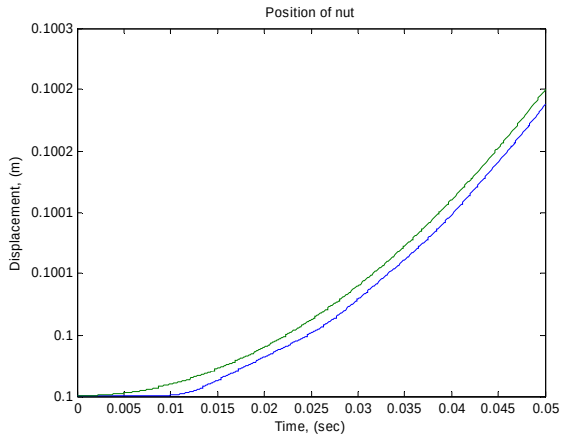


Figure 5.12 - *Ball nut displacement*

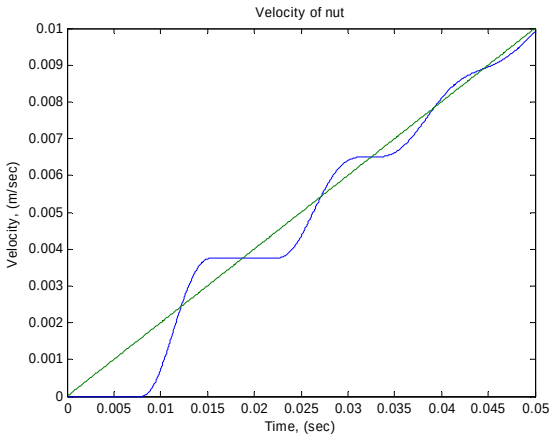


Figure 5.13 - *Ball nut velocity*

reduces the torque as speed is achieved. As the screw picks up speed, with the inertia of the system reduced by not having to carry the nut, it again catches up with nut with a second but smaller bump. This also causes the controller to increase the torque resulting in what appears to be a momentary loss of contact between the screw and the nut. Thereafter the nut and screw settle down to a steady acceleration. Figures 5.12 and 5.13 show the motion of the nut. Hand calculations based on classical elastic theory show that the force required to accelerate the table is 71.8 N. This gives rise to an axial displacement of the screw at its point of contact with the nut of 0.962×10^{-7} m. Figures 5.14 and 5.15 show the motion of 5 points distributed equally along the screw. Hand calculations (see Appendix 5.2) reveal that the axial deflections to be expected are:-

- Drive end bearing 0.717×10^{-7} m
- Ballscrew ¼ point 0.791×10^{-7} m
- Ballscrew ½ point 0.558×10^{-7} m
- Ballscrew ¾ point 0.328×10^{-7} m
- Tail end bearing 0.093×10^{-7} m.

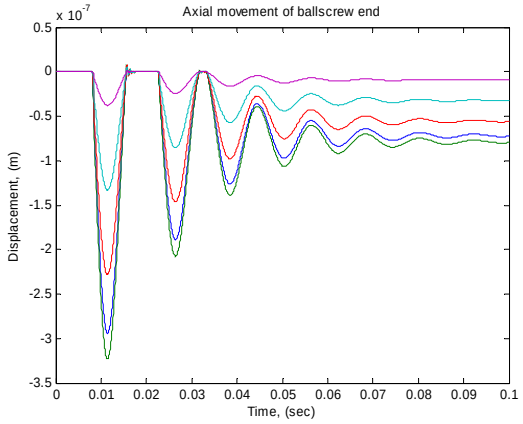


Figure 5.14 – *Ballscrew axial displacement*

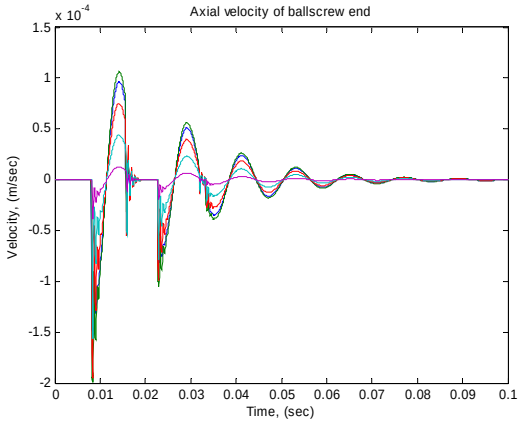


Figure 5.15 – *Ballscrew axial velocity*

As can be seen (Figures 5.14 and 5.15) the axial deflection of the drive end of the ballscrew, which is close to the drive end bearing, (blue trace, 4th line from the top) is settling down to about 0.75×10^{-7} m. The axial deflection of the ¼ point of the ballscrew (green trace, bottom line) to 0.8×10^{-7} m, the mid-point (red trace, third line from the top) to 0.55×10^{-7} m, and the ¾ point (cyan trace, second line from the top) to 0.3×10^{-7} m. Finally the axial deflection of the tail end of the ballscrew, which is close to the tail end bearing (magenta trace, top line) to 0.1×10^{-7} m. Therefore the axial behaviour of the screw as predicted by the model corresponds to that predicted by classical theory.

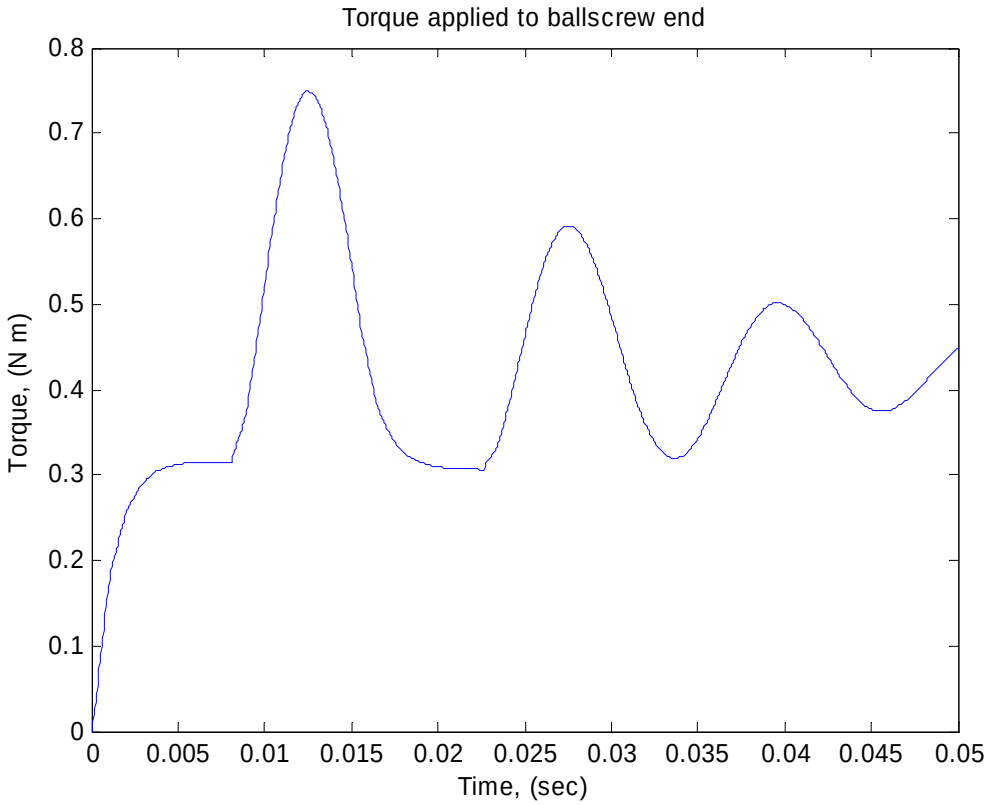


Figure 5.16 – *Torque applied at the driven end of the ballscrew*

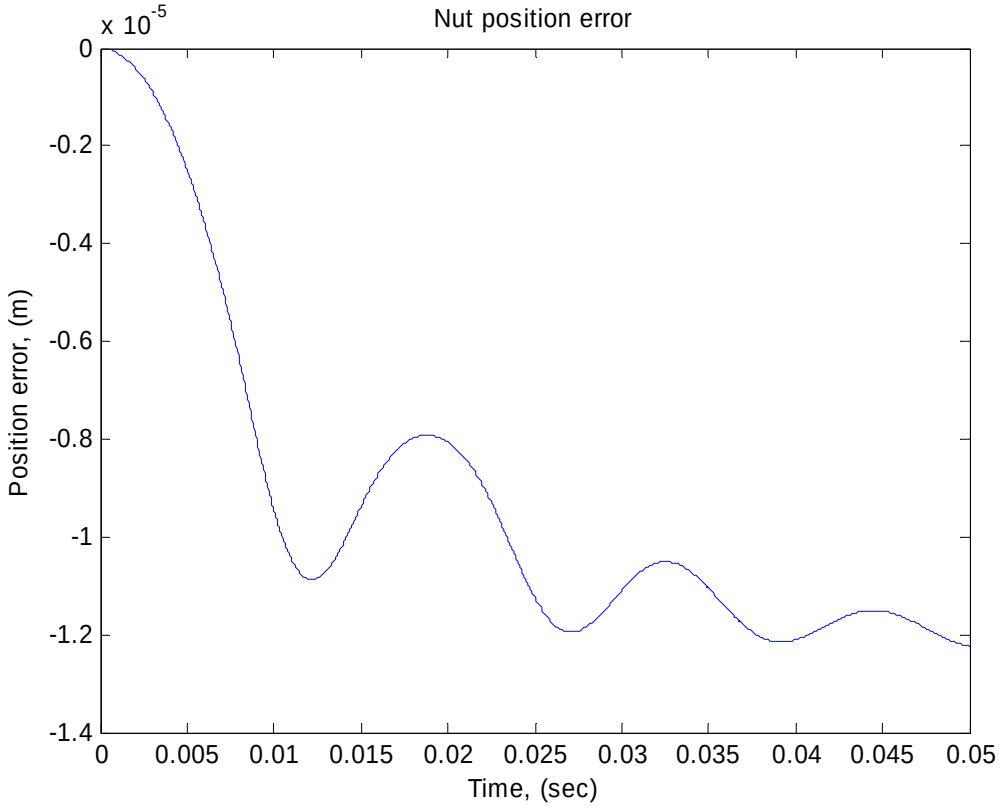


Figure 5.17 – *Ball nut position error*

The torque required to accelerate the table is 0.114 N.m and that needed to accelerate the screw is 0.300 N.m. The torque necessary to overcome viscous drag in the bearings is 0.004 N.m giving a total torque of 0.423 N.m. Figure 5.16 shows the torque settling down to around 0.42 N.m. The "square corner" at about 0.008 seconds and the "blip" at about 0.023 seconds coincide with the bumps, (c.f. Figures 5.14 and 5.15). These most likely arise from delays in response caused by the cycle time of the solution process which is typically 50 micro-seconds combined with the small time needed to compress the nut stiffness sufficient to establish full effective contact between the nut and the screw.

Figure 5.17 shows the position error of the nut. Hand calculations reveal that the contributions to this error can be expected to be – axial deflection of the screw 0.1 μm, axial deflection within the nut 0.7 μm, torsional wind up of the screw ~0 μm, backlash 5.0 μm and following error of the controller 6.7 μm, making 12.5 μm in total. The position error of the nut can be seen to be settling down to about 12 μm. The model thus checks out well against classical theory.

In order to include transverse vibration in the model it is necessary to include four extra degrees of freedom for each node of the ballscrew and to extend the requirements of the bearing sub-models. The phenomena of sagging of the shaft under gravity and pre-tension are also considered in Chapter 6.

Chapter 6 - TRANSVERSE CASE FOR MOVING MASS MODEL

The main function of a ballscrew is to transform rotary motion of the screw into linear motion of the nut. The degrees of freedom in which this motion takes place are the linear degrees of freedom along the direction of the screw ("x") and the rotational degrees of freedom about the axis of the screw ("θ_x"). In Chapter 5 a technique was developed which is capable of modelling a ballscrew's behaviour in these degrees of freedom. Transverse vibratory behaviour, although it does not impact directly on drive system errors, can affect the life of ballscrew components and the surface finish of products made on the machine tool of which the ballscrew is part. This chapter deals with these aspects of the dynamic behaviour of a ballscrew system.

6.1 Modelling the transverse behaviour of the screw

In order to account for transverse vibration, extra degrees of freedom must be accommodated. In its simplest case transverse vibration is a "two dimensional" phenomenon. The displacements are transverse displacement y and tilt angle θ. The inertia terms are mass and tilt inertia which can be represented by a matrix **M** such as

$$\mathbf{M}=\begin{pmatrix} m & 0 \\ 0 & J_{tilt} \end{pmatrix} \tag{6.1}$$

where m is the mass and J_{tilt} is the tilt inertia.

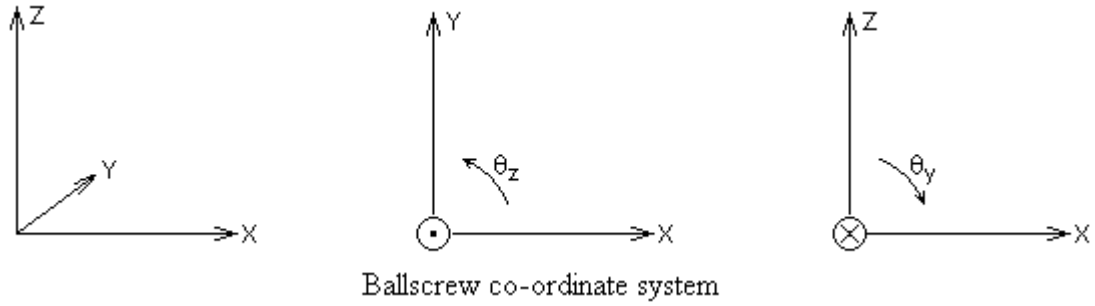


Figure 6.1 *The global reference axes*

Using a right hand set of Cartesian axes (see Figure 6.1), the elastic behaviour of a cantilever beam can be represented by the following set of equations, derived from classical beam theory [**100**]:-

$$y = \frac{l^3}{3EI}F + \frac{l^2}{2EI}\Gamma_z \tag{6.2a}$$

$$\theta_z = \frac{l^2}{2EI}F + \frac{l}{EI}\Gamma_z \tag{6.2b}$$

where E is Young's modulus, I is the second moment of area and l is the beam or element length. In matrix form:-

$$\mathbf{fl}=\begin{pmatrix} \frac{l^3}{3EI} & \frac{l^2}{2EI} \\ \frac{l^2}{2EI} & \frac{l}{EI} \end{pmatrix} \tag{6.3}$$

where **fl**(i,j) is the deflection of the *i*th degree of freedom caused by applying a force at the *j*th degree of freedom.

The stiffness of a cantilever beam can be represented by a 2 × 2 matrix such as **K** derived by inverting the flexibility matrix **fl**

$$\mathbf{K}=\begin{pmatrix} \frac{12EI}{l^3} & -\frac{6EI}{l^2} \\ -\frac{6EI}{l^2} & \frac{4EI}{l} \end{pmatrix} \tag{6.4}$$

Considering the forces at both ends of a beam element which are set up by applying a unit deflection at each degree of freedom in turn a 4 × 4 stiffness matrix representing the whole beam can be derived. A matrix of this kind forms a submatrix which represents an individual element in the system stiffness matrix typically as follows:-

$$\begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \mathbf{K} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} = \begin{pmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \frac{12EI}{l^3} & \frac{6EI}{l^2} & -\frac{12EI}{l^3} & \frac{6EI}{l^2} & \cdots \\ \cdots & \frac{6EI}{l^2} & \frac{4EI}{l} & -\frac{6EI}{l^2} & \frac{2EI}{l} & \cdots \\ \cdots & -\frac{12EI}{l^3} & -\frac{6EI}{l^2} & \frac{12EI}{l^3} & -\frac{6EI}{l^2} & \cdots \\ \cdots & \frac{6EI}{l^2} & \frac{2EI}{l} & -\frac{6EI}{l^2} & \frac{4EI}{l} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix} \tag{6.5}$$

One such submatrix should be added in for each spring element modelled.

6.2 Combining the transverse behaviour of the screw with the axial and torsional

Combining both Y and Z transverse degrees of freedom with the axial and torsional case dealt with in Chapter 5 leads to a 6 degree of freedom system. The mass matrix of the i^{th} node of the screw is given by

$$\mathbf{M}_i = \begin{pmatrix} m_i & 0 & 0 & 0 & 0 & 0 \\ 0 & m_i & 0 & 0 & 0 & 0 \\ 0 & 0 & m_i & 0 & 0 & 0 \\ 0 & 0 & 0 & J_{tors\,i} & 0 & 0 \\ 0 & 0 & 0 & 0 & J_{tilt\,i} & 0 \\ 0 & 0 & 0 & 0 & 0 & J_{tilt\,i} \end{pmatrix} \tag{6.6}$$

where J_{tors} is the torsional moment of inertia and J_{tilt} is the tilt or rocking moment of inertia (i.e. that about the transverse axes). The stiffness submatrix of a 6 degree of freedom spring element is shown in Figure 6.2, where G the shear modulus, A the cross-sectional area and K the torsional stiffness constant.

The bearings are modelled by attaching a set of springs at a node axially located at the centre of the bearing and "earthed" at the other end. The additional terms in the stiffness matrix are:-

$$\mathbf{K}_{+i} = \begin{pmatrix} k_{ax\,i} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{tr\,y\,i} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{tr\,z\,i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{tilt\,\theta y\,i} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{tilt\,\theta z\,i} \end{pmatrix} \tag{6.7}$$

where $k_{ax\,i}$ is the axial or thrust stiffness of the i^{th} bearing, $k_{tr\,y\,i}$ is the transverse or radial stiffness in the Y direction and $k_{tr\,z\,i}$ is that in the Z direction and $k_{tilt\,\theta y\,i}$, $k_{tilt\,\theta z\,i}$ are the tilt or rocking stiffnesses. i is the bearing number and can be 1 or 2.

In the case of the transverse degrees of freedom the stiffness matrix needs to be modified to avoid the phenomenon of discontinuous behaviour at element boundaries described in Chapter 5. The method of solution is essentially the same. If k_n is the radial stiffness of the nut and ϕ_n the tilt stiffness, then using the terminology of Chapter 5

$$\mathbf{p_{red}} = \mathbf{K} + (\mathbf{K_1}k_n + \mathbf{K_2}\phi_n + \mathbf{K_3}k_n\phi_n)/D \tag{6.8}$$

where

$$D = 1 + f^3(1-f)^3\frac{l^3}{3EI}k_n + f(1-f)(1-3f+3f^2)\frac{l}{EI}\phi_n + f^4(1-f)^4\frac{l^4}{12E^2I^2}k_n\phi_n \tag{6.9}.$$

and $\mathbf{K}$, $\mathbf{K_1}$, $\mathbf{K_2}$ and $\mathbf{K_3}$ are taken from Figures 6.2 and 6.3.

One of the most common ways of operating a ballscrew is the one where the shaft is rotated and the nut moves axially. This application is the one considered in this investigation.

$$\begin{pmatrix} \dots & \dots & \dots \\ \dots & \mathbf{K} & \dots \\ \dots & \dots & \dots \end{pmatrix} = \begin{pmatrix} \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & \frac{12EI_{zz}}{l^3} & 0 & 0 & 0 & \frac{6EI_{zz}}{l^2} & 0 & -\frac{12EI_{zz}}{l^3} & 0 & 0 & 0 & \frac{6EI_{zz}}{l^2} & \dots \\ \dots & 0 & 0 & \frac{12EI_{yy}}{l^3} & 0 & -\frac{6EI_{yy}}{l^2} & 0 & 0 & 0 & -\frac{12EI_{yy}}{l^3} & 0 & -\frac{6EI_{yy}}{l^2} & 0 & \dots \\ \dots & 0 & 0 & 0 & \frac{GK}{l} & 0 & 0 & 0 & 0 & 0 & -\frac{GK}{l} & 0 & 0 & \dots \\ \dots & 0 & 0 & -\frac{6EI_{yy}}{l^2} & 0 & \frac{4EI_{yy}}{l} & 0 & 0 & 0 & \frac{6EI_{yy}}{l^2} & 0 & \frac{2EI_{yy}}{l} & 0 & \dots \\ \dots & 0 & \frac{6EI_{zz}}{l^2} & 0 & 0 & 0 & \frac{4EI_{zz}}{l} & 0 & -\frac{6EI_{zz}}{l^2} & 0 & 0 & 0 & \frac{2EI_{zz}}{l} & \dots \\ \dots & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 0 & -\frac{12EI_{zz}}{l^3} & 0 & 0 & 0 & -\frac{6EI_{zz}}{l^2} & 0 & \frac{12EI_{zz}}{l^3} & 0 & 0 & 0 & -\frac{6EI_{zz}}{l^2} & \dots \\ \dots & 0 & 0 & -\frac{12EI_{yy}}{l^3} & 0 & \frac{6EI_{yy}}{l^2} & 0 & 0 & 0 & \frac{12EI_{yy}}{l^3} & 0 & \frac{6EI_{yy}}{l^2} & 0 & \dots \\ \dots & 0 & 0 & 0 & -\frac{GK}{l} & 0 & 0 & 0 & 0 & 0 & \frac{GK}{l} & 0 & 0 & \dots \\ \dots & 0 & 0 & -\frac{6EI_{yy}}{l^2} & 0 & \frac{2EI_{yy}}{l} & 0 & 0 & 0 & \frac{6EI_{yy}}{l^2} & 0 & \frac{4EI_{yy}}{l} & 0 & \dots \\ \dots & 0 & \frac{6EI_{zz}}{l^2} & 0 & 0 & 0 & \frac{2EI_{zz}}{l} & 0 & -\frac{6EI_{zz}}{l^2} & 0 & 0 & 0 & \frac{4EI_{zz}}{l} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \quad [103]$$

Figure 6.2 Six degree of freedom matrix for a beam element representing part of the ballscrew

$$\begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \mathbf{K} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} = \begin{pmatrix} \frac{12EI_{zz}}{l^3} & \frac{6EI_{zz}}{l^2} & -\frac{12EI_{zz}}{l^3} & \frac{6EI_{zz}}{l^2} & 0 & 0 \\ \cdots & \frac{4EI_{zz}}{l} & -\frac{6EI_{zz}}{l^2} & \frac{2EI_{zz}}{l} & 0 & 0 \\ \cdots & \cdots & \frac{12EI_{zz}}{l^3} & -\frac{6EI_{zz}}{l^2} & 0 & 0 \\ \cdots & \cdots & \cdots & \frac{4EI_{zz}}{l} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & k_n & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \phi_n \end{pmatrix}$$

Figure 6.3a Y and θ_z degree of freedom matrix for the beam element representing part of the ballscrew in contact with nut
Terms in I_{zz}, k_n and ϕ_n

$$\begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & \mathbf{K}_1 & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} = \begin{pmatrix} (1-f)^3(1+3f) & f(1-f)^3(1+f)l & 3f^2(1-f)^2 & -f^2(1-f)^3 l & -(1-f)^2(1+2f) & 0 \\ \cdots & \frac{1}{3}f^2(1-f)^3(3+f)l^2 & f^3(1-f)^2 l & -\frac{1}{3}f^3(1-f)^3 l^2 & -f(1-f)^2 l & 0 \\ \cdots & \cdots & f^3(4-3f) & -f^3(1-f)(2-f)l & -f^2(3-2f) & 0 \\ \cdots & \cdots & \cdots & \frac{1}{3}f^3(1-f)^2(4-f)l^2 & f^2(1-f)l & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix} k_n / D$$

Figure 6.3b Y and θ_z degree of freedom matrix for the beam element representing part of the ballscrew in contact with nut
Additional terms in k_n

where $D = 1 + f^3(1-f)^3 \frac{l^3}{3EI} k_n + f(1-f)(1-3f+3f^2)\frac{l}{EI}\phi_n + f^4(1-f)^4 \frac{l^4}{12E^2I^2} k_n \phi_n$.

$$\begin{pmatrix} \dots & \dots & \dots \\ \dots & \mathbf{K}_2 & \dots \\ \dots & \dots & \dots \end{pmatrix} = \begin{pmatrix} \frac{12f(1-f)}{l^2} & \frac{6f^2(1-f)}{l} & -\frac{12f(1-f)}{l^2} & \frac{6f(1-f)^2}{l} & 0 & \frac{6f(1-f)}{l} \\ \dots & (1-f)(1-3f+3f^2+3f^3) & -\frac{6f^2(1-f)}{l} & 3f^2(1-f)^2 & 0 & (1-f)(3f-1) \\ \dots & \dots & \frac{12f(1-f)}{l^2} & -\frac{6f(1-f)^2}{l} & 0 & -\frac{6f(1-f)}{l} \\ \dots & \dots & \dots & f(4-12f+12f^2-3f^3) & 0 & f(2-3f) \\ \dots & \dots & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & 0 \end{pmatrix} \phi_n / D$$

Figure 6.3c Y and θ_z degree of freedom matrix for the beam element representing part of the ballscrew in contact with nut

Additional terms in ϕ_n

[105]

$$\begin{pmatrix} \dots & \dots & \dots \\ \dots & \mathbf{K}_3 & \dots \\ \dots & \dots & \dots \end{pmatrix} = \begin{pmatrix} f(1-f)^4\frac{l}{EI_{zz}} & f^2(1-f)^4\frac{l^2}{2EI_{zz}} & 0 & 0 & -f(1-f)^4\frac{l}{EI_{zz}} & f^2(1-f)^4\frac{l^2}{2EI} \\ \dots & f^3(1-f)^4\frac{l^3}{3EI_{zz}} & 0 & 0 & -f^2(1-f)^4\frac{l^2}{2EI_{zz}} & f^3(1-f)^4\frac{l^3}{6EI} \\ \dots & \dots & f^4(1-f)\frac{l}{EI_{zz}} & -f^4(1-f)^2\frac{l^2}{2EI_{zz}} & -f^4(1-f)\frac{l}{EI_{zz}} & -f^4(1-f)^2\frac{l^2}{2EI} \\ \dots & \dots & \dots & f^4(1-f)^3\frac{l^3}{3EI_{zz}} & f^4(1-f)^2\frac{l^2}{2EI_{zz}} & f^4(1-f)^3\frac{l^3}{6EI} \\ \dots & \dots & \dots & \dots & f(1-f)(1-3f+3f^2)\frac{l}{EI_{zz}} & f^2(1-f)^2(2f-1)\frac{l^2}{2EI} \\ \dots & \dots & \dots & \dots & \dots & f^3(1-f)^3\frac{l^3}{3EI_{zz}} \end{pmatrix} k_n\phi_n / D$$

Figure 6.3d Y and θ_z degree of freedom matrix for the beam element representing part of the ballscrew in contact with nut

Additional terms in $k_n \phi_n$

In addition to the degrees of freedom considered in Chapter 5, others need to be included to account for the transverse movement of the nut. In total the degrees of freedom used in modelling the nut are:-

- An axial degree of freedom to represent movement of the nut in the X direction,
- Two transverse degrees of freedom to represent movement in each of the Y and Z directions, and
- Two rotational degrees of freedom to represent tilt movement about the Y and Z axes

The transverse support stiffness of the slideways which carry the nut's weight is modelled by adding $k_{slide\ y}$ to the diagonal term of the global stiffness matrix which relates to the degree of freedom representing Y movement of the nut and $k_{slide\ z}$ to the corresponding Z term.
Similarly, the tilt support stiffness is modelled by adding $k_{slide\ \theta y}$ and $k_{slide\ \theta z}$ to the respective diagonal term relating to the degrees of freedom representing tilting motion about the Y and Z axes.

6.3 Initial conditions

Since most ballscrews are used close to the surface of the earth they are subject to a gravitational field of one "g". Using the global axes shown earlier, this acts in the –Z direction. One consequence of this is that the screw sags when at rest. The extent of this sag can be determined by solving the equation

$$\mathbf{Kx} = \mathbf{Mg} \tag{6.10}$$

where

- **K** is the system stiffness matrix
- **x** is the global displacement vector, a column containing the displacements of each degree of freedom in the system. For the screw this is a succession of Cartesian coordinates $\{.., x_i, y_i, z_i, \theta_{xi}, \theta_{yi}, \theta_{zi}, ..\}$ for each node of the screw.
- **M** is the mass matrix is built up from terms like those defined in equation (6.6)
- **g** is the gravitational field vector, a column consisting of $-g$ for all the Z degrees of freedom and zero elsewhere. g is, of course, the acceleration due to gravity.

A detailed examination of the stiffness matrix reveals that it can be partitioned into one $\mathbf{K_z}$ involving the X, Z, Θ_x and Θ_y degrees of freedom and one $\mathbf{K_r}$ involving "the rest" as follows:-

$$\begin{pmatrix} \mathbf{K_z} & \mathbf{0} \\ \mathbf{0} & \mathbf{K_r} \end{pmatrix} \begin{pmatrix} \mathbf{x_z} \\ \mathbf{x_r} \end{pmatrix} = \begin{pmatrix} \mathbf{M_z} & \mathbf{0} \\ \mathbf{0} & \mathbf{M_r} \end{pmatrix} \begin{pmatrix} \mathbf{g_z} \\ \mathbf{0} \end{pmatrix} \tag{6.11}$$

The diagonal mass matrix **M** is likewise partitioned into $\mathbf{M_z}$ and $\mathbf{M_r}$ and so also with the vector **g**. This gives the equations

$$\begin{aligned} \mathbf{K_z x_z} &= \mathbf{M_z g_z} = \mathbf{F} \\ \mathbf{K_r x_r} &= \mathbf{M_r 0} \end{aligned} \tag{6.12a) and (6.12b}$$

F is a force vector consisting of $-m_i \times g$ at all the Z degrees of freedom and zero elsewhere. This gives a means of deriving the sag displacement (Z freedoms) and slope (Θ_y freedoms) using a standard method of solving linear algebraic equations. The one used in this investigation is LU decomposition [**101**].

At first glance it would appear that only the Z and Θy degrees of freedom need be included in $\mathbf{K_z}$. The X and Θ_z degrees of freedom are included because the tilt of the ballscrew in the nut causes the saddle or table to tilt about the slideways. This can cause an axial movement of the nut which in turn can cause a rotation of the screw. Although these effects are small, if neglected they can give rise to a small "kick" at the beginning of the motion predicted by the model.

The remaining displacements $\mathbf{x_r}$ are zero if $|\mathbf{K_r}| \neq 0$ and the diagonal terms of $\mathbf{M_r}$ are non-zero. The two conditions are true for physically realistic systems and the result is to be expected.

When there is pre-tension in the ballscrew, tilt moments occur. The extent of these moments is dependent on the deflected shape of the screw, but the moments themselves have an effect on the deflected shape. For the *i*th node in the ballscrew the titling moment $M_{y\,i}$ is given by:-

$$M_{y\,i} = T_+ \times (z_i - z_{i-1}) + T_- \times (z_{i+1} - z_i) \tag{6.13}$$

where T_+ is the tension in the element on the "upstream" or driving end side of the node and T_- is the tension in the element on the "downstream" side. If a pre-tension T of the typical order used in machine tool applications is set into the ballscrew, then $T_- \approx T_+ \approx T$. The small difference which can arise between T_- and T_+ occur because of axial transient or vibratory behaviour of the screw. When calculating the initial condition, $T_- = T_+ = T$.

The deflected shape under pre-tension is derived as follows. First the deflected shape with no pre-tension $\{z_i\}_0$ is calculated using equation (6.12a). Then the tilt moments are derived and included in the load vector. Equation (6.12a) is solved again to give $\{z_i\}_1$. A measure of

the difference between the two shapes is obtained by adding the square of the difference in deflection for each of the nodes:-

$$S_{dz2} = \sum (z_{i1} - z_{i0})^2 \tag{6.14}$$

and a measure of the "size" of the shape is based on:-

$$S_{z2} = \sum z_{i0}^{\ 2} \tag{6.15}.$$

$\{z_i\}_0$ is an over-estimate of the required deflections and $\{z_i\}_1$ is an under-estimate. An improved estimate can be obtained by using the tilt moments derivable from $\{z_i\}_1$ and solving again. The difference measure is now:-

$$S_{dz2} = \sum (z_{i1} - z_{i0})^2 \tag{6.16}$$

The deflected shape can be improved progressively, a satisfactory solution being considered to have been obtained when:-

$$\frac{S_{dz2}}{S_{z2}} < \varepsilon = 10^{-20} \tag{6.17}.$$

The sag displacements are used as the "initial values" in calculating the dynamic behaviour of the ballscrew system.

6.4 Developing the transverse motion of the ballscrew system

The sag induced by gravity sets up a distribution of strains and associated potential energy within the body of the screw. When the screw rotates these strains are carried round with it. Therefore, one approach to the analysis of the subsequent behaviour of the screw is to use a set of local axes which rotate with the screw, (see Figure 6.4). In the cases considered here there are more degrees of freedom associated with the screw than with the rest of the system put together. To start with, it is therefore considered to be more convenient, mathematically speaking at least, to rotate the universe than to rotate the screw. This means that the following changes have to be made to the terms in the modelling process.

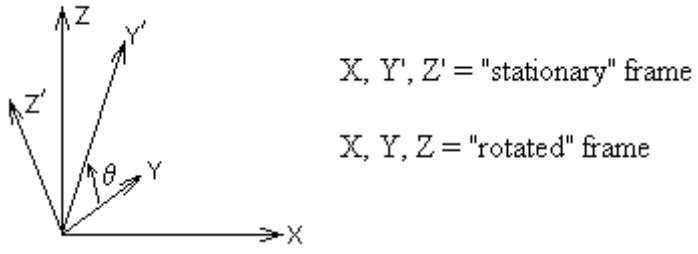


Figure 6.4 *Axes rotated with ballscrew*

First the bearing stiffness must be rotated in a direction opposite to the screw:-

$$\mathbf{K}_{+ib} = \begin{pmatrix} \mathbf{K}_{\mathbf{lin}} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{\mathbf{ang}} \end{pmatrix} \tag{6.18a}$$

where

$$\mathbf{K}_{\mathbf{lin}} = \begin{pmatrix} k_{ax\,ib} & 0 & 0 \\ 0 & k_{tr\,y\,ib}c2\theta + k_{tr\,z\,ib}s2\theta & -(k_{tr\,y\,ib} - k_{tr\,z\,ib})sc\theta \\ 0 & -(k_{tr\,y\,ib} - k_{tr\,z\,ib})sc\theta & k_{tr\,y\,ib}s2\theta + k_{tr\,z\,ib}c2\theta \end{pmatrix}$$

$$\mathbf{K}_{\mathbf{ang}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & k_{tilt\,\theta\,y\,ib}c2\theta + k_{tilt\,\theta\,z\,ib}s2\theta & -(k_{tilt\,\theta\,y\,ib} - k_{tilt\,\theta\,z\,ib})sc\theta \\ 0 & -(k_{tilt\,\theta\,y\,ib} - k_{tilt\,\theta\,z\,ib})sc\theta & k_{tilt\,\theta\,y\,ib}s2\theta + k_{tilt\,\theta\,z\,ib}c2\theta \end{pmatrix} \tag{6.18b}$$

and

$$\begin{aligned} c2\theta &= \cos^2\theta_{x\,ib} \\ s2\theta &= \sin^2\theta_{x\,ib} \\ sc\theta &= \sin\theta_{x\,ib}\cos\theta_{x\,ib} \end{aligned} \tag{6.18c}$$

$\theta_{x\,ib}$ is the angular position of the ballscrew in the bearing.

Similarly the terms representing the support of the slideways must also be "rotated".

$$\mathbf{K}^{+}_{slide} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & k_{slide\,y}c2\theta + k_{slide\,z}s2\theta & -(k_{slide\,y} - k_{slide\,z})sc\theta \\ 0 & -(k_{slide\,y} - k_{slide\,z})sc\theta & k_{slide\,y}s2\theta + k_{slide\,z}c2\theta \end{pmatrix} \tag{6.19}$$

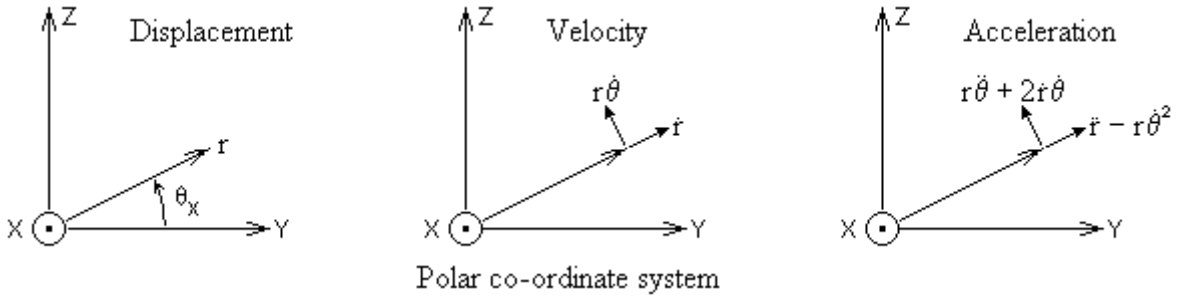


Figure 6.5 Polar coordinate system

The degrees of freedom of the screw have to be treated specially because they are in a "rotating world". Three consequences arise.

- First, considering the rate of change of angular momentum of a node gives an expression for torque Γ about the X axis, (see Figure 6.5)

$$\begin{aligned} \Gamma &= \frac{\partial}{\partial t}(J_{xx}\dot{\theta}_x) = \frac{\partial}{\partial t}((J_0 + mr^2)\dot{\theta}_x) = J_0\ddot{\theta}_x + m\dot{\theta}_x\frac{\partial}{\partial t}(r^2) \\ &= J_{xx}\ddot{\theta}_x + 2mr\dot{r}\dot{\theta} \end{aligned} \tag{6.20}$$

  or

$$J_{xx}\ddot{\theta}_x = \Gamma - 2m\,r\,\dot{r}\,\dot{\theta}$$
$$\text{where } r = \sqrt{y^2 + z^2} \tag{6.21}$$

where m is the mass, J_0 is the torsional moment of inertia about an axis through the centre of mass and parallel to the X axis and J_{xx} is the moment of inertia about the X axis. Therefore the inertia term has to be increased as the centre of the node moves off centre and a "Coriolis" term has to be introduced into the force vector

- Secondly the additional terms in the acceleration vector give rise to extra terms in the "y" and "z" columns of the force vector at each node of the ballscrew
- Thirdly the gravitational force term needs to be "rotated" as the screw rotates.

The combined effect of these factors gives the force vector for node i as

$$\mathbf{F}_i = \begin{pmatrix} F_{a\,i} \\ m_i g \sin\theta_{x\,i} + y_i\dot{\theta}_{x\,i}^2 + y_i\ddot{\theta}_{x\,i} + 2\dot{y}_i\dot{\theta}_{x\,i} \\ -m_i g \cos\theta_{x\,i} + z_i\dot{\theta}_{x\,i}^2 - z_i\ddot{\theta}_{x\,i} - 2\dot{z}_i\dot{\theta}_{x\,i} \\ \Gamma_{e\,i} \\ 0 \\ 0 \end{pmatrix} \tag{6.22}$$

where

- $F_{a\,i}$ is the external axial force
- m_i is the mass at the node
- y_i, z_i are the components of the displacement r_i in the local X and Y directions respectively. Their derivatives with respect to time are the corresponding velocities
- $\theta_{x\,i}$ is the torsional position. The first and second derivatives with respect to time are the angular velocity and acceleration. It should be noted that the position is in terms of the global (i.e. stationary) axes. When this method of analysis is being used the acceleration is calculated by using the velocities from the previous two cycles of analysis. Provided the results are stable and the time step is small, this should not introduce much error
- $\Gamma_{e\,i}$ is the external torque.

The approach to rotating the shaft outlined above was found to have a significant drawback, namely that it assumes that the screw rotates about the "x" axis. This is not normally the case. At rest the screw assumes a deflected shape dependent on its weight and the support provided by the bearings. If the screw turns slowly, its centre line stays in the same deflected shape (apart from changes caused by repositioning of the support provided by the nut). As speed

increases there is an increased tendency for the screw to move like a skipping rope in which case the centre of rotation does move closer to the "x" axis.

Because the screw does not, in general, rotate around the "x" axis, it is better to use a reference frame which is fixed relative to earth. The transformations described in equations (6.18), (6.19) and (6.22) are not therefore needed.

6.5 Gyroscopic effect

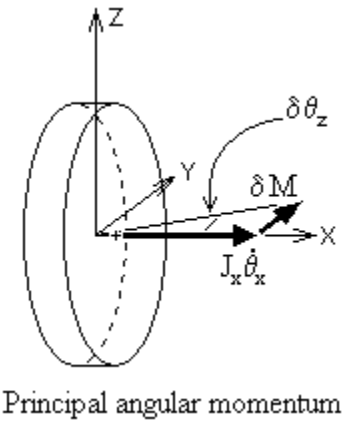


Figure 6.6 *Gyroscopic effects*

A further consideration is the so-called "gyroscopic effect". When a shaft is rotating, if it is subject only to small transverse vibrations, by far the greatest component of the angular momentum vector is that parallel to the axis of rotation, (see Figure 6.6). In this case the angular momentum M_i is given by

$$M_i = J_{xi}\dot{\theta}_{xi} \tag{6.23}$$

where J_x is the polar moment inertia of the node.

Now consider a small change of the angular momentum vector δM. Let this be in the Y direction and take place over time interval δt. This gives rise to a torque Γ_y as follows:-

$$\delta M = M.\delta\theta_z = \Gamma_y \delta t \tag{6.24}$$

Dividing throughout by δt and letting $\delta t \to 0$ gives:-

$$\Gamma_y = M\frac{\partial\theta_z}{\partial t} = J_x\frac{\partial\theta_x}{\partial t}\frac{\partial\theta_z}{\partial t} \tag{6.25}$$

If δM occurs in the Z direction then:-

$$\delta M = M.-\delta\theta_y = \Gamma_z \delta t \tag{6.26}$$

and:-

$$\Gamma_z = -M\frac{\partial\theta_y}{\partial t} = -J_x\frac{\partial\theta_x}{\partial t}\frac{\partial\theta_y}{\partial t} \tag{6.27}$$

These torques are "active" on the node inertia, therefore to allow for gyroscopic effects extra terms need to be added to the force vector as follows:-

$$\mathbf{F}_{gyr} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -J_x\dfrac{\partial\theta_x}{\partial t}\dfrac{\partial\theta_z}{\partial t} \\ J_x\dfrac{\partial\theta_x}{\partial t}\dfrac{\partial\theta_y}{\partial t} \end{pmatrix} \tag{6.28}$$

6.6 Ballscrew orbits

It has been found that the trajectory of a node on the ballscrew is a curve that can be approximated to an ellipse, which in turn can be thought of as the sum of a displacement in the YZ plane, a transverse vibration along some line in that plane, and a circular motion.

In the case of transverse vibration, the acceleration vector is parallel to the velocity vector, but in the case of circular motion, acceleration acts at right angles to velocity. This gives a means of obtaining a local centre of motion. First the acceleration amplitude a_n and its direction ϕ_a from the Y axis of a node are derived from the components in the Y and Z directions:-

$$a_{n\,i} = \sqrt{\ddot{y}_i^2 + \ddot{z}_i^2} \tag{6.29}$$

$$\phi_a = \tan^{-1}\frac{\ddot{z}_i}{\ddot{y}_i} \tag{6.30}$$

Then similarly, the velocity amplitude v_n and direction angle ϕ_v are obtained:-

$$v_{n\,i} = \sqrt{\dot{y}_i^2 + \dot{z}_i^2} \tag{6.31}$$

$$\phi_v = \tan^{-1}\frac{\dot{z}_i}{\dot{y}_i} \tag{6.32}$$

The angle between them enables the "circular" component of the acceleration a_{circ} to be derived from:-

$$a_{circ\,i} = a_{n\,i}\sin(\phi_a - \phi_v) \tag{6.33}$$

The radius r_{circ} of the motion then follows from:-

$$r_{circ\,i} = \frac{a_{circ\,i}}{\omega^2} \qquad (6.34),$$

where ω is the angular speed of the screw.

It had been intended to us this information to up-date the moment of inertia of the screw nodes, and introduce a centrifugal force component to the dynamic equations. However, it was found that for the examples used to test the C program these effects were quite small and were neglected.

6.7 Bearing cap vibrations

One final consideration. It is often the case that vibration measurements using accelerometers or velocity transducers are taken on parts of a structure close to the bearing caps in order to monitor a machine's condition. This can be more convenient to measure. For example, a shaft order vibration of 0.1 μm at 1000 rpm could prove difficult to detect, but this gives an acceleration of $0.1\times10^{-6}\times\left(\frac{2\pi\times1000}{60}\right)^2\times1000 = 1.10$ mm/sec^{-2} which should be quite easy.

In order that bearing support movement can be predicted additional terms should be included in the model, (see Figure 6.7).

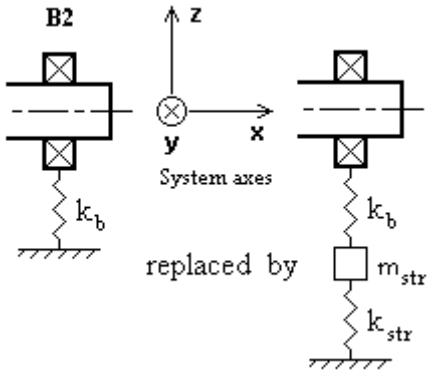


Figure 6.7 *Modelling bearing cap*

At each bearing support point five additional degrees of freedom are included – y, z, θy, θz. The mass matrix is extended to include an effective mass $m_{str\,x}$, $m_{str\,y}$ and $m_{str\,z}$ and an effective tilt inertia $j_{str\,y}$ and $j_{str\,z}$.

$$\mathbf{M}_{str\,i} = \begin{pmatrix} m_{x\,str\,i} & 0 & 0 & 0 & 0 & 0 \\ 0 & m_{y\,str\,i} & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{z\,str\,i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & j_{y\,str\,i} & 0 \\ 0 & 0 & 0 & 0 & 0 & j_{z\,str\,i} \end{pmatrix} \tag{6.35}$$

Including the $\mathbf{K_{+i}}$ matrix defined by equation (6.7), the additional terms in the stiffness matrix which represent the bearings and their support structures are

$$\begin{pmatrix} +\mathbf{K}_{+\mathbf{i}} & -\mathbf{K}_{+\mathbf{i}} \\ -\mathbf{K}_{+\mathbf{i}} & \mathbf{K}_{+\mathbf{i}} + \mathbf{K}_{\mathbf{str\,i}} \end{pmatrix} \tag{6.36}$$

where

$$\mathbf{K}_{str\,i} = \begin{pmatrix} k_{x\,str\,i} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{y\,str\,i} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{z\,str\,i} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{\theta y\,str\,i} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{\theta z\,str\,i} \end{pmatrix} \tag{6.37}$$

6.8 Out-of-balance excitation

An important form of excitation for transverse vibration is out-of-balance of the screw. This can be introduces at any node *i* by an additional force term of:-

$$\mathbf{F}_i = \begin{pmatrix} 0 \\ B_i \dot{\theta}_{x\,i}^{\,2} \cos\theta_{x\,i} \\ B_i \dot{\theta}_{x\,i}^{\,2} \sin\theta_{x\,i} \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{6.38}$$

where B_i is the out-of-balance. It has dimension of "mass×distance". A mass of 50 g at a radial distance of 20 mm from the centre of the ballscrew, which is typical of the levels of out-of-balance used on the test rig, was used. The behaviour shown on Figure 6.8 is predicted by the model. On this figure five trajectories of the ballscrew centre are plotted in the transverse (YZ) plane. These are:-

- ballscrew driven end (blue trace),
- ballscrew 1/4 point (green trace)
- ballscrew mid-point (red trace)

- ballscrew 3/4 point (cyan point)
- ballscrew non-drive end (black trace)

The motion starts with the screw sagging under its own gravitational load. Thus the points all lie in the y=0 plane. As the screw picks up speed, the driven end, the 1/4 point and the 3/4 point follow spiral trajectories. The movement predicted for the non-drive end and the mid-point, which is restrained by the ball nut, is very small.

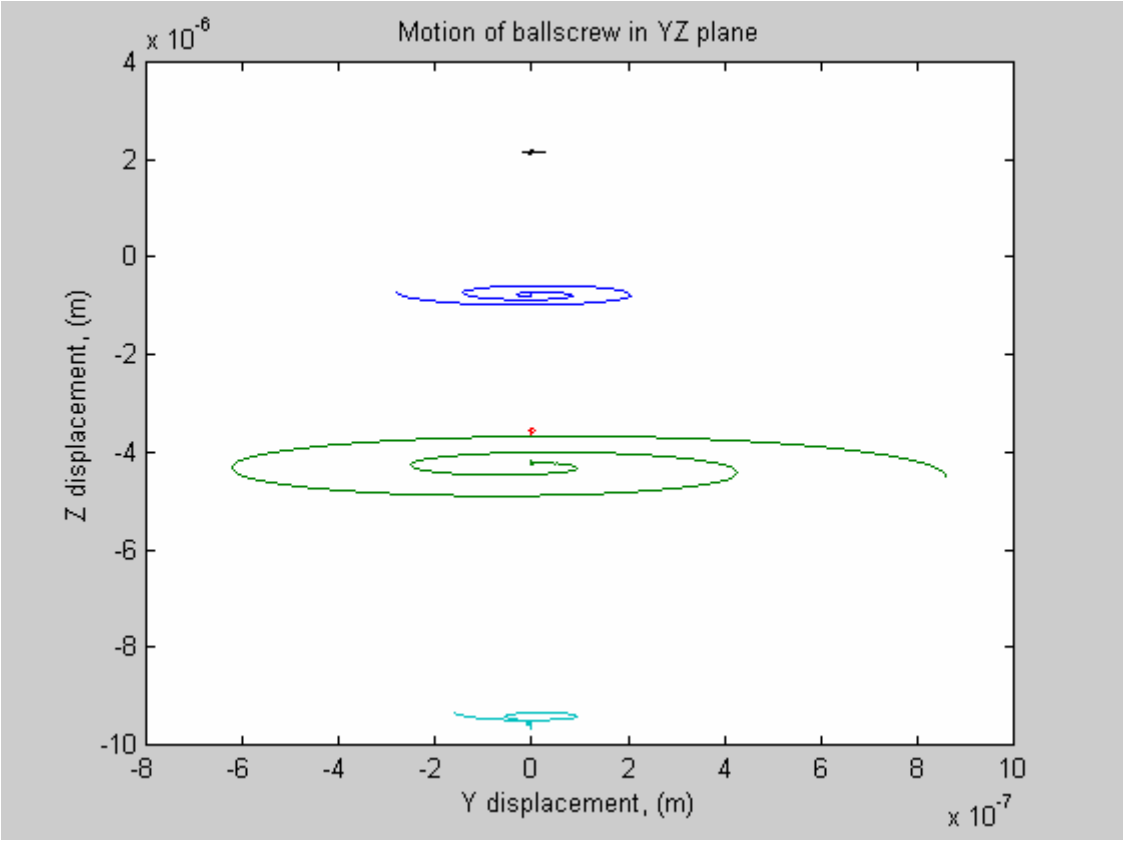


Figure 6.8 *Trajectory of ballscrew centre under out-of-balance load*

Comparison with measured vibration levels is deferred to Chapter 8.

The basic layout of the C program which predicts axial, torsional and transverse dynamic behaviour of a ballscrew drive system is included in Appendix 6.1, together with a typical example of the data used.

The models of the dynamic behaviour described in Chapter 5 and this chapter include features which calculate energy dissipated by the ballscrew as it operates. This energy is converted to heat and can be used as energy sources in the thermal model described in the next chapter.

Chapter 7 - THERMAL CONSIDERATIONS

A thermal model of a ballscrew system needs to take into account where and how heat is generated in the system, where and how heat is dissipated and how the heat gets from one part of the system to another. The heat sources are the ballscrew nut and the screw support bearings, and the heat sinks are the machine part in which the bearings are mounted, the machine part to which the nut is attached and the surrounding air. In addition, the means by which the screw is driven may be either a source or a sink or both.

In the case of the ball nut, the heat generation mechanisms are expected to be friction in the rolling action, hysteresis associated with repeated compression and decompression of the nut, screw and ball material, viscous effects, churning of the lubricant and seal friction. In addition, the acceleration and deceleration of the balls as they pass from the recirculation paths to the screw helix also dissipates mechanical energy as heat. These factors in turn are affected by the number, size and quality of the rolling elements, the surface finish and circularity of the screw and nut races, the lubrication type, performance and temperature, pre-load and pre-tension, misalignment and the type of seal.

In the case of the bearings also, the heat generation mechanisms are expected to be friction in the rolling action, hysteresis associated with repeated compression and decompression of the race and ball or roller material, viscous effects, churning of the lubricant and seal friction. Again, these factors in turn are affected by the bearing type (ball, roller, angular contact etc), the number, size and quality of the rolling elements, the surface finish and circularity of the races, the effect of caging, the lubrication type, performance and temperature, pre-load, misalignment and the type of seal.

The means of dissipating the heat are radiation, convection and conduction. The principal means of transferring heat from one part of the system to another is conduction.

Dealing first with **radiation**; heat is a form of energy which in matter - solid, liquid or gas - is the kinetic energy of the particles which make up the matter. This energy can also take the form of electromagnetic particle/wave quanta with a characteristic wavelength in the range of 10^{-4} m to 7×10^{-7} m [**102**]. In this form heat travels through space at the speed of light. Such energy is known as **radiant heat** and the law of physics which governs radiation from a surface of a body is the Stefan Boltzmann law [**103**], which can be stated as:-

$$q = \varepsilon \sigma T^4 - \alpha \sigma T_a^4 \tag{7.1}$$

where q is the nett heat flux from the surface and T is the absolute temperature of the surface under consideration. T_a is a temperature representative of the surroundings into which the

surface is radiating heat. In the case of a machine tool drive, typically a ballscrew is mounted inside a bed, saddle or column surrounded by guarding designed to keep swarf away from the machine's mechanism. In such cases the average temperature of the surfaces surrounding the ballscrew would serve. ε is the emissivity of the surface, that is the heat radiating ability of a surface compared to that of an ideal black body, and α is the absorptivity of a surface, which is the surface's heat absorbing ability compared to that of an ideal black body. Both ε and α are measures of the efficiency of a surface in transferring energy between the kinetic energy of the molecules of the body and quanta of radiant heat. $\sigma = 5.672\times10^{-8}$ W/(m^2 K^4) [**104**] is the Stefann-Boltzmann constant, a fundamental property of the universe.

Commonly, because the temperature of the emitting surface is significantly higher than that of the surroundings, and therefore the range of wavelengths in which the greater part of the emitted energy is different from that of the energy absorbed, the emissivity is different from the absorptivity [**105**]. However, since the temperature reached by the surface of the ballscrew is only of the order of tens of degrees above that of the cavity in which it operates, the absorptivity can be regarded as the same as the emissivity and so equation (7.1) can be rewritten as:-

$$q = \varepsilon\,\sigma\,(T^4 - T_a^4) \tag{7.2}$$

Factorising $(T^4 - T_a^4)$ a heat transfer coefficient h_r can be derived as follows:-

$$h_r = \frac{q}{\Delta T} = \varepsilon\,\sigma\,(T^2 + T_a^2)(T + T_a)(T - T_a)/\Delta T \tag{7.3a}$$

If $(T - T_a) << (T + T_a)$, equation (7.3a) can be replaced by:-

$$h_r = \frac{q}{\Delta T} \approx 4\varepsilon\,\sigma\,(\frac{T + T_a}{2})^3 \tag{7.3b}$$

Using a typical value of 0.20 for the emissivity of steel in the polished state which is normal in the case of a ballscrew [**106**], assuming an ambient temperature of 20°C and a ballscrew surface temperature of 35 C gives:-

$$h_r = 4\times0.70\times5.672\times10^{-8}\times\left(\frac{(273+35)+(273+20)}{2}\right)^3 = 4.31 \text{ W/(m}^2\text{ K)} \tag{7.4}$$

When a surface of a solid body is covered by fluid which is at a different temperature to that of the solid close to the surface, heat is conducted from the solid into the fluid. This causes the fluid close to the surface to be at a temperature different from that of the bulk of the fluid. The resulting changes in density cause differential buoyancy effects which set the fluid in motion. The effect of such motion is to replace the fluid whose temperature is different from the bulk temperature of the fluid by some whose temperature is closer to the bulk temperature. The nett

effect of this is to transport heat to or from the surface of the solid at a rate greater than could be achieved by conduction. This process is known as **natural convection** and the motion in the fluid is called a convection current.

If the solid body is moving through the fluid, or if the fluid is caused to move across the body's surface, the motion of the fluid is at a rate greater than that which would be set up by differential buoyancy, and the rate of heat transfer to the fluid can be expected to be greater. This process is known as **forced convection**.

Analysis of these phenomena is fraught with difficulties. The convection currents can be **laminar** where adjacent layers within the fluid slide over one another, or **turbulent** where the flow includes lots of eddies acting in a chaotic manner. Generally speaking laminar flow occurs at low fluid speeds and turbulent flow at high speed. Especially in the case of natural convection, the amount of space in which convection currents can be established has an effect on their speed and behaviour. The density, viscosity and thermal conductivity of the fluid all have an influence on convective behaviour. All of these physical properties are affected by temperature, viscosity especially so. The flow rate at which turbulent behaviour is expected to start is determined using the dimensionless Reynolds number, *Re*. For flow where the speed of the bulk of the fluid is u_∞ and ν the kinematic viscosity, for a surface of characteristic length L

$$\mathrm{Re} = \frac{u_\infty L}{\nu} \tag{7.5}$$

In physical terms, the Renolds number can be thought of as the ratio of the inertia force to the viscous force. The onset of turbulence can occur when $10^5 < \mathrm{Re} < 4 \times 10^6$. For a flat plate the critical Reynolds number is commonly taken to be 5×10^5 [**107**].

In many convection problems, a **boundary layer** is considered to exist close to the surface in which the flow is laminar. The flow in the fluid beyond this layer is considered to be turbulent. The thickness of this layer has an important influence on the heat dissipating ability of a surface.

The behaviour of a ballscrew system is especially complex. Convection can be expected to occur at the surface of the bearing housings, the ball nut and the ballscrew itself. The ballscrew rotates, thus generating relative motion between itself and the surrounding air. The nut also moves through the air generating a flow across its surface as well as displacing air in front of it and entraining air behind it. Nevertheless, the mounting arrangements of the nut flange are often such that the side of the nut closest to the body it is mounted to is sheltered to some extent from windage generated by movement of the nut. Lots of work has been done on the convective properties of flat plates at various orientations to the vertical, on cylindrical tubes and on banks

of cylindrical tubes. The case of a ballscrew does not fit conveniently into any of these categories and so estimates of the convective behaviour have had to be made.

Heat in a body is stored in the kinetic energy of its constituent particles, the more vigorous the motion of the atoms etc., the higher the temperature. If one part of the body is at a higher temperature than another, there is a general tendency for the internal energy of the body to become shared out more evenly. This means that, in effect, energy is transferred from hotter regions to colder regions. This process is known as thermal diffusion or **thermal conduction**.

For a body made of a material of density ρ, specific heat at constant pressure C_p and thermal conductivity k the heat diffusion equation can be stated in a general form as:-

$$\rho C_p \frac{\partial T}{\partial t} = H\rho + \nabla.(k \nabla T) \tag{7.6a}$$

or:-

$$\frac{\partial T}{\partial t} = \frac{H}{C_p} + \frac{1}{\rho C_p} \nabla.(k \nabla T) \tag{7.6b}$$

where H is the heat production rate density (watt/kg) of the part of the body under consideration. In the case of ballscrew systems, the materials of which their parts are made are isotropic and so equation (7.b) can be simplified to:-

$$\frac{\partial T}{\partial t} = \frac{H}{C_p} + \kappa \nabla^2 T \tag{7.7}$$

where the thermal diffusivity κ is given by

$$\kappa = \frac{k}{\rho C_p} \tag{7.8}.$$

(It is of interest to note that the units of thermal diffusivity are those of area per unit time, the same as those of kinematic viscosity.) Where there are no heat sources, equation (7.7) can be simplified further to:-

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T = \frac{k}{\rho C_p} \nabla^2 T \tag{7.9}$$

7.1 **Modelling of screw – justification of one dimensional approach**

The solution of equation (7.9) in the three-dimensional case can be very difficult. It would be a lot simpler to model the ballscrew considering the temperature as being dependent only on the axial position along the screw. Intuitively this seems reasonable because steel, the material of

which the ballscrew is made, is a good conductor of heat, and the typical distance from a piece of ballscrew material from the surface is small compared to the length of the screw.

In order to test the reasonableness of this assumption, the following approach was undertaken. A round bar of a size typical of a ballscrew in a medium sized machine tool was taken and modelled as a string of N "thermal masses" connected by conductivity elements of cross-sectional area A_c, that is to say, the material was considered to act at N nodes equally spaced along the length L of the bar. The elements between the nodes pass heat according to the Fourier equation:-

$$\frac{dQ_c}{dt} = k\frac{A_c}{\delta x}\delta T \tag{7.10}$$

where $\frac{dQ_c}{dt}$ is the rate of heat conducted along the element and δT is the temperature difference across it. Its length δx is given by:-

$$\delta x = \frac{L}{N-1} \tag{7.11}.$$

Each node has a thermal capacity tc_{jx} given by:-

$$tc_{jx} = C_p m_{jx} \tag{7.12}$$

where the mass m_{jx} is given by:-

$$m_{jx} = \rho A_c \frac{L}{2N} \text{ for } jx = 1 \text{ and } jx = N \tag{7.13a}$$

and

$$m_{jx} = \rho A_c \frac{L}{N} \text{ for } 2 \le jx \le N-1 \tag{7.13b}.$$

Heat losses from the cylindrical surface of the bar and from its end are achieved by radiation. Heat is supplied at a rate of Q at node jQ for a period of initial heating $0 \le t \le t_h$ and the system is allowed to cool for the period $t_h < t \le t_{stop}$. The analysis proceeds by setting the first row of a time history array of temperatures $Th_{itime\ ix}$ to the ambient temperature T_a then calculating the subsequent rows at time interval Δt. This was done by defining a smaller time interval δt by $\delta t = \Delta t / N$ (7.14), then at each of these small intervals the changes in temperature dT_{jx} were calculated for each node as follows:-

- For $jx = 1$:-

$$\delta T_1 = (k\, A_c \frac{T_2 - T_1}{\delta x} - \varepsilon \sigma (As_1 + A_c)(T_1^4 - T_a^4)) / tc_1 \times \delta t \tag{7.15a}$$

- for $2 \le jx \le jQ-1$ and for $jQ+1 \le jx \le N-1$:-

$$\delta T_{jx} = (k\, A_c \frac{T_{jx-1} - 2T_{jx} + T_{jx+1}}{\delta x} - \varepsilon\, \sigma\, As_{jx} (T_{jx}^4 - T_a^4)) / tc_{jx} \times \delta t \qquad (7.15b)$$

- for $jx = jQ$:-

$$\delta T_{jQ} = (Q + k\, A_c \frac{T_{jQ-1} - 2T_{jQ} + T_{jQ+1}}{\delta x} - \varepsilon\, \sigma\, As_{jQ} (T_{jQ}^4 - T_a^4)) / tc_{jQ} \times \delta t \qquad (7.15c)$$

- and for $jx = N$:-

$$\delta T_N = (k\, A_c \frac{T_{N-1} - T_N}{\delta x} - \varepsilon\, \sigma\, (As_N + A_c)(T_N^4 - T_a^4)) / tc_N \times \delta t \qquad (7.15d)$$

where As_i is the cylindrical surface area of the ith element.

The temperature at row *itime* in the temperature history array *Th* was then given by

$$Th_{itime\ jx} = Th_{itime-1\ jx} + \sum_{ix=1}^{N} \delta T_{jx} \quad \text{for } 1 \le jx \le N \qquad (7.16)$$

Selected rows from the temperature history array could then be plotted on a temperature versus axial position plot to illustrate the development of the temperature distribution along the rod as time passes. This model is **one-dimensional**, that is to say the temperature is dependent only on the axial position of a point on the body and on time.

In fact, a ballscrew is a three-dimensional object and, when it is running, heat from the bearings and nut will be applied in an approximately symmetrical manner at the surface. In order to study the effect of this the model was extended to two dimensions by considering the temperature to be a function of the radial distance from the axis of the rod as well as of the axial position. This was achieved by splitting each of the nodes tc_{jx} into five nodes with the same axial coordinate. One is a central cylinder with a diameter a fifth of that of the rod, the remaining four are hollow cylinders of progressively larger size with the last one having an outside diameter equal to that of the rod. The axial conductivity elements are split up in a similar manner, and four radial conductivity elements have been included to allow flow of heat from one layer to another. Of course, heating and cooling can only be applied to the outer layer. The surface area of the inner and outer cylindrical surfaces of the nodes Asi_{jx}, Aso_{jx} are derived from the outer surface of the rod, the thermal capacity of the nodes are derived from those of the corresponding one-dimensional nodes and the cross sectional area of the axial conductivity elements are derived from that of the bar as shown in Table 7.1.

Equations (7.15) need to be modified to allow for radial conduction between the outer layer and the next one in as follows:-

- For $jx = 1$:-

$$\delta T_{11} = (k\, A_{c\,1} \frac{T_{1\,2} - T_{11}}{\delta x} + k\, Asi_1 \frac{T_{2\,1} - T_{11}}{d/10} - \varepsilon \sigma (As_1 + A_{c\,1})(T_{11}^4 - T_a^4)) / tcr_1 \times \delta t \qquad (7.17a)$$

- for $2 \le jx \le jQ - 1$ and for $jQ + 1 \le jx \le N - 1$:-

$$\delta T_{1\,jx} = (k\, A_{c\,1} \frac{T_{1\,jx-1} - 2T_{1\,jx} + T_{1\,jx+1}}{\delta x} + k\, Asi_{jx} \frac{T_{2\,jx} - T_{1\,jx}}{d/10} - \varepsilon \sigma\, As_{jx} (T_{1\,jx}^4 - T_a^4)) / tcr_{jx} \times \delta t \qquad (7.17b)$$

- for $jx = jQ$:-

$$\delta T_{1\,jQ} = (Q + k\, A_{c\,1} \frac{T_{1\,jQ-1} - 2T_{1\,jQ} + T_{1\,jQ+1}}{\delta x} + k\, Asi_{jQ} \frac{T_{2\,jQ} - T_{1\,jQ}}{d/10} - \varepsilon \sigma\, As_{jQ} (T_{1\,jQ}^4 - T_a^4)) / tcr_{jQ} \times \delta t \qquad (7.17c)$$

- and for $jx = N$:-

$$\delta T_{1\,N} = (k\, A_{c\,1} \frac{T_{1\,N-1} - T_{1\,N}}{\delta x} + k\, Asi_N \frac{T_{2\,N} - T_{1\,N}}{d/10} - \varepsilon \sigma (As_N + A_{c\,1})(T_{1\,N}^4 - T_a^4)) / tcr_N \times \delta t \qquad (7.17d)$$

Table 7.1 – *Derivation of 2D thermal model data*

Layer	1 (outer)	2	3	4	5 (inner)	
Ac_i	9/25	7/25	5/25	3/25	1/25	× Ac
Asi_{jx}	4/5	3/5	2/5	1/5	-	× As_{jx}
Aso_{jx}	-	4/5	3/5	2/5	1/5	× As_{jx}
tcr_{jx}	9/25	7/25	5/25	3/25	1/25	× tcr_{jx}

In the case of the inner hollow layers, heat can be conducted between a layer further out from the centre as well as one further in. Equations (7.17) have to be developed further for $2 \le ir \le 4$ as:-

- For $jx = 1$:-

$$\delta T_{ir\,1} = (k\, A_{c\,ir} \frac{T_{ir\,2} - T_{ir\,1}}{\delta x} - k\, Aso_1 \frac{T_{ir\,1} - T_{ir-1\,1}}{d/10} + k\, Asi_1 \frac{T_{ir+1\,1} - T_{ir\,1}}{d/10} - \varepsilon \sigma\, A_{c\,ir} (T_{ir\,1}^4 - T_a^4)) / tcr_1 \times \delta t \qquad (7.18a)$$

- for $2 \le jx \le N - 1$:-

$$\delta T_{ir\ jx} = (k\,A_{c\,ir}\frac{T_{ir\ jx-1} - 2T_{ir\ jx} + T_{ir\ jx+1}}{\delta x} - k\,Aso_{jx}\frac{T_{ir\ jx} - T_{ir-1\ jx}}{d/10} + k\,Asi_{jx}\frac{T_{ir+1\ jx} - T_{ir\ jx}}{d/10}) / tcr_{jx} \times \delta t \quad (7.18b)$$

- and for $jx = N$:-

$$\delta T_{ir\ N} = (k\,A_{c\,ir}\frac{T_{ir\ N-1} - T_{ir\ N}}{\delta x} - k\,Aso_{N}\frac{T_{ir\ N} - T_{ir-1\ N}}{d/10} + k\,Asi_{N}\frac{T_{ir+1\ N} - T_{ir\ N}}{d/10} - \varepsilon\sigma A_{c\,ir}(T_{ir\ N}^4 - T_a^4)) / tcr_N \times \delta t \quad (7.18c)$$

In the case of the central row of elements, heat can only be conducted between a layer further out and so equations (7.18) can be simplified to:-

- For $jx = 1$:-

$$\delta T_{51} = (k\,A_{c\,5}\frac{T_{52} - T_{51}}{\delta x} - k\,Aso_1\frac{T_{51} - T_{41}}{d/10} - \varepsilon\sigma A_{c\,5}(T_{51}^4 - T_a^4)) / tcr_1 \times \delta t \quad (7.19a)$$

- for $2 \le jx \le N-1$:-

$$\delta T_{5\ jx} = (k\,A_{c\,5}\frac{T_{5\ jx-1} - 2T_{5\ jx} + T_{5\ jx+1}}{\delta x} - k\,Aso_{jx}\frac{T_{5\ jx} - T_{4\ jx}}{d/10}) / tcr_{jx} \times \delta t \quad (7.19b)$$

- and for $jx = N$:-

$$\delta T_{5\ N} = (k\,A_{c\,5}\frac{T_{5\ N-1} - T_{5\ N}}{\delta x} - k\,Aso_{N}\frac{T_{5\ N} - T_{4\ N}}{d/10} - \varepsilon\sigma A_{c\,5}(T_{5\ N}^4 - T_a^4)) / tcr_N \times \delta t \quad (7.19c)$$

An average temperature distribution Tav_{jx} was given by:-

$$Tav_{jx} = \frac{9}{25}T_{1\ jx} + \frac{7}{25}T_{2\ jx} + \frac{5}{25}T_{3\ jx} + \frac{3}{25}T_{4\ jx} + \frac{1}{25}T_{5\ jx} \quad (7.20)$$

These two methods were coded in MATLAB routines "`therm2a`" and "`therm3a`" respectively. They were both run to predict how the temperature distribution would develop with time and the results were compared. A steel bar of 38.4 mm diameter and 1.362 m long was used, this being the size of the ballscrew in the linear guide rig. This was heated for 10 seconds at the centre by a 500 W source and then allowed to cool for a further 50 seconds. Figure 7.1 shows the difference between a succession of temperature distributions predicted by a one-dimensional thermal model of a ballscrew and the corresponding average temperature distributions predicted by a two-dimensional model, (a three dimensional model with radial symmetry).

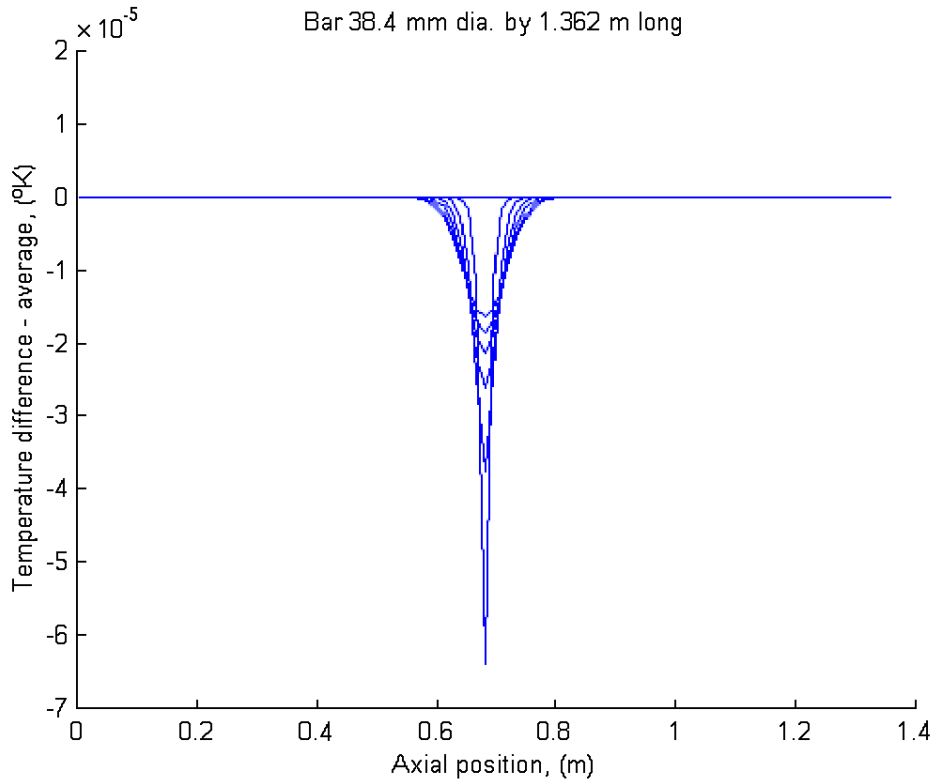


Figure 7.1 – *A comparison of one-dimensional thermal and two-dimensional thermal models of a ballscrew, $2D_{ave}$ – 1D*

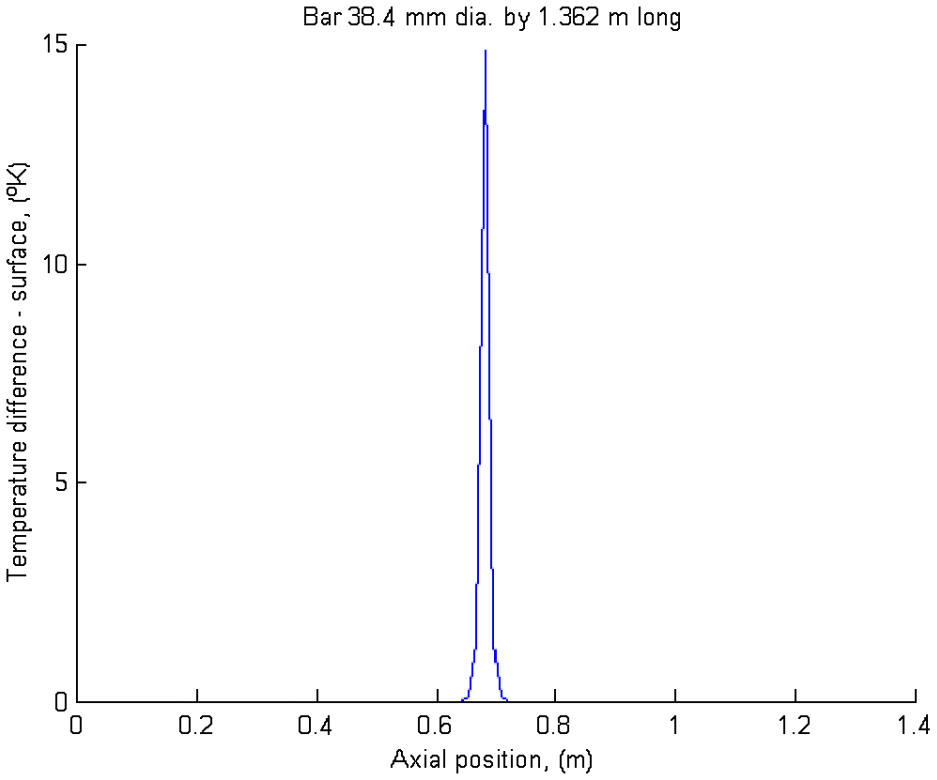


Figure 7.2 – *A comparison of one-dimensional thermal and two-dimensional thermal models of a ballscrew, $2D_{surf}$ – 1D, heating*

The results show that the difference was very small when comparing the average 2D temperatures with the 1D values. This indicates that a one-dimensional approach can be used with confidence for modelling thermal growth effects which are dependent on average temperatures. When the temperatures predicted at the surface are compared, a temperature difference of 15ºC occurs under the heating element at the end of the heating phase (Figure 7.2), but the temperature differences during the cooling phase are no more than 0.03ºC (Figure 7.3).

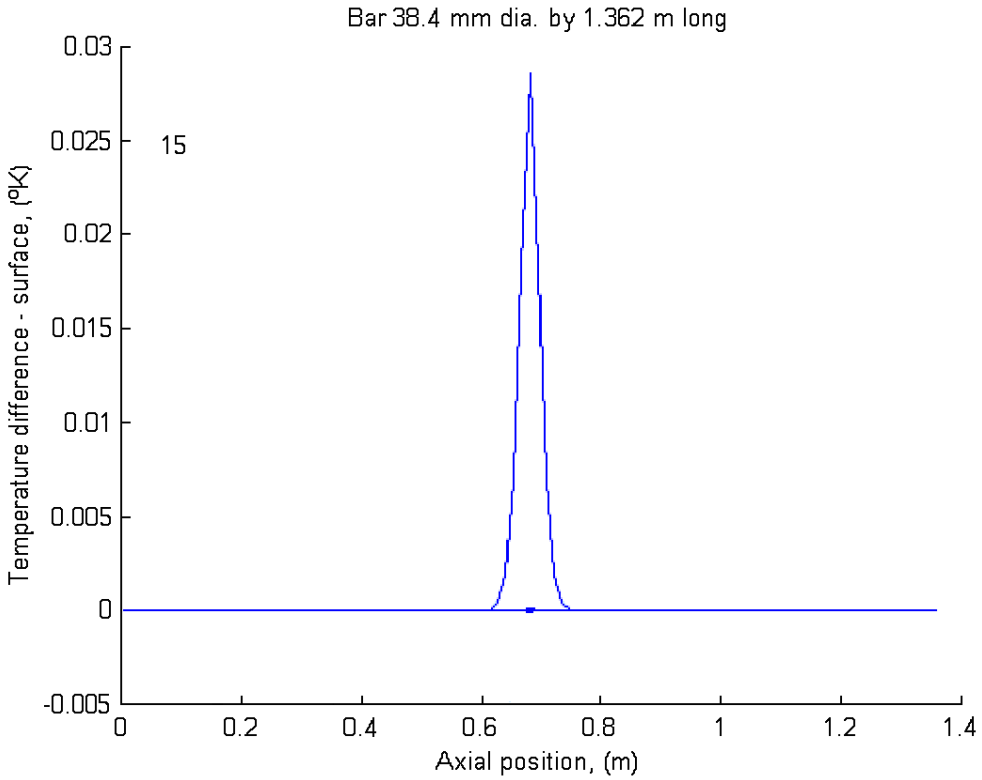


Figure 7.3 – *A comparison of one-dimensional thermal and two-dimensional thermal models of a ballscrew, $2D_{surf}$ – 1D, cooling*

It should be borne in mind that in these models heating was applied locally at one node, whereas in the models developed to predict the thermal behaviour of the whole ballscrew system account is taken of the fact that the ballscrew covers a finite axial length and therefore such highly localised temperature rises should not arise. The small temperature differences in the cooling phase should not materially affect the predictions of heat dissipated from the ballscrew surface. Therefore a one-dimensional approach can be used to model the screw.

7.2 **Thermal behaviour of bearings**

The bearings which support the ballscrew are commonly of the rolling element type. Such bearings have an inner sleeve, usually known as a race, which mounts on a suitably machined diameter of the ballscrew, an outer race which fits into a housing held by the machine part into

which the ballscrew system is mounted, and a set of rolling elements which roll between the inner and outer races. The rolling elements might be balls, cylindrical rollers, taper rollers, or a form of long thin cylindrical rollers known as needles. In most types of bearing the rolling elements are kept in place by some sort of "cage", commonly a thin sleeve with a set of near radial holes to constrain the rolling elements. Such bearings can be lubricated by oil supplied from a centralised supply, or grease which is packed into the bearing during manufacture. Arrangements whereby the grease can be replenished from time to time are common. Bearings on machine tool drives normally have lip seals or similar to keep the lubricant in and foreign matter out.

As the ballscrew rotates the rolling elements roll between the inner and outer races. The load carried by the bearing puts the rolling elements in compression, and a set of Hertzian compression zones are set up on the rolling elements and the raceways. The size and shape of these zones depend on the load being carried and the relative radii of curvature of the surfaces involved. The zones on the inner race will have different characteristics from those on the outer race. Taking the case of a ball bearing as an example and defining convex curvature as positive, the inner race/ball contact will involve three positive curvatures and one negative, whereas the ball/outer race contact will involve two positive curvatures and two negative. The local compression will push the ball and race away from the shapes in which they would ideally roll and therefore the rolling action will in practice entail a small amount of sliding with its attendant friction. This **rolling friction** is a source of heat.

As the bearing rotates the regions of material that are under compression change with time. A particular piece experiences a time when it is under little strain, it is then given energy in the form of strain potential energy as it comes under load, this energy is then released as the load is removed. Some of this energy is dissipated as heat. The process is known as hysteresis and involves the material in the races and the rolling elements which come into the compression zones.

The lubricant can be subject to viscous shear as it passes through a compression zone and churning of lubricant being pushed aside by the rolling action also occurs. Both these effects give rise to heat.

Finally a lip seal normally has a circumferential spring which helps the lip of the seal keep in contact with the shaft being sealed. The friction effect associated with the lip contact force is also a source of heat.

7.3 **Thermal behaviour of nut**

In the case of the ball nut, the heat generation can take place in all of the ways described in Section 7.2. In addition, as the balls roll between the screw and the nut, eventually they come close to the end of the nut. At this stage they need to be collected and fed back to the "front" of the nut as described in Chapter 3. While the balls are working in the nut, they contain kinetic energy of rotation, kinetic energy associated with the "whole body" movement of their centre of gravity, and strain potential energy associated with the compressive load which they carry. Much of this energy is dissipated as heat as the balls leave the working part of the nut. Since the ball collection arrangements are part of the nut, it is expected that this heat will pass into the nut.

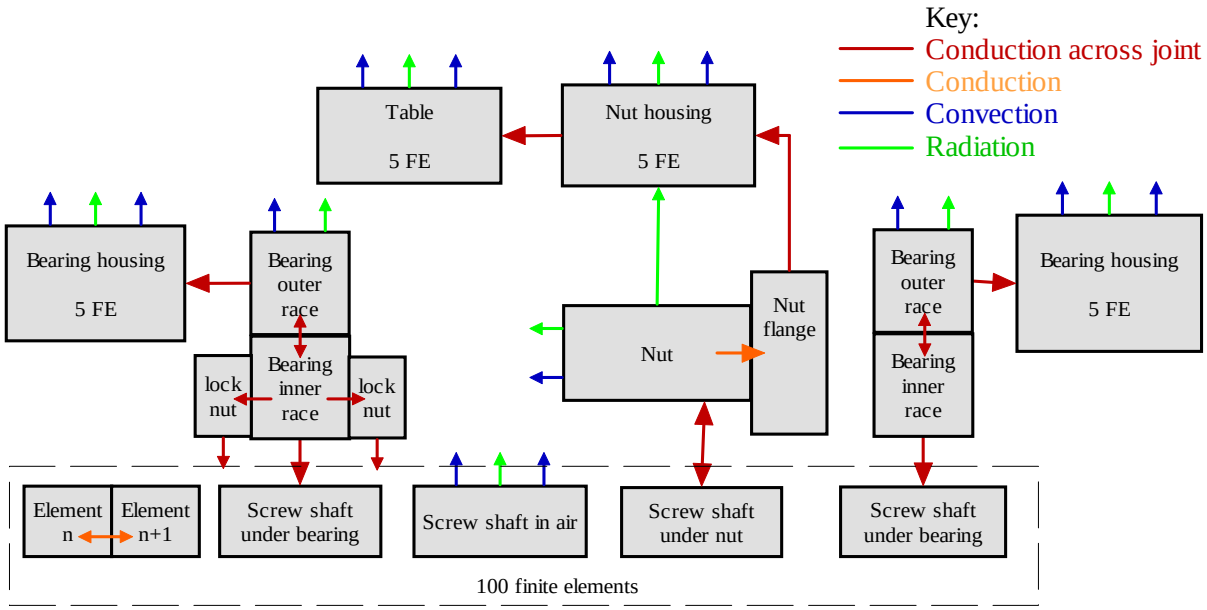


Figure 7.4 - *Heat transfer elements of ballscrew system model* [**108**]

7.4 **Ballscrew system thermal model**

The Ultra Precision Engineering group at the University of Huddersfield has undertaken extensive research into the thermal behaviour of machine tool drives. Based on this work Dr Simon Fletcher [**108**] has developed a model of the thermal characteristics of a ballscrew system based on the physical behaviour of the following components of a ballscrew drive:-

- heat generated by the ballscrew nut action
- heat generated by the support bearings
- heat conducted along the body of the ballscrew shaft
- heat conducted through the ballscrew
- heat conducted through the bearings
- heat dissipated from the ballscrew shaft
- heat dissipated from the ballscrew housings, and

- heat dissipated from the bearing and ballscrew nut mounting arrangements, (see Figure 7.4).

The model, which uses finite element techniques, can predict the thermal behaviour under a variety of conditions.

The ballscrew on the Linear Guide Rig has been the subject of several thermal tests undertaken in a recent Research Project carried out at the University of Huddersfield [**109**]. One of the results of this work has been to find that the dissipative effect of the ballscrew nut is approximately independent of load, in the sense of mass of the saddle, and only slightly dependent on speed. The friction torque of the nut and bearings has been measured by noting the motor current which is proportional to torque. This has shown that the nut torque drag on the ballscrew is 1.1 N m at low speed increasing to 1.2 N m at high speed [**110**]. A feature, which enables speed-dependent empirical torque data to be used, has been incorporated into the C program which models the mechanical system (see Chapter 6 and Appendix 6.1). This feature has been shown to work correctly by checking against hand calculations.

The C program also computes the energy dissipated, and therefore the heat generated from 7 effects as follows:-

1. drive motor bearings, Coulomb friction plus viscous drag
2. drive coupling (coupling element, drive belt, chain, gear mesh), viscous effects
3. ballscrew drive end bearings, Coulomb friction plus viscous drag
4. ballscrew shaft, viscous effects
5. ballscrew nut, to suit empirical data OR calculated from screw theory using an equivalent coefficient of friction
6. ballscrew tail end bearings, Coulomb friction plus viscous drag
7. saddle / table support bearings, Coulomb friction plus viscous drag

It did not prove possible to build a comprehensive model incorporating mechanical and thermal effects, because of the incompatibility of the time scales of the two effects. In order to cope adequately with mechanical effects, especially with transients, it was necessary to use small time steps, typically of the order of 50 μsec, whereas thermal models' time steps are typically of the order of minutes. The method developed in this Thesis is capable of modelling dissipative effects, including non-linear ones, but the output would best be converted to heating rates to be input into a thermal model. If the thermal model suggested that parameters in the mechanical model had changed over the order of the time scales in which the thermal model operates, the mechanical model would have to be run again to generate a set of new thermal inputs, and so on to cover the period covered by a full thermal analysis.

Experience has shown that a fully coupled thermal / stress analysis is rarely required, very few parameters need up-dating on a small time scale, so it is acceptable to use data generated from a “brief” simulation to provide them.

7.5 **Ballscrew cooling**

Thermal expansion of a ballscrew relative to the component in which it is mounted can induce significant positional errors especially if the position of the ballscrew is determined by a rotary encoder on the motor which drives the screw. Many ballscrews are set up with some degree of pre-tension. Expansion of the screw changes this pre-tension, possibly into compression. Loss of pre-tension can give rise to backlash in the bearings. If the screw is mounted in single-row bearings, changes in pre-tension can also effect the bearing stiffness and therefore the dynamic characteristics of the drive of which the ballscrew is part.

When a ballscrew runs heat is generated. Although it is desirable to reduce the amount of this heat by maximising the mechanical efficiency of the screw and its supporting bearings nevertheless some heat will still be generated. Cooling therefore suggests itself as a means to counter the effects of the thermal expansion resulting from the heat generated by the ballscrew system. To be effective the cooling system needs to remove sufficient heat, but only sufficient heat, to bring the ballscrew to such a length that its expansion matches that of the support structure. This suggests that any system devised should have two functional sections – cooling and control.

Research has shown [**108**] that the temperature distribution in a ballscrew follows a complex pattern dependent, amongst other things, on the position of the ballscrew in the time recently before the time of interest. It is not going to be practicable to bring the ballscrew to an even temperature along its length between its support bearings with any reasonably simple system. The majority of users of ballscrews are under pressure to minimise costs and maximise reliability, and for those reasons any system devised should be as simple as possible.

Cooling could be applied by supplying coolant to the outside of the screw by a nozzle or spray, or to the inside through a hole or set of holes running parallel to the axis along the length of the screw. Just as the heating effects are not uniformly distributed along the length of the screw, so also the cooling effects will be unevenly distributed. Control therefore is only sought over total expansion.

7.5.1 **Cooling using a moving fluid**

The first thing to note is that as a consequence of the second law of thermodynamics [**111**], which in its simplest form can be stated as “heat can’t flow from a colder to a hotter body”, it is only possible to reduce thermal expansion by the use of cooling media whose temperature is about the ambient temperature of the machine where the ballscrew is being used. In order to bring the temperature of the screw down to ambient the cooling medium must be cooler than ambient.

Considering the two options of external or internal cooling:-

- External cooling involves squirting the cooling medium at the outside surface of the ballscrew or nut. It offers the advantages that it entails little extra equipment, a few nozzles and the associated pipe-work will do, and if some of the nozzles are mounted on the part propelled by the nut, the cooling effort can be concentrated close to the sources of heat generation. However, external cooling has the drawbacks that the surface area of the parts being cooled which is in contact with the cooling medium is small, and, because the coolant quickly falls from or is flung off the surface, the time that the medium is in contact is short. This makes the use of the coolant inefficient. Also if the coolant is water-based, it is not good practice to flood the ballscrew. If the nut is grease lubricated, sufficient coolant to impair lubrication may eventually get past the seal. If the nut is oil lubricated by a pressure system, the lubricant tends to flow out so lubricant contamination should not be a problem. Nevertheless, in order to avoid abrasive damage to the screw and the possibility of abrasive particles finding their way past the seals into the ball mechanism, the coolant would need to be well filtered. Further, care would need to be taken over the siting of the coolant nozzles in order to avoid collisions or restricting the scope of movement of the nut.
- Internal cooling entails passing the cooling medium through one or more passages in the part being cooled. The surface area over which the coolant flows is under the control of the designer and it can be arranged that the coolant remains in contact with the surface being cooled for a significant time.

Because internal cooling arrangements are more amenable to analysis and control, it is these that will be considered in this Chapter.

7.5.1.1 Water cooling of the ballscrew

It will simplify matters a great deal if the “water” used is the water/oil emulsion commonly used on machine tools for cooling and lubricating the cutting process. This has the advantage of

making most use of the services already available on most machines and that the coolant can be allowed to drain into the coolant sump of the machine after it has finished its cooling work.

If the ballscrew is made with an axially central hole, coolant can be introduced at one end and collected mainly at the other. A nozzle arrangement which directs the coolant along the tube and outwards to the surface of the internal passage should ensure a flow along the internal surface of the ballscrew when it is running. A collecting shroud would be fitted over the opposite end of the screw to direct the used coolant towards the coolant drain. A similar shroud should also be fitted around the inlet end to collect any coolant which might overflow past the nozzle and spill out at the inlet end. These arrangements would not have to be "water-tight" since the momentum of the flow from the nozzle should ensure that the greater part of the coolant should flow in the direction intended.

Such arrangements would be satisfactory for ballscrews driven by gears, belt pulleys or chain sprockets. In the case of screws driven in-line by a flexible coupling two arrangements are possible. One would be to have the nozzle mounted internal to the screw at the drive-end and the coolant fed via a sleeve fitted on the outside of the screw. The other would be to mount the nozzle at the non-drive end and arrange drainage via a series of radial holes close to the drive end of the ballscrew. In the case of the ballscrew on the linear guide rig at Huddersfield University, its outside diameter is typically 40 mm, its pitch 16 mm and its maximum speed 40 000 mm/min. Its maximum peripheral speed is therefore:-

$$v_{per} = \frac{40\,000}{16} \times \frac{\pi \times 40}{1000} \times \frac{1}{60} = 5.24 \text{ m/sec} \qquad (7.21).$$

Lip seals should be adequate in this situation for the pressures likely to be involved, although it is recommended by seal manufacturers that the surface against which they bear should be ground if the lip speed exceeds 4 m/sec [**112**].

For laminar flow the Nusselt number, which can be understood as the ratio of the rate of heat transfer by convection to that by conduction, is about 3.66 [**113**]. The relation between the Nusselt number *Nu* and the heat transfer coefficient *h* is given by:-

$$Nu = \frac{hd}{k} \qquad (7.22)$$

where *d* is the diameter of the tube and *k* is the thermal conductivity of the fluid. For the whole ballscrew the heat transfer rate per degree is therefore:-

$$H = \frac{k\ Nu}{d} \times \pi\, d\, l = \pi\, k\ Nu\, l \qquad (7.23)$$

where l is the length of the coolant hole. The thermal conductivity of water at 20°C is 0.597 W/(m K) [**114**] and so the heat transfer rate per unit length of whole hole can be estimated as:-

$$\frac{\partial H}{\partial x} = \pi \times 0.597 \times 3.66 = 6.86 \text{ W/(m K)} \tag{7.24}$$

Consider water flowing through the hole with an average speed of v_w. Let the temperature of the water at the inlet (x = 0) end of the cooling duct be T_{wi}, that of the water as it travels through the screw be T_w and the temperature of the ballscrew material be T_b. The change in temperature of the water as it passes through an element of the screw from x to $x + \delta x$ is δT. Balancing the heat δQ which passes from the water into the ballscrew metal in time interval δt against the heat gain or loss from the water flowing through the hole gives:-

$$\delta q = \pi k Nu (T_w - T_b) \delta x \delta t = C_p \frac{\pi}{4} d^2 v_w \delta t \rho_w - \delta T_w \text{ joules} \tag{7.25}$$

where ρ_w is the density of water. This gives the temperature gradient in the coolant as:-

$$\frac{\partial T_w}{\partial x} = -\frac{4k Nu}{C_p \rho_w d^2 v_w} (T_w - T_b) \text{ K/m} \tag{7.26}.$$

Since the thermal conductivity of the screw material is much greater than that of water, assume for simplicity that T_b is uniform throughout the screw. Then the temperature of the water is given by:-

$$T_w = (T_{wi} - T_b) e^{-\frac{4k Nu}{C_p \rho_w d^2 v_w} x} + T_b \text{ ºC} \tag{7.27}$$

and for a coolant hole of length l the average temperature is:-

$$\overline{T}_w = \frac{T_{wi} - T_b}{\frac{4k Nu}{C_p \rho_w d^2 v_w} l} (1 - e^{-\frac{4k Nu}{C_p \rho_w d^2 v_w} l}) + T_b \text{ ºC} \tag{7.28}$$

For water at 20ºC $$\frac{4k Nu}{C_p \rho_w} = \frac{4 \times 0.597 \times 3.66}{4.183 \times 10^3 \times 1000} = 2.089 \times 10^{-6} \text{ m}^2\text{/sec} \tag{7.29a}$$

and $$\frac{C_p \rho_w}{4k Nu} = 0.4786 \times 10^6 \text{ sec/m}^2 \tag{7.29b}$$

leading to:-

$$\overline{T}_w = 0.4786 \times 10^6 \frac{d^2 v_w}{l} (T_{wi} - T_b) \times (1 - e^{-\frac{2.089 \times 10^{-6}}{d^2 v_w} l}) + T_b \text{ ºC} \tag{7.30}$$

Using the average temperature of the water derived by equation 7.30 and equation 7.23 the rate at which heat can be removed by water passing through the ballscrew can be derived.

Calculations based on this approach (Appendix 7.1) for a typical application show that this method of cooling can remove the heat likely to be generated by the action of the ballscrew and its bearings using reasonable coolant flow rates if a supply of chilled water, that is water at about 10ºC below the ambient temperature of the machine, is available.

If this method is to be used to correct ballscrew thermal errors and to maintain pre-tension, it is necessary to control the heat flow through the coolant. The heat loss rate can be changed by changing the coolant inlet temperature, the flow rate or both. Control would need to be achieved by monitoring the average temperature of the screw.

7.5.2 **Evaporative cooling**

An alternative to water cooling is to use air. It is possible to cool the air by spraying water into the cooling air. As the water evaporates it takes in its latent heat thus causing a drop in temperature of the air.

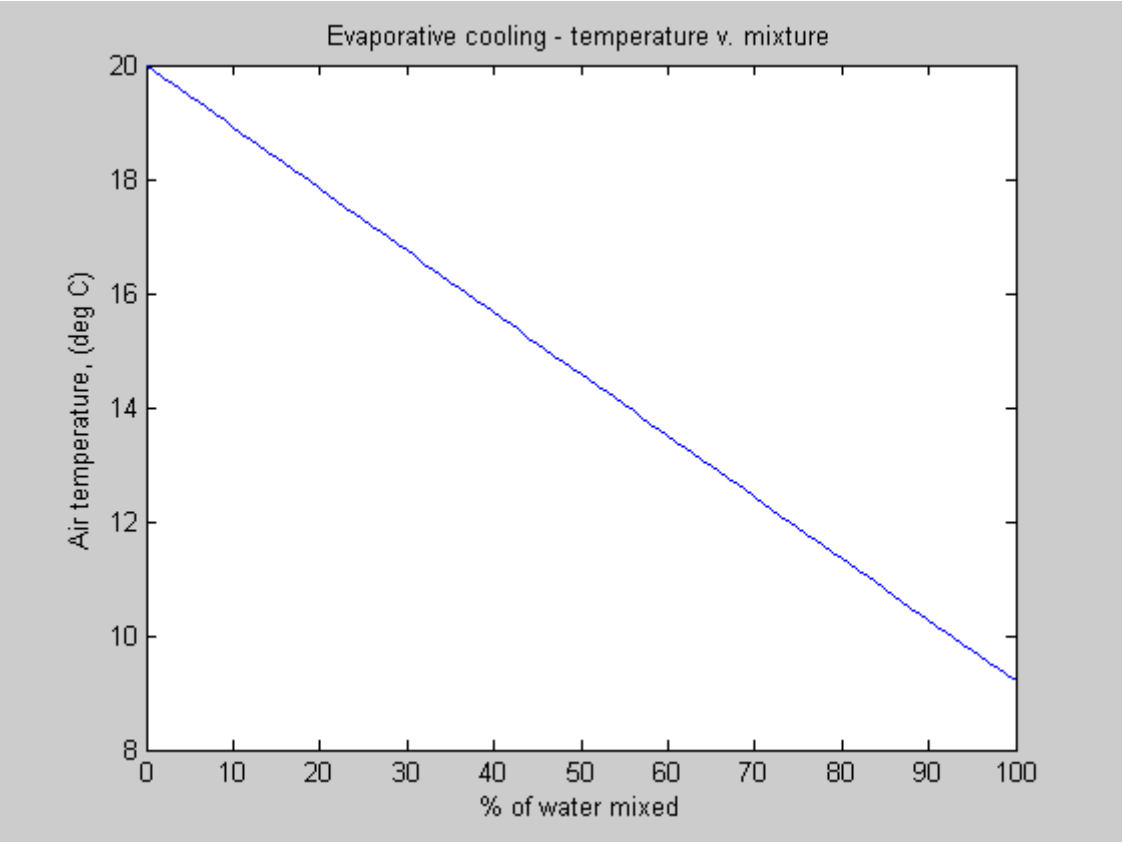


Figure 7.5 – *Evaporative cooling – typical temperature drop*

Compressed air available in typical workshops is obtained by compressing air from the atmosphere and storing it in a pressure vessel. The atmosphere contains a certain amount of water vapour which depends on the weather and the proximity of "steamy" processes. The amount of water vapour it can carry before condensation takes place depends on temperature. Therefore compressing air tends to make it more humid. Commercial air supplies should

therefore be regarded as “wet”, and arrangements need to be made for draining excess water from compressor receivers. Unless the compressor is working in a particularly arid region or special arrangements have been made, the partial pressure of water vapour in the air supply line will be near to the saturated vapour pressure at the temperature of the workshop.

If compressed air is expanded down to atmospheric pressure it dries and has the capacity to absorb moisture from a water spray and achieve a cooling effect. Calculations using a typical workshop air supply pressure (Appendix 7.2) show that a cooling effect of about 12ºC can be achieved, (see Figure 7.5). However, the rate of air flow needed to remove the heat at the required rate is likely to be regarded as excessive by many users and so of the two options water cooling is better in most cases.

7.6 **Comparison with measured data**

The dynamic models keep an account of all the dissipation mechanisms built into them. At each stage of the analysis the rate of energy dissipation is calculated for any Coulomb friction, viscous damping and any other dissipative process which is incorporated into the mechanical model. As presently set up, the dissipative effects are formed into seven sums as follows:-

- motor bearings
- drive coupling
- ballscrew driving end bearing
- ballscrew shaft (material dissipation)
- ballscrew nut
- ballscrew driven end bearing
- saddle or table slideways

The rates of dissipation of each of these groups is integrated to give a total energy. This energy is mainly in the form of heat.

As the analysis proceeds, the energy dissipated by the various mechanisms is added to the potential energy in the springs and the kinetic energy in the inertias. This total energy is balanced against the energy put in by the motor. This is a check on the validity of the model and shows that the system obeys the first law of thermodynamics. A sample result is shown in Figure 7.6. As can be seen a balance is achieved within 0.5%. One source of energy input which is not considered in the energy balance is that involved in the vertical movement of the gravitational forces as the nut changes position in the run. Though the movements are small, the forces are significant. If the model were to be refined to include this energy term, even the 0.5% might be improved upon.

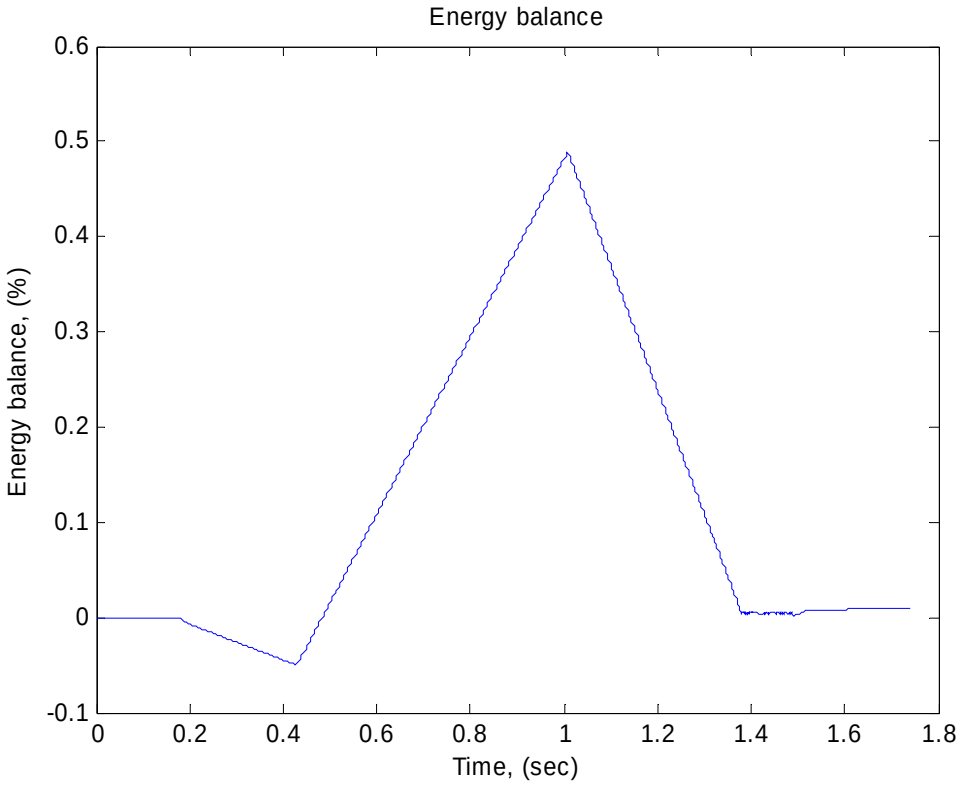


Figure 7.6 – *Energy balance in a typical ballscrew movement*

The dynamic model includes two broad options for accounting for dissipation. One is based on considering the detailed mechanical environment of each part of the mechanical system. The other involves the use of measured data (e.g. measured friction torques) where available.

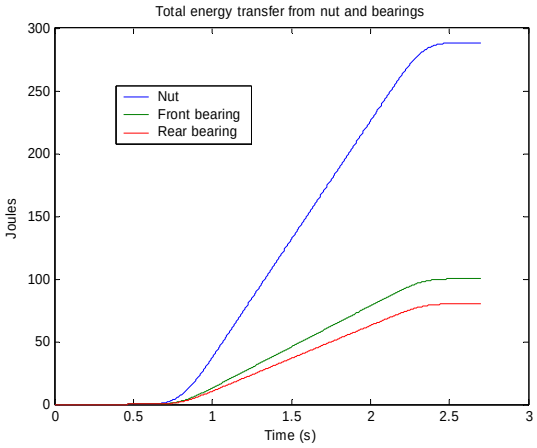


Figure 7.7 – *Energy dissipated in nut and bearings – thermal model*

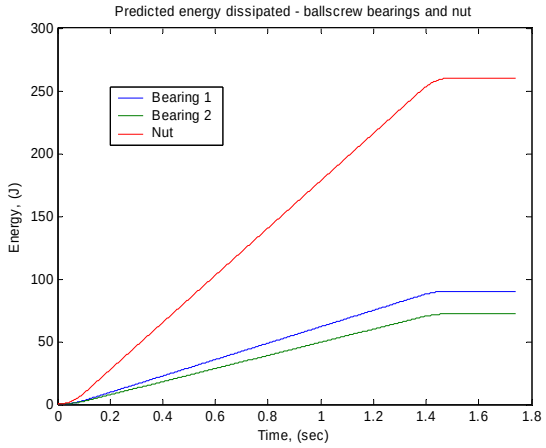


Figure 7.8 – *Energy dissipated in nut and bearings – dynamic model*

A comparison has been made between the energy used in a model used to account successfully for some thermal measurements on the Linear Guide Rig [**108**] and the energy dissipation calculated by the dynamic model based on measured friction data. Figure 7.7 shows

the energy involved in the thermal model, and Figure 7.8 gives the corresponding result from the dynamic model.

The figures compared are one single run with the dynamic model compared with a "half cycle" for the thermal model. The pattern of energy dissipation is the same and the differences in final energy level are approximately 10%. This degree of correlation shows that the dissipative mechanisms included in the dynamic model can give valid inputs to a thermal model without the need to take measurements in a situation where the mechanisms of dissipation are well understood.

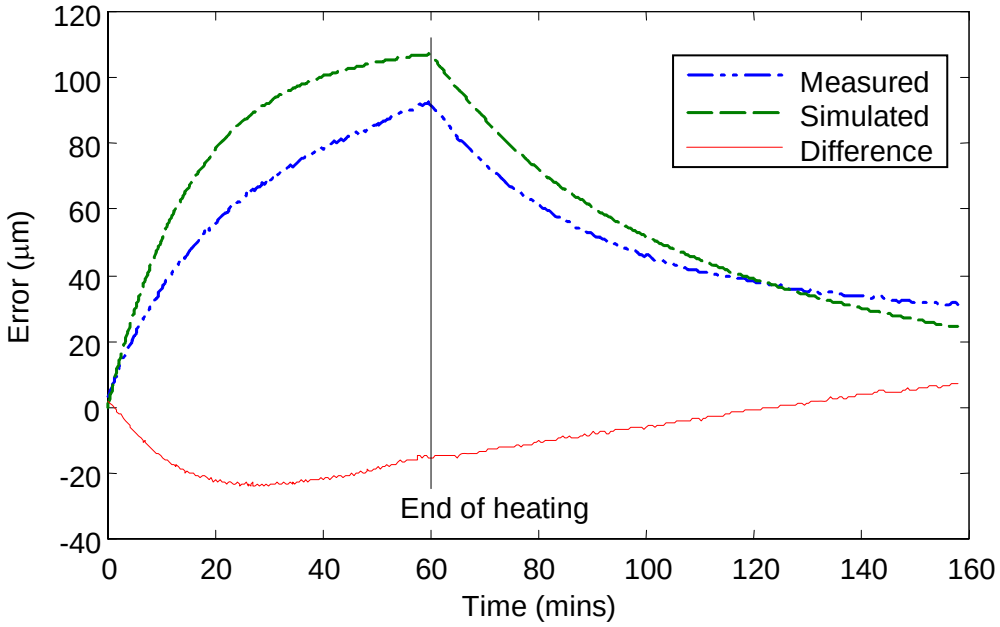


Figure 7.9 - *Simulated and measured error between axis positions 0 and -460 during a heating and cooling cycle* [**108, Fig. 9**]

Figure 7.9 shows that errors of the order of 100 µm can easily occur due to ballscrew heating. Such errors can cause loss of pre-load, added wear and pre-tension becoming compression. Direct measurements of the terms involved in heat generation can be difficult to make. Therefore, a method of generating heating terms from the dynamic performance of the screw coupled with knowledge of the dissipation mechanisms involved is a useful addition to the analytical tools available. If an understanding of how the dissipation effects vary with temperature can be built up, the dynamic model can be used in conjunction with a thermal model to predict changes in dynamic response as a system warms up. Two models would need to be used because of the different time scales involved as discussed in Section 7.4.

Chapter 8 - EXPERIMENTAL VERIFICATION

The experimental verification of the dynamic model has been undertaken using the linear guideway rig, (see Figure 8.1).

This consists of a cast iron milling machine base with a cast iron saddle carried on two linear guideways. The saddle is driven by a ballscrew mounted in the base. The ballscrew nut is attached to the underside of the saddle by a set of screws. The ballscrew itself is held between two sets of angular contact ball bearings. Each bearing set is mounted in a bearing housing into which the outer races of two bearings have been pressed. The inner races are a sliding fit on the ballscrew and are held in place by a locking nut. The degree of preload applied to a bearing set is determined by the torque applied to the locking nut with an upper limit being reached when the axial gap between the inner races has been closed. Each bearing housing is held in a carrier block which in turn is fixed to the base casting by a set of screws and dowels.

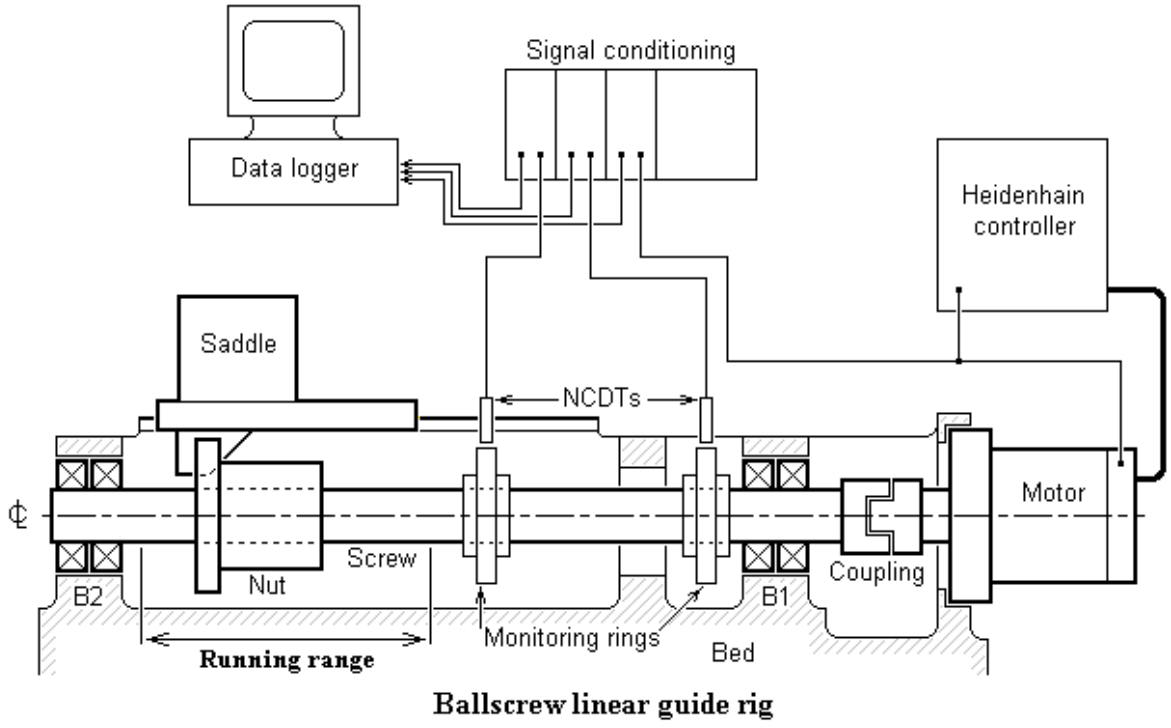


Figure 8.1 – *Ballscrew linear guide rig with controller and vibration monitoring equipment*

The ballscrew is driven by a DC motor which is fixed to one end of the base via a mounting plate. Control of the motor is achieved using a Heidenhain controller located close to the rig. Electronic signals which give an indication of the position of the ballscrew nut are generated by a rotary encoder attached to the motor and are fed to the controller. Similar

information can also be obtained from a linear encoder on the side of the base with a sensor attached to the saddle.

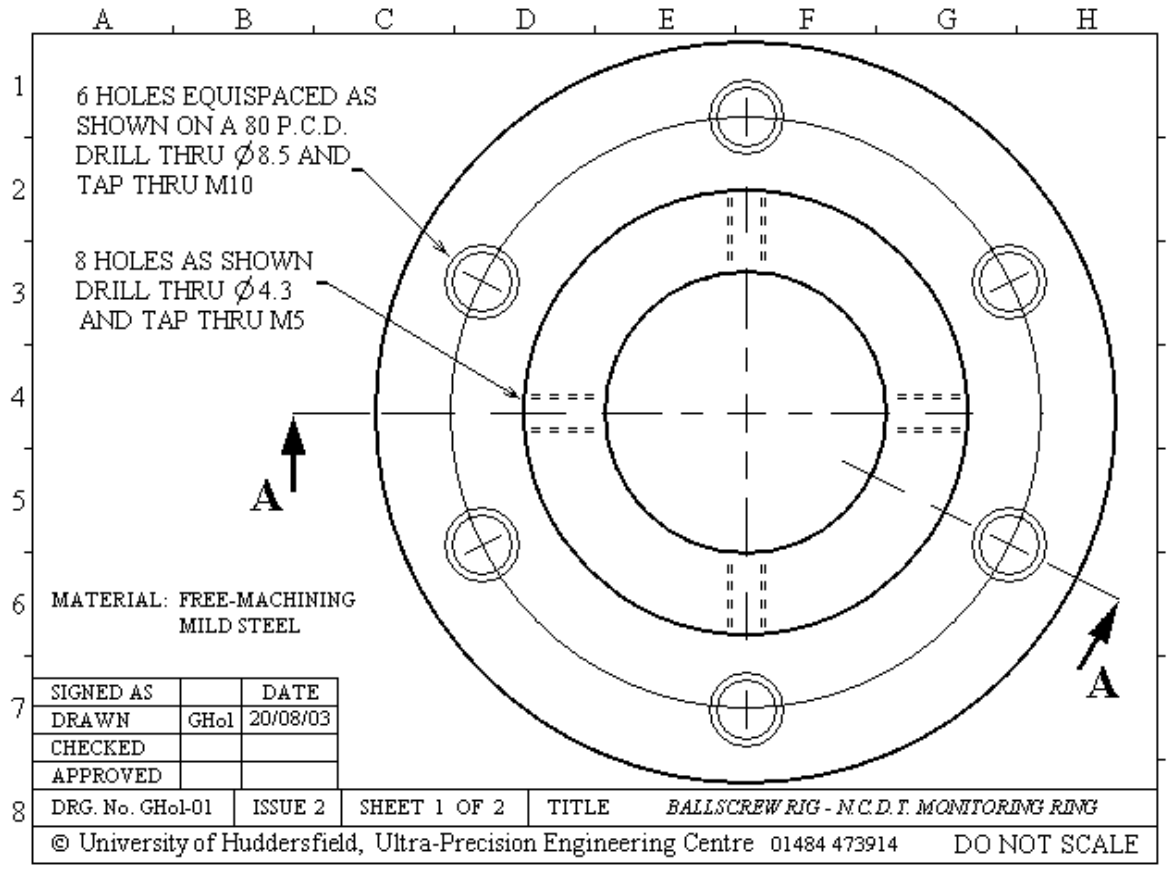
Pre-tension can be applied to the screw by stretching it. This is achieved by adjusting the position of a lock nut at the non-drive end of the screw.

Since the ballscrew is manufactured to a good standard of precision, and since the ballscrew is limited to a speed of 2500 rpm (40 m/min nut speed), it is not expected that this particular screw should generate high levels of vibration by itself. Other applications involving long ballscrews may do so, it is hoped that the mathematical models developed in this Thesis will be of help in designing or helping to diagnose problems with such systems. But in order to demonstrate the model's ability to predict levels of transverse vibration it is necessary to subject the screw to artificially high levels of vibration excitation. It is also necessary to measure the vibration produced. To this end two 'monitoring rings' have been designed and fitted to the screw. The drawings of the rings are included in Figure 8.2. Their function is to provide a cylindrical surface concentric with the centre of the ballscrew which can be used in conjunction with non-contacting displacement transducers (NCDT's) to give an indication of the radial position of the ballscrew, and to provide a means of carrying a set of off-centre masses whose centripetal acceleration generates a transverse force proportional to the square of the speed of the shaft.

The NCDT's are held in place using mounting pieces which are either carried by magnetic clamps attached to the base or bonded to the base using epoxy resin. The signals from the NCDT's are carried via special cables to a set of matched amplifiers the output of which is a voltage signal proportional to the distance of the monitoring ring surface from the probe end. When the probes are set at the correct distance from the monitoring ring surface, the sensitivity of the system is 10mV/μm.

A signal originating from the rotary encoder on the drive motor is taken from the controller to give the angular position of the ballscrew. This signal takes the form of two series of pulses in quadrature.

The signals obtained are fed into the University of Huddersfield's "Data Logger" system. This system consists of a personal computer (PC) fitted in this case with a digital scalar card to handle the signal from the rotary encoder and a Blue Chip PCI analogue to digital interface card to handle each of the signals from the NCDT's. The signals are processed using data logging software that operates under DOS to achieve high rates of data digitisation. The rate depends on the number and type of interface cards used and the speed of the PC's processor. In the present case a sampling rate greater than 6000 Hz was achieved. The output of the data



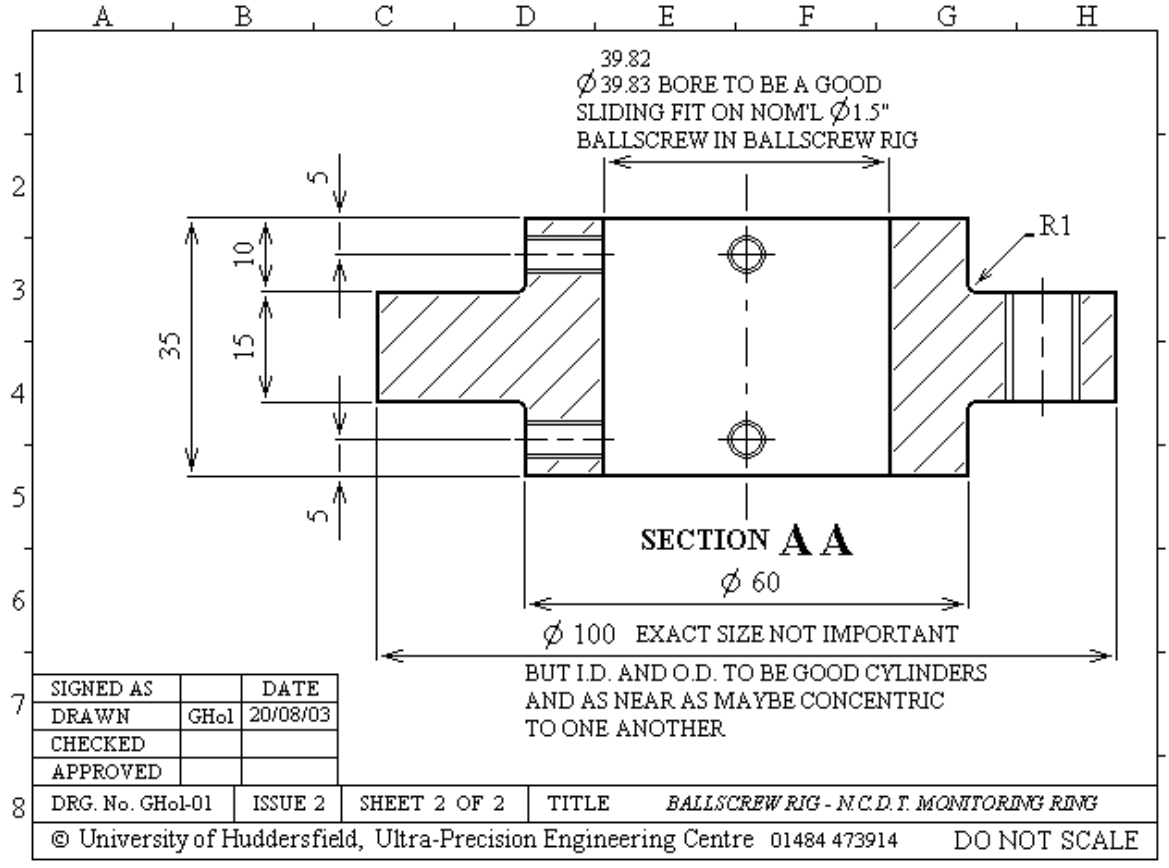


Figure 8.2 – *Ring used as a target for an N.C.D.T and to carry out-of-balance weights*

logger is a set of computer files each of which contains a record of the configuration data for the hardware set up used and a time series of data values for each channel of information monitored.

8.1. **Static measurements**

The support stiffness of the bearings has a significant influence on the dynamic behaviour of the ballscrew. Data given by bearing manufacturers are often “typical” and do not cover the range in which values of particular bearings can lie. Also the stiffness offered to the screw by the bearings also includes that of the support structure, in this case the rig base, bearing housings etc. It was therefore decided to try to derive the bearing support stiffness from the deflected shape of the screw under a known set of forces.

To achieve this the ballscrew nut was detached from the saddle and parked near to the tail end of the ballscrew. In order to prevent the screw moving axially in subsequent tests two split clamping rings were fitted to the screw, one at each side of the nut. Two monitoring rings were fitted to the screw, one as close as possible to the drive end bearing. This first ring had to be fitted between the bearing support block and the part of the base which would have supported the bottom of the column had the base been used as part of a milling machine. This was the closest we could get to the bearing. Another monitoring ring was fitted between the column support bridge and the ballscrew nut. In the case of this second ring it was possible to fit it in a variety of positions. One of the rings was fitted with a pair of screws in one of the holes which could be used for carrying balance weights. A wire was slung over the workshop’s crane’s hook with a 0 - 100 N spring balance at each end, the lower end of the balances being attached to the screws in the monitoring ring. Load could be applied to the screw by pulling up using a crane.

Deflection of the ballscrew was measured by mounting an NCDT close to the ballscrew next to a cylindrical portion of its surface, that is avoiding the helical grooves where the balls run. The probe was held by bracketry carried on a magnetic clamp attached to the saddle. Since the ballscrew nut was not connected to the saddle, it was possible to move the saddle along the guides by hand. By this means the radial position of the ballscrew surface could be monitored at a variety of axial locations along the screw. Since the loads are applied statically, that is, the loads are held steady while deflection readings are taken, the output from the NCDT’s was measured using a digital voltmeter.

8.1.1 **Preliminary considerations**

To get a feel for the order of magnitude of the deflections likely to be encountered consider the case of a cylindrical bar supported at each end. Neglecting the ballscrew nut, the clamping rings, the monitoring rings and half of the flexible coupling between the ballscrew and the drive motor, the ballscrew can be represented approximately by a hollow cylindrical bar of length l = 1.182 m, the distance between the bearing pack centres.

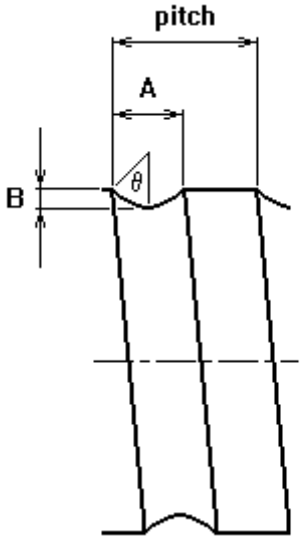


Figure 8.3 *Dimensions A and B*

The screw has an outside diameter d_o of 39.95 mm and inside diameter d_i of 10 mm. The ballscrew groove is a single start helix with a pitch p of 16 mm. The axial length of the cylindrical portion between the groove has been measured to be 8.52 mm using Vernier callipers. This gives dimension A, Figure 8.3, to be 7.48 mm. Dimension B has been measured to be 2.95 mm. Applying Pythagoras' theorem [**115**] gives:-

$$\left(\frac{A}{2}\right)^2 + \left(r_g - B\right)^2 = {r_g}^2 \tag{8.1}$$

where r_g is the radius of the groove.

Hence
$$\frac{A^2}{4} + {r_g}^2 - 2r_g B + B^2 = {r_g}^2$$
$$2r_g B = \frac{A^2}{4} + B^2$$

and
$$r_g = \frac{A^2}{8B} + \frac{B}{2} \tag{8.2}$$

Putting in the measured values gives

$$r_g = \frac{7.48^2}{8 \times 2.95} + \frac{2.95}{2} = 3.846 \text{ mm} \tag{8.3}$$

Angle θ can be derived from

$$\sin\theta = \frac{A/2}{r_g} = \frac{A/2}{\frac{A^2}{8B} + \frac{B}{2}} = \frac{4AB}{A^2 + 4B^2} \tag{8.4}$$

Substituting

$$\sin\theta = \frac{4 \times 7.48 \times 2.95}{7.48^2 + 4 \times 2.95^2} = 0.9725 \Rightarrow \theta = 1.3357 \text{ rad} \tag{8.5}$$

Using standard formulae [**116**], the cross sectional area of the groove is:-

$$A_g = \frac{1}{2} r_g^2 (2\theta - \sin 2\theta) = 16.407 \text{ mm}^2 \tag{8.6}$$

and the distance from the outer cylinder to the groove's centre of area is:-

$$y_g = r_g \times \left(\frac{4}{3} \frac{\sin^3 \theta}{2\theta - \sin 2\vartheta} - \cos\theta \right) = 1.230 \text{ mm} \tag{8.7}$$

The volume of material in a length of screw one pitch long will therefore be about:-

$$V_p = \frac{\pi}{4} \times (39.95^2 - 10^2) \times 16 \quad - \quad 16.407 \times \sqrt{(2 \times \pi \times (\frac{39.95}{2} - 1.230))^2 + 16^2} = 16849 \text{ mm}^3 \tag{8.8}$$

Replacing the grooved cylinder by a uniform one would need a cylinder of outside diameter given by:-

$$d_{o\,eq} = \sqrt{\frac{4V_p}{\pi\, p} + d_i^2} = \sqrt{\frac{4 \times 16849}{\pi \times 16} + 10^2} = 37.96 \text{ mm} \tag{8.9}$$

Considering first the "stiff" case when the bearings give "built in" support, standard beam theory gives [**117**] the maximum deflection to be:-

$$y_{\max} = \frac{-wl^4}{384EI} \tag{8.10}$$

where the weight per unit length w is given by:-

$$w = \frac{\pi}{4} (d_o^2 - d_i^2) \rho\, g \tag{8.11}$$

and the second moment of area I is given by:-

$$I = \frac{\pi}{64} (d_o^4 - d_i^4) \tag{8.12}$$

The ballscrew is made of steel of which the density ρ = 7860 kg/m^3 and the modulus of elasticity E = 206×10^9 N/m^2 [**118**]. The acceleration due to gravity g = 9.80665 m/sec^2 [**119**]. The maximum deflection is therefore:-

$$y_{\max} = \frac{-\frac{\pi}{4} \times (0.03796^2 - 0.010^2) \times 7860 \times 9.80665 \times 1.182^4}{384 \times 206 \times 10^9 \times \frac{\pi}{64} \times (0.03796^4 - 0.010^4)} \times 10^6 = -19.75 \text{ μm} \tag{8.13}$$

Considering secondly the "flexible" case when the bearings give "simple" support, standard beam theory gives the maximum deflection to be:-

$$y_{\max} = \frac{-5wl^4}{384EI} \tag{8.14}$$

The maximum deflection is therefore:-

$$y_{\max} = \frac{-5 \times \frac{\pi}{4} \times (0.03796^2 - 0.010^2) \times 7860 \times 9.80665 \times 1.182^4}{384 \times 206 \times 10^9 \times \frac{\pi}{64} \times (0.03796^4 - 0.010^4)} \times 10^6 = -99.77 \ \mu\text{m} \quad (8.15)$$

The maximum deflection of the ballscrew under its own weight should be between 20 and 100 μm.

The total weight of the ballscrew using this simple model is:-

$$W = \frac{\pi}{4} \times (0.03796^2 - 0.010^2) \times 1.182 \times 7860 \times 9.80665 = 95.95 \ \text{N} \quad (8.16)$$

The deflections predicted by simple beam theory and those predicted by the moving mass model refer to points on an imaginary line at the centre of the screw. A measurement taken at the surface of the screw would follow the centre if the screw surface were to be perfectly cylindrical (except for the ball groove of course). In practice the surface deviates from the ideal. Manufacturing imperfections in the inner and outer diameters of the monitoring rings also contribute to errors in the readings taken from their surface. Also the fact that the probe is mounted indirectly to the saddle also gives rise to errors because the linear guides are not perfectly parallel to the line joining the centres of the two bearing pairs which support the ballscrew. For a given angular position of the ballscrew, the measured deflections can therefore be understood as the sum of the deflection of the centre, which is dependent on the loading pattern and axial position along the screw, and the errors which are dependent only on axial position:-

$$\{y_{m\,i}(x_j)\} = \{y_i(x_j, geom, matl, load)\} + \{\varepsilon(x_j)\}, \ j = 1 \cdots N_m \quad (8.17)$$

where y_m are the elements of a set of N_m members which make up the *i*th set of readings, each element being the radial deflection as measured at the axial position x_j on the ballscrew. y are the elements of the corresponding set of deflections of the ballscrew centre and ε are the elements of the set of radial errors. The terms *geom*, *matl* and *load* indicate that these deflections are dependent on the shape and size of the screw, the material of which it is made and the pattern of loads to which it is subject. If the ballscrew is subject to several loading patterns, the set of the deflections derived by subtracting the corresponding elements of the sets obtained from any pair of loading patterns is independent of the errors.

8.1.2 **Measurement strategy**

Therefore the measurement strategy was to set the second monitoring ring in a particular axial position and to subject the screw to three sets of loads, one pulling up with a load of 20 N, the second with 110 N and the third with 200 N and to measure the deflection of the ballscrew cylindrical surface at as many “lands” between grooves as were accessible to the NCDT probe. (The reason why a minimum of 20 N load was maintained was to prevent the link arrangement from coming loose and moving as measurements were under way.)

The original idea had been to set the load and then work through all the axial positions in turn to get a deflection curve, move on to the next load, take another set of deflection readings and so on. However it became clear that the readings were subject to a significant drift of values with time. The rig is situated under a fan driven convector heater which comes on and goes off from time to time and so concern was expressed about thermal errors. In the event it

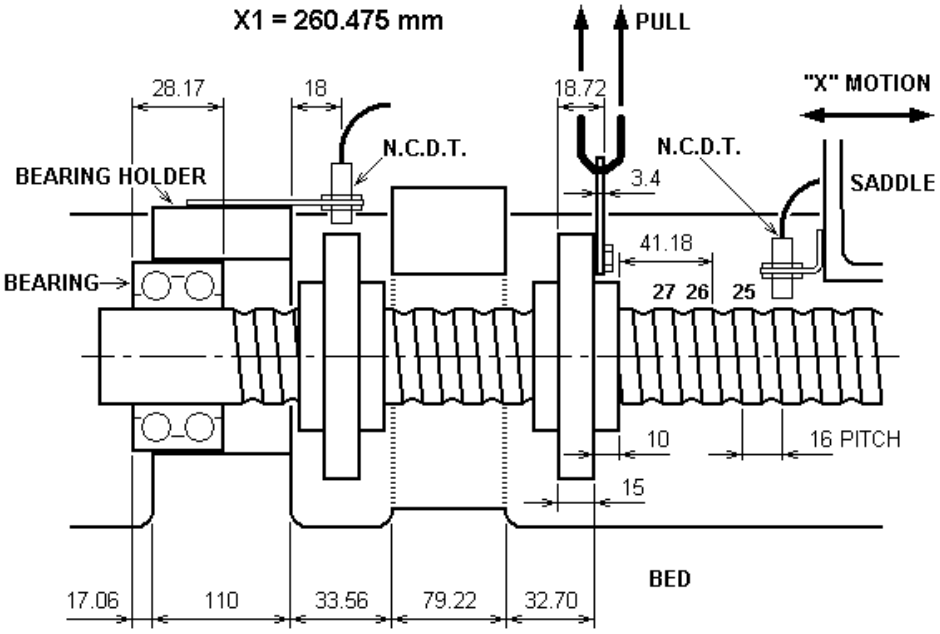


Figure 8.4 – *Static tests, Position 1*

was found that the drift was most likely due to the stretching of the insulator covered copper wire being used to suspend the spring balances on the crane hook. This was replaced by a stouter steel one and the readings were then found to be much more stable. Also to avoid thermal errors the NCDT probe was set at one axial position and successively increasing loads (20 N, 110 N, 200 N) were applied (say), then the probe was moved on a land and successively decreasing loads (200 N, 110 N, 20 N) were applied, then the probe was moved on another land and the process repeated. In this way values which were to be subtracted were taken within about a minute of time of one another thus reducing substantially the probability of significant thermal errors.

Three loading positions were used, (see Figures 8.4, 8.5 and 8.6 respectively):-

1. Monitoring ring 2 set close to the column base bridge, pulling on monitoring ring 2
2. Monitoring ring 2 set close to the ballscrew centre, pulling on monitoring ring 2
3. Monitoring ring 2 set close to the column base bridge, pulling on monitoring ring 1.

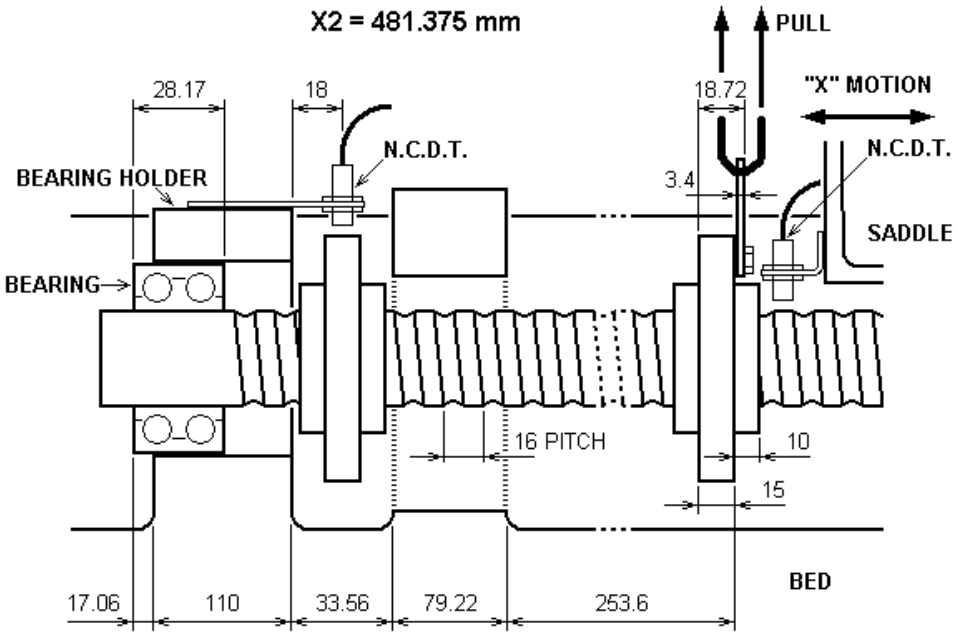


Figure 8.5 – *Static tests, Position 2*

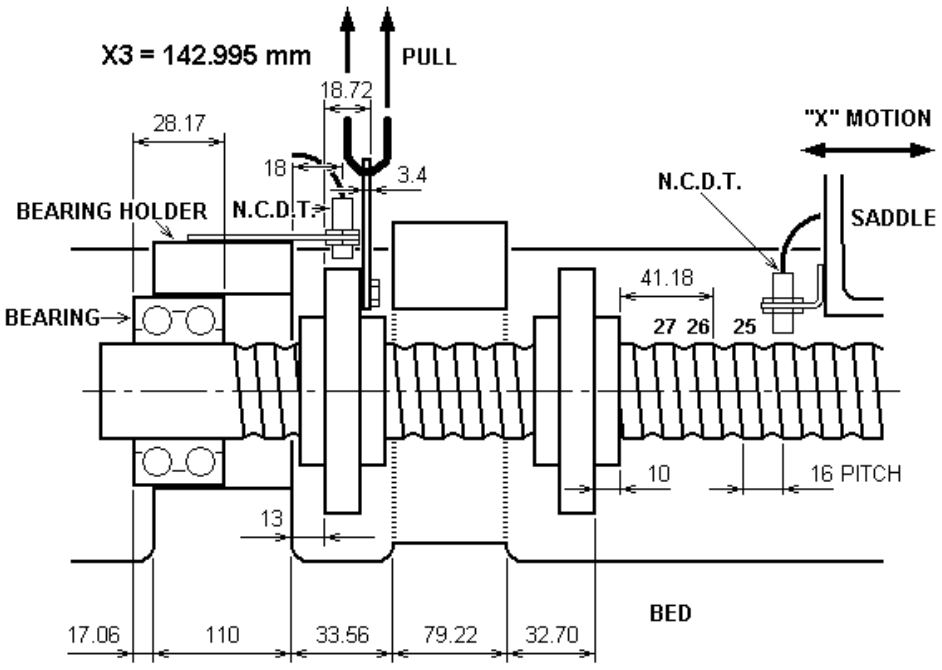


Figure 8.6 – *Static tests, Position 3*

8.1.3 Measurement results

The results are summarised in Table 8.1. The data for Position 1 is plotted on Figure 8.7

Table 8.1 – Static measurements – Raw data

	Position 1			Position 2			Position 3		
Land	20 N	110 N	200 N	20 N	110 N	200 N	20 N	110 N	200 N
1	5.527	5.286	5.046	5.451	4.960	4.471	5.361	5.254	5.148
2	5.541	5.296	5.048	5.465	4.961	4.461	5.463	5.354	5.244
3	5.559	5.302	5.045	5.471	4.956	4.438	5.478	5.364	5.249
4	5.580	5.317	5.053	5.485	4.964	4.439	5.505	5.389	5.272
5	5.572	5.303	5.029	5.485	4.955	4.421	5.494	5.374	5.253
6	5.557	5.278	5.004	5.469	4.925	4.381	5.478	5.355	5.231
7	5.553	5.270	4.989	5.455	4.901	4.353	5.465	5.338	5.210
8	5.535	5.247	4.958	5.435	4.879	4.328	5.456	5.325	5.195
9	5.522	5.227	4.937	5.427	4.866	4.306	5.448	5.317	5.183
10	5.514	5.219	4.919	5.421	4.857	4.291	5.437	5.304	5.168
11	5.531	5.228	4.923	5.433	4.865	4.298	5.426	5.289	5.152
12	5.456	5.148	4.841	5.365	4.797	4.231	5.392	5.251	5.113
13	5.410	5.099	4.787	5.317	4.750	4.184	5.332	5.190	5.049
14	5.394	5.080	4.766				5.315	5.175	5.032
15	5.384	5.068	4.754				5.305	5.161	5.015
16	5.341	5.026	4.711				5.270	5.124	4.978
17	5.149	4.832	4.511				5.084	4.935	4.786
18	5.166	4.848	4.527				5.081	4.933	4.785
19	5.160	4.848	4.531				5.093	4.945	4.798
20	5.146	4.834	4.523				5.065	4.920	4.772
21	5.150	4.841	4.535				5.057	4.911	4.764
22	5.127	4.823	4.517				5.039	4.893	4.747
23	5.105	4.805	4.505				5.057	4.915	4.771
24	5.054	4.760	4.466				4.987	4.845	4.701
25	5.033	4.749	4.461				4.982	4.843	4.701
26	5.055	4.776	4.497				5.017	4.877	4.738
27	5.080	4.814	4.546				5.036	4.903	4.767
MR1	5.941	5.819	5.694	6.688	6.558	6.426	5.864	5.788	5.711

The results tabulated are NCDT amplifier output in volts. Scaling factor: 1 V = 100 µm

The results are processed in the case of each loading position by first subtracting the data in the columns obtained from a 110 N pull from data in the columns obtained from a 20 N pull and then subtracting the 200 N data from the 110 N data. The result is a set of data which gives the effect of pulling with a force of 90 N in zero gravitational field. (This is because the effect of gravity, the self weight forces, are the same in all cases, as are the geometrical errors.) A MATLAB function, "`funcs_2_dat`" has been written which does this. Figures 8.7 and 8.8 show this process applied to the results for Position 1.

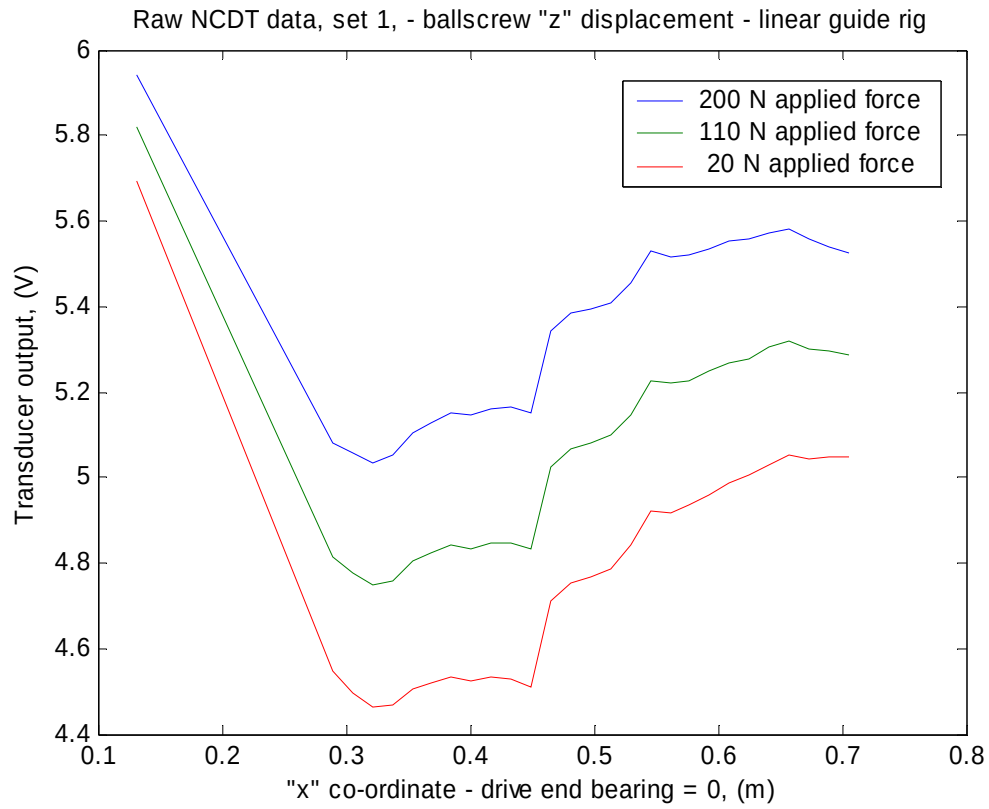


Figure 8.7 – *Raw data, static tests, Position 1*

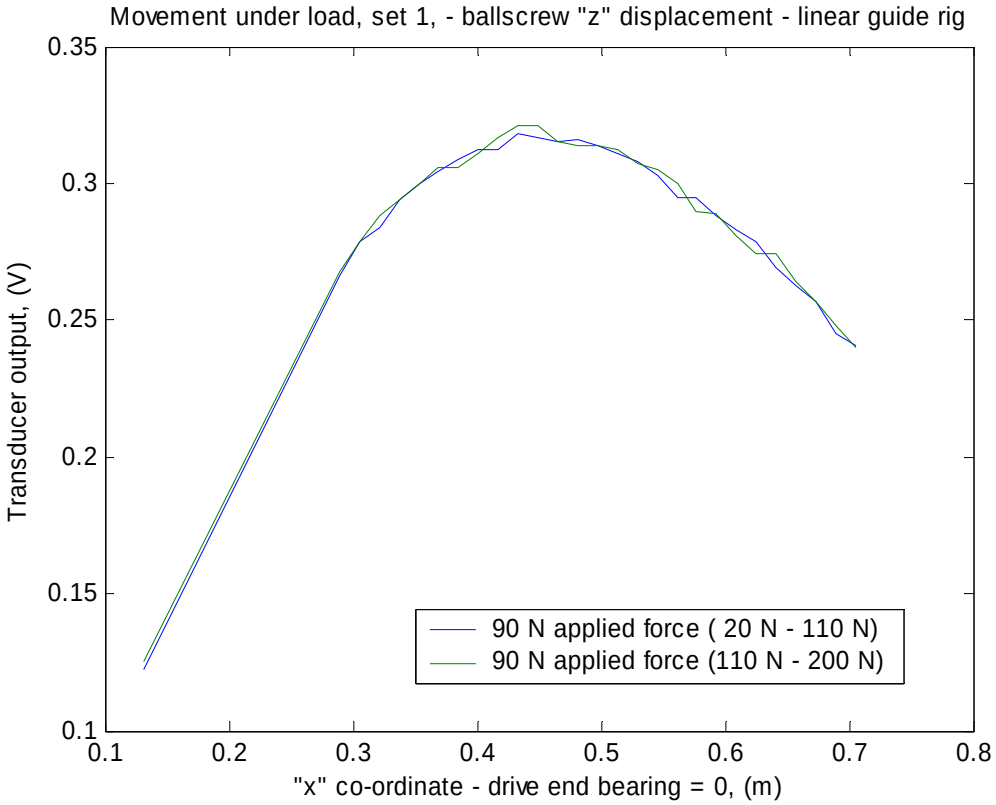


Figure 8.8 – *Transducer output for 90 N transverse pull, static tests, Position 1*

The next stage is to formulate a model which predicts the behaviour of the ballscrew. The final stage is to extract by an optimisation process the support stiffness which gives a predicted behaviour closest to the observed behaviour.

8.1.4 Static deflection models

In these models only the static behaviour is considered. The ballscrew is considered as an elastic beam supported at each end by a bearing whose radial stiffness is K_i and whose tilt stiffness is Φ_i, i being one at the drive end of the screw and two at the non-drive end.

In the first instance the pre-tension T in the ballscrew is considered to be zero, or, to be more precise, T is considered to be small enough to make no measurable difference to the behaviour of the screw to the precision to which the deflection is being measured. As shown in Appendix 8.1, the deflected shape of the ballscrew can be represented by the equation:-

$$y = -\frac{\mu g}{24EI}x^4 + \frac{A_{4(s-1)+1}}{6}x^3 + \frac{A_{4(s-1)+2}}{2}x^2 + A_{4(s-1)+3}x + A_{4(s-1)+4}, \text{for } s = 1\cdots 4 \qquad (8.18)$$

where A_i are constants determined by the boundary conditions and s is the screw section number.

In the case where the pre-tension is tensile (positive) it is shown in Appendix 8.2 that the deflected shape of the ballscrew is given by:-

$$y = A_{4(s-1)+1} + A_{4(s-1)+2}x - \frac{\mu g}{2T}x^2 + A_{4(s-1)+3}\cosh\lambda x + A_{4(s-1)+4}\sinh\lambda x, \text{for } s = 1\cdots 4 \qquad (8.19)$$

where $\lambda = \sqrt{\frac{T}{EI}}$.

For any particular geometrical configuration and loading condition equation (8.18) or equation (8.19) can be solved to give a set $\{A_i\}$ which can be used to give a predicted set of $\{y_i\}$ to compare with the measured data. A set of MATLAB functions have been written which give a predicted y for any x for the constants A given by equations (8.18) or (8.19).

(In the case of compressive or negative pre-tension, the deflected shape can be expected to take the form:

$$y = B_{4(s-1)+1} + B_{4(s-1)+2}x - \frac{\mu g}{2T}x^2 + B_{4(s-1)+3}\cos\lambda x + B_{4(s-1)+4}\sin\lambda x, \text{for } s = 1\cdots 4 \qquad (8.20),$$

where B_i are constants determined by the boundary conditions. However, a method of calculating these constants was not derived because ballscrews normally have positive pre-tension.)

The best estimate of the bearing stiffness has been extracted from the measured data by a least squares method. Because the equations giving rise to $\{A_i\}$ are dependent on the stiffness in a non-linear way, a routine based on the Levenburg-Marquardt method was prepared [**120**]. This seeks to minimise an error measure χ^2 by a method which seeks to vary smoothly between the inverse-Hessian approach and the steepest descent method. It achieves this by use of a special non-dimensional factor λ. When this factor is small the method leans towards being an inverse-Hessian approach using a matrix of partial derivatives. When λ gets bigger the method leans towards a steepest descent. The method is recommended because it is unlikely to fail due to a zero pivot in the Hessian matrix, although it can wander round in shallow valleys (in a multi-dimensional sense). The method has been coded in MATLAB functions "`Mrq_min`" and "`Mrq_cof`". The model was incorporated in function "`funcs_2`" and since some problems were experienced with singular matrices when using the MATLAB standard coding for processing matrices, a function "`GaussJ`" was prepared which uses Gauss-Jordan elimination [**121**]. Since it did not prove possible to come up with a reliable method of terminating the search for the minimum automatically, some manual intervention was necessary to arrive at a reasonable answer.

The fitted curves are shown in Figures 8.9, 8.10 and 8.11.

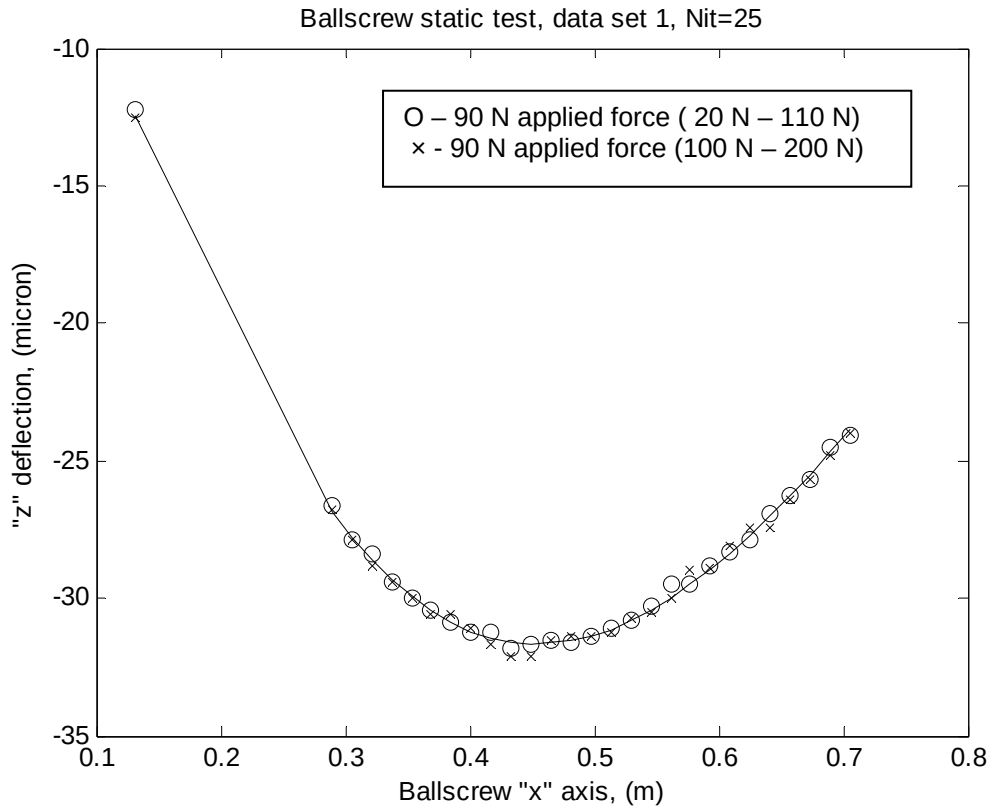


Figure 8.9 – *Fit of modelled curve to measured data, Position 1*

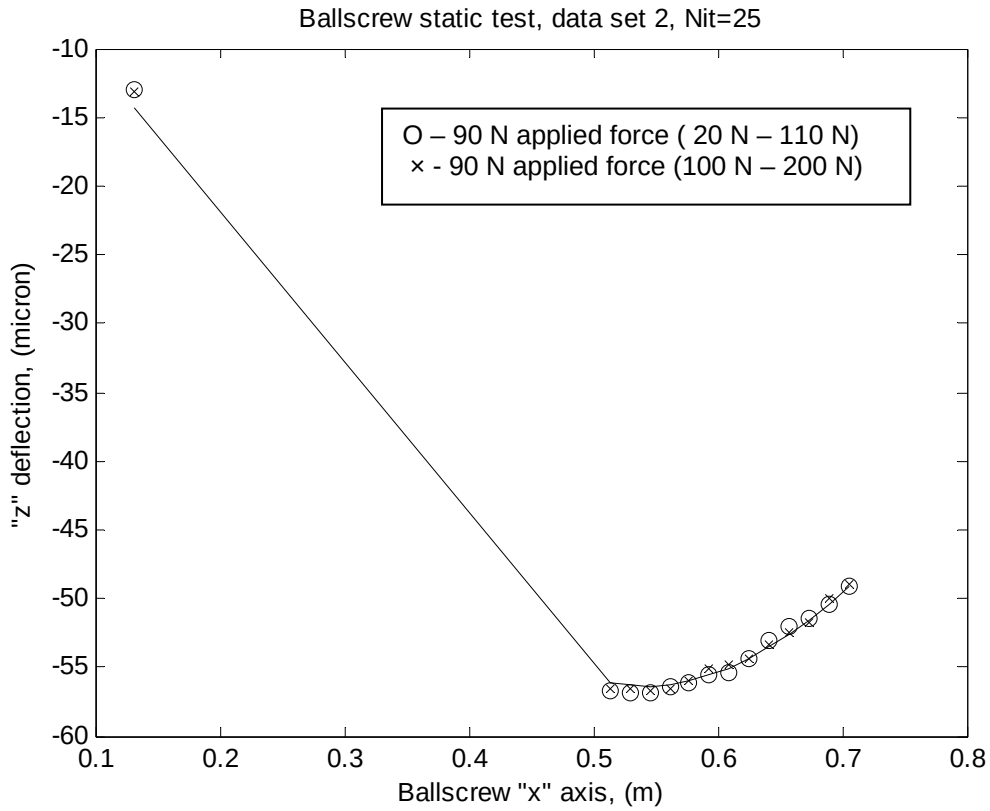


Figure 8.10 – *Fit of modelled curve to measured data, Position 2*

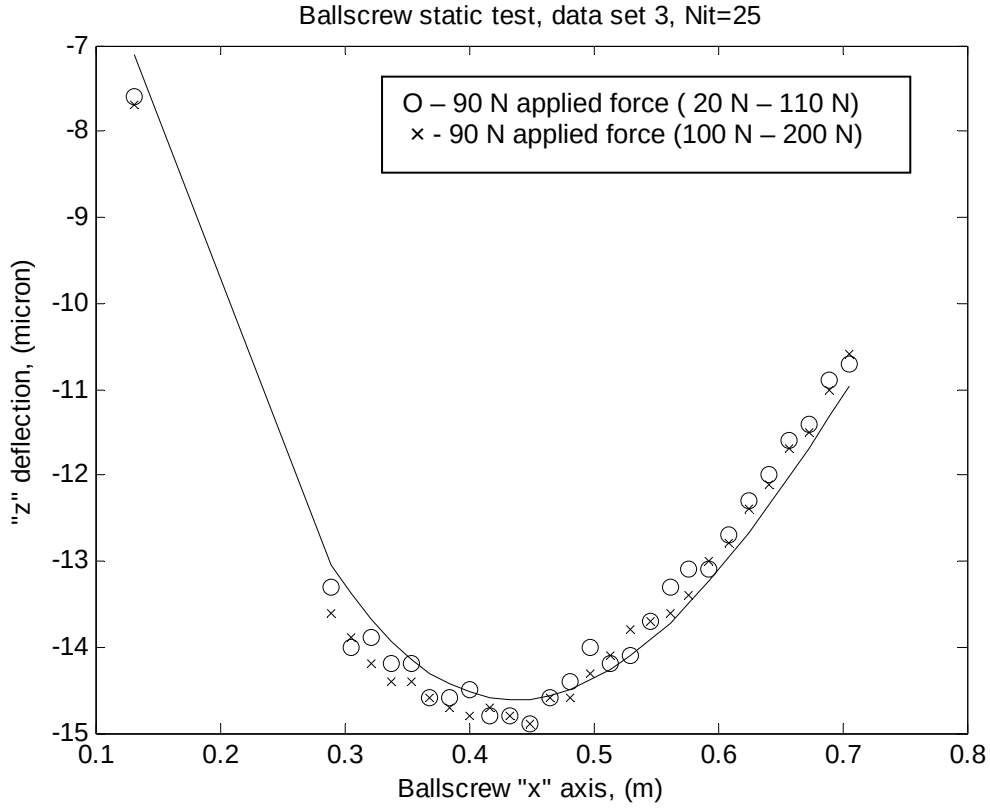


Figure 8.11 – *Fit of modelled curve to measured data, Position 3*

It had been hoped at the outset that this method would yield the effective diameter of the ballscrew, the transverse and tilt stiffness for each bearing, and the pre-tension from the best fit data. In the event the three sets of data did not give a consistent set of results for all six of these parameters. It was therefore decided to estimate the effective diameter of the shaft starting from the value of 37.96 mm given by equation 8.9. The portions of the ballscrew where its support bearings are located are 30 mm in diameter which suggests a smaller effective diameter would be appropriate. These portions are however, quite short and have bearing inner races or portions of coupling attached to parts of them. A weightier consideration is that the stiffness of a shaft is determined by the second moment of area which is a function of the fourth power of the diameter. Equation 8.9 is based on volume which is dependent on the square of the diameter, it was therefore thought that the ball groove would bring the effective diameter down to lower than 37.96 mm. A value of 37.5 mm was chosen in the absence of a detailed analysis based on the fourth power. The pre-tension was believed to be low at the time of these tests, and so it was assumed that it was zero. Consideration of the forces and deflections involved suggests that the tilt stiffness of the bearings has a greater influence on transverse behaviour than does the transverse stiffness. Further, the tilt stiffness of the structures supporting the bearings at both ends appear to be greater than the bearings themselves, and since the bearings were of the same type at both ends, it was assumed that the bearing stiffness would be the same. The results from the curve fitting method then gave the values set out in the following table:-

Table 8.2 – Static measurements - Results

Position	1	2	3	Average *	
Transverse stiffness	4.25×10^8	4.25×10^8	4.25×10^8	4.250×10^8	N/m
Tilt stiffness	1.70×10^5	1.55×10^5	1.92×10^5	1.743×10^5	N m/rad
χ^2	429.3877	149.4193	129.1332		

* weighted by χ^2.

A summary of data used in the dynamic models is given in Appendix 8.6.

8.2. **Dynamic measurements – nut in fixed position**

During these measurements the ballscrew nut was detached from the saddle and clamped in position near to the tail end of the ballscrew as in the case of the static measurements. Two monitoring rings were fitted to the screw, one as close as possible to the drive end bearing. The other monitoring ring was fitted between the column support bridge and the ballscrew nut

and was used to carry the eccentric weights used to apply radial force excitation to the ballscrew. In the case of this second ring it was possible to fit it in a variety of positions.

Deflection of the ballscrew was measured by mounting an NCDT close to each monitoring ring. The probe for the second NCDT was held by bracketry carried on a magnetic clamp attached to the bed slide. The probe for the NCDT closest to the drive end bearing was held on a mounting plate which was bonded to the bed.

8.2.1 **Preliminary considerations**

It has been shown in Appendix 8.3 that any persistent transverse vibration of a rotating ballscrew is likely to be caused by external forces, as opposed to a mode generated by its deflections under centrifugal loading.

In the dynamic case, for a given angular position of the ballscrew, the measured deflections can be understood as the sum of the deflection of the centre, which is dependent on the loading pattern and axial position along the screw, and the errors which are dependent on the axial position of the monitoring ring and the angular position of the screw:-

$$\{y_{m\,i}(x_j)\} = \{y_i(x_j, geom, matl, load)\} + \{\varepsilon(x_j, \theta_{x\,j})\},\ j = 1 \cdots N_m \qquad (8.21)$$

where y_m are the elements of a set of N_m members which make up the *i*th set of readings, each element being the radial deflection as measured at the axial position x_j on the ballscrew. y are the elements of the corresponding set of deflections of the ballscrew centre and ε are the elements of the set of radial errors. The terms *geom*, *matl* and *load* indicate that these deflections are dependent on the shape and size of the screw, the material of which it is made and the pattern of loads to which it is subject. Given that the dynamic loads are applied by an eccentric weight, it was anticipated that *load* will depend on the square of the speed of the screw. If the speed of the ballscrew is low, *load* will be close to zero. This should give a means of subtracting out the effects of the errors.

8.2.2 **Measurement strategy**

Therefore the measurement strategy was to set the second monitoring ring in a particular axial position and to subject the screw to two sets of out-of balance loading, one with the eccentric weights applied and one without. The maximum load was fixed by running the screw with the out-of-balance weights fitted at a speed which gave rise to vibration levels which appeared to an experienced operator of the rig to be at the upper end of the tolerable range. Using this speed as a maximum four other speeds were determined so as to give 20%, 40%, 60% and 80% of the dynamic force applied at the maximum speed. A final speed was

chosen to give a “slow roll” the results of which might be used to subtract out the effects of the run-out of the monitoring ring and other errors.

In order to make comparison of the different data sets as easy as possible, it was decided to arrange the data to be read as close as possible to the same angular positions for each run. To this end, the speeds chosen were in part determined by considerations of which sampling times could be set most conveniently in the configuration file of the Data Logger.

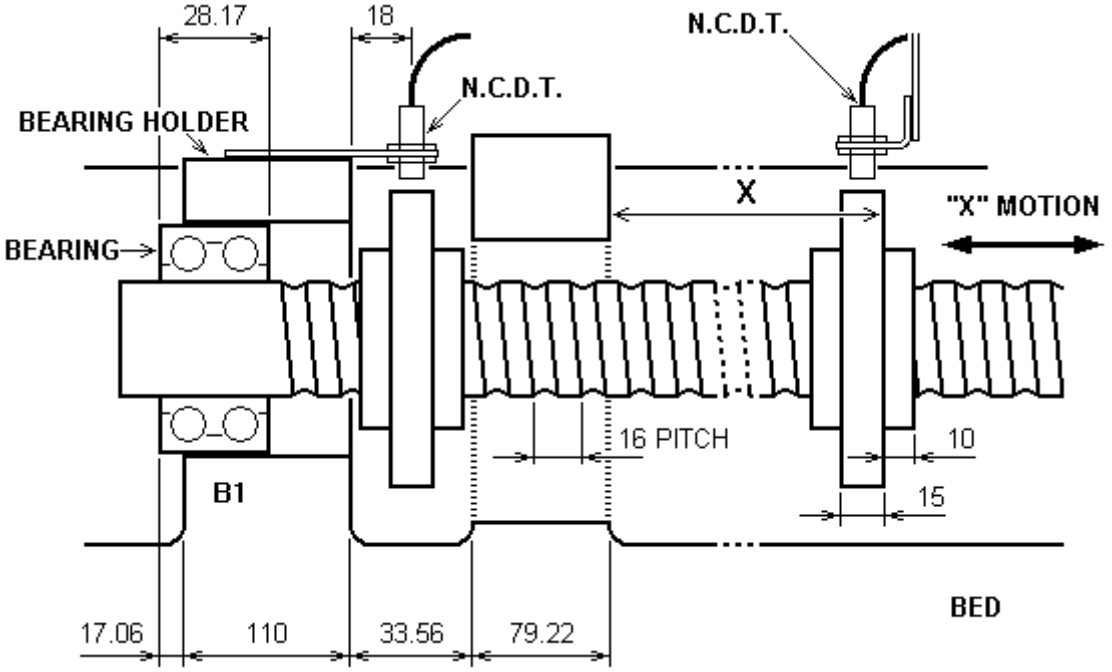


Figure 8.12 – *Dimension X in the dynamic tests*

Three loading positions were used. These were such that Dimension X (see Figure 8.12) was 410 mm, 235 mm and 60 mm respectively.

Table 8.3 - *Dynamic Tests 1 – Speeds and Sampling times*

Speed			Sample time	Nominal travel	Fraction of excitation	Error
	rpm	mm/min (nom’l)	sec	mm	%	cycles
S6	5000	32000	0.16667	320	100	0.072
S5	5362.5	28600	0.18648	320	79.9	-0.0036
S4	4650	24800	0.21505	320	60.1	-0.063
S3	3787.5	20200	0.26403	320	39.8	0.04905
S2	1936.5	14300	0.37296	320	20.0	-0.0036
S1	500.0625	2666	2.0005	320	0.7	-0.000225

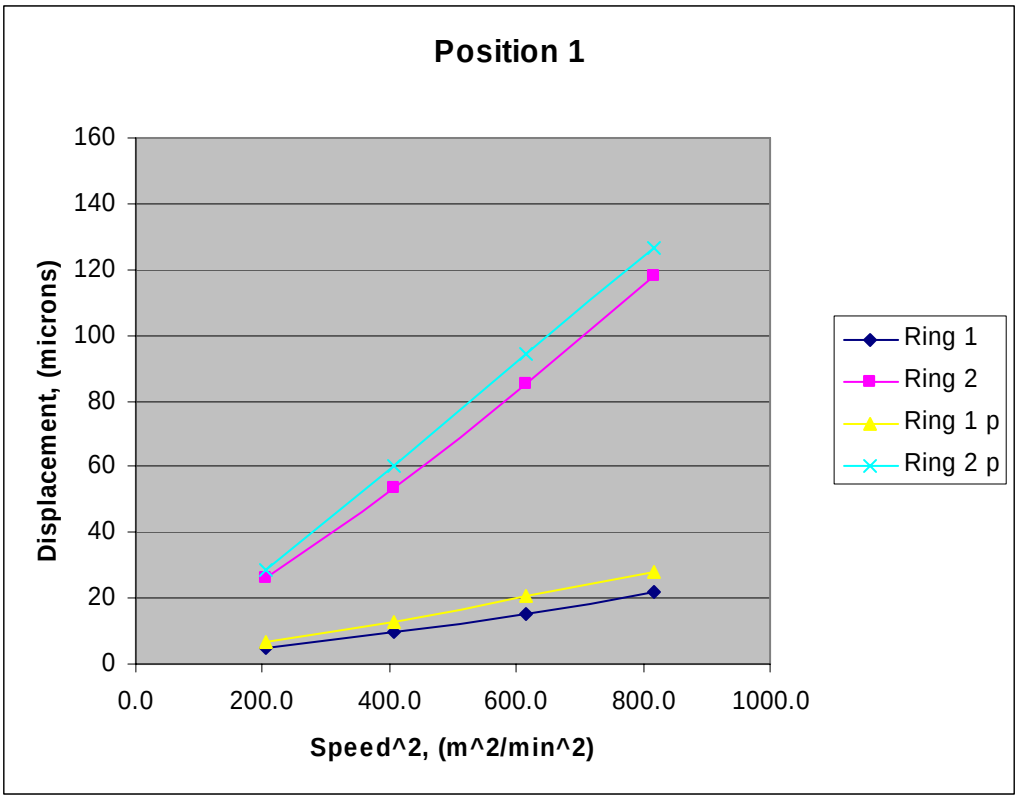


Figure 8.13 – *Comparison of measured and predicted vibration levels Monitoring ring position 1*

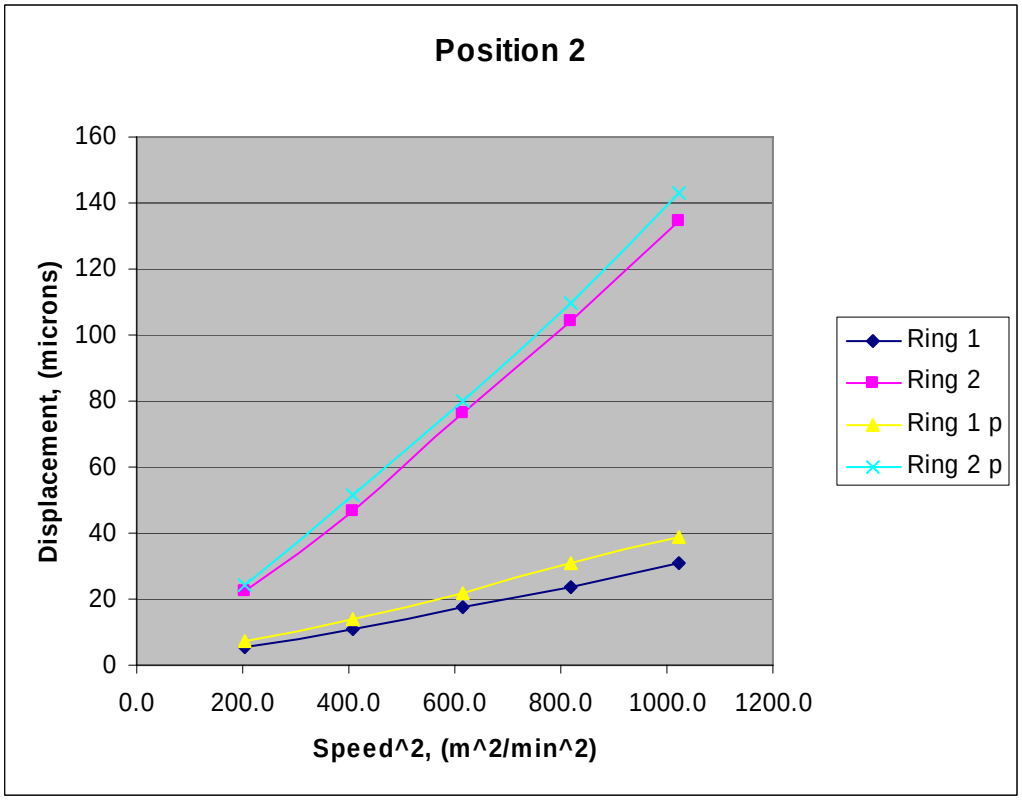


Figure 8.14 – *Comparison of measured and predicted vibration levels Monitoring ring position 2*

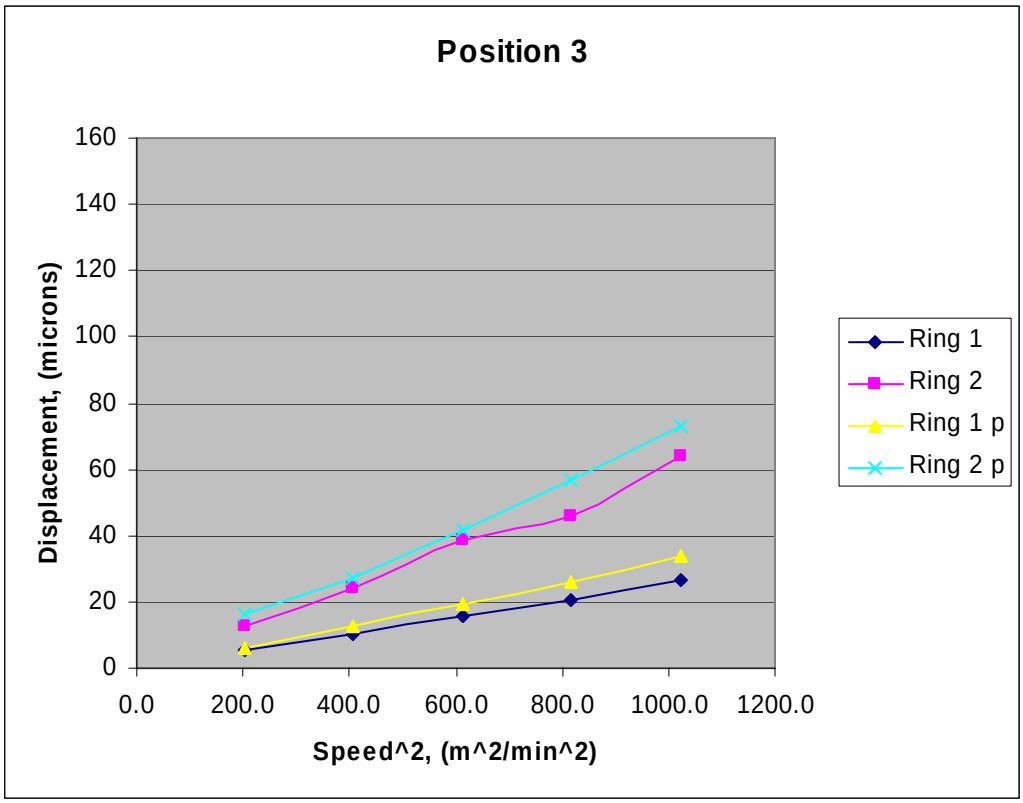


Figure 8.15 – *Comparison of measured and predicted vibration levels Monitoring ring position 3*

8.2.3 **Measurement results**

For each test, the raw data recorded by the data logger was a time series of the vertical displacement of the monitoring rings as sensed by the NCDT's, and the angular position of the screw as seen by the drive motor's rotary encoder. The data recorded at the "full speed" of the run (e.g. Appendix 8.4, Figure A8.4.1) was extracted and plotted as a function of angular position (Figure A8.4.2), and an average of ring surface vertical position as a function of angle was formed (Figure A8.4.3). This was done both with and without excitation weights. The difference between the two plots was taken, thus giving the effect of the weights alone (Figure A8.4.4). This curve was Fourier analysed to extract the shaft order amplitude, which is the one caused by out-of-balance excitation. Grouping these values by monitoring ring position enables a graph to be plotted of observed amplitude versus speed for each position. The results are summarised in tabular form in Appendix 8.4.

The C program was run with a slow acceleration to full speed for each monitoring ring position. This gave a plot of vibration amplitude versus speed which can be compared directly to the measured results, (see Figures 8.13 to 8.15). In these graphs "Ring 1" and "Ring 2" refer to data extracted from measured values and "Ring 1 p" and "Ring 2 p" refer to values predicted by the C program.

Table 8.4 – *"Fixed nut" dynamic tests – correlation of measured and predicted results*

(See Appendix 8.4 and Appendix 6.1)

Position	1	2	3	Overall	
Ring 1, rms deviation	4.5	5.5	4.6	4.9	µm
Ring 2, rms deviation	7.1	5.4	6.8	6.4	µm

As shown in Table 8.4, overall the model predicted vibration levels which agreed with experiment to within about 5 to 6.5 µm. Given that the ballscrew was modelled as a single section shaft and the bearings as simple springs, this shows that the modelling method is basically sound.

8.3 **Dynamic measurements – nut moving along screw**

After taking the measurements with the ballscrew nut clamped close to the non-drive end bearing, the clamps were removed and the nut was re-attached to the saddle. As for the "fixed nut" measurements, two monitoring rings were fitted to the screw, one as close as possible to the drive end bearing, the other being fitted between the column support bridge and the ballscrew nut and was used to carry the eccentric weights used to apply radial force excitation to the ballscrew. In the case of this second ring it was possible to fit it in a variety of positions. The controller was set to run the nut from close to the non-drive end of the screw to close to the second monitoring ring.

Deflection of the ballscrew was measured using NCDT's as in the case of "fixed nut" measurements.

8.3.1 **Preliminary considerations**

As for the "fixed nut" case, the measured value can be expressed by:-

$$\{y_{m\,i}(x_j)\} = \{y_i(x_j, geom, matl, load)\} + \{\varepsilon(x_j, \theta_{x\,j})\},\ j = 1 \cdots N_m \qquad (8.22).$$

In this case the term *geom* should be understood to include effects caused by the different axial positions occupied by the nut as the screw turns. Again, given that the dynamic loads are applied by an eccentric weight, it is anticipated that *load* will depend on the square of the speed of the screw. If the speed of the ballscrew is low, *load* will be close to zero.

8.3.2 **Measurement strategy**

Therefore the measurement strategy was to set the second monitoring ring in a particular axial position and to subject the screw to two sets of out-of balance loading, one with the eccentric weights applied and one without. Because of the limited time at which vibrations were being generated, the maximum speed used was increased to 40 000 mm/min. Using this

speed as a maximum four other speeds were determined so as to give 40%, 60%, 80% and 90% of the dynamic force applied at the maximum speed. A final speed was chosen to give a "slow roll" the results of which might possibly be used to subtract out the effects of the run-out of the monitoring ring and other errors.

In order to make comparison of the different data sets as easy as possible as the speeds were reduced the data sampling period was increased.

Three loading positions were used. These were such that Dimension X (see Figure 8.12) was 57.2 mm, 230 mm and 400 mm respectively.

8.3.3 **Measurement results**

As in the case of the "fixed nut" investigation, the raw data recorded by the data logger for each test was a time series of the vertical displacement of the monitoring rings as sensed by the NCDT's (e.g. Appendix 8.5, Figure A8.5.1), and the position of the screw as seen by the drive motor's rotary encoder (Figure A8.5.2). The data recorded at the "full speed" of the run was extracted and plotted as a function of angular position (Figure A8.5.3), and a picture was built up of how the ring surface vertical position as a function of angle developed with time (Figure A8.5.4). This was done for both with and without excitation weights. The difference between the two plots was taken, thus giving the effect of the weights alone. These curves were Fourier analysed and the first five shaft orders were extracted and plotted against time. It was found that the first order was the largest of the five and showed a level which changed consistently with nut position. The other orders were significantly lower than the first order and fluctuated as the position of the nut changed (Figure A8.5.5).

The C program was run with the monitoring ring set at each of the three positions. To illustrate its capabilities three samples are shown here. The first order measured vibration levels are being compared with levels predicted by the C program.

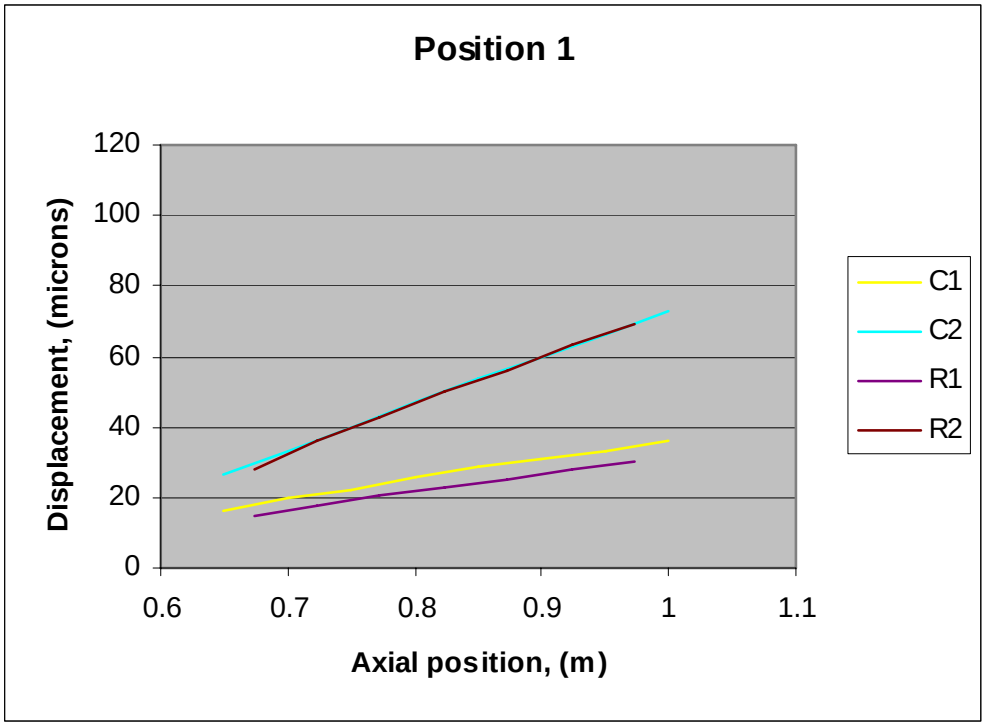


Figure 8.16 – *Comparison of measured and predicted vibration levels Monitoring ring position 1, speed 5*

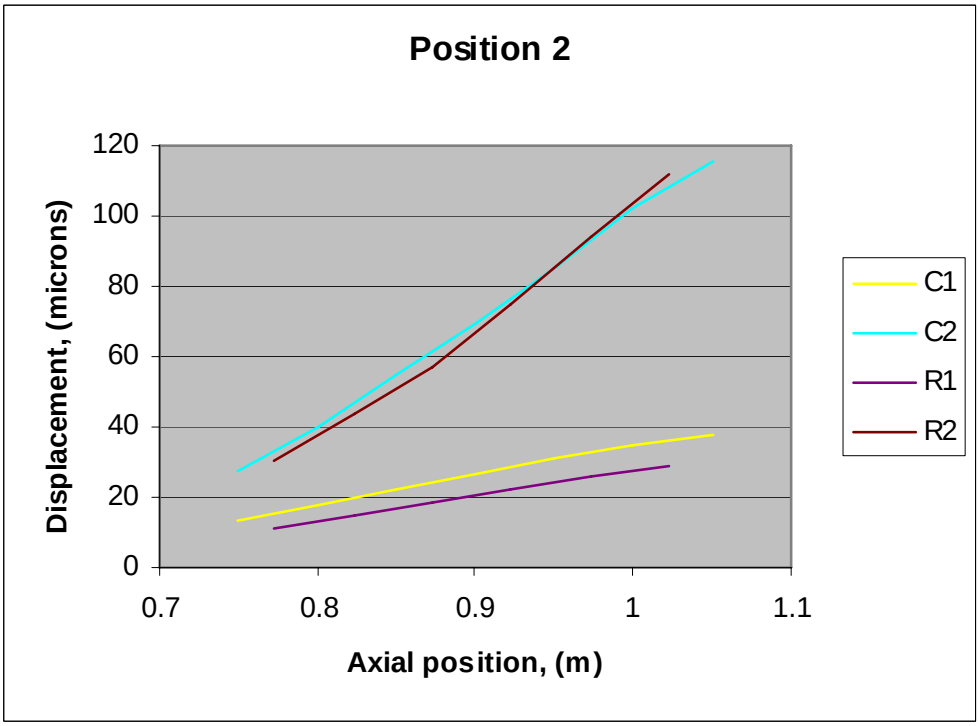


Figure 8.17 – *Comparison of measured and predicted vibration levels Monitoring ring position 2, speed 5*

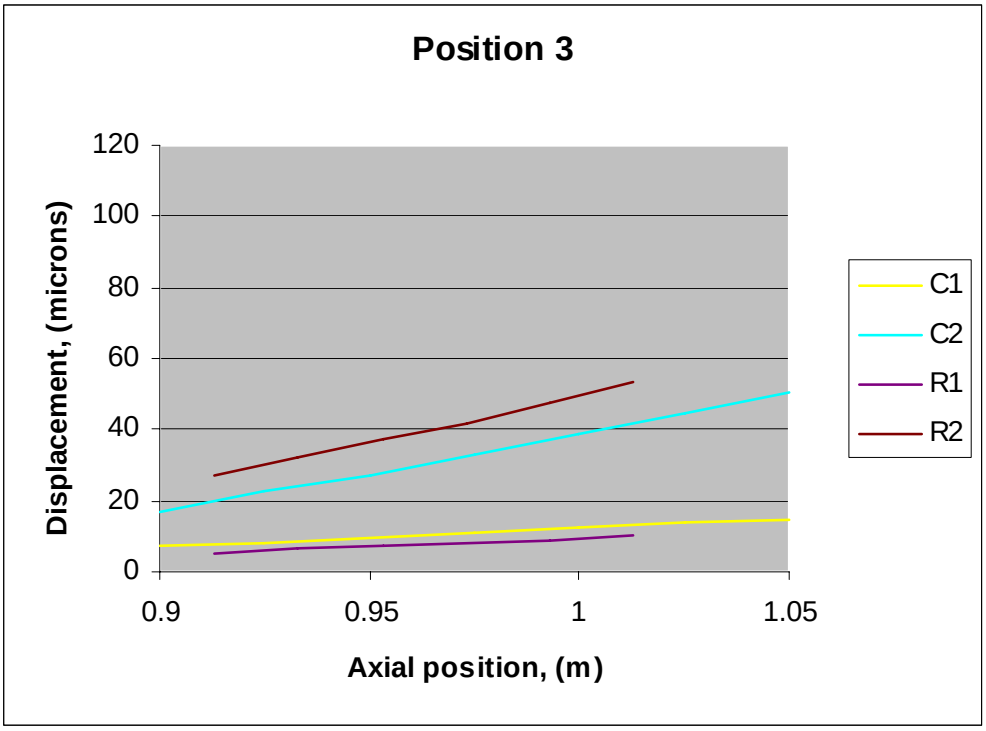


Figure 8.18 – *Comparison of measured and predicted vibration levels*
Monitoring ring position 3, speed 3

In the keys to Figures 8.16 to 8.18 "R1" and "R2" refer to data extracted from measured values for monitoring rings 1 and 2 respectively, and "C1" and "C2" refer to the corresponding values predicted by the C program.

Table 8.5 – *"Moving nut" dynamic tests – correlation of measured and predicted results*

(See Appendix 8.5 and Appendix 6.1)

Position	1	2	3	Overall	
Ring 1, rms deviation	5.2	4.7	3.0	4.4	µm
Ring 2, rms deviation	0.6	2.8	8.7	5.6	µm

For most of the positions tested an excellent degree of agreement was obtained, (see Table 8.5), though the results presented for Positions 1 and 2 (Figures 8.16 and 8.17) were generally better than those for Position 3 (Figure 8.18). Again, bearing in mind the simplifications used in setting up the ballscrew model, the results show the method to be well founded.

Chapter 9 - ERROR REDUCTION

A ballscrew drive converts an angular position θ of the motor to a linear position x of the machine part attached to the nut. The ideal position of the nut is given by the equation

$$x = \frac{N_s p}{2\pi}\theta = R\theta \tag{3.1}$$

where the ratio of the nut R is defined by the pitch p and the number N_s of helices in the ballscrew.

Errors between the ideal position and the actual position arise because of errors in manufacture of the ballscrew, errors in installation, deflection of the drive components under load, thermal movements and errors that arise due to the action of the drive's control system.

Attempts can be made to decrease these errors by improvements in design and installation, or by means of compensation.

9.1 Error reduction through design

Errors could, in principle, be reduced by improving the precision of manufacture, but ballscrews are already offered in a variety of grades of precision, up to JIS c0 [**122**] or better. These are at the forefront of precision manufacture, and can be used if the application justifies the costs involved.

9.1.1 Effects on stiffness of detailed geometry

The precision of a ballscrew drive could be improved by increasing its stiffness because the deflections of the drive would be lower for a given applied force. The components which contribute to the overall flexibility are the nut, the ball-mechanism, the screw, the support bearings and the structure in which the ballscrew is mounted.

Using manufacturer's data [**26**], a range of precision ballscrews mounted in a typical support bearing arrangement has been considered. The screw sizes varied from 12 mm to 50 mm in diameter with bearings to match. The screw lengths considered were the maximum available in accuracy grade c1. By applying a unit torque in each case it was found that the proportion of the resulting axial deflection arising from the various components of the drive was as follows:-

Table 9.1 – *Distribution of ballscrew drive flexibility*

Element of flexibility	Range	Average
Equivalent of torsional wind-up of screw	1.2 – 12.5%	4%
Movement in nut	35.0 – 54.9%	43%
Screw deflection – average of both sides of nut	30.2 – 50.4%	43%
Bearing deflection - average of both bearings	8.8 – 11.1%	10%

Of these, typically half of the total deflection is made up of deflection of the screw itself, and the only way of increasing the stiffness of this component is to make the screw bigger. This would have the effect of increasing the cost, the running speed of the bearings and ball mechanism, which would have thermal implications, and increasing system inertia, which would have energy consumption and control implications.

About 10% of total deflection is in the bearings. This can be reduced by putting pre-tension in the screw, but at the expense of greater energy consumption, heat generation and a greater tendency to bearing wear.

Just under half of the total deflection was found to occur in the nut. The flexibility of this component is dependent to a significant degree on the Hertzian contact between the circulating balls and the nut and screw. A normal ballscrew has its races packed as full as possible with balls, and the races are typically of gothic arc section with a conformity of about 55%. (This means that the radius of the race profile is 1.1× that of the balls.) This suggests that provided the nut contains as many ball races as can be reasonably fitted in the space available, there is little scope for increasing the stiffness beyond increasing the length of the nut or the pre-load. Increasing the pre-load also increases energy consumption, heat generation and the tendency to wear.

9.1.2 **Thermal considerations**

Wherever possible, a machine which is being used to produce high-precision components should be run in a temperature-controlled environment in a location where it does not receive direct sunshine. This should prevent environmental effects distorting the components of the machine of which the drive is part.

As the ballscrew is run some heat is generated, the more so if high levels of pre-tension and pre-load are used. It is possible to bring the temperature nearer to ambient by cooling the ballscrew, the nut or the bearing housing, though control of such cooling would require careful consideration due to the time scales involved.

9.1.3 **Material considerations**

The stiffness of a component part is dependent on the material's elastic modulus, as well as on its shape. The elastic modulus for pure metals ranges from 11.0 GPa for indium to 517 GPa for iridium [**123**]. Those for commonly used engineering materials range from 15.9 GPa for lead to 226 GPa for Nimonic 90. Carbon steel at 206 GPa is close to the upper end of this range [**118**]. Ceramic materials typically have values of 275 – 350 GPa for alumina [**124**], 160 – 350 GPa for zirconia [**125**], 600 GPa for tungsten carbide [**126**], 170 – 290 GPa for silicon

nitride and 390 – 450 GPa for silicon carbide [127]. Machineable ceramics, however, have elastic moduli of only 19 – 66.9 GPa.

Iridium is one of the platinum group metals and is the most corrosion resistant metal known. It is, however, very hard and brittle, making it very hard to machine, form, or work [128]. This makes it an undesirable choice of material for a ballscrew. Ceramics are a possibility, bearings with ceramic balls have already been developed and are used in certain applications. The contribution of material compliance to Hertzian flexibility is represented by the factor C_E, (see §3.5.1, equation 3.16).

For steel on steel, $$C_E = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} = 2 \times \frac{1-0.29^2}{206 \times 10^9} = 8.89 \times 10^{-12} \text{ m}^2/\text{N} \qquad (9.1)$$

For tungsten carbide on steel,

$$C_E = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} = \frac{1-0.26^2}{600 \times 10^9} + \frac{1-0.29^2}{206 \times 10^9} = 6.00 \times 10^{-12} \text{ m}^2/\text{N}. \qquad (9.2)$$

This represents a 48% increase in the ball action stiffness when compared with steel balls.

If the nut were to be made of Macor, a machineable ceramic, and the balls of tungsten carbide, for ball – nut contact

$$C_E = \frac{1-\nu_1^2}{E_1} + \frac{1-\nu_2^2}{E_2} = \frac{1-0.26^2}{600 \times 10^9} + \frac{1-0.29^2}{66.9 \times 10^9} = 15.24 \times 10^{-12} \text{ m}^2/\text{N}. \qquad (9.3)$$

The screw – ball contact would still have a C_E of 6.00×10^{-12} m²/N, this would give an average C_E value for the screw – nut engagement of 10.62×10^{-12} m²/N which is a 16% reduction in stiffness compared to an all steel ballscrew.

9.2 Error reduction through compensation

9.2.1 Compensation for dynamic effects

Correction of spatial dependent errors

The axial pitch errors can be measured by laser. The data can be stored in a look up table. The errors can therefore be corrected for by modifying the position error signal in the controller. This is a well-established method [46].

Correction of time dependent errors

The axial and torsional errors can be represented by a simple model. Basically the position error which is caused by the torque applied to the ballscrew can be closely approximated as a simple function of load and axial position. The load can be determined from the drive motor current, a measure of which can be taken from the controller, and the position can be determined from the output of the rotary encoder.

For example, in simple static terms, the torsional deflection at the centre of the nut θ_{bs} is given by:-

$$\begin{aligned}\theta_{bs} &= fl_{sys} T_{m\,net} \\ &= \left(\frac{1}{k_{ms}} + \frac{1}{k_{dr}} + \frac{1}{k\theta_{bs}}\right)\end{aligned} \tag{9.4}$$

where

- fl_{sys} = the total system flexibility referred to the motor shaft between the motor and the ballscrew nut,
- $T_{m\,net}$ = the motor output torque, and
- k_{ms}, k_{dr} and $k\theta_{bs}$ = the stiffness of the motor shaft, drive coupling and ballscrew.

The motor output torque can be approximated by:-

$$T_{m\,net} = \frac{J_{tot} - J_m}{J_{tot}} T_m \tag{9.5}$$

where

- J_{tot} = the total referred inertia of the ballscrew drive system,
- J_m = the motor inertia, and
- T_m = the torque applied to the motor rotor by its electrical fields.

The torsional stiffness of the ballscrew is given approximately by

$$k\theta_{bs} = \frac{\pi G(d_o^4 - d_i^4)}{32x} \tag{9.6}$$

where

- G = the shear modulus of the ballscrew material,
- d_o and d_i = the outer and inner diameters of the ballscrew, and
- x = the position of the nut.

Similarly, the axial deflection at the centre of the nut δx_{bs} is given by:-

$$\begin{aligned}\delta x_{bs} &= fl_{bs} F_{nut} \\ &= \frac{F_{nut}}{\dfrac{1}{1/k_{ax\,B1} + x/EA_{bs}} + \dfrac{1}{1/k_{ax\,B2} + (l_{bs} - x)/EA_{bs}}}\end{aligned} \tag{9.7}$$

where

- fl_{bs} = the axial flexibility of the ballscrew,
- F_{nut} = the axial force delivered by the ballscrew to the nut,
- $k_{ax\,B1}$ and $k_{ax\,B2}$ = axial stiffness of the ballscrew support bearings,

- l_{bs} = length of ballscrew between its supporting bearings,
- E = the modulus of elasticity of the nut material,

$$A_{bs} = \frac{\pi}{4}(d_o^2 - d_i^2) \tag{9.8},$$

and force on the ballscrew nut can be approximated by:-

$$F_{nut} = m_{load} R \frac{T_m}{J_{tot}} \tag{9.9}$$

where

- m_{load} = the mass of the nut and saddle and/or table, and
- R = the ballscrew ratio.

The total deflection to be compensated for δx_{comp} is then given by

$$\delta x_{comp} = R\theta_{bs} + \delta x_{bs} \tag{9.10}.$$

By a similar approach, if the tilt stiffness of the slideway which carries the load being driven by the ballscrew is known, a term which compensates for tilting of the load can be included.

The program described in Chapters 5 and 6 of this Thesis has been modified to include such compensation.

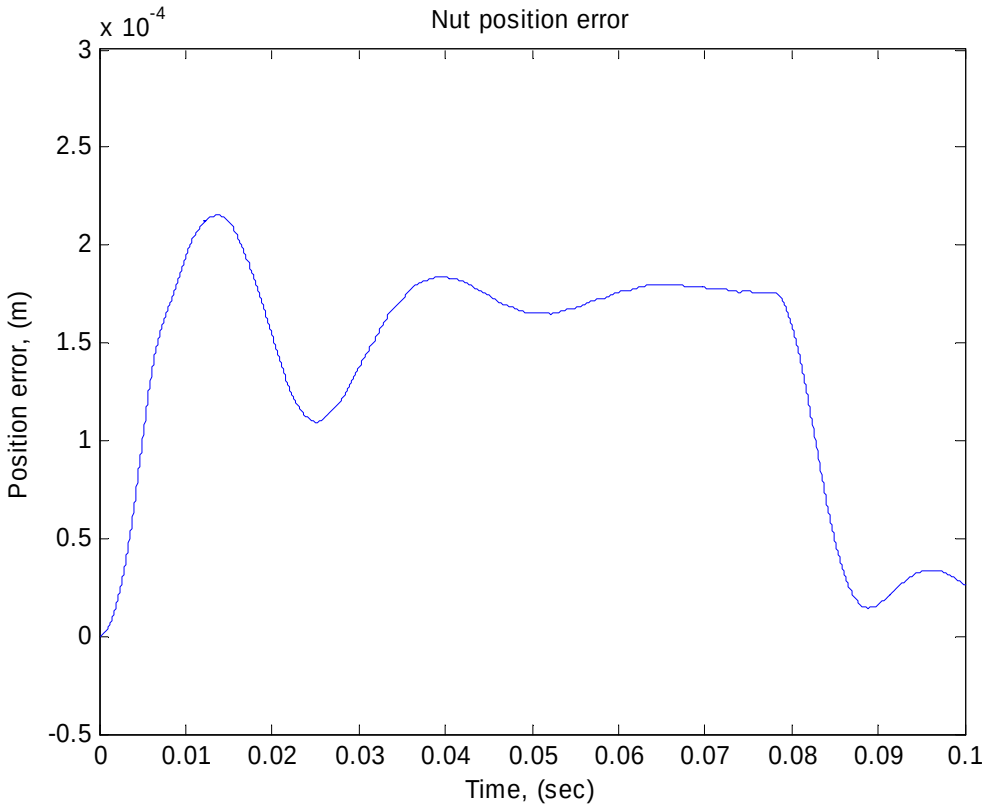


Figure 9.1 *Nut position error without compensation*

Figure 9.1 shows the nut position error predicted by the model during the acceleration phase of a trapezoidal velocity demand cycle. If the various easily predictable mechanical deflections are compensated for, the position error can be reduced to the values shown in Figure 9.2.

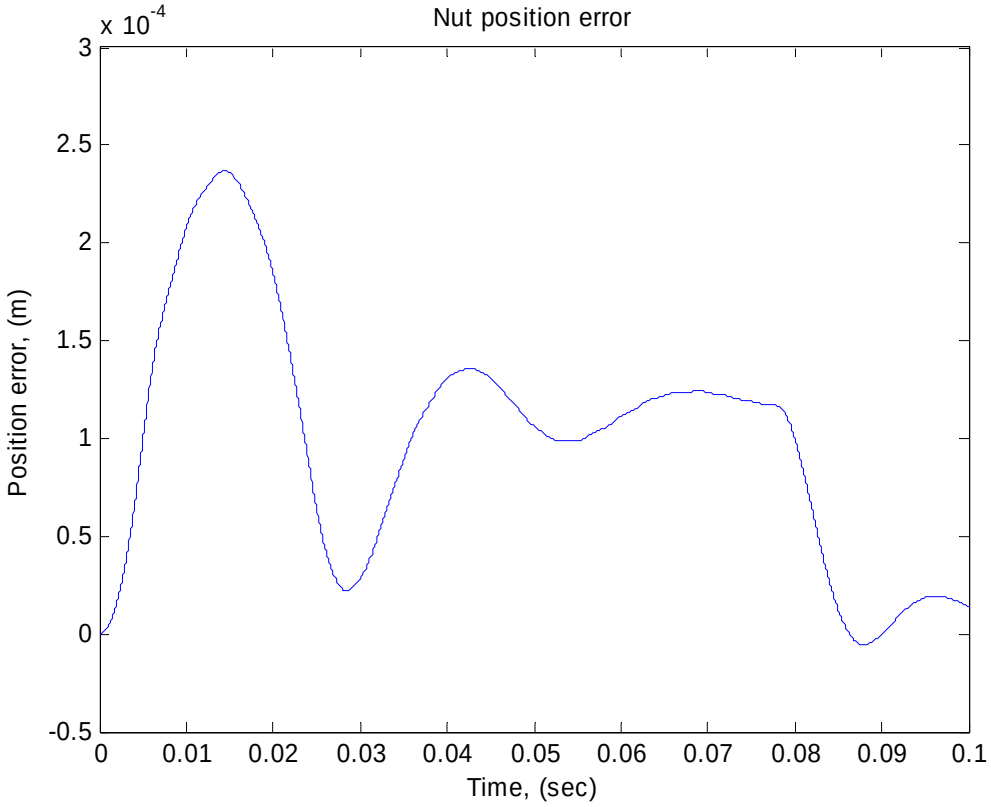


Figure 9.2 *Nut position error with compensation for flexibility*

The characteristics of the controller play a significant role in the behaviour of a ballscrew drive. In the case of a simple analogue PI (proportional integral) velocity controller, the "integral" term, together with the motor constant, have the same effect as a mechanical spring, and introduces an error during the acceleration phase. (Similarly, the "proportional" term has a damping effect.) Once known, these terms can be easily compensated for.

However, modern machine tool controllers have more sophisticated logic than a simple PI device, and include velocity feed forward components and other features which, when tuned correctly, compensate for acceleration lag error. As an example, the Heidenhain controller used on the linear guide test rig at the University of Huddersfield has been studied in detail, the results of which are set out in Appendix 9.1. Were this controller to work in analogue form, it should be able to accelerate a load with little error. The reality is that it works in a "quantised" manner, that is, its feedback loops are changed at fixed time steps. This introduces both delay errors and vibration excitation.

A compensation strategy which can compensate for such errors needs therefore to take account of the detailed characteristics of the controller. An attempt to compensate for effects other than mechanical deformation using the techniques derived in this Thesis gives the result shown in Figure 9.3.

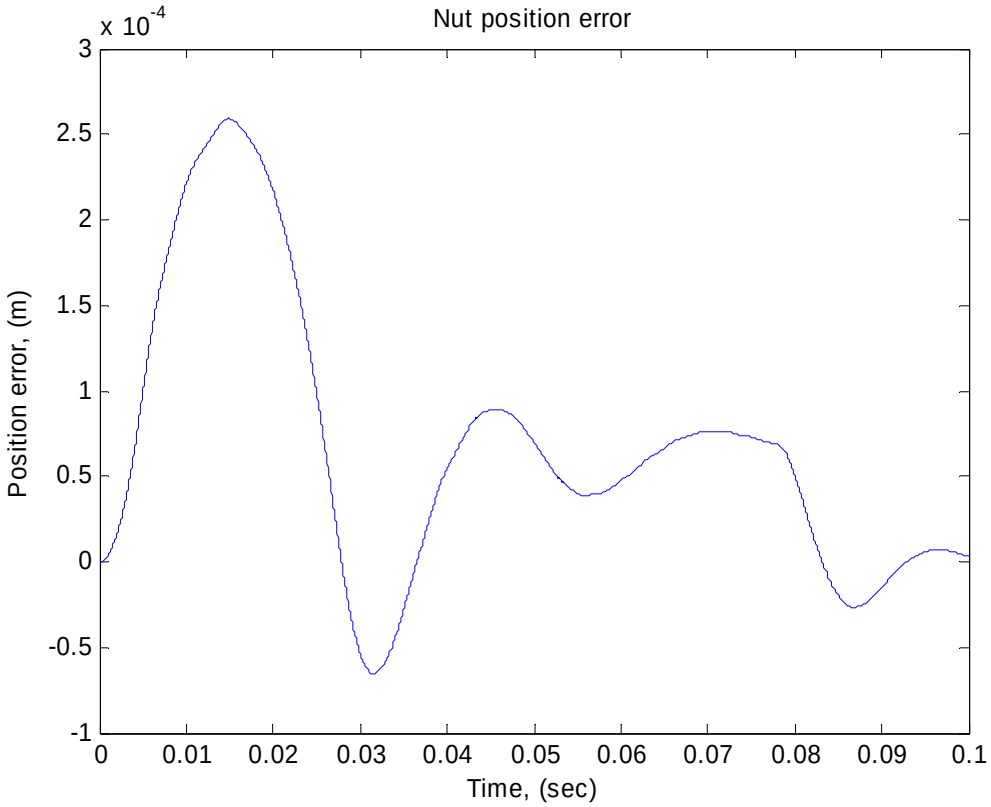


Figure 9.3 *Nut position error with compensation for flexibility and controller effects*

As can be seen, although some improvement in "steady" state errors has been achieved, transient responses at start-up and at the end of the acceleration phase remain largely unchanged.

9.2.2 **Compensation for thermal effects**

Thermal errors in machine tool drives have been the subject of a lot of research. Effective compensation strategies have been devised which reduce substantially the effect of thermal errors [**38**].

In conclusion,

- ballscrews have reached an advanced state of development and can already reach a high standard of precision
- spatial errors which arise in their use can be compensated for
- thermal errors can be compensated for
- "steady state" errors arising from mechanical loading can be compensated for

- transient errors cannot be compensated for but can be reduced by incorporating a suitable level of damping in the mechanical system, and by adopting "smooth" methods of control, e.g. "$\sin^2$" demand characteristics, wherever possible.

Chapter 10 - CONCLUSIONS

During the course of the programme of research covered by this Thesis the geometric, thermal and load errors commonly encountered on CNC machine tools have been studied. Several mathematical models have been developed or extended which enable a deeper understanding of the interaction between these errors, various details of ballscrew design and the dynamic behaviour of ballscrew driven systems. Further, a strategy for compensation of some of these errors has been devised.

Results

The results of the research are detailed as follows:-

- The static elastic theory relevant to ballscrew machine tool drives has been brought together to form a group of formulae useful for estimating the static stiffness characteristics of such systems.
- The relevant dynamic elastic theory and the theory of thermal modelling have been applied to develop a detailed understanding of the behaviour of a ballscrew driven system. A set of working models have been assembled which cover the dynamic and thermal aspects of ballscrew behaviour.
- The continuous matter approach to the dynamics of a ballscrew using wave solutions was developed. Although this succeeded in giving satisfactory results for the case of axial and torsional behaviour, it did not prove to be a useful method for dealing with transverse vibratory behaviour.
- Several models based on the discrete matter or “lumped mass” approach were devised. Using matrix solutions, three useful models were developed as follows.
- The first of these models extends the classical eigenvalue method for finding the natural frequencies and other dynamic characteristics of ballscrew systems to include viscous damping effects by using a generalised eigenvalue approach. This not only gives estimates of the natural frequencies, but also the damping coefficient of each predicted vibration mode. This feature enables many of the natural frequencies predicted by standard undamped natural frequency analyses to be dismissed as being of little consequence to the vibratory behaviour of the system and is a useful contribution to knowledge in this field.

  A development of this modelling method gives the sensitivity of the system to changes in stiffness and damping characteristics. When a mode of vibration is considered to be potentially detrimental to a system’s performance, the design parameters which have

greatest influence on the mode can be identified. Changes in these parameters can be tested quickly by the model and improved dynamic characteristics found before going on to the stage of detailed design changes. This is a tool useful both at the preliminary design stage of a ballscrew system, and for helping to decide the most convenient remedy to vibration problems which may occur in service.

With some modification, the approach can be applied to many electro-mechanical systems. Specifically, the controller parameters can be included in the model giving a means of investigating how changes in such parameters can be used to modify the drive system's behaviour.

By using a suitable means of deriving a set of viscous damping coefficients which are equivalent, in defined circumstances, to non-linear damping effects, it is possible to extend this method to include non-linear behaviour.

- The second set of lumped-mass models is specially developed to take account of the changes in the configuration of the system with time as the nut moves along the screw. These deal only with the axial and torsional degrees of freedom of the system, which have the most direct impact on the response of a ballscrew driven system. A special difficulty arises when the position of the nut does not coincide with a node in the model. Using a “simple” approach to connecting the nut with the screw gave rise to a response that was dependent on the structure of the model. This problem has been successfully addressed by using a “massless node” as the connection point. Implementation of this approach entailed the development of an algorithm based on sophisticated matrix algebra, which has succeeded in advancing knowledge by giving a seamless dynamic response.
- These models can predict the position error of the ballscrew nut.
- The third set of models includes the transverse degrees of freedom as well as the axial and torsional ones. These give a means of predicting the vibration amplitudes which might be expected as a system “runs through” a natural frequency, either because the system parameters change as the nut moves along the screw or because the speed of the screw is changing.
- The second and third sets of models include the non-linear phenomena of backlash and Coulomb friction as well as time dependent stiffness and damping characteristics. Both of these models are capable of being extended to include other non-linear phenomena, such as stiffness which is dependent on the state of strain (e.g. Hertzian

contact stiffness characteristics) and drag terms which are dependent on the square of the speed.

- The three groups of lumped mass models all include features which calculate the energy converted to heat by all the energy dissipative mechanisms in the model. The Engineering Control and Machine Performance Group at the University of Huddersfield have already produced a model which predicts the thermal behaviour of a ballscrew system. This model has been validated experimentally. The heat output from a dynamic model has been shown to correlate well with the heat levels used in this proven thermal model. Therefore dynamic models and a suitably adapted thermal model, used together, could be used to predict the thermal behaviour of a ballscrew system.
- Various methods by which cooling parts of the ballscrew system can be used to reduce thermal errors have been investigated. The use of a flow of chilled water based coolant is the one most likely to best in the majority of cases.
- A novel experimental approach has been devised which uses beam theory as a basis for measuring support stiffness characteristics which are not possible to measure directly unless the bearing system has been designed with this in mind. The method allows measurements to be taken on accessible parts of the system and represents a contribution to knowledge.
- Comparison of the model predictions with well-established modelling methods and with data obtained from a machine tool and from a special test rig show that the modelling approaches are valid.
- The effect of modifying the design of precision ballscrews has been investigated. The existing designs incorporate many features which minimise errors. This programme of research suggests that the basic design is sound, and that efforts to reduce substantially the remaining errors arising from design or manufacture should be concentrated on understanding better the details of their behaviour in the applications in which they are used.
- An algorithm which will correct for the errors due to the expected elastic deformations of the ballscrew system has been devised.
- The position error can be reduced further by correcting for the lags induced by the controller

Further work

The following fields of investigation are suggested by the results of the work covered by this Thesis:-

- The methods devised in this investigation enable the effects of damping to be assessed at the early stages of the design of a machine tool drive. It is recommended that the generalised eigenvalue approach be extended to include the electrical aspects of the system and the controller. This will enable hybrid model parameters to be optimised, and will enhance the understanding of the influence of controller characteristics on the vibratory behaviour of ballscrew-driven systems.
- The dynamic models enable not only the behaviour of the ballscrew nut to be predicted, but also that of the motor, of all parts of the screw, including ballscrew wind-up and transient torsion and tension deformation, and the three-dimensional behaviour of the machine tool component driven by the nut. It is recommended that this approach be incorporated into further research into improving overall control of a ballscrew system. Specifically, this modelling approach enables the output from a rotary encoder on the motor, and that from a rotary encoder on the non-drive end of the screw, and that from a linear encoder mounted close to a slideway all to be predicted. The method can therefore be used as a basis of investigating a multi-variable control strategy whose aim would be to reduce transient errors in machine tool drives.
- The temperature dependency of dissipative mechanisms should be investigated further with a view to developing an interacting pair of models which predict thermal behaviour of a system together with mechanical behaviour (sampled) over long time scales.
- The techniques developed here can be extended to devise a control system that reduces thermal errors by regulating the supply of chilled coolant to the ballscrew centre, the nut, and possibly the ballscrew support bearing housings.
- The methods of dynamic analysis used in this research are not limited to ballscrew driven systems. They can, for example, be used to investigate the dynamics of gear tooth excitation where the conditions of engagement change continuously as the teeth move through the mesh.

REFERENCES

1 MTTA, "British Machine Tool Industry – Basic Facts 2002", Tables 2 and 3

2 Aomori H., "Metal cutting machine tools, Japan", Industry Sector Analysis, U. S. Department of Commerce - National Trade Data Bank, November 3, 2000.

3 Lee J., "Overview and Perspectives on Japanese Manufacturing Strategies and Production Practices in Machinery Industry", International Journal of Machine Tools and Manufacture, Vol. 37 No.10, 1997, pp.1449-1463

4 Fletcher S., "Computer aided System for Intelligent Implementation of Machine Tool Error Reduction Methodologies", PhD thesis, University of Huddersfield, 2001.

5 Bryan J., (Lawrence Livermore Nat. Lab.), "International status of thermal error research". Annals of the CIRP, Vol. 39, N2, P645-656, 1990.

6 Donmez A., "A general methodology for machine tool accuracy enhancement - theory, application and implementation", PhD thesis, Purdue University, 1985.

7 Blake M. D., "Investigation into load effects on machine tool accuracy", MPhil thesis, University of Huddersfield, 1995.

8 Blake M. D., Ford D. G., Postlethwaite S. R., Morton D., "Analysis of Machine Tool Non-Rigid Error Components", Laser Metrology and Machine Performance II, P297-308, 1995.

9 Ford D. G., Blake M. D, Postlethwaite S. R., "The identification of non-rigid errors in a vertical machining centre", Proceedings of the Institution of Mechanical Engineers, part B, Vol. 213, pp 555-566, 1999.

10 Ford D. G., Postlethwaite S. R., White A., Pislaru C, "Time and Spatial Error Correction in CNC Machine Tools", keynote lecture: IEE (Yorkshire Region) Control and Manufacturing Sections, 8 December 1998

11 Mou J., "A Systematic Approach to Enhance machine Tool Accuracy for Precision Manufacturing", International Journal of Machine Tools Manufacturing. Vol. 37, No. 5, pp. 669-685, 1997

12 Slocum A. H., "Precision Machine Design", Prentice Hall, Englewood Cliffs, NJ, 1992, (ISBN 0 87263 492 2)

13 Pislaru C, Ford D. G., Freeman J. M., "Identification of Modal Parameters Affecting the Dynamic Performance of a CNC Machine Tool", pp. 161-170, LAMDAMAP 2003, Ed. Prof D. G. Ford, WIT Press, (ISBN 1 85312 990 9)

14 Pislaru C., Ford D. G., Holroyd G., "Modelling Non-linearities from CNC Machine Tool Drives", Proc. of the 8th Internat. Conf. on Optimization of Electrical and Electronic Equipments OPTIM 2002, Brasov, Romania pp. 553 – 558 (ISBN 973-635-004-5)

15 Pislaru C., "Parameter Identification and Hybrid Mathematical Modelling Techniques Applied to Non-Linear Control Systems", Ph.D. Thesis, 2001, University of Huddersfield

16 Holroyd G., Pislaru C., Ford D. G., "Determination of stiffness and damping sensitivity for computer numerically controlled machine tool drives", Proc. Instn. Mech. Engrs Vol 217, Part C: J. Mechanical Engineering Science, pp 1165-1177, 2003

17 Diodorus Siculus, "Bibliotheke", Book V, 37.3-4, c. first century BC

18 Oldfather C. H. (translator), in "Diodorus Siculus", Library of History, Volume III, Loeb Classical Library, Harvard University Press, Cambridge, 1939, (ISBN 0-674-99375-6)

19 Mulcahy David E., "Materials Handling Handbook", McGraw-Hill, Ch 3, Pages 3.200-3.205, 1998, (ISBN 0 070 44014 X)

20 Woodbury Robert S., "Studies in the history of machine tools. History of the lathe to 1950", The M.I.T. Press, Ch II, Page 47, 1972, (ISBN 0 262 73033 2)

21 Lipkin H.; Duffy J., "Sir Robert Stawell Ball and methodologies of modern screw theory", Proceedings of the I MECH E, Part C, Journal of Mechanical Engineering Science, 1 January 2002, vol. 216, no. 1, pp. 1-11(11)

22 Ball Sir Robert Stawell, "A Treatise on the Theory of Screws", Cambridge University Press, 1900, (paperback edition 1998, ISBN 0-521-636550-7)

23 Pfister F., "Spatial point contact kinematics and parallel transport", Proceedings of the I MECH E Part C Journal of Mechanical Engineering Science, 1 January 2002, vol. 216, no. 1, pp. 33-45(13)

24 Neumann C., "Über die rollende Bewegung eines Körpers auf ciner gegebenen Horizontal-Ebene unter dem Einfluss der Schwere", berichtee über die Verhandlungen der königlich sächsischen Gesellschaft der Wissenschaften zu Leipzig, Mathematisch-Physische Classe, **37**, 352–378, 1885

25 Richter M., "Über die Bewegung eines Körpers auf einer Horizontal-Ebene", Inaug. Dissertation, University of Leipzig, Leipzig, 1887 (Metzger und Wittig)

26 "LM system ball screws", catalog. no.200-2BE, THK Co. Ltd., chapter 4, 1999, (© THK Ltd, 04 990425)

27 Ford D. G., Postlethwaite S. R., Allen J. P., Blake M. D., "Compensation algorithms for the correction of time and spatial errors in a vertical machining centre", Proc. Inst. Mech. Engrs., Vol 214, Part B, pp221-234, 2000

28 Postlethwaite S. R., Ford D. G., Morton D., "Geometric errors from dynamics data – a novel calibration technique", Proc. 2nd Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '95, Southampton Institute, 1995, pp 139-148, (ISBN 1 85312 355 2)

29 Postlethwaite S. R., Ford D. G., "Geometric error analysis software for CNC machine tools", Proc. 3rd Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '97, University of Huddersfield, 1997, pp305-316, (ISBN 1 85312 536 9)

30 Freeman J. M., Ford D. G., "The analysis of geometrical and thermal errors in non-Cartesian machine structures", Proc. 3rd Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '97, University of Huddersfield, 1997, pp371-379, (ISBN 1 85312 536 9)

31 Ford D. G., Postlethwaite S. R., "Accuracy improvement for three axis CNC machining by geometric, load and thermal error compensation", Proc. 4th Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '99, Longhirst Hall, Northumberland, 1999, pp113-124, (ISBN 1 85312 661 6)

32 White A., J., Postlethwaite S. R., Ford D. G., "An investigation into the relative accuracy of ballscrews and linear encoders over a broad range of application configurations and usage conditions", Proc. 4th Int. Conf. on Laser Metrology and

Machine Performance – LAMDAMAP '99, Longhirst Hall, Northumberland, 1999, pp345-355, (ISBN 1 85312 661 6)

33 Frank A., Ruech F., "Thermal behaviour of position measurement systems: linear transducer – rotary encoder", Proc. EUSPEN Conf., Bremen, Germany, 31st May – 4th June 1999, pp124-127

34 Chen J. S., "A study of thermally induced machine tool errors in real cutting conditions". *International Journal of Machine Tools and Manufacture*, Vol. 36/12, pp. 1401-1411, 1996.

35 Wang Y., Zhang G., Moon K. S., Sutherland J. W., "Compensation for the thermal error of a multi-axis machining center", *Journal of Materials Processing Technology*, Vol. 75, pp. 45-53, 1998

36 Fraser S., Attia M. H., Osman M. O. M., "Modelling, identification and control of thermal deformation of machine tool structures, part 1: concept of generalized modelling", *Transactions of the ASME, Journal of Manufacturing Science and Engineering*, Vol. 120, pp. 623-631, August 1998

37 Jan H.-K., Chu C. N., Liu C. R., "A configuration independent error model of machine tools: hyperpatch model and metrology pallet", Robotics and Computer-Integrated Manufacturing, Vol. 9/3, pp. 201-210, 1992

38 White A. J., Postlethwaite S. R., Ford D. G., "A general purpose thermal error compensation system for CNC machine tools", LAMDAMAP 2001, University of Huddersfield, UK, pp. 3-13, (Lam2001_2.doc), (ISBN 1 85312 890 2)

39 White A. J., Postlethwaite S. R., Ford D. G., "Measuring and modelling thermal distortion on CNC machine tools", LAMDAMAP 2001, University of Huddersfield, UK, pp 69-79, (Lam2001_1.doc), (ISBN 1 85312 890 2)

40 Freeman J. M., White A. and Ford D. G, "Ball-screw thermal errors – a finite element simulation for on-line estimation", LAMDAMAP 2001, Precision Engineering Research Group, The School of Engineering, University of Huddersfield, England, pp 269-278, (Lamd 2001 MF.doc), (ISBN 1 85312 890 2)

41 Postlethwaite S. R., Allen J. P., Ford D. G., "The use of thermal imaging, temperature and distortion models for machine tool thermal error reduction",

Proc. Instn Mech. Engrs Vol 212 Part B: J. of Engineering Manufacture, pp 671-679, 1998

42 Postlethwaite S. R., Allen J., Ford D. G., "Practical application of thermal error correction – 4 case studies", Proc. 3rd Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '97, University of Huddersfield, 1997, pp359-369, (ISBN 1 85312 536 9)

43 Postlethwaite S. R., Allen J. P., Ford D. G., "Machine tool thermal error reduction – an appraisal", Proc. Instn Mech. Engrs Vol 213 Part B: J. of Engineering Manufacture, pp 1-10, 1999

44 White A., J., Postlethwaite S. R., Ford D. G., "An identification and study of mechanisms causing thermal errors in CNC machine tools", Proc. 4th Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '99, Longhirst Hall, Northumberland, 1999, pp101-112, (ISBN 1 85312 661 6)

45 Pahk H. J., Lee S. W., "Development of a thermal error measurement –real time correction system for the thermal error in CNC machine tools", Proc. EUSPEN Conf., Bremen, Germany, 31st May – 4th June 1999, pp92-95

46 Postlethwaite S. R., Ford D. G., "A practical system for 5 axis volumetric compensation", Proc. 4th Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '99, Longhirst Hall, Northumberland, 1999, pp 379-388, (ISBN 1 85312 661 6)

47 Postlethwaite S. R., Ford D. G., Morton D., "Dynamic calibration of CNC machine tools", Int. Journal of Machine Tools and Manufacture, vol 37, No 3, pp 287-294, 1997

48 Ford D. G., Postlethwaite S. R., Pislaru C., "High accuracy feedback transducer incorporating a real time error compensation system", Proc. EUSPEN Conf., Bremen, Germany, 31st May – 4th June 1999, pp100-103

49 Weck M., Hilbing R., "Characterisation and compensation of machine vibrations with non-linear methods", Proc. EUSPEN Conf., Bremen, Germany, 31st May – 4th June 1999, pp167-170

50 Pislaru C., Ford D. G., Freeman J. M., "A new 3D model for evaluating the performance of CNC machine tool axis drives", Proc. EUSPEN Conf., Bremen, Germany, 31st May – 4th June 1999, pp72-75

51 Pislaru C., Ford D. G., Freeman J. M., "A new approach to the modelling and simulation for a CNC machine tool axis drive", Proc. 4th Int. Conf. on Laser Metrology and Machine Performance – LAMDAMAP '99, Longhirst Hall, Northumberland, 1999, pp335-343, (ISBN 1 85312 661 6)

52 Erkorkmaz K. and Altintas Y., "High speed CNC system design. Part II: modelling and identification of feed drives", International Journal of Machine Tools and Manufacture, Vol 41 (10), August 2001, pp 1487-1509

53 Ro P. I., Shim W., Jeong S., "Robust friction compensation for submicrometer positioning and tracking for a ball-screw-driven slide system", Precision Engineering, April 2000, vol. 24, no. 2, pp. 160-173(14)

54 Dequidt A., Castelain J.-M., Valdes E., "Mechanical pre-design of high performance motion servomechanisms", Mechanism and Machine Theory, 1 August 2000, vol. 35, no. 8, pp. 1047-1063(17)

55 Braasch J., "Position measurement on machine tools: by linear encoder or ballscrew and rotary encoder?", Modern Machine Shop, http://www.mmsonline.com/articles/1198sup.htr, accessed 04/10/04

56 Huang S. C., "Analysis of a model to forecast thermal deformation of ball screw feed drive systems", Int. J. Mach. Tools Manufact., vol. 35, no. 8, 1995, pp. 1099-1104

57 Yun W. S., Kim S. K., Cho D. W., "Thermal error analysis for a CNC lathe feed drive system", Int. J. of Mach. Tools Manufact., vol. 39, 1999, pp. 1087-1101

58 Kim S. K., Cho D. W., "Real-time estimation of temperature distribution in a ball-screw system", Int. J. Mach. Tools Manufact., vol. 37 no. 4, 1997, pp. 451-464

59 Lin M. C., Velinsky S. A., Ravani B., "Design of the ball screw mechanism for optimal efficiency", Trans. ASME, vol. 116, Sept. 1994, pp. 856-861

60 Markhauser A. W., "Preloading ball screws", Machine Design, March 16, 1967, pp. 207-211

61 Schmitt T., "Modell der Wärmeübertragungs-vorgänge in der mechanischen Struktur von CNC-gesteuerten Vorschubsystemen, {A model of the heat transmission processes in the mechanical structures of CNC-controlled form feed systems}", Darmstädter Forschungs-berichte für Konstruktion und Fertigung {Darmstädt Research Reports for Design and Manufacture}, Ph. D. thesis, 1995, (ISBN 3-8265-1476-9).

62 De Silva C. W., "Vibration – fundamentals and practice", CRC Press, ch 7, pp 349 – 398, 1999, (ISBN 0-8493-1808-4)

63 Pestel C. P., Leckie F. A., "Matrix methods in elastomechanics", McGraw-Hill, 1963, (Library of Congress Catalog Card Number 62-19249)

64 Holroyd G., "Identification of damping elements in a CNC machine tool drive", MSc project dissertation, University of Huddersfield School of Engineering, September 2000

65 Holroyd G., Pislaru C. and Ford D. G., "Identification of damping elements in a CNC machine tool drive". In Proceedings of 5th International Conference on *Laser Technology, Machine Tool, CMM and Robot Performance* LAMDAMAP 2001 (Ed. WIT Press), Birmingham, UK, July 2001, pp. 289-299, (ISBN 1 85312 890 2)

66 Adhikari S., "Calculation of derivative of complex modes using classical normal modes", Computers & Structures, vol 77 (6), 15 August 2000, pp 625-63

67 Nelson R. B., "Simplified calculation of eigenvector derivatives". *AIAA J* **14** 9 (1976), pp 1201–1205

68 Adelman H. M., and Haftka R. T., "Sensitivity analysis of discrete structural system". *AIAA J* **24** 5 (1986), pp 823–832

69 Murthy D. V. and Haftka R. T., "Derivatives of eigenvalues and eigenvectors of a general complex matrix". *Int J Numer Meth Engng* **26** (1988), pp 293–311

70 Zeng Q. H., "Highly accurate modal method for calculating eigenvector derivatives in viscous damping systems". *AIAA J* **33** 4 (1995), pp 746–751

71 Cronin D. L., "Eigenvalue and eigenvector determination for non-classically damped dynamic systems". *Comp Struct* **36** 1 (1990), pp 133–138

72 Fresen J. L., Juritz J. M., "A note on Foss's method of obtaining initial estimates for exponential curve fitting by numerical integration", Biometrics, vol 42, no 4, pp 821-827, December 1986

73 Woodhouse J., "Linear damping models for structural vibration". *J Sound Vibration* **215** 3 (1998), pp 547–569

74 ISO Standard 3048, Part 4, "Ball screws — Part 4: Static axial rigidity", International Standards Organisation document, ISO TC 39/WG7 N 1408, 29/07/2003

75 "LM system ball screws", catalog. no.200-2BE, THK Co. Ltd., section 6.1, 1999, (© THK Ltd, 04 990425).

76 SKF, "General catalogue – 4000/IV E", SKF Group, pp 14-25, 1994, (Reg. 47 24000 1994-12).

77 "LM system ball screws", catalog. no.200-2BE, THK Co. Ltd., sections 5.2-5.4, 1999, (© THK Ltd, 04 990425).

78 Warren C. Young, "Roark's formulas for stress and strain", 6th edition, McGraw-Hill, 1989, Ch 13, Table 33, Item 4, P 652, (ISBN 0-07-072541-1).

79 Hertz, H., "Über die Berührung fester elastischer Körper", Journal für die reine und angewandte Mathematik, Vol. 92, pp. 156-171, 1882.

80 Warren C. Young, "Roark's formulas for stress and strain", 6th edition, McGraw-Hill, 1989, Ch 12, Table 32, pp 638-640, (ISBN 0-07-072541-1).

81 Brändlein J., Eschmann P., Hasbergen L., Weigand K., "Ball and roller bearings – Theory, design and application", John Wiley & Sons, 1999, Ch 4, Pp 214-224, (ISBN 0 471 98452 3).

82 Brändlein J., Eschmann P., Hasbergen L., Weigand K., "Ball and roller bearings – Theory, design and application", John Wiley & Sons, 1999, Ch 4, Table 4.1, P 218, (ISBN 0 471 98452 3).

83 Brändlein J., Eschmann P., Hasbergen L., Weigand K., "Ball and roller bearings – Theory, design and application", John Wiley & Sons, 1999, Ch 3, Table 3.2, P 151, (ISBN 0 471 98452 3).

84 Hooke R, "A description of Helioscopes and some other Instruments", (in anagram form), 1676; "De Potentia Bestitutiva" (the key), 1678

85 Newton I., "Philosophiae Naturalis Principia Mathematica", pp 12-13, 1687

86 De Silva C. W., Vibration – fundamentals and practice, CRC Press, ch 6, pp 299 - 302, 1999, (ISBN 0-8493-1808-4)

87 Coulson Charles A., "Waves", Oliver and Boyd, Ch IV, 1955

88 De Silva C. W., Vibration – fundamentals and practice, CRC Press, ch 6, pp 290-299, 1999, (ISBN 0-8493-1808-4)

89 Holroyd G., Pislaru C. and Ford D. G., Identification of damping elements in a CNC machine tool drive. In Proceedings of 5th International Conference on *Laser Technology, Machine Tool, CMM and Robot Performance* LAMDAMAP 2001 (Ed. WIT Press), Birmingham, UK, July 2001, pp. 289-299, (ISBN 1 85312 890 2)

90 Pislaru C, Ford D. G. and Holroyd G., "Hybrid modelling and simulation of a computer numerical control machine tool feed drive", Proc. Instn Mech. Engrs Vol 218 Part I: J. Systems and Control Engineering, pp 111-120, 2004

91 Tan X. and Rogers R. J., "Equivalent viscous damping models of Coulomb friction in multi-degree of freedom vibration systems", J. of Sound and Vibration, 185(1), pp 33-50

92 Harris C. M., Shock and vibration handbook, 3rd edition (Ed. McGraw-Hill, New York), ch 1, p 1-22, 1988, (ISBN 0-07-026801-0).

93 Quayle J. P., Kempe's engineers' year book, 1984, 89th edition (Ed. Morgan Grampian), pp **A2**/37-8, 1984.

94 James M. L., Smith G. M., Wolford J. C. and Whaley P. W. Vibration of mechanical and structural systems: with microcomputer applications (Ed. Harper & Row), 1989, (ISBN 0 060 43261 6)

95 Newland D. E., Mechanical vibration analysis and computation (Ed. Longman Scientific & Technical), 1989, (ISBN 0 582 02744 6)

96 Press W. H., Teukolsky S. A., Vetterling W. T. and Flannery B. P., Numerical recipes in C – The art of scientific computing (Ed. Cambridge University Press), Ch 11, pp 456-495, 1992, (ISBN 0 521 43108 5).

97 Gourlay A. R. and Watson G. A., Computational methods for matrix eigenproblems (Ed. John Wiley, Chichester), 1973, (ISBN 0 471 31915 5)

98 Wilkinson M. A., The algebraic eigenvalue problem (Ed. Clarendon), 1965, (ISBN 0 198 53403 5)

99 Harris C. M., Shock and vibration handbook, 3rd edition (Ed. McGraw-Hill, New York), ch 2, pp 2-6 – 2-25, 1988, (ISBN 0-07-026801-0).

100 Young W. C., "Roark's formulas for stress and strain", 6th edition, McGraw-Hill, 1989, ch 7, pp 93-112, (ISBN 0-07-072541-1)

101 Press W. H., Teukolsky S. A., Vetterling W. T. and Flannery B. P., Numerical recipes in C – The art of scientific computing (Ed. Cambridge University Press), Ch 11, pp 43-48, 1992, (ISBN 0 521 43108 5).

102 "The electromagnetic spectrum", http://csep10.phys.utk.edu/astr162/lect/light/spectrum.html

103 In 1879 Jožef Stefan (1835-1893) determined that the amount of radiation given off by a body through heating is proportional to the fourth power of its temperature. In 1884 Ludwig Boltzmann (1844-1906) succeeded in theoretically deriving the radiation law found by Stephan. See http://www-history.mcs.st-and.ac.uk/history/Mathematicians/Stefan_Josef.html, and http://www-history.mcs.st-and.ac.uk/history/Mathematicians/Boltzmann.html

104 Özişik M. Necati , "Heat transfer – A basic approach", 1st edition, McGraw_Hill, Appendix A, pp 733-735, 1985

105 Özişik M. Necati , "Heat transfer – A basic approach", 1st edition, McGraw_Hill, ch 1, pp 9-13, 1985

106 Özişik M. Necati , "Heat transfer – A basic approach", 1st edition, McGraw_Hill, Appendix C, pp 756-761, 1985

107 Özişik M. Necati , "Heat transfer – A basic approach", 1st edition, McGraw_Hill, ch 6, pp 226-269, 1985

108 Fletcher S., Ford D. G., "Measuring and modelling heat transfer and thermal errors on a ballscrew feed drive system". In Proceedings of 6th International Conference on *Laser Technology, Machine Tool, CMM and Robot Performance*

LAMDAMAP 2003 (Ed. WIT Press), Huddersfield, UK, July 2003, pp. 349-359, (ISBN 1-85312-990-9)

109 "Reduction of errors in the design and integration of CNC machine tools", EPSRC Grant: GR/L05624/01

110 "Reduction of errors in the design and integration of CNC machine tools", Disseminated technical report for collaborators

111 http://www.physlink.com/Education/AskExperts/ae280.cfm, consulted on 14/02/05.

112 SKF Group, "General catalogue", Catalogue 4000/IV E, Reg. 47 – 24 000 – 1994-12, p 145, 1994

113 Özişik M. Necati , "Heat transfer – A basic approach", 1st edition, McGraw_Hill, ch 7, p 291, 1985

114 Özişik M. Necati , "Heat transfer – A basic approach", 1st edition, McGraw_Hill, Appendix B, pp 736-755, 1985

115 Euclid, "Elements", book I, proposition 47, c300 B.C. (Apparently Pythagoras himself didn't write anything, see "The internet encyclopedia of philosophy", section on Pythagoras, fl. 530 B.C. http://www.utm.edu/research/iep/p/pythagor.htm)

116 Quayle J. P. ed, "Kempe's engineers' year book , 1984", 89th edition, Morgan Grampian, p **A3**/17, 1984

117 Young W. C., "Roark's formulas for stress and strain", 6th edition, McGraw-Hill, ch 7, table 3, cases 2d and 2e, pp 104-105, 1989

118 Quayle J. P. ed, "Kempe's engineers' year book , 1984", 89th edition, Morgan Grampian, Table 5, p **A1**/19, 1984

119 Quayle J. P. ed, "Kempe's engineers' year book , 1984", 89th edition, Morgan Grampian, p **A1**/5, 1984

120 Press H. W., Teukolosky S. A., Vetterling W. T., Flannery B. P., "Numerical recipes in C", 2nd edition, Cambridge University Press, ch 15, pp 681-687, 1992

121 Press H. W., Teukolosky S. A., Vetterling W. T., Flannery B. P., "Numerical recipes in C", 2nd edition, Cambridge University Press, ch 2, pp 36-43, 1992

122 http://www.kuroda-precision.co.jp/e-top/newproducts/bal/bal07w.htm, 30/05/06

123 Quayle J. P., Kempe's engineers' year book, 1984, 89th edition (Ed. Morgan Grampian), Tables 4-6, pp **A1**/17-20, 1984.

124 http://www.dynacer.com/PDF/dynalloxproperties.pdf, accessed 11/07/06

125 http://www.dynacer.com/PDF/technoxproperties.pdf, accessed 11/07/06

126 http://www.goodfellow.com/csp/active/gfMaterialInfo.csp?text=*T&MATID=W%2060&material=1, accessed 18/02/07

127 http://www.dynacer.com/PDF/nonoxideproperties.pdf, accessed 11/07/06

128 http://www.webelements.com/webelements/elements/text/Ir/key.html, accessed 11/07/06

129 Young W. C., "Roark's formulas for stress and strain", 6th edition, McGraw-Hill, ch 7, p 95, 1989

130 Felippa C. A., "A historical outline of matrix structural analysis: a play in three acts", *Computers & Structures* **79** (14): pp 1313 – 1324, 2001

131 http://www.algor.com/, accessed 01/12/07

132 http://www.ansys.com/, accessed 01/12/07

133 http://www.simulia.com/, accessed 01/12/07

134 http://www.lusas.com/, accessed 01/12/07

135 http://www.arup.com/DYNA/software/ls-dyna/ls-dyna.htm, accessed 01/12/07

136 http://www.mscsoftware.com/products/msc_nastran.cfm, accessed 01/12/07

137 http://www.nenastran.com/, accessed 01/12/07

138 http://www.samtech.fr/, accessed 01/12/07

139 http://www.quantech.es/QuantechATZ/Stampack.html, accessed 01/12/07

140 "LM system ball screws", catalog. no.200-2BE, THK Co. Ltd., chapter 2, pp 10-11, 1999, (© THK Ltd, 04 990425)

141 http://www/colorado.edu/engineering/CAS/courses.d/IFEM.d/IFEM.ch31.d/IFEM.Ch31.pdf, accessed on 07/11/07

142 Castaneda V. Y. M., "Parameter Identification and Transmission Line Modelling applied to Non-Linear Control Systems", Ph.D. Thesis, 2006, University of Huddersfield, Ch 4, pp 44-89 and Appendix E, pp 214-220

Appendix 4.1 – **Validation of continuous matter test model**

A method of predicting the dynamic behaviour of a mechanical system based on wave theory is set out in Section 4.1 of Chapter 4. This appendix validates the model by comparing a sample of the results from the model with those calculated from classical theory. Using the following data:-

Model	Simulink "wave_7e.mdl"	
Basic data:-		
Mass of lump masses	0.1	kg each
Length of spring	1	m
Cross sectional area of spring	1	m^2
Young's modulus of spring material	1	N/m^2
Wave velocity	5	m/sec
Damping coefficient of viscous damper, c	0.1	N sec/m
Derived data:-		
Spring constant, $k = EA/l$	1	N/m
Density of spring, c.f. equation (4.2)	0.04	kg/m^3
Mass of spring	0.04	kg
Forcing terms:-		
A force of 1 N for the first second of motion and zero thereafter		

the pattern of motion predicted by the model is:-

- The first mass moves farther than the second in the first 1.3 seconds or so.
- After the first 0.7 seconds, the two masses execute lightly damped relative motion with a frequency of 0.671 Hz and a logarithmic decrement of 1.405.
- The system settles down to a speed of 4.16 m/sec.
- The total energy increases for the first second while the external force is being applied, then remains constant for the remainder of the motion. The energy values settle out at 0.87 J for each mass and 0.35 J for the spring with 0.32 J being dissipated by the damper.

The results from the model are shown on Figures A4.1.1 to A4.1.6.

In a spring system of this nature it is common practice to add a third of the mass of the spring to the lumped mass when estimating the fundamental natural frequency [**87**]. Following this approach the damped natural frequency ω_d is predicted to be 0.654 Hz and the logarithmic decrement Δ 1.350 as follows.

The natural frequency of system comprising two masses with an interconnecting spring is calculated by first reducing the system to a single "mass on a spring" system of stiffness *k* and reduced mass μ given by:-

$$\mu = \frac{m_1 m_2}{m_1 + m_2} = \frac{0.1 \times 0.1}{0.1 + 0.1} = 0.05 \text{ kg}$$

The total equivalent mass including the spring is therefore:-

$$\mu_{tot} = 0.05 + \frac{0.04}{2 \times 3} = 0.056667 \text{ kg}$$

and the undamped natural frequency is then:-

$$\omega_n = \sqrt{\frac{k}{\mu_{\text{tot}}}} = \sqrt{\frac{1}{0.056667}} = 4.2008 \text{ rad/sec} = 0.66858 \text{ Hz}.$$

The fraction of critical damping is:-

$$\zeta = \frac{c}{2\sqrt{k\mu}} = \frac{0.1}{2\sqrt{1 \times 0.056667}} = 0.21004$$

The undamped natural frequency is then:-

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} = 0.66858 \times \sqrt{1 - 0.21004^2} = 0.65367 \text{ Hz}.$$

and the logarithmic decrement is:-

$$\Delta = \frac{2\pi\zeta}{\sqrt{1 - \zeta^2}} = \frac{2 \times \pi \times 0.21004}{\sqrt{1 - 0.21004^2}} = 1.3498.$$

If the system were a rigid body, the velocity at the end of the "forcing" stage of the motion would be:-

$$v = \frac{F}{\sum m} t = \frac{1}{0.1 + 0.04 + 0.1} \times 1 = 4.1667 \text{ m/sec}$$

and the kinetic energy:-

$$KE(Mass\ 1) = KE(Mass\ 1) = \frac{1}{2} \times 0.1 \times 4.1667^2 = 0.868 \text{ J}.$$

$$KE(spring) = \frac{1}{2} \times 0.04 \times 4.1667^2 = 0.347 \text{ J}.$$

The value for the energy in the spring checks out against the kinetic energy of the spring calculated on a "½ mass × velocity^2" basis. When the system settles down there is a continual small interchange of energy between the masses and the spring at a frequency of about 2.7 Hz. The reason for this is not clear.

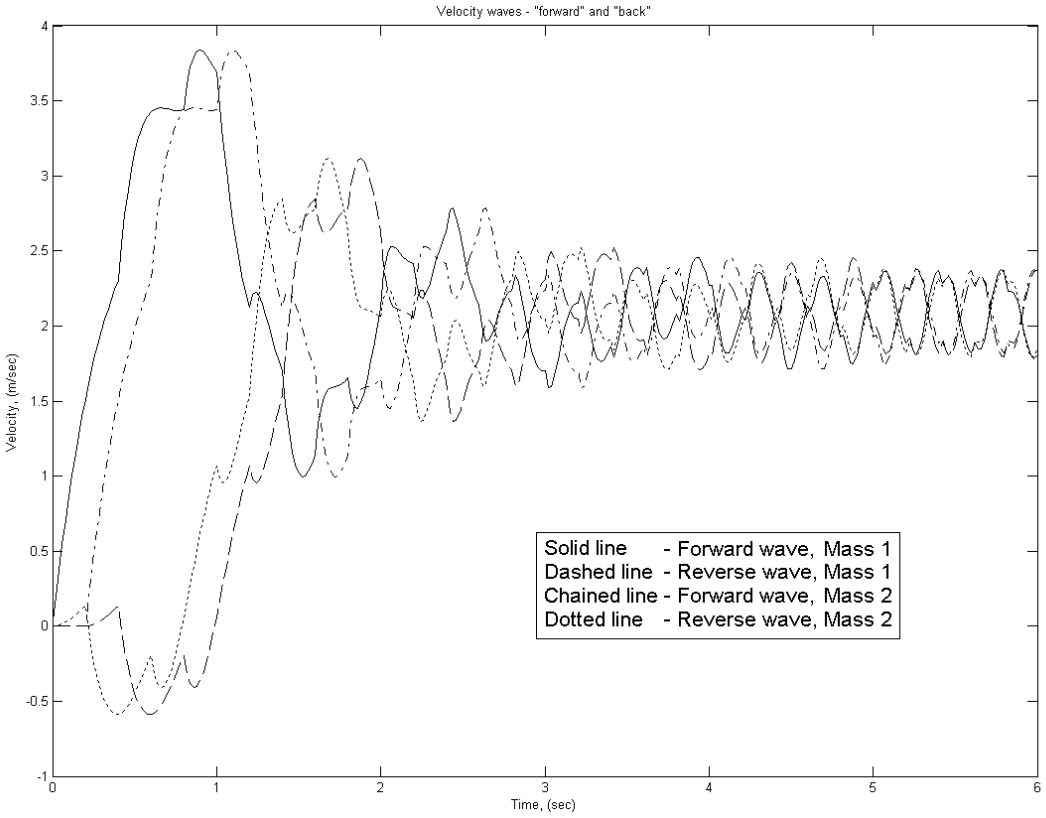


Figure A4.1.1 - *Wave model – wave amplitude (m/sec) v. time (sec)*

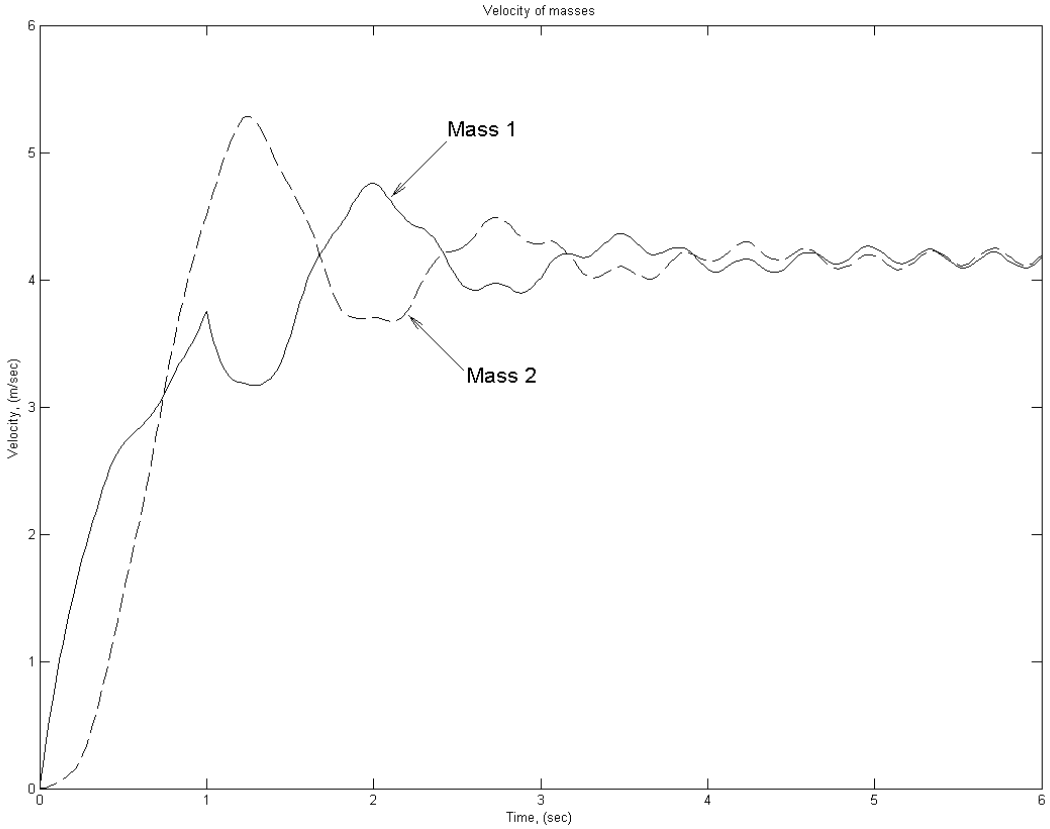


Figure A4.1.2 - *Wave model – mass velocities (m/sec) v. time (sec)*

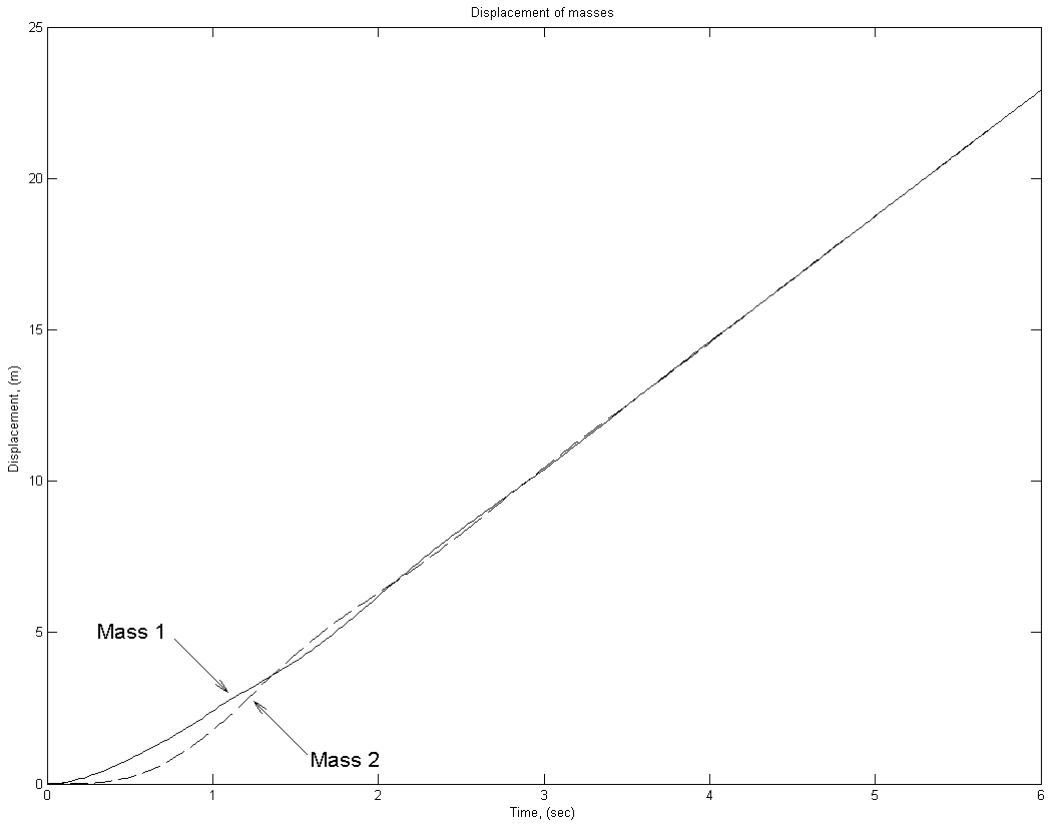


Figure A4.1.3 - *Wave model – mass displacements (m) v. time (sec)*

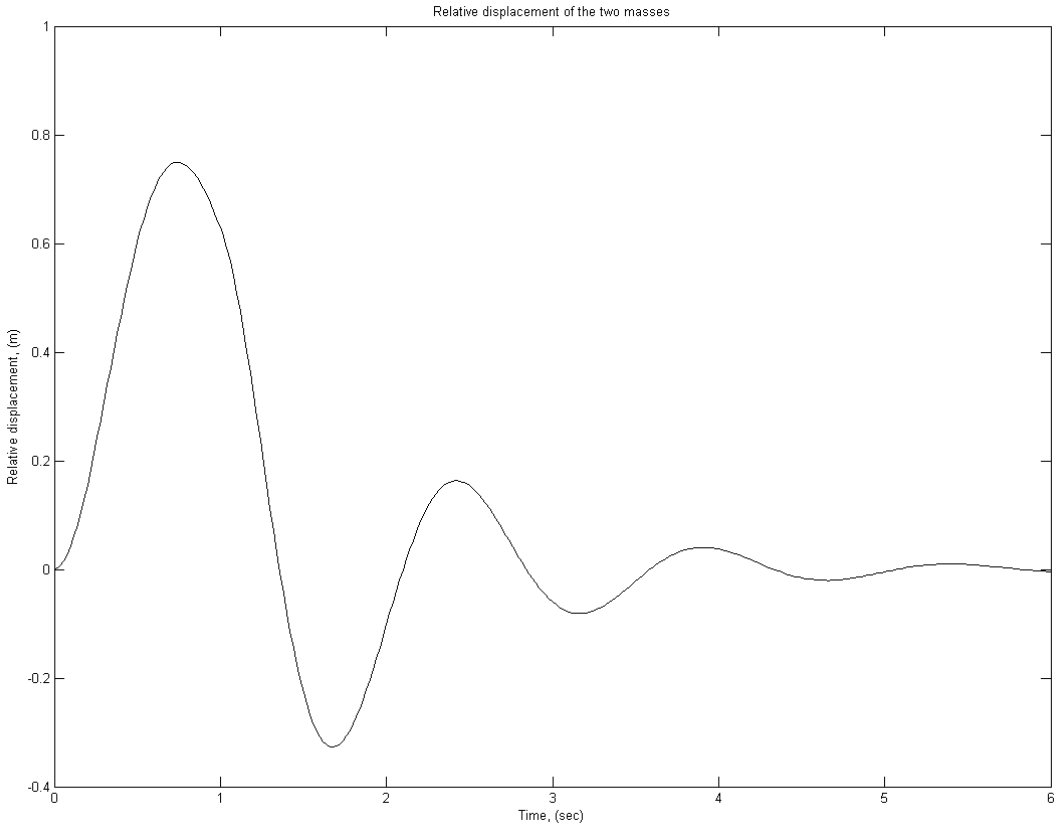


Figure A4.1.4 - *Wave model – relative displacement of masses (m) v. time (sec)*

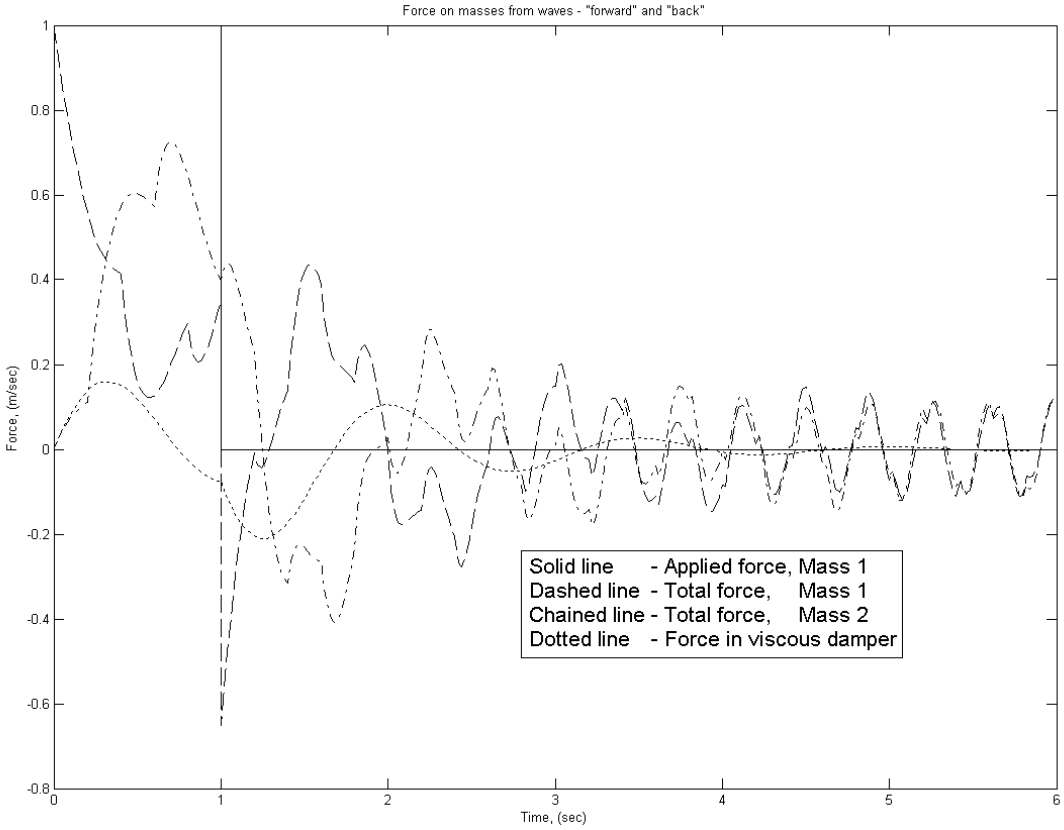


Figure A4.1.5 - *Wave model – forces (N) v. time (sec)*

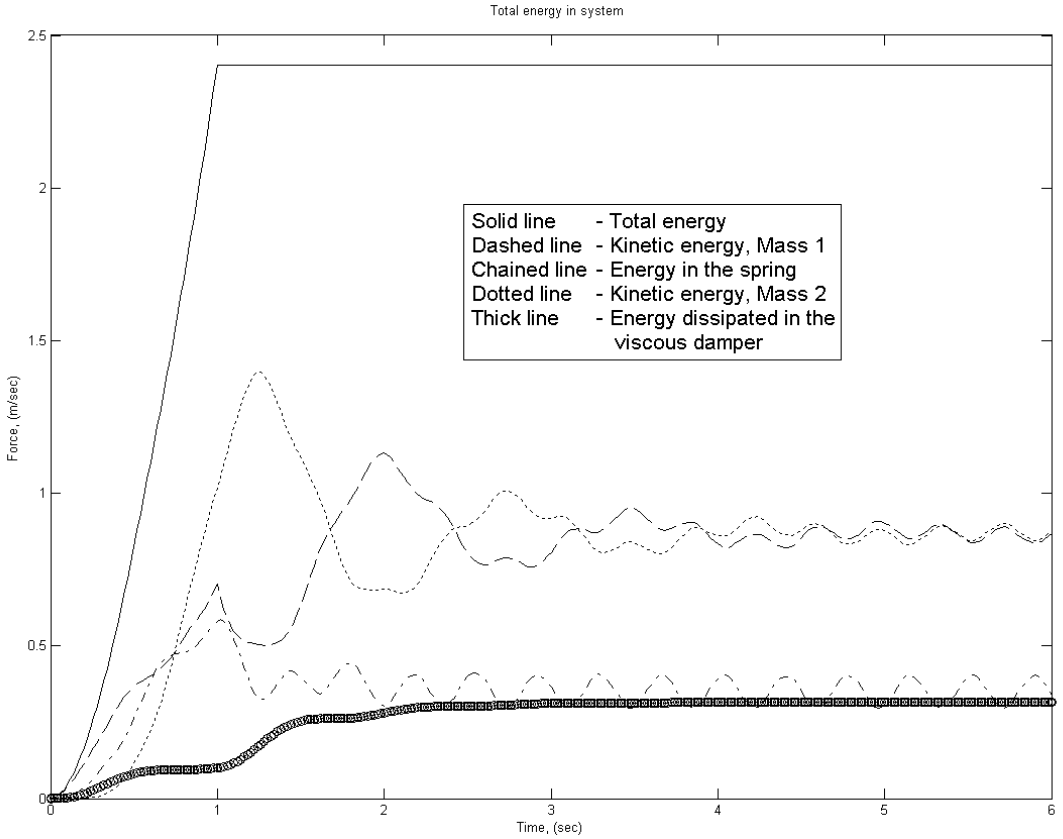


Figure A4.1.6 - *Wave model – energies (J) v. time (sec)*

Appendix 4.2 - **Details of Y-axis drive for Beaver VC 35 CNC machine tool**

J_{bs}	rotational inertia of the ball-screw		=0.002724 kg m^2
J_m	rotational inertia of the drive motor		= 0.0048 kg m^2
J_{ms}	rotational inertia of the shaft inertia		=0.0000075 kg m^2
J_{p1}	rotational inertia of the driving pulley		=0.0004986 kg m^2
J_{p2}	rotational inertia of the driven pulley		= 0.009524 kg m^2
k_{ax}	axial stiffness of the ball-screw and its support bearings		= 482.7e6 N/m
k_{bl}	torsional stiffness of the belt drive		= 1167 Nm/rad
k_{bs}	torsional stiffness of the ball-screw		= 9735 Nm/rad
k_{ms}	torsional stiffness of the motor shaft		= 5466 Nm/rad
k_{nut}	axial stiffness of the ball-screw nut		= 5.e8 N/m
m_{bm}	a third of the mass of the ball-screw		= 5 kg
m_{tab}	mass of the table and saddle		= 677 kg
R	ball-screw ratio	for 10 mm/turn	= 0.0015915 m/rad
u	drive belt ratio		=2

Appendix 4.3 – **MATLAB function for calculating undamped natural frequencies**

```
function[natf,V]=natf_dr2(drive,N,iL,ipl)

% This routine computes the undamped natural frequencies and mode shapes
% for a motor driven ballscrew system.
%
% The degrees of freedom considered are:-
%  1        motor           torsional
%  2        driving pulley  torsional
%  3        driven pulley   torsional
%  4 – N+3  ballscrew       torsional
%  N+4      ballscrew       axial
%  N+5      table / saddle  axial
%
% Output variables
%     natf()    natural frequency array
%     V(,)      eigenvector matrix
%
% Input variables
%     drive     select 'x' or 'y' drive
%     N         number of elements in ballscrew
%     iL        node number of load position
%     ipl       node to plot
%
% Other variables
%     D(,)      square of natural frequency matrix
%     i         for loop counter
%     ii        for loop counter
%     j         for loop counter
%     J()       inertia array
%     Jbs       moment of inertia ballscrew
%     Jm        moment of inertia motor
%     Jms       moment of inertia motor shaft
%     Jp1       moment of inertia driving pulley
%     Jp2       moment of inertia driven pulley
%     K(,)      stiffness matrix
%     k_()      stiffness array
%     k_ax      stiffness ballscrew axial
%     k_bl      stiffness drive belt
%     k_bs      stiffness ballscrew
%     k_ms      stiffness motor shaft
%     k_nut     stiffness ball nut
%     m_bm      mass of ballscrew
%     m_tab     table ('x' drive), saddle and table ('y' drive)
%     natf_     "hold" value for natural frequency
%     pi        pi
%     R         ballscrew ratio
%     rsum2     root mean square of mode shape amplitudes
%     sum2      sum of squares of mode shape amplitudes
%     u         belt drive ratio
%     V_()      "hold" array for mode shape amplitudes
%     x()       "x" array for plots
%     y()       "y" array for plot data
%
% Subroutines used
% MATLAB - atan eig int2str num2str plot sqrt title xlabel ylabel
```

```
% UoH    -

pi=4.*atan(1.);

Jm=0.0048;% kg m^2            Data
Jms=0.0000075;% kg m^2
Jp1=0.0004986;% kg m^2
Jbs=0.002724;% kg m^2
m_bm=5.;% kg
k_ms=5466.;% N m/rad
k_bs=9735.;% N m/rad
k_nut=5.e8;% N/m
k_ax=482.7e6;% N/m
u=2.;
R=0.0015915;% m/rad
if (drive=='x'|drive=='X')
   Jp2=0.017779;% kg m^2
   m_tab=359.;% kg
   k_bl=1188.;% N m/rad
else
   Jp2=0.009524;% kg m^2
   m_tab=677.;% kg
   k_bl=1167.;% N m/rad
end

J(1)=Jm + Jms/2.;             %Assemble inertia array
J(2)=Jp1 + Jms/2.;
J(3)=Jp2 + Jbs/(N*2.);
for i=4:N+3
   J(i)=Jbs/N;
end
J(N+3)=J(N+3)/2.;
J(N+4)=m_bm;
J(N+5)=m_tab;

k_(1)=k_ms;                   %Assemble stiffness array
k_(2)=k_bl;
for i=3:N+2
   k_(i)=(N*k_bs);
end
k_(N+3)=k_nut;
k_(N+4)=k_ax;

for i=1:N+5                   %Zeroise upper half of  matrix
   for j=i:N+5
      K(i,j)=0.;
   end
end

K(1,1)=k_(1);                 %Assemble upper half of K matrix
K(1,2)=(-1.)*k_(1);
K(2,2)=k_(1)+k_(2);
K(2,3)=(-1.)*u*k_(2);
K(3,3)=u^2*k_(2)+k_(3);
K(3,4)=(-1.)*k_(3);
for i=4:N+2
   K(i,i)  =k_(i-1)+k_(i);
```

```
   K(i,i+1)=(-1.)*k_(i);
end
K(iL,iL)  =K(iL,iL)+R^2*k_(N+3);
K(iL,N+4) =R*k_(N+3);
K(iL,N+5) =(-1.)*R*k_(N+3);
K(N+3,N+3)=k_(N+2);
K(N+4,N+4)=k_(N+3)+k_(N+4);
K(N+4,N+5)=(-1.)*k_(N+3);
K(N+5,N+5)=k_(N+3);

for i=1:N+4                  %Fill out K matrix
   for j=i+1:N+5
      K(j,i)=K(i,j);
   end
end

for i=1:N+5                  %Convert K matrix to eigenvalue form
   for j=1:N+5
      K(i,j)=K(i,j)/sqrt(J(i));
   end
end
for j=1:N+5
   for i=1:N+5
      K(i,j)=K(i,j)/sqrt(J(j));
   end
end

[V,D]=eig(K);                %Solve eigenvalue equation

for i=1:N+5                  %Prepare mode shape vectors
   x(i)=i;
   natf(i)=sqrt(D(i,i))/(2.*pi);
   for j=1:N+5
      V(j,i)=V(j,i)/sqrt(J(j));
   end
end

for i=1:N+5                  %Normalise mode shape vectors
   sum2=0.;
   for j=1:N+5
      sum2=sum2 + V(j,i)^2;
   end
   rsum2=sqrt(sum2);
   for j=1:N+5
      V(j,i)=V(j,i)/rsum2;
   end
end

for ii=1:N+4                 %Order mode shape vectors by frequency
   for i=2:N+6-ii
      if (natf(i-1)>natf(i))
         natf_=natf(i-1);
         natf(i-1)=natf(i);
         natf(i)=natf_;
         for j=1:N+5
            V_(j)=V(j,i-1);
            V(j,i-1)=V(j,i);
```

```
            V(j,i)=V_(j);
         end
      end
   end
end

for i=1:N+5                  %Select mode shape
   y(i)=V(i,ipl);
end

y(1)=y(1)/u;                 %Set vector to ballscrew equivalent
y(2)=y(2)/u;
y(N+4)=y(N+4)/R;
y(N+5)=y(N+5)/R;

plot(x,y,'k+-.')             %Plot mode shape
title([drive,' drive, mode shape ',int2str(ipl),', f = ',num2str(natf(ipl)),'
Hz'])
xlabel('Motor to ballscrew')
ylabel('Mode shape ')
```

Appendix 4.4 – **MATLAB function for calculating transverse natural frequencies**

```
function[natf,x,yp,Op]=natf_tr1(N,ip)

% This routine computes the natural frequencies and mode shapes
% for the free vibration of a round bar and plots the first four
% non-zero frequency modes.

% Output variables
%     x()       "x" array for plots
%     yp()      "y" array for pth mode shape
%     Op()      "dy/dx" array for pth mode shape
%
% Input variables
%     N         number of elements in ballscrew
%     ip        mode number of output mode
%
% Other variables
%     D(,)      square of natural frequency matrix
%     d_bs      diameter of ballscrew
%     E         modulus of elasticity
%     i         for loop counter
%     ii        for loop counter
%     I_bs      second moment of area of shaft
%     j         for loop counter
%     Jt()      tilt inertia array
%     K(,)      stiffness matrix
%     l_bs      length of ballscrew
%     M()       mass array
%     MJ()      square root of inertia array
%     natf()    natural frequency array
%     natf_     "hold" value for natural frequency
%     O1()      "dy/dx" array for 1st mode shape
%     O2()      "dy/dx" array for 2nd mode shape
%     O3()      "dy/dx" array for 3rd mode shape
%     O4()      "dy/dx" array for 4th mode shape
%     pi        pi
%     rho       density
%     rsum2     root mean square of mode shape amplitudes
%     sum2      sum of squares of mode shape amplitudes
%     V(,)      mode shape matrix
%     V_()      "hold" array for mode shape amplitudes
%     y1()      "y" array for 1st mode shape
%     y2()      "y" array for 2nd mode shape
%     y3()      "y" array for 3rd mode shape
%     y4()      "y" array for 4th mode shape
%
% Subroutines used
% MATLAB - atan eig figure hold plot sqrt subplot xlabel ylabel
% UoH    -

pi=4.*atan(1.);

E=207.e9;        % Material properties
rho=7600.;
```

```
d_bs=38.1e-3;    % Dimensions of bar
l_bs=1.675;

                 % Mass and tilt inertia of the nodes
M(1)=(pi/4.*d_bs^2*l_bs*rho/(2.*N));   % first node
Jt(1)=M(1)*(d_bs^2/16.+(l_bs/(2.*N))^2/3.);
for i=2:N
   M(i)=(pi/4.*d_bs^2*l_bs*rho/N);     % central N-1 nodes
   Jt(i)=M(i)*(d_bs^2/16.+(l_bs/N)^2/12.);
end
M(N+1)=(pi/4.*d_bs^2*l_bs*rho/(2.*N)); % last node
Jt(N+1)=M(N+1)*(d_bs^2/16.+(l_bs/(2.*N))^2/3.);

I_bs=pi/64.*d_bs^4; % second moment of area of bar

                 % Stiffness matrix
for i=1:2*N
   for j=i:2*N
      K(i,j)=0.; % zeroise matrix
   end
end
K(1,1)= 12.*E*I_bs/(l_bs/N)^3; % first element
K(1,2)=  6.*E*I_bs/(l_bs/N)^2; % upper half
K(1,3)=-12.*E*I_bs/(l_bs/N)^3;
K(1,4)=  6.*E*I_bs/(l_bs/N)^2;
K(2,2)=  4.*E*I_bs/(l_bs/N);
K(2,3)= -6.*E*I_bs/(l_bs/N)^2;
K(2,4)=  2.*E*I_bs/(l_bs/N);
for i=2:N
   K(2*i-1,2*i-1)= 12.*E*2.*I_bs/(l_bs/N)^3; % central N-2 elements
   K(2*i-1,2*i+1)=-12.*E*I_bs/(l_bs/N)^3;    % upper half
   K(2*i-1,2*i+2)=  6.*E*I_bs/(l_bs/N)^2;
   K(2*i  ,2*i  )=  4.*E*2.*I_bs/(l_bs/N);
   K(2*i  ,2*i+1)= -6.*E*I_bs/(l_bs/N)^2;
   K(2*i  ,2*i+2)=  2.*E*I_bs/(l_bs/N);
end
K(2*N+1,2*N+1)= 12.*E*I_bs/(l_bs/N)^3; % last element
K(2*N+1,2*N+2)= -6.*E*I_bs/(l_bs/N)^2; % upper half
K(2*N+2,2*N+2)=  4.*E*I_bs/(l_bs/N);
for i=1:2*N+2
   for j=i+1:2*N+2
      K(j,i)=K(i,j); % fill lower half of matrix
   end
end

for i=1:N+1
   MJ(2*i-1)=sqrt(M(i));  % combined mass and inertia array
   MJ(2*i  )=sqrt(Jt(i)); % (square root)
end

for i=1:2*N+2
   for j=1:2*N+2
      K(i,j)=K(i,j)/MJ(j); % divide rows of K by MJ
   end
end
for i=1:2*N+2
   for j=1:2*N+2
```

```
      K(i,j)=K(i,j)/MJ(i); % divide columns of K by MJ
   end
end

[V,D]=eig(K); % solve eigenvalue equation to generate
              % mode shape and square of natural
              % frequency matrices

for i=1:2*N+2
   natf(i)=sqrt(D(i,i))/(2.*pi); % natural frequency array
   for j=1:N+1
      V(2*j-1,i)=V(2*j-1,i)/sqrt(M(j));  % mode shapes corrected
      V(2*j  ,i)=V(2*j  ,i)/sqrt(Jt(j)); % by square root of mass
   end
end

for i=1:2*N+2    % Normalise mode shapes
   sum2=0.;
   for j=1:2*N+2
      sum2=sum2 + V(j,i)^2;
   end
   rsum2=sqrt(sum2);
   for j=1:2*N+2
      V(j,i)=V(j,i)/rsum2;
   end
end

                 % Order natural frequencies and mode shapes in
for ii=1:2*N+1   % ascending order of natural frequency
   for i=2:2*N+3-ii
      if (natf(i-1)>natf(i))
         natf_=natf(i-1);
         natf(i-1)=natf(i);
         natf(i)=natf_;
         for j=1:2*N+2
            V_(j)=V(j,i-1);
            V(j,i-1)=V(j,i);
            V(j,i)=V_(j);
         end
      end
   end
end

for i=1:N+1      % Define first four mode shapes for plotting
   x(i)=i;
   y1(i)=V(2*i-1,3); % amplitude
   O1(i)=V(2*i  ,3); % slope
   y2(i)=V(2*i-1,4);
   O2(i)=V(2*i  ,4);
   y3(i)=V(2*i-1,5);
   O3(i)=V(2*i  ,5);
   y4(i)=V(2*i-1,6);
   O4(i)=V(2*i  ,6);
   yp(i)=V(2*i-1,ip); % mode shape for output
   Op(i)=V(2*i  ,ip);
end
```

```
                % Plot mode shapes
figure          % amplitude
subplot(2,1,1),plot(x,y1,x,y2,x,y3,x,y4)
xlabel('Motor to ballscrew')
ylabel('y amplitude')
title('Mode shapes')

hold on         % slope
subplot(2,1,2),plot(x,O1,x,O2,x,O3,x,O4)
xlabel('Motor to ballscrew')
ylabel('y slope')
```

Appendix 4.5 – **MATLAB function for calculating damped natural frequencies**

```
function[natf,damp]=natf_dr3(drive,N,iL,ipl)

% This routine computes the damped natural frequencies and mode shapes
% for a motor driven ballscrew system.
%
% The degrees of freedom considered are:-
%  1        motor           torsional
%  2        driving pulley  torsional
%  3        driven pulley   torsional
%  4 – N+3  ballscrew       torsional
%  N+4      ballscrew       axial
%  N+5      table / saddle  axial
%
% Output variables
%     natf()    natural frequency array
%     damp()    damping data array
%
% Input variables
%     drive     select 'x' or 'y' drive
%     N         number of elements in ballscrew
%     iL        node number of load position
%     ipl       node to plot
%
% Other variables
%     A0(,)     A0 sub-matrix
%     A1(,)     A1 sub-matrix
%     A2(,)     A2 sub-matrix
%     B1(,)     B1 sub-matrix
%     B2(,)     B2 sub matrix
%     c_()      damping coefficient array
%     c_ax      damping coefficient ballscrew axial
%     c_bl      damping coefficient drive belt
%     c_brg_bs  damping coefficient ballscrew bearing
%     c_brg_ms  damping coefficient motor bearing
%     c_bs      damping coefficient ballscrew
%     c_ms      damping coefficient motor shaft
%     c_nut     damping coefficient ball nut
%     D(,)      natural frequency matrix
%     D_=       "hold" value for square of natural frequency
%     damp_=    "hold" value for damping data
%     damping   description of type of damping
%     i         for loop counter
%     ipl_      node number
%     I(,)      unit matrix
%     ii        for loop counter
%     j         for loop counter
%     J()       inertia array
%     Jbs       moment of inertia ballscrew
%     Jm        moment of inertia motor
%     Jms       moment of inertia motor shaft
%     Jp1       moment of inertia driving pulley
%     Jp2       moment of inertia driven pulley
%     K(,)      stiffness and damping combined matrix
%     k_()      stiffness array
%     k_ax      stiffness ballscrew axial
%     k_bl      stiffness drive belt
%     k_bs      stiffness ballscrew
%     k_ms      stiffness motor shaft
%     k_nut     stiffness ball nut
```

```
%     m_bm      mass of ballscrew
%     m_tab     table ('x' drive), saddle and table ('y' drive)
%     natf_     "hold" value for natural frequency
%     natf_max  maximum natural frequency
%     O(,)      zero matrix
%     pi        pi
%     R         ballscrew ratio
%     rsum2     root mean square of mode shape amplitudes
%     sum2      sum of squares of mode shape amplitudes
%     u         belt drive ratio
%     V(,)      mode shape matrix
%     V_()      "hold" array for mode shape amplitudes
%     x()       "x" array for plots
%     y()       complex "y" array for plot data
%     y1()      "y" array for "in-phase" plots
%     y2()      "y" array for "quadrature" plots
%     y3()      "y" array for "amplitude" plots
%     y4()      "y" array for "phase" plots
%     y_save    "hold" value for member of "y" array
%     y_norm    renormalising value for "y" array
%     zeta      damping ratio
%
% Subroutines used
% MATLAB - abs angle atan diag eig eye (figure) (hold) imag int2str inv
%          num2str plot real sqrt subplot title xlabel ylabel zeros
% UoH    -

pi=4.*atan(1.);

                              % Data common to both drives
Jm=0.0048;%kg m^2
Jms=0.0000075;%kg m^2
Jp1=0.0004986;%kg m^2
Jbs=0.002724;%kg m^2
m_bm=5.;%kg
k_ms=5466.;%N m/rad
k_bs=9735.;%N m/rad
k_nut=5.e8;%N/m
k_ax=482.7e6;%N/m
c_ms=0.001;%N m sec/rad
c_bs=0.0295;%N m sec/rad
c_nut=58060.;%N sec/m
c_ax=8326.;%N sec/m
c_brg_ms=0.668e-3;%N m sec/rad
c_brg_bs=0.668e-3;%N m sec/rad
u=2.;
R=0.0015915;%m/rad
if (drive=='x'|drive=='X')
   Jp2=0.017779;%kg m^2       Data particular to X drive
   m_tab=359.;%kg
   k_bl=1188.;%N m/rad
   c_bl=0.797;%N m sec/rad
else
   Jp2=0.009524;%kg m^2       Data particular to Y drive
   m_tab=677;%kg
   k_bl=1167.;%N m/rad
   c_bl=0.452;%N m sec/rad
end

J(1)=Jm + Jms/2.;             % Assemble inertia array
J(2)=Jp1 + Jms/2.;
```

```
J(3)=Jp2 + Jbs/(N*2.);
for i=4:N+3
   J(i)=Jbs/N;
end
J(N+3)=J(N+3)/2.;
J(N+4)=m_bm;
J(N+5)=m_tab;

k_(1)=k_ms;                % Assemble stiffness array
k_(2)=k_bl;
for i=3:N+2
   k_(i)=N*k_bs;
end
k_(N+3)=k_nut;
k_(N+4)=k_ax;

c_(1)=c_ms;                % Assemble damping array
c_(2)=c_bl;
for i=3:N+2
   c_(i)=N*c_bs;
end
c_(N+3)=c_brg_ms;
c_(N+4)=c_brg_bs;
c_(N+5)=c_nut;
c_(N+6)=c_ax;

O=zeros(N+5);              % Define zero matrix

I=eye(N+5);                % Define unity matrix

A0=diag(J);                % Define A0 matrix

A1=O;                      % Define A1 matrix
A1(1,1)=c_(1)+c_(N+3);     % Assemble upper half of A1 matrix
A1(1,2)=(-1.)*c_(1);
A1(2,2)=c_(1)+c_(2);
A1(2,3)=(-1.)*u*c_(2);
A1(3,3)=u^2*c_(2)+c_(3)+c_(N+4);
A1(3,4)=(-1.)*c_(3);
for i=4:N+2
   A1(i,i)  =c_(i-1)+c_(i);
   A1(i,i+1)=(-1.)*c_(i);
end
A1(iL,iL)  =A1(iL,iL)+R^2*c_(N+5);
A1(iL,N+4) =R*c_(N+5);
A1(iL,N+5) =(-1.)*R*c_(N+5);
A1(N+3,N+3)=c_(N+2)+c_(N+4);
A1(N+4,N+4)=c_(N+5)+c_(N+6);
A1(N+4,N+5)=(-1.)*c_(N+5);
A1(N+5,N+5)=c_(N+5);
for i=1:N+4                % Fill out A1 matrix
   for j=i+1:N+5
      A1(j,i)=A1(i,j);
   end
end

A2=O;                      % Define A2 matrix
A2(1,1)=k_(1);             % Assemble upper half of A2 matrix
A2(1,2)=(-1.)*k_(1);
A2(2,2)=k_(1)+k_(2);
A2(2,3)=(-1.)*u*k_(2);
```

```
A2(3,3)=u^2*k_(2)+k_(3);
A2(3,4)=(-1.)*k_(3);
for i=4:N+2
   A2(i,i)  =k_(i-1)+k_(i);
   A2(i,i+1)=(-1.)*k_(i);
end
A2(iL,iL)  =A2(iL,iL)+R^2*k_(N+3);
A2(iL,N+4) =R*k_(N+3);
A2(iL,N+5) =(-1.)*R*k_(N+3);
A2(N+3,N+3)=k_(N+2);
A2(N+4,N+4)=k_(N+3)+k_(N+4);
A2(N+4,N+5)=(-1.)*k_(N+3);
A2(N+5,N+5)=k_(N+3);
for i=1:N+4                 % Fill out A2 matrix
   for j=i+1:N+5
      A2(j,i)=A2(i,j);
   end
end

B1=-inv(A0)*A1;             % Sets up K matrix for eigenvalue equation
B2=-inv(A0)*A2;
K=[O, I; B2, B1];

[V,D]=eig(K);               % Solve eigenvalue equation

x=1:N+5;                    % Prepare natural frequency arrays
natf_max=0.;
for i=1:2*N+10
   natf(i)=imag(D(i,i))/(2.*pi);
   damp(i)=real(D(i,i));
   if natf_max<natf(i)
      natf_max=natf(i);
   end
end

for i=1:2*N+10              % Normalise mode shape vectors
   sum2=0.;
   for j=1:N+5
      sum2=sum2 + V(j,i)*V(j,i)';
   end
   rsum2=sqrt(sum2);
   for j=1:N+5
      V(j+N+5,i)=V(j+N+5,i)/D(i,i);
   end
   for j=1:2*N+10
      V(j,i)=V(j,i)/rsum2;
   end
end

for ii=1:2*N+9              % Order mode shape vectors by frequency
   for i=2:2*N+11-ii
      if (natf(i-1)>natf(i))
         natf_=natf(i-1);
         natf(i-1)=natf(i);
         natf(i)=natf_;
         damp_=damp(i-1);
         damp(i-1)=damp(i);
         damp(i)=damp_;
         for j=1:2*N+10
            V_(j)=V(j,i-1);
            V(j,i-1)=V(j,i);
```

```
            V(j,i)=V_(j);
         end
         D_=D(i-1,i-1);
         D(i-1,i-1)=D(i,i);
         D(i,i)=D_;
      end
   end
end

if (ipl<0)                  % Select mode shape
   ipl_=N+6+ipl;
elseif (0<ipl)
   ipl_=N+5+ipl;
else
   disp('Invalid mode number - must be non-zero')
   return
end
for i=1:N+5
   y(i)=V(i,ipl_);
end

y_save=abs(y(1));           % Renormalise mode shape
y_norm=y(1)/abs(y(1));
for i=1:N+5
   if y_save<abs(y(i))
      y_save=abs(y(i));
      y_norm=y(i)/abs(y(i));
   end
end
for i=1:N+5
   y(i)=y(i)/y_norm;
end

y(1)=y(1)/u;                % Set vector to ballscrew equivalent
y(2)=y(2)/u;
y(N+4)=y(N+4)/R;
y(N+5)=y(N+5)/R;

for i=1:N+5                 % Plot mode shape
   y1(i)=real(y(i));
   y2(i)=imag(y(i));
   y3(i)=abs(y(i));
   y4(i)=angle(y(i))*180./pi;
end
if abs(natf(ipl_))<natf_max*1.e-5
   damping='critical or over-damping';
else
   zeta=sqrt((damp(ipl_))^2/((damp(ipl_))^2+(imag(D(ipl_,ipl_)))^2));
   damping=['zeta = ',num2str(zeta)];
end
subplot(4,1,1)
plot(x,y1,'k-.')
title([drive,' drive, mode shape ',int2str(ipl),', f =
',num2str(natf(ipl_)),' Hz, ',damping])
ylabel('In phase')
subplot(4,1,2)
plot(x,y2,'k-.')
ylabel('Quadrature')
subplot(4,1,3)
plot(x,y3,'k-.')
ylabel('Amplitude')
```

```
subplot(4,1,4)
plot(x,y4,'k-.')
ylabel('Phase (deg)')
xlabel('Motor to ballscrew')
```

Appendix 4.6 – **MATLAB routine for calculating sensitivity of a ballscrew system to changes in stiffness**

Function “natf_dr3_k” calculates the damped natural frequencies. It is based of function “natf_dr3” (see Appendix 4.5). The input argument list includes the stiffness parameters which may be changed and the argument list includes all the natural frequencies and damping terms. The plotting section has been omitted.

```
function[dnf_dk,dzt_dk]=natf_dr4(drive,N,iL)

% This routine computes the sensitivity of damped natural frequencies to
% changes in stiffness of various parts of a motor driven ballscrew system.

% The degrees of freedom considered for the natural frequencies are:-
%  1        motor           torsional
%  2        driving pulley  torsional
%  3        driven pulley   torsional
%  4 – N+3  ballscrew       torsional
%  N+4      ballscrew       axial
%  N+5      table / saddle  axial
%
% The component stiffnesses considered are:-
%           motor shaft          torsional
%           drive belt           torsional equivalent
%           ballscrew            torsional
%           ballscrew + support  axial
%           ballscrew nut        axial
%
% Output variables
%     dnf_dk(,) sensitivity of natural frequency to changes in stiffness
%     dzt_dk(,) sensitivity of damping ratio to changes in stiffness
%
% Input variables
%     drive     select 'x' or 'y' drive
%     N         number of elements in ballscrew
%     iL        node number of load position
%
% Other variables
%     damp()    damping ??
%     dm_ddm()  damping ?? with ith stiffness parameter incremented by 1%
%     i         for loop counter
%     j         for loop counter
%     k         damping ?? for ith mode
%     k_ax      stiffness ballscrew axial
%     K_ax      k_ax incremented by 1%
%     k_bl      stiffness drive belt
%     K_bl      k_bl incremented by 1%
%     k_bs      stiffness ballscrew
%     K_bs      k_bs incremented by 1%
%     k_ms      stiffness motor shaft
%     K_ms      k_ms incremented by 1%
%     k_nut     stiffness ball nut
%     K_nut     k_nut incremented by 1%
%     natf()    natural frequency array
%     nf_dnf    natural frequency with ith stiffness parameter
```

```
%                incremented by 1%
%     pi         pi
%     w          natural frequency of ith mode (rad/sec)
%     zeta()     damping ratio array
%     zet_dz()   damping ratio array with ith stiffness parameter
%                incremented by 1%

% Subroutines used
% MATLAB - atan sqrt
% UoH    - natf_dr3_k(*)
%
pi=4.*atan(1.);

k_ms=5466.;%N m/rad         % Data common to both drives
k_bs=9735.;%N m/rad
k_nut=5.e8;%N/m
k_ax=482.7e6;%N/m
if (drive=='x'|drive=='X')
   k_bl=1188.;%N m/rad        Data particular to X drive
else
   k_bl=1167.;%N m/rad        Data particular to Y drive
end

% Determine damped natural frequencies – original stiffnesses
[natf,damp]=natf_dr3_k(drive,k_ms,k_bl,k_bs,k_nut,k_ax,N,iL);

for i=1:N+5
   k          = damp(i+N+5);
   w          = 2.*pi*natf(i+N+5);
   zeta(i)    =sqrt(k^2/(k^2+w^2));
end

for i=1:5                   % Assemble increased stiffness array
   K_ms=k_ms;
   K_bl=k_bl;
   K_bs=k_bs;
   K_nut=k_nut;
   K_ax=k_ax;
   if i==1 K_ms=k_ms*1.01;end
   if i==2 K_bl=k_bl*1.01;end
   if i==3 K_bs=k_bs*1.01;end
   if i==4 K_nut=k_nut*1.01;end
   if i==5 K_ax=k_ax*1.01;end

% Determine damped natural frequencies % – ith stiffness increased by 1%
   [nf_dnf,dm_ddm]=natf_dr3_k(drive,K_ms,K_bl,K_bs,K_nut,K_ax,N,iL);

   for j=1:N+5
      dnf_dk(i,j)=(nf_dnf(j+N+5)/natf(j+N+5)-1.)*100.;
                            % .... natural frequency partial derivative
      k          = dm_ddm(j+N+5);
      w          = 2.*pi*nf_dnf(j+N+5);
      zet_dz(j)  =sqrt(k^2/(k^2+w^2));
      dzt_dk(i,j)=(zet_dz(j)/zeta(j)-1.)*100.;
                            % .... damping ratio partial derivative
   end
end
```

Appendix 4.7 – **MATLAB routine for calculating sensitivity of a ballscrew system to changes in damping**

Function “natf_dr3_c” calculates the damped natural frequencies. It is based of function “natf_dr3” (see Appendix 4.5). The input argument list includes the damping parameters which may be changed and the argument list includes all the natural frequencies and damping terms. The plotting section has been omitted.

```
function[dnf_dc,dzt_dc]=natf_dr5(drive,N,iL)

% This routine computes the sensitivity of damped natural frequencies to
% changes in damping coefficient of various parts of a motor driven ballscrew
% system.
%
% The degrees of freedom considered for the natural frequencies are:-
%  1        motor           torsional
%  2        driving pulley  torsional
%  3        driven pulley   torsional
%  4 – N+3  ballscrew       torsional
%  N+4      ballscrew       axial
%  N+5      table / saddle  axial
%
% The component stiffnesses considered are:-
%           motor shaft          torsional
%           motor bearing        torsional
%           drive belt           torsional equivalent
%           ballscrew            torsional
%           ballscrew bearing    torsional
%           ballscrew + support  axial
%           ballscrew nut        axial
%
% Output variables
%     dnf_dc(,) sensitivity of natural frequency to changes in damping
%     dzt_dc(,) sensitivity of damping ratio to changes in damping
%
% Input variables
%     drive     select 'x' or 'y' drive
%     N         number of elements in ballscrew
%     iL        node number of load position
%
% Other variables
%     c_ax      damping coefficient ballscrew axial
%     C_ax      c_ax incremented by 1%
%     c_bl      stiffness drive belt
%     C_bl      c_bl incremented by 1%
%     c_brg_bs  damping coefficient ballscrew bearing
%     C_brg_bs  c_brg_bs incremented by 1%
%     c_brg_ms  damping coefficient motor bearing
%     C_brg_ms  c_brg_ms incremented by 1%
%     c_bs      stiffness ballscrew
%     C_bs      c_bs incremented by 1%
%     c_ms      stiffness motor shaft
%     C_ms      c_ms incremented by 1%
%     c_nut     stiffness ball nut
%     C_nut     c_nut incremented by 1%
```

```
%     damp()    damping ??
%     dm_ddm()  damping ?? with ith stiffness parameter incremented by 1%
%     i         for loop counter
%     j         for loop counter
%     k         damping ?? for ith mode
%     natf()    natural frequency array
%     nf_dnf    natural frequency with ith stiffness parameter
%                incremented by 1%
%     pi        pi
%     w         natural frequency of ith mode (rad/sec)
%     zeta()    damping ratio array
%     zet_dz()  damping ratio array with ith stiffness parameter
%                incremented by 1%

% Subroutines used
% MATLAB - atan sqrt
% UoH    - natf_dr3_c(*)

pi=4.*atan(1.);

                            % Data common to both drives
c_ms=0.001;%N m sec/rad
c_bs=0.0295;%N m sec/rad
c_nut=58060.;%N sec/m
c_ax=8326.;%N sec/m
c_brg_ms=0.668e-3;%N m sec/rad
c_brg_bs=0.668e-3;%N m sec/rad
if (drive=='x'|drive=='X')
   c_bl=0.797;%N m sec/rad    Data particular to X drive
else
   c_bl=0.452;%N m sec/rad    Data particular to Y drive
end

                            % Determine damped natural frequencies
                            % – original damping coefficients
[natf,damp]=natf_dr3_c(drive,c_ms,c_bl,c_bs,c_nut,c_ax,c_brg_ms,c_brg_bs,N,iL)
;

for i=1:N+5
   k          = damp(i+N+5);
   w          = 2.*pi*natf(i+N+5);
   zeta(i)    =sqrt(k^2/(k^2+w^2));
end

for i=1:7%                    Assemble increased damping coefficient array
   C_ms=c_ms;
   C_bl=c_bl;
   C_bs=c_bs;
   C_nut=c_nut;
   C_ax=c_ax;
   C_brg_ms=c_brg_ms;
   C_brg_bs=c_brg_bs;

   if i==1 C_ms=c_ms*1.01;end
   if i==2 C_bl=c_bl*1.01;end
   if i==3 C_bs=c_bs*1.01;end
   if i==4 C_nut=c_nut*1.01;end
```

```
   if i==5 C_ax=c_ax*1.01;end
   if i==6 C_brg_ms=c_brg_ms*1.01;end
   if i==7 C_brg_bs=c_brg_bs*1.01;end

                           % Determine damped natural frequencies
                           % – ith damping coefficient increased by 1%
   [nf_dnf,dm_ddm]
    =natf_dr3_c(drive,C_ms,C_bl,C_bs,C_nut,C_ax,C_brg_ms,C_brg_bs,N,iL);

   for j=1:N+5
      dnf_dc(i,j)=(nf_dnf(j+N+5)/natf(j+N+5)-1.)*100.;
                           % .... natural frequency partial derivative
      k          = dm_ddm(j+N+5);
      w          = 2.*pi*nf_dnf(j+N+5);
      zet_dz(j)  =sqrt(k^2/(k^2+w^2));
      dzt_dc(i,j)=(zet_dz(j)/zeta(j)-1.)*100.;
                           % .... damping ratio partial derivative
   end
end
```

Appendix 4.8 – **MATLAB routine for calculating sensitivity of a ballscrew system to changes in inertia**

Function "natf_dr3_J" calculates the damped natural frequencies. It is based of function "natf_dr3" (see Appendix 4.5). The input argument list includes the inertia parameters which may be changed and the argument list includes all the natural frequencies and damping terms. The plotting section has been omitted.

```
function[dnf_dJ,dzt_dJ]=natf_dr9(drive,N,iL)

% This routine computes the sensitivity of damped natural frequencies to
% changes in inertia of various parts of a motor driven ballscrew system.
%
% The degrees of freedom considered for the natural frequencies are:-
%  1        motor           torsional
%  2        driving pulley  torsional
%  3        driven pulley   torsional
%  4 – N+3  ballscrew       torsional
%  N+4      ballscrew       axial
%  N+5      table / saddle  axial
%
% The component stiffnesses considered are:-
%           motor shaft          torsional
%           motor bearing        torsional
%           drive belt           torsional equivalent
%           ballscrew            torsional
%           ballscrew bearing    torsional
%           ballscrew + support  axial
%           ballscrew nut        axial
%
% Output variables
%     dnf_dJ(,) sensitivity of natural frequency to changes in inertia
%     dzt_dJ(,) sensitivity of damping ratio to changes in inertia
%
% Input variables
%     drive     select 'x' or 'y' drive
%     N         number of elements in ballscrew
%     iL        node number of load position
%
% Other variables
%     damp()    damping ??
%     dm_ddm()  damping ?? with ith stiffness parameter incremented by 1%
%     i         for loop counter
%     j         for loop counter
%     Jbs       moment of inertia ballscrew
%     Jbs_      Jbs incremented by 1%
%     Jm        moment of inertia motor
%     Jm_       Jm incremented by 1%
%     Jms       moment of inertia motor shaft
%     Jms_      Jms incremented by 1%
%     Jp1       moment of inertia driving pulley
%     Jp1_      Jp1 incremented by 1%
%     Jp2       moment of inertia driven pulley
%     Jp2_      Jp2 incremented by 1%
%     k         damping ?? for ith mode
```

```
%     m_bm      mass of ballscrew
%     m_bm_     m_bm incremented by 1%
%     m_tab     table ('x' drive), saddle and table ('y' drive)
%     m_tab_    m_tab incremented by 1%
%     natf()    natural frequency array
%     nf_dnf    natural frequency with ith stiffness parameter incremented by
1%
%     pi        pi
%     w         natural frequency of ith mode (rad/sec)
%     zeta()    damping ratio array
%     zet_dz()  damping ratio array with ith stiffness parameter incremented
by 1%

% Subroutines used
% MATLAB - atan sqrt
% UoH    - natf_dr_J
%
pi=4.*atan(1.);

                              % Data common to both drives
Jm=0.0048;%kg m^2
Jms=0.0000075;%kg m^2
Jp1=0.0004986;%kg m^2
Jbs=0.002724;%kg m^2
m_bm=5.;%kg
if (drive=='x'|drive=='X')
   Jp2=0.017779;%kg m^2       Data particular to X drive
   m_tab=359.;%kg
else
   Jp2=0.009524;%kg m^2       Data particular to Y drive
   m_tab=677.;%kg
end

                              % Determine damped natural frequencies
                              % – original inertias
 [natf,damp]=natf_dr_J(drive,Jm,Jms,Jp1,Jp2,Jbs,m_bm,m_tab,N,iL);

for i=1:N+5
   k          = damp(i+N+5);
   w          = 2.*pi*natf(i+N+5);
   zeta(i)    =sqrt(k^2/(k^2+w^2));
end

for i=1:7                      %Assemble increased inertia array
   Jm_=Jm;
   Jms_=Jms;
   Jp1_=Jp1;
   Jp2_=Jp2;
   Jbs_=Jbs;
   m_bm_=m_bm;
   m_tab_=m_tab;
   if i==1 Jm_=Jm*1.01;end
   if i==2 Jms_=Jms*1.01;end
   if i==3 Jp1_=Jp1*1.01;end
   if i==4 Jp2_=Jp2*1.01;end
   if i==5 Jbs_=Jbs*1.01;end
   if i==6 m_bm_=m_bm*1.01;end
```

```
   if i==7 m_tab_=m_tab*1.01;end

                           % Determine damped natural frequencies
                           % – ith inertia increased by 1%
   [nf_dnf,dm_ddm]=natf_dr_J(drive,Jm_,Jms_,Jp1_,Jp2_,Jbs_,m_bm_,m_tab_,N,iL);

   for j=1:N+5
      dnf_dJ(i,j)=(nf_dnf(j+N+5)/natf(j+N+5)-1.)*100.;
                           % .... natural frequency partial derivative
      k          = dm_ddm(j+N+5);
      w          = 2.*pi*nf_dnf(j+N+5);
      zet_dz(j)  =sqrt(k^2/(k^2+w^2));
      dzt_dJ(i,j)=(zet_dz(j)/zeta(j)-1.)*100.;
                           % .... damping ratio partial derivative
   end
end
```

Appendix 5.1 - **“C” program for axial and torsional degrees of freedom**

```
// Program "T3.cpp"
//         ========
//
// Version dated 16/10/02
// Axial and torsional freedoms only, one node nut, generalised solution method.
// Archive 03 - torque applied to nut driven end of screw.
// "Long double" precision
// Table friction included, (but may need more sophistication).
//
// Standard subroutine libraries
#include <iostream.h>
#include <math.h>
#include <fstream.h>

// Subroutine prototypes
signed int dl_ab(signed int S, signed int N, double dt1);
double dl_dt(signed int S, signed int N, float dt_in);
void d_b0(signed int Msize, signed int j);
void dl_b1(signed int Msize, signed int j, double dt1);
void dl_b2(signed int S, signed int N, double dt1);
void dl_bk(signed int S, signed int N, signed int k);
void d_c0(signed int Msize, signed int j);
void dl_c1(signed int Msize, double dt1);
void dl_ck(signed int S, signed int N, signed int k);
float d_KE(signed int Msize, signed int k);
float dl_m_d(signed int Msize, signed int j);
float dl_max_d(signed int Msize, signed int j);
float d_m_extra(signed int Msize, signed int j);
double dl_PE2(double bs_rat, double blash, double blf1, double blf2,
 signed int Msize, signed int S, signed int j);
void dl_rEnD(double bs_rat, float c_brg_1, float c_brg_2, signed int Msize,
 signed int S, signed int j);
void inc_d(signed int Msize, signed int j, signed int k);
void inc_v(signed int Msize, signed int j, signed int k);
double fric_tab(float m_tab, float g, float mu_tab, double bs_rat, double v_max,
 double blash, double blf2, signed int S, signed int N, signed int j,
 double dt1, signed int mode);
//%
//% Definition of matrix and array sizes
#define SMAX 50         // Maximum number of elements in ballscrew
#define NFS 2           // Number of degrees of freedom per ballscrew node
#define BWS 4           // Bandwidth of ballscrew
#define NMAX 0          // Maximum number of elements in ballscrew nut,
                        //  (0 = 1 node)
#define NFN 1           // Number of degrees of freedom per nut node
#define BWN 1           // Bandwidth of nut
#define KMAX 22         // Maximum number of iterations
#define NBLOCK 1000     // Maximum number of steps in time history block
#define SWRIT 100       // Number of time history steps between screen writes

#define L1 (1+SMAX)
#define L2 (1+NBLOCK)


#define M1 (1+SMAX+1+NMAX+1)
#define M2 (1+NFS*(SMAX+1)+NFN*(NMAX+1))
#define M3 (1+SMAX+1)
#define M4 (BWS+BWN)
```

```
#define M5 (1+BWS+BWN)
#define M6 ((BWS+BWN)*2)
#define M7 (1+KMAX)

//% Output variables
static float x[L1];         // x array for plotting, (m)
static float t1[L2];        // time, (sec)
static float lind[L2][M1];  // linear displacement time history matrix, (m)
static float linv[L2][M1];  // linear velocity time history matrix, (m/sec)
static float tord[L2][M1];  // torsional displacement time history matrix, (rad)
static float torv[L2][M1];  // torsional velocity time history matrix, (rad/sec)
static float KE[L2];        // kinetic energy, (J)
static float PE[L2];        // potential energy, (J)
static float Ei[L2];        // energy input, (J)
static float ED[L2];        // energy dissipated, (J)
static float ED1[L2];       // energy dissipated by table friction, (J)
static float TE[L2];        // total energy, (J)
static double rEnD[M3];     // rate of energy dissipation, (W)
static float R[L2];         // rate of energy dissipation, (W)
//%
//% Common arrays
static double K[M2][M5];    // stiffness matrix, (N/m)
static float K1[M2][M5];    // stiffness matrix - "fixed" version, (N/m)
static double C[M2][M5];    // damping matrix, (N sec/m)
static float C1[M2][M5];    // damping matrix - "fixed" version, (N sec/m)
static float M[M2];         // mass of ballscrew elements, (kg)
static float Mi[M2];        // inverse of mass array, (1/kg)
static long double alpha[M2][M6]; // "-/M*dt1^2" matrix, (1)
static long double beta[M2][M6];  // "-C/M*dt1" matrix, (1)
static float f[M2];         // force vector, (N)
static float f1[L2];        // force vector - screw end torque, (N m)
static float f2[L2];        // force vector - nut, (N)
static float Kn[L2];        // nut stiffness, (N/m)
static float Cn[L2];        // nut damping, (N sec/m)
static long double b[M2][M7]; // terms in time series for displacement,
                              //  (m & rad)
static long double c[M2][M7]; // terms in time series for velocity,
                              //  (m/sec & rad/sec)
static long double d[M2][M3]; // linear displacement time history matrix,
                              //  (m & rad)
static long double v[M2][M3]; // linear velocity time history matrix,
                              //  (m/sec & rad/sec)
static float xn1[L2];       // position of nut - ideal, (m)
static float err[L2];       // controller error, (rad)
static float perr[L2];      // position error, (rad)
//%
/*=========================================================================*/
//%
signed int main(void)
{
//% Input variables
signed int S=20;      // number of elements in ballscrew
signed int N=0;       // number of elements in ballscrew nut, (0 = 1 node)
float dt_in=0.001F;   // input time step, (sec)
float drs=0.00001F;   // damping ratio - shaft
float drb=0.0001F;    // damping ratio - bearings
double drn=0.0001;    // damping ratio - nut
double x1=0.1;        // x coordinate of start point, (m)
double x2=0.2;        // x coordinate of end of acceleration, (m)
double x3=0.5;        // x coordinate of start of deceleration, (m)
double x4=0.6;        // x coordinate of end point, (m)
```

```
double v_max=0.2;     // maximum nut velocity, (m/sec)
signed int mode=1;    // "fric_tab" mode of operation
//%                      0 = "balanced force" mode
//%                      1 = "damper" mode
if (S>SMAX) {cout << "Too many elements in ballscrew - STOP"; return(1);}
if (N>NMAX) {cout << "Too many elements in nut - STOP"; return(1);}
//%
//% Other variables
double a1;            // rate of acceleration, (m/sec^2)
double a2;            // rate of deceleration, (m/sec^2)
double blash=10.e-6;  // backlash, (m)
double blf1;          // backlash factor 1
double blf2;          // backlash factor 2
float A_bs;           // cross-sectional area of ballscrew, (m^2)
double bs_rat;        // ballscrew ratio, (m/rad)
double bs_rat2;       // =bs_rat^2, (m^2/rad^2)
signed int calc;      // power series calculation indicator,
                      //  = 'cont' - continue
                      //  = 'stop' - stop
float c_brg_1=0.668e-3F; // equivalent damping coefficient of bearings,
float c_brg_2=0.668e-3F; // (N m sec/rad)
float d_bs=38.1e-3F;  // diameter of ballscrew, (m)
double dEnD;          // increment in energy dissipated, (J)
double df1;           // stiffness and damping distribution factor 1
double df2;           // stiffness and damping distribution factor 2
double dt;            // time step, (sec)
double dt1;           // time step, inner loop, (sec)
float E=207.0e9F;     // Young's modulus, (N/m^2)
float Ein;            // energy input, (J)
double EnD;           // energy dissipated, (J)
double EnD1;          // energy dissipated by table friction, (J)
double eps=1.0e-9;    // convergence criterion
float error;          // error in position of ballscrew end, (rad)
float error1;         // error - previous step, (rad)
float fd;             // driving torque, (N m)
float fd_max;         // maximum torque at ballscrew enf, (N m)
float g=9.80665F;     // acceleration due to gravity, (m/sec^2)
float G=80.0e9F;      // shear modulus, (N/m^2)
signed int i;         // for loop counter
signed int ib;        // block item counter
signed int ii;        // for loop counter
signed int it;        // for loop counter
signed int i1;        // matrix element subscript
signed int i2;        // matrix element subscript
signed int j;         // for loop counter
float J_bs;           // total inertia of ballscrew, (kg m^2)
signed int jj;        // for loop counter
signed int k;         // index in power series
float k_ax_nut=1.e8F; // axial stiffness of nut, (N/m)
float k_ax_1= 8.0e8F; // axial stiffness of bearings, (N/m)
float k_ax_2=15.6e8F;
float K_bs;           // torsional constant of ballscrew, (m^4)
float Kd=0.F;         // controller differential constant, (N m sec/rad)
float Ki=100.F;       // controller integral constant, (N m/(rad sec))
float Kp=2.F;         // controller proportional constant, (N m/rad)
float l_bs=1.520F;    // length of ballscrew, (m)
float max_d;          // measure of maximum term of displacement vector
                      //  at kth term, (?)
float m_d;            // measure of displacement vector at kth term, (?)
float m_extra;        // measure of last four terms in calculation
                      //  of d vector, (?)
```

```
float m_tab=359.F;    // mass of saddle and table, (kg)
float mu_tab=0.1F;    // coefficient of friction between table and slideways
signed int Nt;        // number of columns in time history matrices
signed int Ns=1;      // number of starts of ballscrew grooves
double Ob;            // angular position of "nut point" on screw, (rad)
double pi;            // pi
float pitch=0.010F;   // ballscrew pitch, (m)
float rho=7600.F;     // density, (kg/m^3)
float S_error;        // integral of error in position of ballscrew end,
                      //  (rad sec)
signed int SN1;       // SN1=S+1+N+1, number of nodes
signed int SN2;       // SN2=NFS*(S+1)+NFN*(N+1), number of degrees of freedom
double t;             // time, (sec)
double t2;            // time of end of acceleration, (sec)
double t3;            // time of start of deceleration, (sec)
double t4;            // time of end point, (sec)
double t5;            // time at end of simulation, (sec)
double vn;            // velocity of nut - ideal, (m/sec)
double xb;            // position of "nut point" on screw, (m)
double xn;            // position of nut - ideal, (m)
//%
//% Subroutines used
//%   1. atan, cout, double, float, fmod, ofstream, return, signed int
//%   2. dl_ab, dl_dt, d_b0, dl_b1, dl_b2, dl_bk, d_c0, dl_c1, dl_ck, d_KE,
//%      dl_max_d, dl_m_d, d_m_extra, dl_PE2, dl_rEnD, inc_d, inc_v
//%
//% Output streams used
//%   os_x, os_t1, os_f1, os_f2, os_Kn, os_Cn, os_xn1, os_lind, os_linv,
//%   os_tord, os_torv, os_KE, os_PE, os_Ei, os_ED, os_ED1, os_TE, os_rEnD,
//%   os_R, os_err, os_perr, os_b1, os_b2, os_Kf, os_b
//%
//% Special variables used
//%   endl, NULL
//%
// Sets up output streams for results files
ofstream os_x("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\x",ios::out);
if (os_x==NULL) {cout << "File 'x' not opened - STOP"; return(1);}
ofstream os_t1("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\t1",ios::out);
if (os_t1==NULL) {cout << "File 't1' not opened - STOP"; return(1);}
ofstream os_f1("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\f1",ios::out);
if (os_f1==NULL) {cout << "File 'f1' not opened - STOP"; return(1);}
ofstream os_f2("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\f2",ios::out);
if (os_f2==NULL) {cout << "File 'f2' not opened - STOP"; return(1);}
ofstream os_Kn("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\Kn",ios::out);
if (os_Kn==NULL) {cout << "File 'Kn' not opened - STOP"; return(1);}
ofstream os_Cn("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\Cn",ios::out);
if (os_Cn==NULL) {cout << "File 'Cn' not opened - STOP"; return(1);}
ofstream os_xn1("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\xn1",ios::out);
if (os_xn1==NULL) {cout << "File 'xn1' not opened - STOP"; return(1);}
ofstream
os_lind("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\lind",ios::out);
if (os_lind==NULL) {cout << "File 'lind' not opened - STOP"; return(1);}
ofstream
os_linv("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\linv",ios::out);
if (os_linv==NULL) {cout << "File 'linv' not opened - STOP"; return(1);}
ofstream
os_tord("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\tord",ios::out);
if (os_tord==NULL) {cout << "File 'tord' not opened - STOP"; return(1);}
ofstream
os_torv("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\torv",ios::out);
if (os_torv==NULL) {cout << "File 'torv' not opened - STOP"; return(1);}
```

```
ofstream os_KE(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\KE”,ios::out);
if (os_KE==NULL) {cout << "File 'KE' not opened - STOP"; return(1);}
ofstream os_PE(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\PE”,ios::out);
if (os_PE==NULL) {cout << "File 'PE' not opened - STOP"; return(1);}
ofstream os_Ei(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\Ei”,ios::out);
if (os_Ei==NULL) {cout << "File 'Ei' not opened - STOP"; return(1);}
ofstream os_ED(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\ED”,ios::out);
if (os_ED==NULL) {cout << "File 'ED' not opened - STOP"; return(1);}
ofstream os_ED1(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\ED1”,ios::out);
if (os_ED1==NULL) {cout << "File 'ED1' not opened - STOP"; return(1);}
ofstream os_TE(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\TE”,ios::out);
if (os_TE==NULL) {cout << "File 'TE' not opened - STOP"; return(1);}
ofstream
os_rEnD(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\rEnD”,ios::out);
if (os_rEnD==NULL) {cout << "File 'rEnD' not opened - STOP"; return(1);}
ofstream os_R(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\R”,ios::out);
if (os_R==NULL) {cout << "File 'R' not opened - STOP"; return(1);}
ofstream os_err(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\err”,ios::out);
if (os_err==NULL) {cout << "File 'err' not opened - STOP"; return(1);}
ofstream
os_perr(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\perr”,ios::out);
if (os_perr==NULL) {cout << "File 'perr' not opened - STOP"; return(1);}
ofstream os_b1(“d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\b1”,ios::out);
if (os_b1==NULL) {cout << "File 'b1' not opened - STOP"; return(1);}
ofstream os_b2( “d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\b2”,ios::out);
if (os_b2==NULL) {cout << "File 'b2' not opened - STOP"; return(1);}

pi=4.*atan(1.);           // pi

bs_rat=double(Ns)*double(pitch)/(2.*pi); // ballscrew ratio
bs_rat2=bs_rat*bs_rat;

SN1=S+1+N+1;             // number of nodes
SN2=NFS*(S+1)+NFN*(N+1); // number of freedoms

// Define inertia array
J_bs=0.F;
M[1]=float(pi)/4.F*(d_bs*d_bs)*l_bs*rho/(2.F*float(S));
M[2]=float(pi)/64.F*(d_bs*d_bs*d_bs*d_bs)*l_bs*rho/(2.F*float(S));
J_bs=J_bs + M[2];
for (i=2;i<=S;i++)
  {
  i1=2*i-1;
  i2=2*i;
  M[i1]=float(pi)/4.F*(d_bs*d_bs)*l_bs*rho/float(S);
  M[i2]=float(pi)/64.F*(d_bs*d_bs*d_bs*d_bs)*l_bs*rho/float(S);
  J_bs=J_bs + M[i2];
  }
M[NFS*S+1]=float(pi)/4.F*(d_bs*d_bs)*l_bs*rho/(2.F*float(S));
M[NFS*S+2]=float(pi)/64.F*(d_bs*d_bs*d_bs*d_bs)*l_bs*rho/(2.F*float(S));
J_bs=J_bs + M[NFS*S+2];
M[NFS*(S+1)+NFN*(N+1)]=m_tab; // only works if N=0

// Define inverse mass array
for (i=1;i<=SN2;i++)
  {
  Mi[i]=1.F/M[i];
  }

// Define stiffness and damping matrices ("fixed" terms),
// only valid for NFS=2 ,axial and torsional
```

```
// Lines marked !£! - only valid for NFN=1 and N=0
A_bs=float(pi)/4.F*(d_bs*d_bs);
K_bs=float(pi)/32.F*(d_bs*d_bs*d_bs*d_bs);

for (i=1;i<=SN2;i++)
  {
  for (j=1;j<=M4;j++)
    {
    K1[i][j]=0.F;
    C1[i][j]=0.F;
    }
  }
K1[1][1]=     E*A_bs/(l_bs/float(S));
K1[1][3]=-1.F*E*A_bs/(l_bs/float(S));
K1[2][1]=     G*K_bs/(l_bs/float(S));
K1[2][3]=-1.F*G*K_bs/(l_bs/float(S));
for (i=2;i<=S;i++)
  {
  i1=2*i-1;
  i2=2*i;
  K1[i1][1]= 2.F*E*A_bs/(l_bs/float(S));
  K1[i1][3]=-1.F*E*A_bs/(l_bs/float(S));
  K1[i2][1]= 2.F*G*K_bs/(l_bs/float(S));
  K1[i2][3]=-1.F*G*K_bs/(l_bs/float(S));
  }
K1[NFS*S+1][1]=E*A_bs/(l_bs/float(S));
K1[NFS*S+2][1]=G*K_bs/(l_bs/float(S));

for (i=1;i<=NFS*(S+1);i++)
  {
  for (j=1;j<=M4;j++)
    {
    C1[i][j]=drs*K1[i][j];
    }
  }

K1[1][1]=K1[1][1] + k_ax_1;
K1[NFS*S+1][1]=K1[NFS*S+1][1] + k_ax_2;
K1[NFS*(S+1)+NFN*(N+1)][1]=k_ax_nut; // !£!
C1[1][1]=C1[1][1] + drb*k_ax_1;
C1[2][1]=C1[2][1] + c_brg_1;
C1[NFS*S+1][1]=C1[NFS*S+1][1] + drb*k_ax_2;
C1[NFS*S+2][1]=C1[NFS*S+2][1] + c_brg_2;
C1[NFS*(S+1)+NFN*(N+1)][1]=K1[NFS*(S+1)+NFN*(N+1)][1]*float(drn); // !£!

// Define time step
for (i=1;i<=SN2;i++)// Zeroise “alpha” and “beta” matrices
  {
  for (j=1;j<=2*BWS+2*BWN-1;j++)
    {
    alpha[i][j]=0.L;
    beta [i][j]=0.L;
    }
  }
dt=dl_dt(S,N,dt_in);
cout << "dt_in = " << dt_in << " sec, dt = " << dt << " sec\n";
if (dt<0.) {cout << "ERROR:  failure to define time step.\n"; return(1);}

// Define trapezoidal velocity profile parameters
a1=(v_max*v_max)/(2.*(x2-x1));
t2=v_max/a1;
```

```
t3=t2 + (x3-x2)/v_max;
a2=-(v_max*v_max)/(2.*(x4-x3));
t4=t3 - v_max/a2;
t5=t4*1.1;

// Estimates maximum torque on ballscrew
fd_max=(J_bs+float(bs_rat2)*m_tab)*float(a1/bs_rat) * 5.F;

xn=x1; // Sets initial position of nut
vn=0.; //

// Sets number of time steps
Nt=signed int(t5/dt)+1; cout << "Nt = " << Nt << "\n";

// Zeroises output arrays and energy variables
t1[0]=0.F; os_t1 << " " << t1[0];
f1[0]=0.F; os_f1 << " " << f1[0];
f2[0]=0.F; os_f2 << " " << f2[0];
Kn[0]=0.F; os_Kn << " " << Kn[0];
Cn[0]=0.F; os_Cn << " " << Cn[0];
xn1[0]=float(x1); os_xn1 << " " << xn1[0];
for (i=1;i<=S+1;i++)
  {
  lind[0][i]=0.F; os_lind << " " << lind[0][i];
  linv[0][i]=0.F; os_linv << " " << linv[0][i];
  tord[0][i]=float(x1/bs_rat); os_tord << " " << tord[0][i];
  torv[0][i]=0.F; os_torv << " " << torv[0][i];
  }
lind[0][SN1]=float(x1); os_lind << " " << lind[0][SN1];
linv[0][SN1]=0.F; os_linv << " " << linv[0][SN1];
tord[0][SN1]=0.F; os_tord << " " << tord[0][SN1];
torv[0][SN1]=0.F; os_torv << " " << torv[0][SN1];
os_lind << endl;
os_linv << endl;
os_tord << endl;
os_torv << endl;

KE[0]=0.F; os_KE << " " << KE[0];
PE[0]=0.F; os_PE << " " << PE[0];
Ein=0.F;
Ei[0]=0.F; os_Ei << " " << Ei[0];
EnD=0.;
EnD1=0.;
ED[0]=0.F; os_ED << " " << ED[0];
ED1[0]=0.F; os_ED1 << " " << ED1[0];
os_TE << " " << 0.;
rEnD[1]=0.; os_rEnD << " " << rEnD[1];
os_R << " " << 0.;
os_err << " " << 0.;
os_perr << " " << 0.;
os_b1 << " " << 0.;
os_b2 << " " << 0.;

// Zeroises displacement, velocity and force arrays
for (i=1;i<=SN2;i++)
  {
  d[i][1]=0.;
  v[i][1]=0.;
  }

// Defines 'x' co-ordinate array for ballscrew
```

```
for (i=0;i<=S;i++)
  {
  x[i]=float(i)*l_bs/float(S); os_x << " " << x[i];
  }
os_x << endl;

// Initialises working variables for main loop
ib=0;
error1=0.F;
S_error=0.F;
dt1=dt/double(S);
t=-dt1;

for (it=1;it<=Nt;it++) // Start main loop
  {
  ib=ib+1;
  error=float(vn/bs_rat-v[2][1]); // velocity control
  S_error=S_error + error*float(dt);
  fd=Kp*error + Ki*S_error + Kd*(error-error1)/float(dt);
  error1=error;
  for (j=1;j<=S;j++) // Start inner loop
    {
    t=t+dt1;
    // Defines trapezoidal velocity profile
    for (i=1;i<=SN2;i++)
      {
      f[i]=0.F;
      }
    if ((0.<=t) && (t<t2))
      {
      xn=x1 + 0.5*a1*(t*t);
      vn=a1*t;
      }
    else if ((t2<=t) && (t<t3))
      {
      xn=x2 + v_max*(t-t2);
      vn=v_max;
      }
    else if ((t3<=t) && (t<t4))
      {
      xn=x3 + v_max*(t-t3) + 0.5*a2*((t-t3)*(t-t3));
      vn=v_max + a2*(t-t3);
      }
    else
      {
      xn=x4;
      vn=0.;
      }

    // Copy "fixed" stiffness and damping matrices into working matrices
    for (ii=1;ii<=SN2;ii++)
      {
      for (jj=1;jj<=M4;jj++)
        {
        K[ii][jj]=double(K1[ii][jj]);
        C[ii][jj]=double(C1[ii][jj]);
        }
      }

    // Define nut/screw connection terms
    for (ii=1;ii<=S;ii++) // (only valid for NFS=2, axial and torsional)
```

```
{
i1=2*ii-1;
i2=2*ii;
if ((x[ii-1]<=xn) && (xn<=x[ii]))
  {
  df1=(x[ii] - xn)*double(S)/double(l_bs);
  df2=(xn-x[ii-1])*double(S)/double(l_bs);
  xb=df1*d[i1][j] + df2*d[i1+NFS][j];
  Ob=df1*d[i2][j] + df2*d[i2+NFS][j];
  if ((xb+bs_rat*Ob)<(d[SN2][j]-blash/2.)) // Defines backlash state
    {
    blf1=1.;
    blf2=-1.;
    }
  else if ((d[SN2][j]+blash/2.)<(xb+bs_rat*Ob))
    {
    blf1=1.;
    blf2=1.;
    }
  else
    {
    blf1=0.;
    blf2=0.;
    }
  K[i1][1]=K[i1][1] + blf1*double(k_ax_nut)*df1;
  K[i1+NFS][1]=K[i1+NFS][1] + blf1*double(k_ax_nut)*df2;
  K[i1][M4]=blf1*double(-k_ax_nut)*df1;
  K[i1+NFS][M4]=blf1*double(-k_ax_nut)*df2;

  K[i2][1]=K[i2][1] + blf1*bs_rat2*double(k_ax_nut)*df1;
  K[i2+NFS][1]=K[i2+NFS][1] + blf1*bs_rat2*double(k_ax_nut)*df2;
  K[i2][M4]=-blf1*bs_rat*double(k_ax_nut)*df1;
  K[i2+NFS][M4]=-blf1*bs_rat*double(k_ax_nut)*df2;

  K[i1][2]=blf1*bs_rat*double(k_ax_nut)*df1;
  K[i1+NFS][2]=blf1*bs_rat*double(k_ax_nut)*df2;

  K[SN2][1]=blf1*K[SN2][1];

  C[i1][1]=C[i1][1] - K[i1][M4]*drn;
  C[i1+NFS][1]=C[i1+NFS][1] - K[i1+NFS][M4]*drn;
  C[i1][M4]=K[i1][M4]*drn;
  C[i1+NFS][M4]=K[i1+NFS][M4]*drn;

  C[i2][1]=C[i2][1] - K[i2][M4]*bs_rat*drn;
  C[i2+NFS][1]=C[i2+NFS][1] - K[i2+NFS][M4]*bs_rat*drn;
  C[i2][M4]=K[i2][M4]*drn;
  C[i2+NFS][M4]=K[i2+NFS][M4]*drn;

  C[i1][2]=K[i1][2]*drn;
  C[i1+NFS][2]=K[i1+NFS][2]*drn;

  C[SN2][1]=blf1*C[SN2][1];

  f[i1]=float(blf2*blash*double(k_ax_nut)*df1/2.); // Modifies force
  f[i1+NFS]=float(blf2*blash*double(k_ax_nut)*df2/2.); // depending on
  f[i2]=float(blf2*bs_rat*blash*double(k_ax_nut)*df1/2.); // state of
  f[i2+NFS]=float(blf2*bs_rat*blash*double(k_ax_nut)*df2/2.); // backlash
  f[SN2]=-float(blf2*blash*double(k_ax_nut)/2.);
  f[2]=f[2] + fd;
  }
```

```
  }

ofstream os_Kf("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\Kf",
                ios::out);
if (os_Kf==NULL) {cout << "File 'Kf' not opened - STOP"; return(1);}
os_Kf << "K matrix:- \n";
for (i=1;i<=NFS*(S+1)+NFN*(N+1);i++)
  {
  os_Kf << "i = " << i ;
  for (jj=1;jj<=M4;jj++)
    {
    os_Kf << " " << K[i][jj];
    }
  os_Kf << endl;
  }
os_Kf << "f matrix:- \n";
for (i=1;i<=NFS*(S+1)+NFN*(N+1);i++)
  {
  os_Kf << "i = " << i << " " << f[i] << endl;
  }

// Increments energy dissipated
dEnD=fric_tab(m_tab,g,mu_tab,bs_rat,v_max,blash,blf2,S,N,j,dt1,mode);
EnD=EnD + dEnD;
// Increments energy dissipated by table friction
EnD1=EnD1 + dEnD;

// Defines "alpha" and "beta" matrices
dl_ab(S,N,dt1);

// Builds up the solution arrays "b" and "c"
// and increments the displacements and velocities
d_b0(SN2,j); inc_d(SN2,j+1,0);
d_c0(SN2,j); inc_v(SN2,j+1,0);
dl_b1(SN2,j,dt1); inc_d(SN2,j+1,1); dl_b2(S,N,dt1);
dl_c1(SN2,dt1); inc_v(SN2,j+1,1);
inc_d(SN2,j+1,2);
dl_ck(S,N,2); inc_v(SN2,j+1,2);
calc='cont';
k=2;
while (calc=='cont')
  {
  for (i=1;i<=4;i++)
    {
    k=k+1;
    if (KMAX<k)
      {
      cout << "ERROR:  power series calculation run over matrix
                size limit.\n";
      return(1);
      }
    dl_bk(S,N,k); inc_d(SN2,j+1,k);
    dl_ck(S,N,k); inc_v(SN2,j+1,k);
    }

  // Tests for convergence of solution
  m_extra=d_m_extra(SN2,k);
  m_d=dl_m_d(SN2,j+1);
  max_d=dl_max_d(SN2,j+1);
  if ((m_extra<m_d*eps)||(max_d<1.e-12F)) calc='stop';
  }
```

```
  ofstream os_b("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\b",ios::out);
  if (os_b==NULL) {cout << "File 'b' not opened - STOP"; return(1);}
  os_b << "b matrix:- \n";
  for (i=1;i<=NFS*(S+1)+NFN*(N+1);i++)
    {
    os_b << "i = " << i ;
    for (jj=1;jj<=k;jj++)
      {
      os_b << " " << b[i][jj];
      }
    os_b << endl;
    }
  os_b << "c matrix:- \n";
  for (i=1;i<=NFS*(S+1)+NFN*(N+1);i++)
    {
    os_b << "i = " << i ;
    for (jj=1;jj<=k;jj++)
      {
      os_b << " " << c[i][jj];
      }
    os_b << endl;
    }

  // Calculates rate of energy dissipation due to viscous damping
  dl_rEnD(bs_rat,c_brg_1,c_brg_2,SN2,S,j+1);
  if (it==Nt) os_rEnD << " " << rEnD[j+1];
  EnD=EnD + (rEnD[j]+rEnD[j+1])/2.*dt1;
  Ein=Ein + float(d[2][j+1]-d[2][j])*-fd;
  } // End of inner loop
os_b1 << " " << blf1;
os_b2 << " " << blf2;

// Up-dates output arrays
t1[ib]=float(t+dt1);
f1[ib]=fd;
f2[ib]=f[SN2];
Kn[ib]=float(K[SN2][1]);
Cn[ib]=float(C[SN2][1]);
err[ib]=error;
xn1[ib]=float(xn);
for (i=1;i<=SN1;i++) // (only valid for NFS=2 ,axial and torsional)
  {
  i1=2*i-1;
  i2=2*i;
  lind[ib][i]=float(d[i1][S+1]);
  linv[ib][i]=float(v[i1][S+1]);
  tord[ib][i]=float(d[i2][S+1] + x1/bs_rat);
  torv[ib][i]=float(v[i2][S+1]);
  }
lind[ib][SN1]=lind[ib][SN1] + float(x1);
perr[ib]=float(d[SN2][S+1]+x1-xn);

// Re-sets displacement, velocity and energy dissipation rate arrays
for (i=1;i<=SN2;i++)
  {
  d[i][1]=d[i][S+1];
  v[i][1]=v[i][S+1];
  }
rEnD[1]=rEnD[S+1];
```

```
  // Up-dates energy arrays
  KE[ib]=d_KE(SN2,S+1);
  PE[ib]=float(dl_PE2(bs_rat,blash,blf1,blf2,SN2,S,S+1));
  Ei[ib]=Ein;
  ED[ib]=float(EnD);
  ED1[ib]=float(EnD1);
  TE[ib]=KE[ib]+PE[ib]+Ei[ib]+ED[ib];
  R[ib]=float(rEnD[S+1]);

  // Write results to file
  if ((ib==NBLOCK)||(it==Nt))
    {
    for (ii=1;ii<=ib;ii++)
      {
      os_t1 << " " << t1[ii];
      os_f1 << " " << f1[ii];
      os_f2 << " " << f2[ii];
      os_Kn << " " << Kn[ii];
      os_Cn << " " << Cn[ii];
      os_xn1 << " " << xn1[ii];
      for (jj=1;jj<=SN1;jj++)
        {
        os_lind << " " << lind[ii][jj];
        os_linv << " " << linv[ii][jj];
        os_tord << " " << tord[ii][jj];
        os_torv << " " << torv[ii][jj];
        }
      os_lind << endl;
      os_linv << endl;
      os_tord << endl;
      os_torv << endl;
      os_KE << " " << KE[ii];
      os_PE << " " << PE[ii];
      os_Ei << " " << Ei[ii];
      os_ED << " " << ED[ii];
      os_ED1 << " " << ED1[ii];
      os_TE << " " << TE[ii];
      os_R << " " << R[ii];
      os_err << " " << err[ii];
      os_perr << " " << perr[ii];
      }
    ib=0; // Reset block counter
    }
  if (signed int(fmod(double(Nt-it),double(SWRIT)))==0)
    {
    cout << Nt-it << " loops to go\n";
    }
  } // End of main loop
cout << "fd_max = " << fd_max << "\n";

// Close files
os_t1 << endl;
os_f1 << endl;
os_f2 << endl;
os_Kn << endl;
os_Cn << endl;
os_xn1 << endl;
os_KE << endl;
os_PE << endl;
os_Ei << endl;
os_ED << endl;
```

```
os_ED1 << endl;
os_TE << endl;
os_rEnD << endl;
os_R << endl;
os_err << endl; cout << endl;
os_perr << endl;
os_b1 << " " << endl;
os_b2 << " " << endl;

return(0);
}

/*=============================================================================*/

signed int dl_ab(signed int S, signed int N, double dt1)
{
//  This subroutine defines the "alpha" and "beta" arrays using the
//  formulas:-
//          alpha = -M * dt^2
//          beta  = -M * dt
//
//  The detailed coding takes into account the special compact layout
//  of the matrices.
//
//% Input variables
// S;                  // number of elements in ballscrew
// N;                  // number of elements in ballscrew nut
// dt1;                // time step, inner loop, (sec)
//%
//% Other variables
signed int i;          // for loop counter
signed int j;          // for loop counter
//%
//% Subroutines used
//%   double, cout, long double
//%
//% Output streams used
//% os_ab
//%
for (i=1;i<=NFS*(S+1);i++)
  {
  for (j=-BWS+1;j<=BWS-1;j++)
    {
    if ((j<0)&&(0<i+j))
      {
      alpha[i][j+BWS+BWN]
       =(-1.L)*long double(Mi[i])*long double(K[i+j][1-j]*(dt1*dt1));
      beta [i][j+BWS+BWN]
       =(-1.L)*long double(Mi[i])*long double(C[i+j][1-j]*dt1);
      }
    if ((0<=j)&&(i+j<=NFS*(S+1)))
      {
      alpha[i][j+BWS+BWN]
       =(-1.L)*long double(Mi[i])*long double(K[i][j+1]*(dt1*dt1));
      beta [i][j+BWS+BWN]
       =(-1.L)*long double(Mi[i])*long double(C[i][j+1]*dt1);
      }
    }
  }
for (i=1;i<=NFS*(S+1);i++)
  {
```

```
  alpha[i][1]=(-1.L)*long double(Mi[NFS*(S+1)+NFN*(N+1)])
                *long double(K[i][BWS+BWN]*(dt1*dt1));
  beta [i][1]=(-1.L)*long double(Mi[NFS*(S+1)+NFN*(N+1)])
                *long double(C[i][BWS+BWN]*dt1);
  alpha[i][2*BWS+2*BWN-1]=(-1.L)*long double(Mi[i])
                           *long double(K[i][BWS+BWN]*(dt1*dt1));
  beta [i][2*BWS+2*BWN-1]=(-1.L)*long double(Mi[i])
                           *long double(C[i][BWS+BWN]*dt1);
  }
alpha[NFS*(S+1)+NFN*(N+1)][BWS+BWN]
 =(-1.L)*long double(Mi[NFS*(S+1)+NFN*(N+1)])
   *long double(K[NFS*(S+1)+NFN*(N+1)][1]*(dt1*dt1)); /*NOT*/
beta [NFS*(S+1)+NFN*(N+1)][BWS+BWN]
 =(-1.L)*long double(Mi[NFS*(S+1)+NFN*(N+1)])
   *long double(C[NFS*(S+1)+NFN*(N+1)][1]*dt1); /*NOT*/

ofstream os_ab("d:\\c_drive_1\\ph_d\ballscrew_model\\dynamic_2\\ab",ios::out);
if (os_ab==NULL) {cout << "File 'ab' not opened - STOP"; return(1);}
os_ab << "alpha matrix:- \n";
for (i=1;i<=NFS*(S+1)+NFN*(N+1);i++)
  {
  os_ab << "i = " << i ;
  for (j=1;j<=2*(BWN+BWS)-1;j++)
    {
    os_ab << " " << alpha[i][j];
    }
  os_ab << endl;
  }
os_ab << "beta matrix:- \n";
for (i=1;i<=NFS*(S+1)+NFN*(N+1);i++)
  {
  os_ab << "i = " << i ;
  for (j=1;j<=2*(BWN+BWS)-1;j++)
    {
    os_ab << " " << beta[i][j];
    }
  os_ab << endl;
  }
return(0);
}

/*=========================================================================*/

double dl_dt(signed int S, signed int N, float dt_in)
{
//  This subroutine defines the time step suitable for use in
//  the solution process.
//
//  The detailed coding takes into account the special compact layout
//  of the matrices.
//
//% Output variables
double dt;            // output time step, (sec)
//%
//% Input variables
// S;                 // number of elements in ballscrew
// N;                 // number of elements in ballscrew nut
// dt_in;             // input time step, (sec)
//%
//% Other variables
double dtest;         // first estimate of time step, (sec)
```

```
double dt1est;        // first estimate of inner loop time step, (sec)
signed int exp;       // exponent of dtest
signed int i;         // for loop counter
signed int j;         // for loop counter
double log10dt;       // log10(1/dtest)
double man;           // mantissa of dtest
double Mmag;          // "magnitude" of matrix, (1/sec^2)
signed int Msize;     // number of rows in K,C matrices
//%
//% Subroutines used
//%   1. cout, double, signed int, log10, pow, return, sqrt
//%   2. dl_ab
//%
Msize=NFS*(S+1)+NFN*(N+1);
for (i=1;i<=Msize;i++)
  {
  for (j=1;j<=M4;j++)
    {
    K[i][j]=double(K1[i][j]);
    }
  }
dl_ab(S,N,1.);
Mmag=0.;
for (i=1;i<=Msize;i++)
  {
  Mmag=Mmag + double(alpha[i][M4]*alpha[i][M4]);
  }
Mmag=sqrt(Mmag);
dt1est=1./sqrt(Mmag);
dtest=dt1est*double(S);
log10dt=-1.*log10(dtest);
exp=-1*signed int(log10dt)-1;
man=dtest/pow(10.,double(exp));
if ((man<=1.)||(10.<man)) return dt=-1.;
else if ((1.<=man)&&(man<2.)) man=1.;
else if ((2.<=man)&&(man<5.)) man=2.;
else man=5.;
dt=man*pow(10.,exp);
if (dt<double(dt_in)) return dt;
else return dt=double(dt_in);
}

/*=========================================================================*/

void d_b0(signed int Msize, signed int j)
{
//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of d[i][j]
//%
//% Other variables
signed int i;         // for loop counter
//%
for (i=1;i<=Msize;i++)
  {
  b[i][0]=d[i][j];
  }
}

/*=========================================================================*/
```

```
void dl_b1(signed int Msize, signed int j, double dt1)
{
//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of d[i][j]
// dt1;               // time step, inner loop, (sec)
//%
//% Other variables
signed int i;         // for loop counter
//%
//% Subroutines used
//%   long double
//%
for (i=1;i<=Msize;i++)
  {
  b[i][1]=v[i][j]*long double(dt1);
  }
}

/*=========================================================================*/

void dl_b2(signed int S, signed int N, double dt1)
{
//% Input variables
// S;                 // number of elements in ballscrew
// N;                 // number of elements in ballscrew nut
// dt1;               // time step, inner loop, (sec)
//%
//% Other variables
signed int i;         // for loop counter
signed int j;         // for loop counter
signed int j1;        // matrix element counter
signed int Msize;     // number of rows in K,C matrices
//%
//% Subroutines used
//%   long double
//%
Msize=NFS*(S+1)+NFN*(N+1);
for (i=1;i<=NFS*(S+1);i++)
  {
  b[i][2]=0.L;
  for (j=2;j<=2*BWS+BWN-1;j++)
    {
    j1=j-BWN-BWS;
    if ((1<=i+j1)&&(i+j1<=NFS*(S+1)))
      {
      b[i][2]=b[i][2] + (beta [i][j]*b[i+j1][1])/2.L
                      + (alpha[i][j]*b[i+j1][0])/2.L;
      }
    }
  b[i][2]=b[i][2] + (beta [i][2*BWS+2*BWN-1]*b[Msize][1])/2.L
                  + (alpha[i][2*BWS+2*BWN-1]*b[Msize][0])/2.L;
  }

b[Msize][2]=0.L;
for (i=1;i<=NFS*(S+1);i++)
  {
  b[Msize][2]=b[Msize][2] + (beta [i][1]*b[i][1])/2.L
                          + (alpha[i][1]*b[i][0])/2.L;
  }
b[Msize][2]=b[Msize][2] + (beta [Msize][BWN+BWS]*b[Msize][1])/2.L
```

```
                        + (alpha[Msize][BWN+BWS]*b[Msize][0])/2.L;

for (i=1;i<=Msize;i++)
  {
  b[i][2]=b[i][2] + long double(Mi[i]*f[i])/2.L*long double(dt1*dt1);
  }
}

/*=========================================================================*/

void dl_bk(signed int S, signed int N, signed int k)
{
//% Input variables
// S;                  // number of elements in ballscrew
// N;                  // number of elements in ballscrew nut
// k;                  // index in power series
//%
//% Other variables
signed int i;          // for loop counter
signed int j;          // for loop counter
signed int j1;         // matrix element counter
signed int Msize;      // number of rows in K,C matrices
//%
//% Subroutines used
//%   long double
//%
Msize=NFS*(S+1)+NFN*(N+1);
for (i=1;i<=NFS*(S+1);i++)
  {
  b[i][k]=0.L;
  for (j=2;j<=2*BWS+BWN-1;j++)
    {
    j1=j-BWN-BWS;
    if ((1<=i+j1)&&(i+j1<=NFS*(S+1)))
      {
      b[i][k]=b[i][k] + (beta [i][j]*b[i+j1][k-1])/long double(k)
                      + (alpha[i][j]*b[i+j1][k-2])/long double((k-1)*k);
      }
    }
  b[i][k]=b[i][k]
           + (beta [i][2*BWS+2*BWN-1]*b[Msize][k-1])/long double(k)
           + (alpha[i][2*BWS+2*BWN-1]*b[Msize][k-2])/long double((k-1)*k);
  }
b[Msize][k]=0.L;
for (i=1;i<=NFS*(S+1);i++)
  {
  b[Msize][k]=b[Msize][k] + (beta [i][1]*b[i][k-1])/long double(k)
                          + (alpha[i][1]*b[i][k-2])/long double((k-1)*k);
  }
b[Msize][k]=b[Msize][k]
             + (beta[Msize][BWN+BWS]*b[Msize][k-1])/long double(k)
             + (alpha[Msize][BWN+BWS]*b[Msize][k-2])/long double((k-1)*k);
}

/*=========================================================================*/

void d_c0(signed int Msize, signed int j)
{
//% Input variables
// Msize;              // number of rows in K,C matrices
// j;                  // column number of d[i][j]
```

```
//%
//% Other variables
signed int i;         // for loop counter
//%
for (i=1;i<=Msize;i++)
  {
  c[i][0]=v[i][j];
  }
}

/*=========================================================================*/

void dl_c1(signed int Msize, double dt1)
{
//% Input variables
// Msize;             // number of rows in K,C matrices
// dt1;               // time step, inner loop, (sec)
//%
//% Other variables
signed int i;         // for loop counter
//%
//% Subroutines used
//%   long double
//%
for (i=1;i<=Msize;i++)
  {
  c[i][1]=2.*b[i][2]/long double(dt1);
  }
}

/*=========================================================================*/

void dl_ck(signed int S, signed int N, signed int k)
{
//% Input variables
// S;                 // number of elements in ballscrew
// N;                 // number of elements in ballscrew nut
// k;                 // index in power series
//%
//% Other variables
signed int i;         // for loop counter
signed int j;         // for loop counter
signed int j1;        // matrix element counter
signed int Msize;     // number of rows in K,C matrices
//%
//% Subroutines used
//%   long double
//%
Msize=NFS*(S+1)+NFN*(N+1);
for (i=1;i<=NFS*(S+1);i++)
  {
  c[i][k]=0.L;
  for (j=2;j<=2*BWS+BWN-1;j++)
    {
    j1=j-BWN-BWS;
    if ((1<=i+j1)&&(i+j1<=NFS*(S+1)))
      {
      c[i][k]=c[i][k] + (beta [i][j]*c[i+j1][k-1])/long double(k)
                      + (alpha[i][j]*c[i+j1][k-2])/long double((k-1)*k);
      }
  }
```

```
  c[i][k]=c[i][k]
          + (beta [i][2*BWS+2*BWN-1]*c[Msize][k-1])/long double(k)
          + (alpha[i][2*BWS+2*BWN-1]*c[Msize][k-2])/long double((k-1)*k);
  }
c[Msize][k]=0.L;
for (i=1;i<=NFS*(S+1);i++)
  {
  c[Msize][k]=c[Msize][k] + (beta [i][1]*c[i][k-1])/long double(k)
                          + (alpha[i][1]*c[i][k-2])/long double((k-1)*k);
  }
c[Msize][k]=c[Msize][k]
            + (beta [Msize][BWN+BWS]*c[Msize][k-1])/long double(k)
            + (alpha[Msize][BWN+BWS]*c[Msize][k-2])/long double((k-1)*k);
}

/*=========================================================================*/

float d_KE(signed int Msize, signed int j)
{
//% Output variables
float KE;             // kinetic energy, (J)
//%
//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of v[i][j]
//%
//% Other variables
signed int i;         // for loop counter
//%
//% Subroutines used
//%   float, return
//%
KE=0.F;
for (i=1;i<=Msize;i++)
  {
  KE=KE + 0.5F*M[i]*float(v[i][j]*v[i][j]);
  }
return KE;
}

/*=========================================================================*/

float dl_m_d(signed int Msize, signed int j)
{
//% Output variables
float m_d;            // measure of displacement vector at kth term, (m)
//%

//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of d[i][j]
//%
//% Other variables
signed int i;         // for loop counter
//%
//% Subroutines used
//%   double, fabs, float, return
//%
m_d=0.F;
for (i=1;i<=Msize;i++)
  {
```

```
  m_d=m_d + float(fabs(double(d[i][j])));
  }
return m_d=m_d/float(Msize);
}

/*=========================================================================*/

float dl_max_d(signed int Msize, signed int j)
{
//% Output variables
float max_d;          // measure of displacement vector at kth term, (m)
//%
//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of d[i][j]
//%
//% Other variables
float d_ij;           // d[i][j]
signed int i;         // for loop counter
//%
//% Subroutines used
//%   double, fabs, float, return
//%
max_d=0.F;
for (i=1;i<=Msize;i++)
  {
  d_ij=float(fabs(double(d[i][j])));
  if (max_d<d_ij) max_d=d_ij;
  }
return max_d;
}

/*=========================================================================*/

float d_m_extra(signed int Msize, signed int k)
{
//% Output variables
float m_extra;        // measure of last four terms in calculation
                      //  of d vector, (m)
//%
//% Input variables
// Msize;             // number of rows in K,C matrices
// k;                 // index in power series
//%
//% Other variables

float extra[M2];      // additional terms in power series, (m)
signed int i;         // for loop counter
//%
//% Subroutines used
//%         fabs        float       return
//%
for (i=1;i<=Msize;i++)
  {
  extra[i]=float(b[i][k-3] + b[i][k-2] + b[i][k-1] + b[i][k]);
  }
m_extra=0.F;
for (i=1;i<=Msize;i++)
  {
  m_extra=m_extra + float(fabs(extra[i]));
  }
```

```
return m_extra=m_extra/float(Msize);
}

/*=========================================================================*/

double dl_PE2(double bs_rat, double blash, double blf1, double blf2,
              signed int Msize, signed int S, signed int j)
{
//% Output variables
double PE;            // potential energy, (J)
//%
//% Input variables
// bs_rat;            // ballscrew ratio, (m/rad)
// blash;             // backlash, (m)
// blf1;              // backlash factor 1
// blf2;              // backlash factor 2
// Msize;             // number of rows in K,C matrices
// S;                 // number of elements in ballscrew
// j;                 // column number of d[i][j]
//%
//% Other variables
signed int ii;        // for loop counter
signed int i1;        // matrix element subscript
signed int i2;        // matrix element subscript
//%
//% Subroutines used
//%   double, return
//%
PE=0.;
for (ii=1;ii<=S;ii++)
  {
  i1=2*ii-1;
  i2=2*ii;
  PE=PE - K[i1][3]*double((d[i1+NFS][j]-d[i1][j])*(d[i1+NFS][j]-d[i1][j]))/2.
        - K[i2][3]*double((d[i2+NFS][j]-d[i2][j])*(d[i2+NFS][j]-d[i2][j]))/2.;
  PE=PE - K[i1][M4]
          *double((d[Msize][j]-d[i1][j]-bs_rat*d[i2][j]+blf2*blash/2.)
                 *(d[Msize][j]-d[i1][j]-bs_rat*d[i2][j]+blf2*blash/2.))/2.*blf1;
  }
return PE;
}

/*=========================================================================*/

void dl_rEnD(double bs_rat, float c_brg_1, float c_brg_2, signed int Msize,
             signed int S, signed int j)
{
//% Input variables
// bs_rat;            // ballscrew ratio, (m/rad)
// c_brg_1;           // equivalent damping coefficient of bearings,
// c_brg_2;           //  (N m sec/rad)
// Msize;             // number of rows in K,C matrices
// S;                 // number of elements in ballscrew
// j;                 // column number of v[i][j]
//%
//% Other variables
signed int ii;        // for loop counter
signed int i1;        // matrix element subscript
signed int i2;        // matrix element subscript
//%
//% Subroutines used
```

```
//%   double
//%
rEnD[j]=0.;
for (ii=1;ii<=S;ii++)
  {
  i1=2*ii-1;
  i2=2*ii;
  rEnD[j]=rEnD[j]
          - C[i1][3]*double((v[i1+NFS][j]-v[i1][j])*(v[i1+NFS][j]-v[i1][j]))
          - C[i2][3]*double((v[i2+NFS][j]-v[i2][j])*(v[i2+NFS][j]-v[i2][j]));
  rEnD[j]=rEnD[j]
          - C[i1][M4]*double((v[Msize][j]-v[i1][j]-bs_rat*v[i2][j])
                            *(v[Msize][j]-v[i1][j]-bs_rat*v[i2][j]));
  }
rEnD[j]=rEnD[j] + c_brg_1*double(v[2][j]*v[2][j]);
rEnD[j]=rEnD[j] + c_brg_2*double(v[2*(S+1)][j]*v[2*(S+1)][j]);
}

/*============================================================================*/

void inc_d(signed int Msize, signed int j, signed int k)
{
//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of d[i][j]
// k;                 // index in power series
//%
//% Other variables
signed int i;         // for loop counter
//%
if (k==0)
  {
  for (i=1;i<=Msize;i++)
    {
    d[i][j]=b[i][0];
    }
  }
else
  {
  for (i=1;i<=Msize;i++)
    {
    d[i][j]=d[i][j] + b[i][k];
    }
  }
}

/*============================================================================*/

void inc_v(signed int Msize, signed int j, signed int k)
{
//% Input variables
// Msize;             // number of rows in K,C matrices
// j;                 // column number of v[i][j]
// k;                 // index in power series
//%
//% Other variables
signed int i;         // for loop counter
//%
if (k==0)
  {
  for (i=1;i<=Msize;i++)
```

```
    {
    v[i][j]=c[i][0];
    }
  }
else
  {
  for (i=1;i<=Msize;i++)
    {
    v[i][j]=v[i][j] + c[i][k];
    }
  }
}

/*==========================================================================*/

double fric_tab(float m_tab, float g, float mu_tab, double bs_rat, double v_max,
                double blash, double blf2, signed int S, signed int N,
                signed int j, double dt1, signed int mode)
{
//% Output variables
double EnD;           // energy dissipated, (J)
//%
//% Input variables
// m_tab;             // mass of saddle and table, (kg)
// g;                 // acceleration due to gravity, (m/sec^2)
// mu_tab;            // coefficient of friction between table and slideways
// bs_rat;            // ballscrew ratio, (m/rad)
// v_max;             // maximum nut velocity, (m/sec)
// blash;             // backlash, (m)
// blf2;              // backlash factor 2
// S;                 // number of elements in ballscrew
// N;                 // number of elements in ballscrew nut, (0 = 1 node)
// j;                 // column number of v[i][j]
// dt1;               // time step, inner loop, (sec)
// mode               // 0 = "balanced force" mode
//                    // 1 = "damper" mode
//%
//% Other variables
double a_max=6.;      // maximum acceleration, (m/sec^2)
double eps=1.e-5;     // criterion for "zero" velocity
float fforce;         // friction force, (N)
signed int i;         // for loop counter
signed int i1;        // matrix element subscript
signed int i2;        // matrix element subscript
signed int Msize;     // number of rows in K,C matrices
signed int N_int=2;   // number of time inervals used to estimate v_frict
float sforce;         // spring force on table, (N)
float sign;           // sign of table velocity
double vabs;          // absolute velocity of table, (m/sec)
double v_frict;       // velocity of full friction force, (m/sec)
//%
//% Subroutines used
//%         cout        fabs        float       return
//%
Msize=NFS*(S+1)+NFN*(N+1);
v_frict=a_max*dt1*double(N_int);
vabs=fabs(v[Msize][j]);
if (v[Msize][j]/vabs<0.) sign=-1.F;
else sign=1.F;
fforce=mu_tab*m_tab*g;
if (mode==0)
```

```
  {
  if (vabs<=eps*v_max)
    {
    sforce=0.F;
    for (i=1;i<=S+1;i++)
      {
      i1=2*i-1;
      i2=2*i;
      sforce=sforce - float(K[i2][M4]
             *(d[Msize][j]-d[i1][j]-bs_rat*d[i2][j]+blf2*blash/2.));
      }
    if (sforce<-1.F*fforce)
      {
      f[Msize]=f[Msize] + fforce;
      }
    else if (fforce<sforce)
      {
      f[Msize]=f[Msize] - fforce;
      }
    else
      {
      f[Msize]=f[Msize] - sforce;
      }
    }
  else
    {
    f[Msize]=f[Msize] - sign*fforce;
    }
  EnD=-1.*(float(f[Msize] + K[Msize][1]*(blf2*blash/2.)))*v[Msize][j]*dt1;
  }
else if (mode==1)
  {
  v_frict=a_max*dt1*double(N_int);
  if (vabs<v_frict)
    {
    f[Msize]=f[Msize] - sign*float(vabs/v_frict)*fforce;
    }
  else
    {
    f[Msize]=f[Msize] - sign*fforce;
    }
  EnD=-1.*(float(f[Msize] + K[Msize][1]*(blf2*blash/2.)))*v[Msize][j]*dt1;
  }
else
  {
  EnD=0.;
  }
return EnD;
}
```

Appendix 5.2 - **Details of calculations used to check Program T3**

A steel ballscrew 38.1 mm in diameter and 1520 mm long was modelled using program T3.cpp. The screw was supported at the drive end with a bearing of 8×10^8 N/m axial stiffness at the tail end with a bearing of 15.6×10^8 N/m axial stiffness. A simple PID controller was coded into the model to enable a trapezoidal velocity profile to be run. This controller had a proportional constant K_p of 2 N m/rad, an integral constant K_i of 100 N m/(rad sec) and a differential constant K_d of zero. Velocity feedback came from the nut. The demand signal is the diagonal straight line on Figure 5.13.

At the start of the motion the nut was positioned at 100 mm from the driven end in the centre of 10 μm of backlash. The first part of the motion entailed an acceleration of the nut of 0.2 m/sec^2. In order to avoid the results being swamped by the friction between the table and the slideways, the coefficient of friction between them was assumed to be zero. A load of 359 kg was carried by the nut.

The hand calculations used to check the results of "C" program T3 are detailed as follows:-

A5.2.1 **Axial deflection**

The axial stiffness from the point of contact with the nut to "earth" was calculated by taking into account the stiffness of the several components (TK model BS_KAX.TK, rules 1-13). The axial force on the screw was derived by considering the force needed to accelerate the nut load (*ibid*, rule 14). The deflection at the nut contact point and at the bearings was then computed by applying the load to the component stiffnesses (*ibid*, rules 15-19).

MiniTK model BS_KAX.TK

```
Variables
# 1 pi      3.141592653589793              pi
# 2 E       2.07E11              N/m^2     Young's modulus
# 3
# 4 d       .0381                m         screw diameter
# 5 l_bs    1.52                 m         ballscrew length
# 6 x1      .101                 m         position of nut
# 7 A       .00114009182796937   m^2       cross sectional area
# 8
# 9 k_brg_1 8E8                  N/m       bearing 1 stiffness
#10 k_brg_2 1.56E9               N/m       bearing 2 stiffness
#11 k_bs_1  2336623845.442174    N/m       ballscrew stiffness 1
#12 k_bs_2  166313607.0399292    N/m       ballscrew stiffness 2
#13 k1      595958957.2941673    N/m       stiffness - drive end
#14 k2      150290900.7518983    N/m       stiffness - tail end
#15 kax     746249858.0460656    N/m       total axial stiffness
#16
```

```
#17 fl_bs_1 4.279678999042158E-10 m/N       ballscrew flexibility 1
#18 fl_bs_2 6.012737128357248E-9  m/N       ballscrew flexibility 2
#19 fl_brg1 1.25E-9               m/N       bearing 1 flexibility
#20 fl_brg2 6.41025641025641E-10  m/N       bearing 2 flexibility
#21 fl1     1.677967899904216E-9  m/N       flexibility - drive end
#22 fl2     6.653762769382889E-9  m/N       flexibility - tail end
#23
#24 m       359                   kg        mass of table
#25 a       .2                    m/sec^2   acceleration of table
#26
#27 F       71.8                  N         force on table
#28 F1      57.33984760246322     N         force on bearing 1
#29 F2      14.46015239753679     N         force on bearing 2
#
#30 dx1     7.167480950307903E-8  m         deflection in bearing 1
#31 dx2     9.269328459959478E-9  m         deflection in bearing 2
#32 dxn     9.621442366233298E-8  m         deflection at nut location

Rules (equations)
# 1 pi=4.*atan(1.)
# 2 A=pi/4.*d^2
# 3 k_bs_1=E*A/x1
# 4 fl_bs_1=1./k_bs_1
# 5 fl_brg1=1./k_brg_1
# 6 fl1=fl_bs_1+fl_brg1
# 7 k1=1./fl1
# 8 k_bs_2=E*A/(l_bs-x1)
# 9 fl_bs_2=1./k_bs_2
#10 fl_brg2=1./k_brg_2
#11 fl2=fl_bs_2+fl_brg2
#12 k2=1./fl2
#13 kax=k1+k2
#14 F=m*a
#15 dxn=F/kax
#16 F1=k1/kax*F
#17 dx1=F1/k_brg_1
#18 F2=k2/kax*F
#19 dx2=F2/k_brg_2
```

The axial deflections at the other points plotted on Figure 5.14 were derived by interpolation as follows:-

$$x_{1/4} = \frac{9.2693\times10^{-9} - 9.6214\times10^{-8}}{1.520 - 0.101} \times (1.520 \times \frac{1}{4} - 0.101) + 9.6214\times10^{-8}$$
$$= 7.9119\times10^{-8}\ m$$
$$x_{1/2} = \frac{9.2693\times10^{-9} - 9.6214\times10^{-8}}{1.520 - 0.101} \times (1.520 \times \frac{1}{2} - 0.101) + 9.6214\times10^{-8}$$
$$= 5.5836\times10^{-8}\ m$$
$$x_{3/4} = \frac{9.2693\times10^{-9} - 9.6214\times10^{-8}}{1.520 - 0.101} \times (1.520 \times \frac{3}{4} - 0.101) + 9.6214\times10^{-8}$$
$$= 3.2816\times10^{-8}\ m$$

A5.2.2 **Driving torque**

The torque needed to drive the screw is the sum of the torque required to accelerate the nut load (TK model BS_RIG_T.TK, rules 1 and 3-5), that to accelerate the screw (*ibid*, rules 6-8) and those to overcome viscous drag in the bearings (*ibid*, rules 9-13).

MiniTK model BS_RIG_T.TK

```
Variables
# 1 pi      3.141592653589793              pi
# 2 E       2.07E11              N/m^2     Young's modulus
# 3 rho     7600                 kg/m^3    density
# 4
# 5 d       .0381                m         screw diameter
# 6 l_bs    1.52                 m         ballscrew length
# 7 x1      .101                 m         position of nut
# 8 A       .00114009182796937   m^2       cross sectional area
# 9
#10 c_brg_1 .000668              N m s/rad bearing 1 damping coefficient
#11 c_brg_2 .000668              N m s/rad bearing 2 damping coefficient
#12
#13 m       359                  kg        mass of table
#14 J       .002389774800487604  kg m^2    inertia of ballscrew
#15
#16 Ns      1                              number of starts of screw
#17 p       .01                  m         pitch of screw
#18 R       .001591549430918954  m/rad     ballscrew ratio
#19
#20 t       .05                  sec       time
#21 a       .2                   m/sec^2   acceleration of table
#22 angacc   125.6637061435917   rad/sec^2 angular acceleration of ballscrew
#23 v        .01                 m/sec     velocity of table
#24 w        6.283185307179587   rad/sec   speed of screw
#25
#26 F        71.8                N         force on table
#27 Tm       .1142732491399809   N m       torque to accelerate screw
#28 Tbs      .3003079582778347   N m       torque to accelerate ballscrew
#29 T_brg_1  .004197167785195964 N m       viscous drag in bearing 1
#30 T_brg_2  .004197167785195964 N m       viscous drag in bearing 2
#31 T        .4229755429882076   N m       total torque

Rules (equations)
# 1 pi=4.*atan(1.)
# 2 A=pi/4.*d^2
# 3 R=Ns*p/(2.*pi)
# 4 F=m*a
# 5 Tm=F*R
# 6 J=pi/32.*d^4*l_bs*rho
# 7 angacc=a/R
# 8 Tbs=J*angacc
# 9 v=a*t
#10 w=v/R
#11 T_brg_1=c_brg_1*w
#12 T_brg_2=c_brg_2*w
#13 T=T_brg_1+Tbs+Tm+T_brg_2
```

A5.2.3 **Nut position error**

The nut position error was calculated using the following data (as well as that in Sections A5.2.1 and A5.2.2 of this appendix).

```
double blash=10.e-6;  // backlash, (m)
float G=80.0e9F;      // shear modulus, (N/m^2)
float k_ax_nut=1.e8F; // axial stiffness of nut, (N/m)
float Kd=0.F;         // controller differential constant, (N m sec/rad)
float Ki=100.F;       // controller integral constant, (N m/(rad sec))
float Kp=2.F;         // controller proportional constant, (N m/rad)
```

A5.2.3.1 **Axial movement of screw/nut contact point**

See Section 1

A5.2.3.2 **Axial deflection within nut**

Based on force needed to accelerate nut (see Section 1) applied to nut stiffness

$$\delta_{k\_ax\_nut} = \frac{71.8}{1.\times 10^{8}} = 0.718\times 10^{-6}\ m$$

A5.2.3.3 **Torsional deflection of screw**

Based on total torque (see Section 2) applied to portion of screw between drive end and screw/nut contact point

$$\theta_{tors} \approx \frac{0.42298}{80\times 10^{9}\times \frac{\pi}{32}\times 0.0381^{4} \Big/ 0.101} = 2.5814\times 10^{-6}\ rad$$

$$\delta_{tors} = 2.5184\times 10^{-6}\times 0.0015915 = 4.1084\times 10^{-9}\ m$$

A5.2.3.4 **Backlash**

$$\delta_{blash} = \frac{10.\times 10^{-6}}{2} = 5.\times 10^{-6}\ m$$

A5.2.3.5 **Controller follower error**

Based on the integral of the velocity error (i.e. position error) needed to generate the torque required to drive the system.

$$\theta_{P\_error} \approx \frac{0.42298}{100} = 4.2298\times 10^{-3}\ rad$$

$$\delta_{P\_error} = 4.2298\times 10^{-3}\times 0.0015915 = 6.7319\times 10^{-6}\ m$$

Appendix 6.1 – **C program for 6 degree-of-freedom mechanical model**

This program generates the time history of ballscrew mechanical behaviour. Various routines written in MATLAB are used to produce graphical output.

```
// Program "LG_rig.cpp"
//         ============
//
// Standard function header files

// Basic solution parameters defined

// Matrix and array sizes
// Subroutine prototypes
//%
//% Output variables allocated static arrays including:-
// "x" array for plotting graphs, (m), (output as "float")
// time, (sec)
// linear "x", "y" and "z" displacement and velocity time histories, (m, m/sec)
// torsional "θx" displacement and velocity time histories, (rad, rad/sec)
// tilt "θy" and "θz" displacement and velocity time histories, (rad, rad/sec)
// kinetic and potential energy, energy input, energy dissipated and total
// energy, (J)
//%
//% Common vectors and matrices allocated static arrays including:-
// mass, stiffness and damping matrices, force vectors
// time series coefficients, and ”alpha” and “beta” matrices
// short-term displacement, velocity and acceleration matrices
// controller error and its integral, and position error of nut
//%
//% Common data block
//% ================
// Defined or allocated as static variables are:-
//% Fundamental constants – g and pi
//% Material properties
//% Dimensional and inertia data etc. relating to motor, drive coupling,
ballscrew and nut
//% Stiffness data and damping data relating to the bearings and slideways
//% Common matrix data
//%
//% Controller constants defined and variables are allocated as static variables
//%
int main(void)
{
// Main program
//% Input variables defined including:-
// number of elements in ballscrew and nut
// node numbers of bearings
// details of demand path, position and velocity
// backlash, (m)
// pre-tension, (m/m)
// applied out-of-balance data, (kg m)
//%
//% Other variables used by program allocated

//% Open output streams for the output variables
if (timp==1)
      {
```

```
      // Opens input streams in white noise mode
      }

// Initialisation section
// ======================
// Defines ballscrew "x" axis and element lengths

// Defines mass of system

// Initialises demand state of nut
// Initialises solution process working arrays and output arrays

// Sets the angular position of the ballscrew

// Sets up out-of-balance excitation in in-phase/quadrature form
// Initialises saved motor torque
// Initialises energy put into system

// Sets up “static” stiffness matrix

// Sets time step to ensure convergence of solution series

// Updates the stiffness and damping matrices for initial position of nut

if (timp==1)
      {
// Sets number of iterations for white noise mode
      }
else
      {
// Initialises trapezoidal demand profile
      }

// Initialises block counter and time

// Initialises controller

// Time series solution section
// ============================
for (it=1;it<=Nt;it++) // Start main loop
      {
      // Controller updates motor torque
      for (j=1;j<=S;j++) // Start inner loop
            {
            // Demand position, velocity and acceleration updated

            // Defines a representative position for ballscrew

            // Zeros a force vector for use as a "saved" one and zeros the
energy dissipation arrays

            // Updates the stiffness and damping matrices

            // Calculates the energy dissipation rates due to various sources of
            // friction, and applies the relevant forces
            // - motor bearings
            // - ballscrew bearing 1
            // - ballscrew bearing 2
            // - saddle/table
```

```
            // Updates forces due to dead weight, pre-tension and out-of-balance

            // Updates "alpha" and "beta" matrices

            // Defines first three terms of solution array "b" and builds them
into
            // the displacement array "d", the velocity array "v" and the
acceleration array "a"

            // Defines the remainder of the terms of solution array "b" and
builds them into
            // the arrays "d", "v" and "a"

            // Calulates the rate of energy dissipation by the viscous
            // damping elements

            // Adds in energy input due to motor torque
            // Adds in energy input due to movement of internal forces due
            // to pre-tension and external forces due to dead weight
            } // End of inner loop

      // Puts results of analysis into arrays for output

      // Resets state variables for next inner loop
      // Puts last energy dissipation values into first position
      // ready for next inner loop

// Ouput section
// =============
      // Write results to file

      } // End of main loop

// Close files
return(0);
}
```

Typical data set

```
// Basic solution parameters
#define MDR 2                // Number of degrees of freedom in motor drive
                                 (in addition to ballscrew)
#define SMAX 50              // Maximum number of elements in ballscrew
#define NFS 6                // Number of degrees of freedom per ballscrew node
#define BWS 12               // Bandwidth of ballscrew
#define NMAX 0               // Maximum number of elements in ballscrew nut,
                                 (0 = 1 node)
#define NFN 5                // Number of degrees of freedom per nut node
#define BWN 5                // Bandwidth of nut
#define KMAX 42              // Maximum number of iterations
#define NBLOCK 1000          // Maximum number of steps in time history block
#define SWRIT 100            // Number of time history steps between screen
                                 writes
#define TINY 1.0e-20         // Used in subroutine "ludcmp"
#define NSOURCE 8            // Number of heat sources
```

```
//% Common data block
//% =================
//% Fundamental constants
static float g=9.80665F;     // acceleration due to gravity, (m/sec^2)
//%
//% Material properties
static float E=207.0e9F;     // Young's modulus, (N/m^2)
static float G=80.0e9F;      // shear modulus, (N/m^2)
static float rho=7860.F;     // density, (kg/m^3)
//%
//% Dimensional data etc.
//% Motor
//% -----
static float J_mot=0.0030F;  // drive motor inertia, (kg m^2)
static float J_ms=0.0000075F; // drive motor shaft inertia, (kg m^2)
static float T_mot_max=42.F; // maximum motor torque, (N m)
static float n_mot_max=4700.F; // maximum rated motor speed, (rpm)
static float i_mot_max=41.0F; // peak motor current, (A)
//% Drive coupling
//% --------------
static float J_dr1=194.2e-6F; // driving pulley,
                                 coupling half etc. inertia, (kg m^2)
static float m_dr2=0.310F;   // driven pulley, coupling half etc. mass, (kg)
static float J_dr2=194.2e-6F; // driven pulley,
                                coupling half etc. rotational inertia, (kg m^2)
static float Jt_dr2=143.8e-6F; // driven pulley,
                                coupling half etc. transverse inertia, (kg m^2)
static long double u=1.L;    // drive ratio
//% Ballscrew
//% ---------
static float d_bsO=37.5e-3F; // outer diameter of ballscrew, (m)
static float d_bsI=10.0e-3F; // inner diameter of ballscrew, (m)
static float dp_bs=42.0e-3F; // pitch diameter of ballscrew, (m)
static float l_bs1=0.109F;   // length of ballscrew pulley to bearing, (m)
static float l_bs2=1.183F;   // length of ballscrew between bearings, (m)
static float l_bs3=0.052F;   // length of ballscrew between bearing to end, (m)
static signed int Ns=1;      // number of starts of ballscrew grooves
static float pitch=16.0e-3F; // ballscrew pitch, (m)
static signed int SignB=+1;  // hand of helix, -1 = left hand screw
                             //                +1 = right hand screw
static float mu_brg=0.01F;   // nominal coefficient of friction of bearings
static float d_trace=0.045F; // diameter of bearing thrust race, (m)
//% Nut
//% ---
static float mu_scr=0.019F;  // nominal coefficient of friction of screw nut
static float m_tab=352.F;    // mass of saddle/table, (kg)
static float Jy_tab=0.546F;  // inertia of saddle/table about "y" axis, (kg m^2)
static float Jz_tab=120.F;   // inertia of saddle/table about "z" axis, (kg m^2)
static float z_tab=125.e-3F; // height of saddle/table above ballscrew, (m)
static float z_slid=75.e-3F; // height of saddle/table above slideway, (m)
static float mu_tab=0.0341F; // coefficient of friction between saddle and
                                slideways
static double Absnf=1.44;    // constant term in "Mode 0" ballscrew friction
                                model, (N m)
static double Bbsnf=0.;      // speed-dependent term in "Mode 0" ballscrew
                                friction model, (N m sec/rad) //%
```

```
//% Monitoring ring
//% ---------------
static float m_Mring=1.029F; // mass of monitoring ring, (kg)
static float Jtor_Mring=0.001289F; // torsional inertia of monitoring ring,
                                      (kg m^2)
static float Jtil_Mring=0.000701F; // tilt inertia of monitoring ring, (kg m^2)
static float rOoB=40.e-3F;   // radius of out-of-balance weight, (m)
//%
//% Stiffness data
static float k_ms=5466.F;    // torsional stiffness of motor shaft, (N m/rad)
static float k_dr=10100.F;   // torsional stiffness of drive referred to driving
                                shaft, (N m/rad)
static float k_ax_1=850.e6F; // axial stiffness of bearings, (N/m)
static float k_ax_2=850.e6F;
static float ky_rad_1=1275.e6F; // radial stiffness of bearings
                                    - "y" direction, (N/m)
static float ky_rad_2=1275.e6F;
static float kz_rad_1=1275.e6F; // radial stiffness of bearings
                                    - "z" direction, (N/m)
static float kz_rad_2=1275.e6F;
static float ky_tilt_1=160.e3F; // tilt stiffness of bearings
                                    - "Oy" direction, (N m/rad)
static float ky_tilt_2=160.e3F;
static float kz_tilt_1=160.e3F; // tilt stiffness of bearings
                                    - "Oz" direction, (N m/rad)
static float kz_tilt_2=160.e3F;
static float k_ax_nut=1.2e9F; // axial stiffness of nut, (N/m)
static float ky_nut=1.2e9F;  // radial stiffness of nut - "y" direction, (N/m)
static float kz_nut=1.2e9F;  // radial stiffness of nut - "z" direction, (N/m)
static float kOy_nut=0.7e5F; // tilt stiffness of nut
                                - "Oy" direction, (N m/rad)
static float kOz_nut=0.7e5F; // tilt stiffness of nut
                                - "Oz" direction, (N m/rad)
static float ky_slide=1.e9F; // stiffness of slides supporting nut
                                - "y" direction, (N/m)
static float kz_slide=1.e9F; // stiffness of slides supporting nut
                                - "z" direction, (N/m)
static float kOy_slide=10.4e6F; // stiffness of slides supporting nut
                                - "Oy" direction, (N m/rad)
static float kOz_slide=10.4e6F; // stiffness of slides supporting nut
                                - "Oz" direction, (N m/rad)
//%
//% Damping data
static float c_brg_m=0.656e-3F; // equivalent damping coefficient of bearings,
                                   (N m sec/rad)
static float c_ms=0.01F;   // torsional damping coefficient of motor shaft,
                                   (N m sec/rad)
static float c_dr=0.797F;   // torsional damping coefficient of drive
                                referred to driving shaft, (N m sec/rad)
static float c_brg_1=0.668e-3F; // equivalent damping coefficient of bearings,
                                   (N m sec/rad)
static float c_brg_2=0.668e-3F;
static float drb=0.0005F;    // damping ratio - bearings
static float drs=0.00002F;   // damping ratio - shaft
static float drn=0.0005F;    // damping ratio - nut
static float drsl=0.0005F;   // damping ratio - slide
// Damping reference -
http://www.mom.gov.sg/MOM/OHD/Others/N&VGUIDELINES04(chap6).pdf. p84 - damping
ratio for spring steel = 0.005
//%
```

```
//% Controller constants
static double K_v=4.*1000./60.; // position controller constant = 4 (m/min)/mm,
                                   (1/sec)
static double K_k1=1./double(pitch); // constant, (1/m)
static double K_vff=K_k1;    // gain - velocity feed forward, (1/m)
static double K_vel;         // constant =1./(2.*pi), (rev/rad)
static double K_aff=0.039;   // gain - acceleration feed forward, (A sec^2/rev)
static double K_p=26.;       // proportional gain - velocity control,
                                (A sec/rev)
static double K_i=6300.;     // integral gain - velocity control, (A/rev)
static double LA=0.0062;     // motor armature inductance, (H)
static double RA=0.68;       // motor armature resistance, (ohm)
static double K_cp=50.;      // proportional gain - current control, (V/A)
static double lamda=0.7639;  // flux, (V sec/rad)
static signed int N_poles=4; // number of pairs of motor poles
static double Kt=1.28F;      // motor torque time constant, (N m/A)
//%
//% Input variables
signed int S=25;      // number of elements in ballscrew
signed int N=0;       // number of elements in ballscrew nut, (0 = 1 node)
signed int B1=1;      // node number at Bearing 1, (node number of first node in
ballscrew = 0)
signed int B2=24;     // node number at Bearing 2
double xs=1073.e-3F;  // "x" coordinate of start point, (m)
double xf=xs-480.e-3F; // "x" coordinate of end of deceleration, (m)
double v_max=40./60.; // maximum nut velocity, (m/sec)
double t1_=0.;        // time of start of acceleration, (sec)
double tacc=1./9.;    // acceleration time, (sec)
double tdec=1./9.;    // deceleration time, (sec)
float blash=0.F;      // backlash, (m)
float PT=0.e-6F;      // pre-tension, (m/m)
float xMR1=l_bs1+133.e-3F; // “x” coordinate of monitoring ring 1
float xMR2=l_bs1+276.e-3F; // “x” coordinate of monitoring ring 2
mOoB1=0.F;            // mass of out-of-balance weight 1, (kg)
phOoB1=0.F;           // phase of out-of-balance weight 1, (deg)
mOoB2=124.e-3F;       // mass of out-of-balance weight 2, (kg)
phOoB2=0.F;           // phase of out-of-balance weight 2, (deg)
float dt_in=50.e-6F;  // input time step, (sec)
plst=3.0e-3;          // controller position loop up-date time, (sec)
vlst=0.6e-3;          // controller velocity loop up-date time, (sec)
ilst=0.2e-3;          // controller current loop up-date time, (sec)
// "Mode 0" bearing friction model
//  constant term   speed-dependent
    Abrg[0]=0.5453; Bbrg[0]=0.001208; // motor bearing
    Abrg[1]=0.5;    Bbrg[1]=0.;       // ballscrew bearing 1
    Abrg[2]=0.4;    Bbrg[2]=0.;       // ballscrew bearing 2
//   (N m)           (N m sec/rad)
```

Appendix 7.1 – **Water cooling the ballscrew**

Consider a ballscrew of the size used on the Linear Guide rig. Let it be furnished with four cooling passages of Ø5 mm each, arranged so that the cooling water (chilled coolant) makes four passes of the ballscrew before draining into the sump of the machine. Applying equations (7.23) and (7.30) to a coolant flow supplied at 10 C against a ballscrew running at 20ºC gives the estimated cooling performance for different rates of flow as shown in Figure A7.1:-

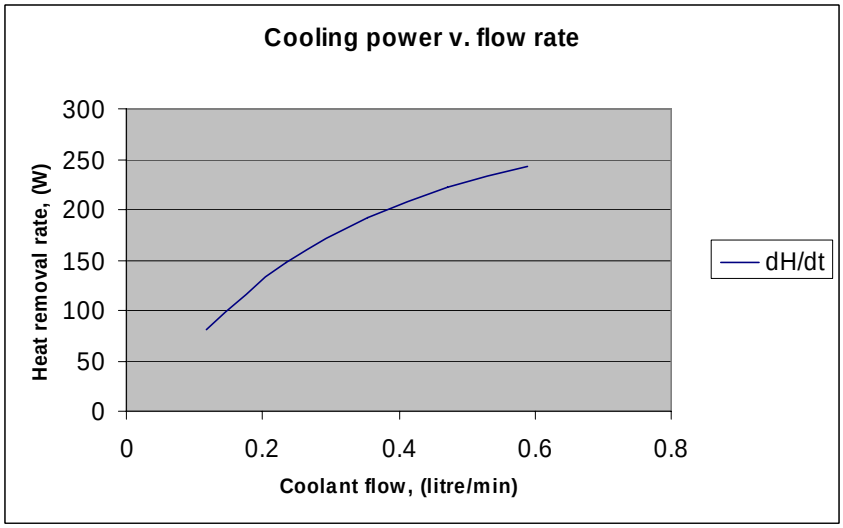


Figure A7.1 – *Heat removal rate v. coolant flow for a water cooled ballscrew*

Fixing the flow rate at 0.5 l/min and 1 l/min, gives the estimated performance with varying temperature as shown in Figure A7.2:-

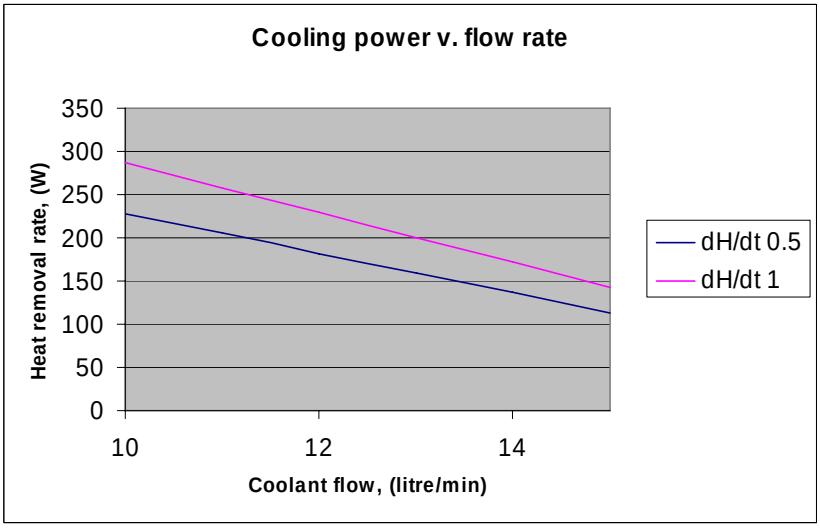


Figure A7.2 – *Heat removal rate v. coolant temperature a water cooled ballscrew*

Appendix 7.2 **– Matlab model "cool2.m"**

```
function[x,T,RH]=cool2(T_in,p_res,mw,N)
%
% Output - x     = proportion of mixture for plots, [%]
%          T     = temperature after evaporation, [deg C]
%          RH    = relative humidity after evaporation
%
% Input  - T_in  = input temperature, [deg C]
%          p_res = pressure of compressed air reservoir, [bar]
%          mw    = mass of water, [kg] (for 1 m^3 at 1 bar at T_in)
%          N     = number of increments
%
Cp_air=1.00;     % specific heat of air at constant pressure, [kJ/kg K]
rho_air_0=1.293; % density of air at 0 deg C, [kg/m^3]

T1     =[  0.01,  5.  , 10.  , 15.  , 20.  , 25.  , 30.  , 35.  , 40.  ];
          % temperature, [deg C]
Cpl_dat=[ 4.210, 4.204, 4.193, 4.186, 4.183, 4.181, 4.179, 4.178, 4.179];
          % specific heat of water, [kJ/kg K]
Cpv_dat=[ 1.86 , 1.86 , 1.86 , 1.87 , 1.87 , 1.88 , 1.88 , 1.88 , 1.89 ];
          % specific heat of steam at constant pressure, [kJ/kg K]
hlv_dat=[2500.8,2488.9,2477.2,2465.5,2453.7,2441.8,2430.0,2418.2,2406.2];
          % latent heat, water to steam, [kJ/kg]
T2     =[  0.01,  7.0 , 13.0 , 17.5 , 21.1 , 24.1 , 29.0 , 34.6 , 40.3 ];
          % temperature, [deg C]
vg_dat =[206.1 ,129.2 , 87.98, 67.01, 54.26, 45.67, 34.80, 25.77, 19.24];
          % specific volume steam, [m^3/kg]
pv_dat =[0.0061,0.010 ,0.015 ,0.020 ,0.025 ,0.030 ,0.040 ,0.055 ,0.075 ];
          % saturated vapour pressure water, [bar]

% For 1 m^3 of air saturated at 'T_in' deg C and 'p_res' bar pressure then expanded
% isothermally to 1 bar:-
RH(1)=1./p_res;             % relative humidity
pv=interp1(T2,pv_dat,T_in); % vapour pressure at T_in deg C, [bar]
m_air=(1.-RH(1)*pv)*rho_air_0*(273.15/(T_in+273.15)); % mass of air, [kg]
vg=interp1(T2,vg_dat,T_in); % specific volume at T_in deg C, [m^3/kg]
m_vap=RH/vg;                % mass of water vapour, [kg]

% Now add mw kg of water dm kg at a time
dm=mw/N;                        % mass of increment of water, [kg]
T(1)=T_in;                      % temperature set to initial value, [deg C]
x(1)=0;                         % number of mass increments set to zero
for i=1:N
   pv=interp1(T2,pv_dat,T(i));   % vapour pressure at T deg C, [bar]
   vg=interp1(T2,vg_dat,T(i));   % specific volume at T deg C, [m^3/kg]
   m_vap_max=1./vg;              % mass of water vapour at saturation, [kg]
   hlv=interp1(T1,hlv_dat,T(i)); % latent heat at T deg C, [kJ/kg]
   Cpv=interp1(T1,Cpv_dat,T(i)); % specific heat of water vapour at T deg C, [kJ/kg K]
   Tavl=(T_in+T(i))/2.;          % average temperature of water sprayed in, [deg C]
   Cpl=interp1(T1,Cpl_dat,Tavl); % specific heat of water at Tavl deg C, [kJ/kg K]
   dQlv=hlv*dm;                  % heat absorbed by evaporation, [kJ]
   dQl=Cpl*dm*(T_in-T(i));       % heat given out by cooling water in spray, [kJ]
   m_vap=m_vap+dm;               % mass of water vapour after evaporation of dm, [kg]
   C=Cp_air*m_air + Cpv*m_vap;   % thermal capacity of mixture, [kJ/K]
   dT=(-1.)*(dQlv-dQl)/C;        % temperature change, [K]
   T(i+1)=T(i)+dT;               % temperature after evaporation of dm, [deg C]
   RH(i+1)=m_vap/m_vap_max;      % relative humidity after evaporation of dm
   x(i+1)=i;                     % update number of mass increments
end
x=x/N*100.; % convert mass increments to proportion of total mass added
```

Appendix 7.3 – **The ballscrew thermal model**

```
function [time,z1,z2,z3,z4]=bsmodel(heatTime,coolTime,TI,xx)

% BSMODEL: My model with nut and bearing temperature modelled.
%
% Use: [time,z1,z2,z3,z4]=bsmodel(heatTime,coolTime,TI,xx)
% Set xx = 1 to suppress plots

% Output variables
%  time(,)     time, ['nt' × 1], (sec?)
%  z1(,)       screw element temperature, ['nt' × nss], (degC)
%  z2(,)       temperature of nut and bearings, ['nt' × 3], (degC)
%  z3(,)       total error? less its starting value, ['nt' × 2], (micron)
%  z4(,)       column 1, difference in total error of bearings,
%              column 2, difference in thermal error of bearings,
              ['nt' × bhs], (W)
%
% Input variables
%  heatTime    time taken running ballscrew round a trapezoidal velocity
%              load cycle, (min), (redefined later to a whole number of runs)
%  coolTime    time taken in cooling after loading, (min)
%  TI          time interval ?, (sec?)
%  xx          supress plots (1), plot (otherwise)
%
% Other variables
%  Abf         bearing mounting face area, (m^2)
%  AbSection   area of bearing transverse cross section, (m^2)
%  AbSurface   surface area of outer half bearing, (m^2)
%  Ah()        bearing housing surface area, [1 × 2], (m^2), rough (not used)
%  Ahs()       surface area of housing section, [1 × 2], (m^2)
%  Ahsf()      area of housing FE face, [1 × 2], (m^2)
%  AnfSurface  surface area of nut flange, (m^2)
%  AnSection   area of nut section, (m^2)
%  AnSurface   nut surface area, (m^2)
%  AnhSection  nut housing cross section area, (m^2)
%  AnhSurface  nut housing element surface area, (m^2)
%  AnsbContact estimated surface contact between nut and screw via oil film,
              (m^2?)
%  AnutMount   nut mounting flange area, (m^2)
%  AsSection   cross-sectional area of screw, (m^2)
%  AsSurface   outer surface area of screw FE, (m^2)
%  AsVolume    volume of screw FE, (m^3), (not used)
%  AtblMount   nut housing to table mounting surface contact area, (m^2)
%  AtblSection area of table cross section, (m^2)
%  AtblSurface surface area of table element, (m^2)
%  A1          area of a surface 38 mm diameter and length of nut, (m^2)
%  A2          area of a surface 44 mm diameter and length of nut, (m^2)
%
%  Ballnum     number of balls in nut, (not used)
%  B1ss        index of screw element under bearing 1, [1 × 2]
%  B2ss        index of screw element under bearing 2, [1 × 2]
%  b           for loop counter on bearing housings
%  barh        handle to waitbar
%  bhs         number of bearing housing elements
%  brgArrange  '0'=PT and axial, '1'=PT no axial, '2'=free-ended
```

```
%  brgDis      distance between bearings, (m)
%
%  cd          current directory as a string, MATLAB special function
%  cSteel      specific heat capacity of steel, (J/kg/K),
%              (approx. from various sources)
%
%  DensitySteel
%              density of steel, (kg/m^3)
%  Dist        traverse distance, (m)
%  DnInner     nut inner diameter, (m)
%  DnOuter     nut outer diameter, (m)
%  DnutFlange  diameter of nut flange, (m)
%  dv()        'average' velocity, d'y()'/d'x()', [? × 1], (m/sec)
%
%  E_err()     error due to strain at 'y = 0.' and 'y = -460.',
               [1 × 2], (micron)
%  ErrPos1ss() front of position 1
%  ErrPos2ss() front of position 2
%  Esteel      Young's modulus of steel, (kN/m^2)
%  Exp_change  change in thermal expansion of shaft, (micron?)
%  ExpCoeff    thermal expansion coefficient of shaft
               (AISI 8620 not exactly known)
%  ePaint      surface painted white emissivity, (not used)
%  eSteel      steel emissivity
%  expansion   total expansion of shaft, (micron?)
%  extra(,)    column 1, difference in total error of bearings,
%              column 2, difference in thermal error of bearings,
               ['nt' × bhs], (W)
%
%  F           Feedrate (m/min) CHECK UNITS!
%  F_change    change in axial force in shaft, (kN)
%  F_current   current axial force in shaft, (kN)
%  F_start     axial force in shaft at start, (kN)
%  f           interchange factor for radiation, (not used)
%  filename    'det-' ? which file ?
%  froot       current directory as a string
%
%  Hbhs(,)     heat flow - bearing housing elements, [2 × bhs], (W)
%  Hc          free convective heat transfer coefficient, (W/m^2.K) (2-25)
%  Hcf         forced convection by screw rotation/axis traverse,
               (25-250 W/m^2.K)
%  Hnhs()      heat flow - nut housing elements, [nhs], (W)
%  Hss()       heat flow - screw elements, [nss], (W)
%  Htbl()      heat flow - table elements, [nts], (W)
%  hs          for loop counter on housing elements
%  h1 - h12    handles for graphs
%
%  I           friction torque ?, (Nm?)
%  Ia          friction torque ? due to ?, (Nm?)
%  IDb         bearing face inner diameter, (m)
%  IRb         bearing inner radius, (m)
%  Im          friction torque ? due to motor, (Nm?)
%  Ims         screw shaft moment of inertia, (kg m^2), (not used)
%  i           for loop counter on time
%
%  j           counter, but is it used for anything?
```

```
%  K           coefficient of conductivity of steel, (W/m.K)
              (45 for nut) (250 from b3??)
%  Kbrg()      axial (?) stiffness of front and rear bearing assemblies,
               [1 × 2], (kN/m?)
%  KbrgJoint   coeff. of conduction across bearing mounting joint, (W/m^2.K),
%               good surface & pressure
%  KbrgRace    coeff. of conduction across balls between races, (W/K)
%  KbrgShaft   coeff. of conduction across bearing/shaft joint, (W/m^2.K),
               interferance fit.
%  KnhJoint    coeff. of conduction across nut/nut housing joint, (W/m^2.K)
%  KnutShaft   coeff. of conduction nut to shaft based on
               test 'b3nutc1' data, (W/m^2.K)
%  KtblJoint   coeff. of conduction across nut housing/table joint, (W/m^2.K)
%  k           time interval counter
%
%  Lb          bearing length, (m)
%  Lhs()       length of each housing FE (conduction calc.), [1 × 2], (m)
%  Ln          length of nut, (m)
%  Lnc         length of nut in contact with shaft, (m)
%  Lnh         length of nut housing, (m)
%  Lnhs        length of nut housing element, (m)
%  LnutFlange  length of nut flange, (m)
%  Lprebrg1    length of screw shaft before bearing 1, (m)
%  Ls          total shaft length, (m)
%  Lss         length of screw element, (m)
%  Ltbl        length of table, (m)
%  Ltbls       length of table elements, (m)
%
%  Mbi         mass of bearing inner race, (kg)
%  Mbo         mass of bearing outer race, (kg)
%  Mh()        bearing housing mass, [1 × 2], (kg), (not used)
%  Mhs()       mass of housing FE, [1 × 2], (kg)
%  Mn          mass of nut, (kg)
%  Mnhs        mass of nut housing element, (kg)
%  MnutFlange  mass of nut flange, (kg)
%  Mposs       vector of targets corresponding to ??, [? × ?], (?)
%  Ms          mass of shaft, (kg)
%  Mss         mass of screw FE, (kg)
%  MTss        !! is this variable defined ??
%  Mtbl        mass of table element, (kg)
%  m           counter (not used)
%  meanRs      mean radius of entire screw shaft, (m)
%  motorc()    something to do with calculating Im, [1 × 2]
%  motorpars()  - ditto - , [1 × 2]
%
%  NumRuns     number of runs in heating cycle
%  n           for loop counter on nut elements
%  nargin      number of arguments in input list, MATLAB special variable
%  newtitle    time in hours : minutes : seconds used when storing images
%  nhs         number of nut housing elements
%  nss         number of finite element screw elements
%  nts         number of table elements
%  numss       number of screw elements covered by nut
%
%  ODb         bearing outer diameter, (m)
%  ORb         bearing outer radius, (m)
```

```
%  origPT      pre-tension (microns/m).  If zero, presume free-ended.
%
%  Pnut        nut position, (m)
%  pi          pi, MATLAB special variable
%  pitch       ballscrew pitch (m)
%  plottype    graphical display (1) or progress bar (otherwise)
%  poss        vector of targets corresponding to screw, [? × ?], (?)
%  ptc()       something to do with calculating Ia, [1 × 2]
%  ptpars()     - ditto - , [1 × 2]
%
%  Req_exp     shaft expansion required (for what?), (micron?)
%  Rs          outer radius of screw shaft ?, (m)
%  recrows     row index for recorded data
%  rectime     time index for recording data
%
%  SBC         Stefan Boltzmann Constant, (W/m^2.K^4), (not used)
%  StopDist    distance from start for stopped nut position, (m)
%  savename    MATLAB string variable for image file
%  ss          for loop counter on screw elements
%  startPos    axis datum position from shaft start (flexible coupling), (m)
%  startss     screw element index at start of part covered by nut
%  stop        stop time at traverse limits, (sec)
%  storeimages to store sequence at 'waittime' intervals (1), not (otherwise)
%
%  T           friction torque based on measurement data, (Nm)
%  Tamb        ambient temperature, (degC)
%  Tbf()       bearing friction torque, [1 × 2], (Nm)
%  Tbf2()      bearing friction torque including non-linear friction, [1 × 2],
(Nm)
%  Tbrgi()     bearing temperature - inner race, [1 × 2], (degC)
%  Tbrgo()     bearing temperature - outer race, [1 × 2], (degC)
%  Tconstant   motor torque constant, (Nm/amp)
%  TestTime    total time for test, (min)
%  Th()        bearing housing temperature, [1 × 2], (degC) (not used)
%  Th_exp()    thermal expansion at 'y = 0.' and 'y = -460.',
               [1 × 2], (micron)
%  Ths(,)      bearing housing temperatures, [bhs × 2], (degC)
%  Tnhs()      nut housing temperatures, [nhs], (degC)
%  Tnf         nut friction torque (measured), (Nm)
%  Tnf2        nut friction torque including non-linear friction, (Nm)
%  Tnut        temperature of nut, (degC)
%  TnutFlange  temperature of nut flange, (degC)
%  Tnuthousing temperature of nut housing, (degC)
%  Tnutshaft   temperature of shaft under nut, (degC)
%  Total_err(,)total error?, ['nt' × 2], (micron)
%  Total_exp   total expansion of shaft, (micron?)
%  Tremain     approx. torque of motor and linear guideways, (Nm)
%  Tremain2    ? linear guideways at 5% viscous ?, (Nm), (not used)
%  Trun        time for one return load cycle, (sec)
%  Tss()       screw element temperature, [nss], (degC)
%  Tssb        average temperature of bearing housing, (degC)
%  Ttbl()      temperature of table, [nts], (degC)
%  Ttrav       time for uni-directional traverse, (sec)
%  Tvf         non-linear friction torque (needs sharing), (Nm?)
%  t           time index on trapezoidal position graph
%  tblnuts     centre element of table over nut
%  t0(6)       clock time, [6], (yr, mnth, day, hr, min, sec), (not used)
```

```
%
%  Vh()         bearing housing volume, [1 × 2], (m^3)
%  Vhs()        volume of bearing housing FE, [bhs? × 2], (m^3)
%  Vn           volume of nut, (m^3)
%  Vnhs         volume of nut housing element, (m^3)
%  Vtbl         volume of table element, (m^3)
%
%  W            current rotational speed, (rad/sec)
%  waittime     pause between recorded data, (sec)
%  windage      loss coefficient for cooling due to windage,
%               20m/min = 130rad/sec = max windage, (4*Hcf), (W/m^2.K)
%
%  x()          array of times at 'TI' intervals up to 'Trun', [? × 1], (sec)
%  x1           position of bearing 1, (m)
%  x1_start     start position of bearing 1, (m)
%  x2           position of bearing 2, (m).  Note negative direction
%  x2_start     start position of bearing 2, (m)
%
%  y()          array of positions in a load cycle corresponding to 'x()', [? ×
1], (m)
%
%  z            serial date number used when storing images
%
% Subroutines used
% MATLAB - abs ceil cd clock close datenum datestr drawnow eq exp figure floor
%          gca gcf gradient hold isempty legend mean mktargs not num2str pause
%          plot print round set size sprintf sum title trapmf waitbar warning
%          xlabel ylabel
% UoH    -
%
if nargin < 4; xx=1; end;
% Simulation run details
plottype=1;    % graphical display (1) or progress bar
storeimages=0; % set to '1' to store sequence at 'waittime' intervals.
waittime=0.4;  % pause (sec) between recorded data
filename='det-';
froot=cd;

% General constants
cSteel=450;        % specific heat capacity of steel (J/kg/K),
%                    (approx. from various sources)
DensitySteel=7860; % kg/m^3
Esteel=210000000;  % Young's modulus of steel (kN/m^2)
SBC=5.67E-8;       % Stefan Boltzmann constant (W/m^2.K^4)
pitch=0.016;       % m
Tamb=20;           % ambient temperature (degC)
Tconstant=1.27;    % motor torque constant (Nm/Amp)

% Heat transfer coefficients
Hc=15;  % free convective heat transfer coefficient (W/m^2.K) (2-25)
Hcf=50; % forced convection by screw rotation/axis traverse (25-250 W/m^2.K)
K=50;   % coefficient of conductivity of steel (W/m.K) (45 for nut)
%         (250 from b3??)
KbrgJoint=8000; % coeff. of conduction across bearing mounting joint
$                 (W/m^2.K),  good surface & pressure
KnhJoint=5000;  % coeff. of conduction across nut/nut housing joint (W/m^2.K)
KtblJoint=4000; % coeff. of conduction across nut housing/table joint W/m^2.K)
```

```
KbrgShaft=2000; % coeff. of conduction across bearing/shaft joint (W/m^2.K),
                % interferance fit
KbrgRace=0.2;   % coeff. of conduction across balls between races (W/K)
KnutShaft=66;   % 66 based on test 'b3nutc1' data (W/m^2.K)
AnsbContact=0.0027; % estimated surface contact between nut
                    % and screw via oil film

eSteel=0.70;    % steel emissivity
ePaint=0.95;    % surface painted white emissivity

% Bearing constants
ODb=0.08;       % bearing outer diameter (m)
IDb=0.054;      % bearing face inner diameter (m)
ORb=ODb/2;      % bearing outer radius (m)
IRb=0.015;      % bearing inner radius (m)
Lb=0.028;       % bearing length (m)
AbSection=(pi*ORb^2)-(pi*IRb^2); % m^2
AbSurface=(2*pi*ORb*(Lb/2))+AbSection; % m^2
Abf=(pi/4)*(ODb^2-IDb^2); % bearing mounting face area (m^2)
Mbo=0.6;        % mass of bearing outer race (kg)
Mbi=0.18;       % mass of bearing inner race (kg)
Tbf=[0.5 0.4];  % bearing friction torque (Nm)
%Tbf=[0 0.04];   % to give symmetric heating
Kbrg=[0.54 0.47];   % stiffness of front and rear bearing assemblies
Ah=[0.19 0.08];     % bearing housing surface area (m^2) rough
Vh=[10E-4 4.95E-4]; % bearing housing volume (m^3)
Mh=Vh*DensitySteel; % bearing housing mass (kg)
Tbrgi=[Tamb Tamb];  % bearing temperature (degC)
Tbrgo=[Tamb Tamb];
Th=[Tamb Tamb];     % bearing housing temperature (degC)

% Split bearing housings into n elements.
bhs=5;              % number of bearing housing elements
Vhs=Vh./bhs;        % volume of bearing housing FE (m^3)
Lhs=Vhs.^(1/3);     % length of each housing FE (conduction calc.) (m)
Ahsf=Lhs.^2;        % area of housing FE face (m^2)
Ahs=Ahsf*4;         % surface area of housing element
Mhs=Vhs*DensitySteel; % mass of housing FE
Ths(1:2,1:bhs)=Tamb; % set initial housing temp. to ambient

% Screw shaft constants
ExpCoeff=11;        % thermal expansion coefficient of shaft
                    % (AISI 8620 not exactly known)
Rs=0.020;           % m
meanRs=0.0192;      % mean radius of entire screw shaft
Ls=1.362;           % total shaft length (m)
Ms=(pi*meanRs^2*Ls)*DensitySteel;
Ims=0.5*Ms*meanRs^2; % screw shaft moment of inertia
Lprebrg1=0.11;      % length of screw shaft before bearing 1 (m)
brgDis=1.182;       % distance between bearings (m)
nss=100;            % number of finite element screw elements
Lss=Ls/nss;         % length of screw FE (m)
Mss=Ms/nss;         % mass of screw FE
AsSection=pi*meanRs^2;  % m^2
AsSurface=pi*Rs*2*Lss;  % m^2
AsVolume=AsSection*Lss; % m^3
Tss(1:nss)=Tamb;    % set initial screw temp. to ambient
```

```
% Identify screw elements under bearings
B1ss=round(Lprebrg1/Lss):round((Lprebrg1+Lb)/Lss);
B2ss=round((Lprebrg1+brgDis)/Lss):round((Lprebrg1+brgDis+Lb)/Lss);

% Ballnut constants
Ballnum=118;        % number of balls in nut
Lnc=0.08;           % length of nut in contact with shaft (m)
LnutFlange=0.022;   % length of nut flange (m)
DnutFlange=0.11;    % diameter of nut flange (m)
AnfSurface=pi*DnutFlange*LnutFlange; % surface area of nut flange (m^2)
MnutFlange=((pi*DnutFlange^2)/4)*LnutFlange*DensitySteel;
                    % mass of nut flange (kg)
Ln=0.146-LnutFlange; % length of nut (m)
Tnf=1.2;            % nut friction torque (Nm) (measured)
DnOuter=0.082;      % nut outer diameter (m)
DnInner=0.050;      % nut inner diameter (m)
AnSection=(pi/4)*(DnOuter^2-DnInner^2); % area of nut section (m^2)
AnSurface=(pi*DnOuter*Ln)+AnSection;    % nut surface area (m^2)
Vn=AnSection*Ln;    % volume of nut (m^3)
Mn=Vn*DensitySteel; % mass of nut (kg)
Tnut=Tamb;
TnutFlange=Tamb;

% Ballnut housing constants
AnutMount=(pi/4)*(0.110^2-0.088^2); % nut mounting flange area (m^2)
Lnh=0.14;           % length of nut housing
AnhSection=0.0044;  % nut housing cross section area (m^2)
Tnuthousing=Tamb;   % temperature of nut housing

% Split nut housing into elements
nhs=5;              % number of nut housing elements
Lnhs=Lnh/nhs;       % length of nut housing element
AnhSurface=2*(0.09*Lnhs);
Vnhs=AnhSection*Lnhs;   % volume of nut housing element (m^3)
Mnhs=Vnhs*DensitySteel; % mass of nut housing element (kg)
Tnhs(1,1:nhs)=Tamb;     % set initial nut housing temp to ambient

A1=pi*0.038*Ln; A2=pi*0.044*Ln;
f=(eSteel^2)/(eSteel+((A1/A2)*(eSteel-eSteel^2)));
                    % interchange factor for radiation

% Table constants
nts=5;              % number of table elements
tblnuts=ceil(nts/2); % centre element of table over nut
Ltbl=1.755;         % length of table
Ltbls=Ltbl/nts;     % length of table elements (m)
AtblMount=0.0161;   % nut housing to table mounting surface contact area (m^2)
AtblSection=0.026;  % area of table cross section (m^2)
AtblSurface=0.334;  % surface area of table section (m^2)
Vtbl=AtblSection*Ltbls; % volume of table element (m^3)
Mtbl=Vtbl*DensitySteel; % mass of table element (kg)
Ttbl(1,1:nts)=Tamb;

Tremain=0.3;        % approx. torque of motor and linear guideways

%--------------------------------------------------
```

```
% Set run parameters
coolTime=coolTime*60; % sec
Dist=0.46;          % traverse distance (m)
StopDist=0.46;      % distance from start for stopped nut position
startPos=1.079;     % axis datum position (m) from shaft start (flexible
coupling)
%startPos=0.931;     % gives 460 traverse symmetric about brg centre
%startPos=1.17;
F=20;               % feedrate (m/min)
stop=2;             % stop time at traverse limits (sec)
origPT=14;          % pre-tension (microns/m).  If zero, presume free-ended
brgArrange=0;       % '0'=PT and axial, '1'=PT no axial, '2'=free-ended

% Identify screw elements under error positions (traverse limits)
ErrPos1ss=[round(Lprebrg1/Lss) round(startPos/Lss)]; % front of position 1
ErrPos2ss=[round(Lprebrg1/Lss) round((startPos-Dist)/Lss)];
                                                    % front of position 2

F_start=(((origPT*brgDis)/1000000*Esteel*AsSection)/brgDis);
F_current=F_start;
expansion=0;
x2_start=(F_start/2)/-Kbrg(2) % negative direction
x1_start=(F_start/2)/Kbrg(1);
E_err(1)=((((ErrPos1ss(2)-1)-
ErrPos1ss(1))*Lss)*F_start*1000000)/(AsSection*Esteel);
E_err(2)=((((ErrPos2ss(2)-1)-
ErrPos2ss(1))*Lss)*F_start*1000000)/(AsSection*Esteel);
Total_err(1,1)=E_err(1)+x1_start;
Total_err(1,2)=E_err(2)+x1_start;

Req_exp=x1_start+abs(x2_start)+((F_start*brgDis*1000000)/(AsSection*Esteel))

%-------------------------------------------------------------

% current and feedrate double exponential parameters.

% Ballscrew 2, no pre-tension
% noptpars=[0.1368E-3 -0.0079E-3]; noptc=[-1.1632 2.1778];

% Hollow ballscrew, no pre-tension (parameters relate to feedrate in m/min)
% noptpars=[0.1104 -0.0074]; noptc=[-1.0502 2.4433];

% Hollow ballscrew with pre-tension
ptpars=[0.1879E-3 -0.0091E-3]; ptc=[-1.2431 2.9024];

% Motor only
motorpars=[0.300816 -0.009472]; motorc=[-0.325107 0.464454];

%-------------------------------------------------------------

% Generate trapezoidal axis motion
Ttrav=Dist/(F/60);  % time for uni-directional traverse (sec)
Trun=(2*Ttrav)+(2*stop);
x=(0:TI:Trun)';
y=Dist*(trapmf(x,[stop Ttrav+stop Ttrav+(2*stop) Trun]));
%y=Dist*(dsigmf(x,[stop Ttrav+stop Ttrav+(2*stop) Trun]));
```

```
dv=gradient(y,x);
% plot(x,[y,dv]); pause
% h = plot(460,1,'k*','EraseMode','xor');
NumRuns=floor((heatTime*60)/Trun);
heatTime=NumRuns*Trun;

TestTime=heatTime+coolTime;

if xx~=1 & plottype==1
   % close all
   if storeimages~=1; figure; end;
   poss=mktargs(Tss',0,0,Ls);
   h1  = plot(poss,Tss','k','EraseMode','xor');
   hold on
   % Nut
   h2  = plot([startPos-0.069;startPos-0.069],[Tamb;Tnut],
    '--b','EraseMode','xor');
   h3  = plot([startPos+0.077;startPos+0.077],[Tamb;Tnut],
    '--b','EraseMode','xor');
   h4  = plot([startPos-0.069;startPos+0.077],[Tnut+5;Tnut+5],
    '-r','EraseMode','xor');
   % Front bearing
   h5  = plot([Lprebrg1-(Lb/2);Lprebrg1-(Lb/2)],[20;Tbrgo(1)],
    '--b','EraseMode','xor');
   h6  = plot([Lprebrg1+(Lb/2);Lprebrg1+(Lb/2)],[20;Tbrgo(1)],
    '--b','EraseMode','xor');
   h7  = plot([Lprebrg1-(Lb/2);Lprebrg1+(Lb/2)],[Tbrgo(1);Tbrgo(1)],
    '--r','EraseMode','xor');
   h11 = plot(Lprebrg1,Tbrgi(1),'*g','EraseMode','xor');
   % Rear bearing
   h8  = plot([(Lprebrg1+brgDis)-(Lb/2);(Lprebrg1+brgDis)-(Lb/2)],
              [20;Tbrgo(2)],'--b','EraseMode','xor');
   h9  = plot([(Lprebrg1+brgDis)+(Lb/2);(Lprebrg1+brgDis)+(Lb/2)],
              [20;Tbrgo(2)],'--b','EraseMode','xor');
   h10 = plot([(Lprebrg1+brgDis)-(Lb/2);(Lprebrg1+brgDis)+(Lb/2)],
              [Tbrgo(2);Tbrgo(2)],'--r','EraseMode','xor');
   h12 = plot(Lprebrg1+brgDis,Tbrgi(2),'*g','EraseMode','xor');
   set(gca,'YLim',[19 32]);

   % figure resizing
   set(gca,'FontSize',7);set(gcf,'Position',[324 298 300 230]);

   xlabel('Position along screw (m)');
   ylabel('Temperature');
   % Measured temperature
    Mposs=mktargs(MTss,0,0,Ls); % !! is 'MTss' defined ??
   pause
else
   barh=waitbar(0,'Simulation working ... ');
end

t0 = clock;
j=1;
t=1;                % time index on trapezoidal position graph
rectime=0;          % time index for recording data
k=0;
recrows=1;          % row index for recorded data
```

```
m=0;

for i=0:TI:TestTime
   % Calculate nut position
   if t>size(y,1) t=1; end;
   if xx~=1 & plottype==1 & k>=60 & storeimages~=1;
      legend(h1,num2str(floor(i/60))); k=0;
   else k=k+TI; end;
   if i<=heatTime
      Pnut=startPos-y(t); % nut position (m)
      W=(abs(dv(t))/pitch)*2*pi; % current rotational speed (rad/s)
   else
      Pnut=startPos-StopDist; W=0;
   end
   windage=Hcf*4*(W/130); % 20m/min = 130rad/s = max windage (4*Hcf)
   F=(W/(2*pi))*60*16; % convert to feedrate (mm/min) for equation parameters
   Ia=ptc(1)*exp(-ptpars(1)*(F/1000)) + ptc(2)*exp(-ptpars(2)*(F/1000));
   Im=motorc(1)*exp(-motorpars(1)*(F/1000)) + motorc(2)*exp(-
motorpars(2)*(F/1000));
   I=Ia-Im;               % subtract motor friction
   T=I*Tconstant;         % friction torque based on measurement data (Nm)
   Tvf=T-Tnf-Tbf(1)-Tbf(2)-Tremain;
                          % non-linear friction torque (needs sharing)
   if Tvf < 0; Tvf=0; end
   Tnf2=Tnf+(Tvf*0.8);    % add non-linear friction.
                          % 75% in nut due to ball behaviour
   Tbf2(1)=Tbf(1)+(Tvf*0.1); % ball rack system reduces non-linear ball slip
friction
   Tbf2(2)=Tbf(2)+(Tvf*0.05);
   Tremain2=Tremain+(Tvf*0.05); % linear guideways at 5% viscous
   %set(h,'xdata',y(t));
   %drawnow;

   for b=1:2 % calculate temperature of front (1)
             % and rear (2) bearing housings
      for hs=1:bhs     % heat entering elements (joules/sec)
         if hs==1         % element in contact with bearing
            Hbhs(b,hs) = ((Tbrgo(b)-Ths(b,hs))*KbrgJoint*Abf) -...
               (((K*Ahsf(b))*(Ths(b,hs)-Ths(b,hs+1)))/Lhs(b)) -...
               (Hc*Ahs(b)*(Ths(b,hs)-Tamb));
         elseif hs==bhs % no conduction from last element, more convection etc
            Hbhs(b,hs) = (((K*Ahsf(b))/Lhs(b))*(Ths(b,hs-1)-Ths(b,hs))) - ...
               (Hc*(Ahs(b)+Ahsf(b))*(Ths(b,hs)-Tamb));
         else
            Hbhs(b,hs) = (((K*Ahsf(b))/Lhs(b))*((Ths(b,hs-1)-Ths(b,hs)) - ...
               (Ths(b,hs)-Ths(b,hs+1)))) - (Hc*Ahs(b)*(Ths(b,hs)-Tamb));
         end
      end

      Ths(b,:) = Ths(b,:) + ((TI * Hbhs(b,:)) / (Mhs(b)*cSteel));

      Tbrgo(b) = Tbrgo(b) + ((TI * ((Tbf2(b)*0.45*W) -...
         ((Tbrgo(b)-Tbrgi(b))*KbrgRace) -...      % conduction between races
         ((Tbrgo(b)-Ths(b,1))*KbrgJoint*Abf) -... % conduction into bearing
         (Hc*AbSurface*(Tbrgo(b)-Tamb)))) /...    % housing convection
         (Mbo*cSteel));
```

```
      if b==1; Tssb=mean(Tss(B1ss)); else Tssb=mean(Tss(B2ss)); end;

      Tbrgi(b) = Tbrgi(b) + ((TI * ((Tbf2(b)*0.55*W) -...
         ((Tbrgi(b)-Tbrgo(b))*KbrgRace) -...
         ((Hc+(windage/10))*(AbSurface/4)*(Tbrgi(b)-Tamb)) -... % convection
         ((Tbrgi(b)-Tssb)*KbrgShaft*(pi*Rs*2*Lb)))) /... % conduction to shaft
         (Mbi*cSteel));
   end % end of bearing loop

   startss=round((Pnut-(Lnc/2))/Lss);
   numss=round(Lnc/Lss);
   Tnutshaft=sum(Tss(startss:(startss+numss-1)))/numss; % temperature of shaft
under nut

% Calculate temperature of nut housing elements
   for n=1:nhs          % heat entering/leaving elements (joules/sec)
      if n==1           % element in contact with nut flange
         Hnhs(n) = ((TnutFlange-Tnhs(n))*KnhJoint*AnutMount) -...
                                                % conduction from nut flange
            (((K*AnhSection)/Lnhs)*(Tnhs(n)-Tnhs(n+1))) -...
                                                % conduction across elements
            ((Hc+(windage/10))*AnhSurface*(Tnhs(n)-Tamb));        % convection
      elseif n==nhs
         Hnhs(n) = (((K*AnhSection)/Lnhs)*(Tnhs(n-1)-Tnhs(n))) -...
            ((Hc+(windage/10))*(AnhSection+AnhSurface)*(Tnhs(n)-Tamb));
      else
         Hnhs(n) = (((K*AnhSection)/Lnhs)*((Tnhs(n-1)-Tnhs(n))-(Tnhs(n)-
Tnhs(n+1)))) -...
            ((Hc+(windage/10))*AnhSurface*(Tnhs(n)-Tamb));
      end
   end
   for n=1:nhs
      Tnhs(n) = Tnhs(n) + ((TI * (Hnhs(n) -...
         ((Tnhs(n)-Ttbl(tblnuts))*KtblJoint*(AtblMount/nhs)))) /
         (Mnhs*cSteel));...                    % conduction - housing > table
   end
   Tnuthousing=sum(Tnhs(:))/nhs;

   % Calculate temperature of nut
   Tnut = Tnut + ((TI * ((Tnf2*0.6*W) -...
      (((K*AnSection)/LnutFlange)*(Tnut-TnutFlange)) -...
                                                % conduction to nut flange
      ((Tnut-Tnutshaft)*KnutShaft*AnsbContact) - ...
                                                % conduction to screw
      ((Hc+(windage/10))*(AnSurface/3)*(Tnut-Tamb)) )) /... % convection
      (Mn*cSteel));

   TnutFlange=TnutFlange + ((TI * ((((K*AnSection)/LnutFlange)*(Tnut-
TnutFlange)) -...
      ((TnutFlange-Tnhs(1))*KnhJoint*AnutMount) -...
                                               % conduction to nut housing
      ((Hc+(windage/5))*AnfSurface*(TnutFlange-Tamb))))/ ... % convection
      (MnutFlange*cSteel));

   % Calculate temperature of table
   for n=1:nts
      if n == 1
```

```
         Htbl(n) = (((K*AtblSection)/Ltbls)*(Ttbl(n+1)-Ttbl(n))) -...
            ((Hc+(windage/10))*(AtblSurface+AtblSection)*(Ttbl(n)-Tamb));
      elseif n == nts
         Htbl(n) = (((K*AtblSection)/Ltbls)*(Ttbl(n-1)-Ttbl(n))) -...
            ((Hc+(windage/10))*(AtblSurface+AtblSection)*(Ttbl(n)-Tamb));
      elseif n == tblnuts
         Htbl(n) = ((Tnuthousing-Ttbl(n))*KtblJoint*AtblMount) +...
            (((K*AtblSection)/Ltbls)*((Ttbl(n-1)-Ttbl(n))-(Ttbl(n)-
Ttbl(n+1)))) -...
            ((Hc+(windage/10))*AtblSurface*(Ttbl(n)-Tamb));
      else
         Htbl(n) = (((K*AtblSection)/Ltbls)*((Ttbl(n-1)-Ttbl(n))-(Ttbl(n)-
Ttbl(n+1)))) -...
            ((Hc+(windage/10))*AtblSurface*(Ttbl(n)-Tamb));
      end
   end
   Ttbl(:) = Ttbl(:) + ((TI * Htbl(:)) / (Mtbl*cSteel));

   % Calculate temperature of screw shaft
   for ss=1:nss
      if ss==1         % no conduction from first screw FE, more convection
etc
                       % must consider oil film preventing/aiding heat loss
from shaft
         Hss(ss) = (((K*AsSection) * (Tss(ss+1)-Tss(ss)))/Lss) -...
            ((Hc+windage)*(AsSurface+(AsSection*2))*(Tss(ss)-Tamb));
      elseif ss==nss
         Hss(ss) = (((K*AsSection) * (Tss(ss-1)-Tss(ss)))/Lss) -...
            ((Hc+windage)*(AsSurface+(AsSection*2))*(Tss(ss)-Tamb));
      elseif sum(eq(B1ss,ss))>0 % screw FE under bearing 1
         Hss(ss) = ((Tbrgi(1)-Tss(ss))*KbrgShaft*AsSurface) +...
            (((K*AsSection) * ((Tss(ss+1)-Tss(ss)) -...
            (Tss(ss)-Tss(ss-1)))/Lss)); % joules/sec
      elseif sum(eq(B2ss,ss))>0 % FE under bearing 2
         Hss(ss) = ((Tbrgi(2)-Tss(ss))*KbrgShaft*AsSurface) +...
            (((K*AsSection) * ((Tss(ss+1)-Tss(ss)) -...
            (Tss(ss)-Tss(ss-1))))/Lss); % joules/sec
      elseif ss*Lss >= Pnut-(Lnc/2) & ss*Lss <= Pnut+(Lnc/2)
                                                  % screw FE under nut
         Hss(ss) = ((Tnf2*0.6*W)/(Lnc/Lss)) + (((K*AsSection) *...
            ((Tss(ss+1)-Tss(ss)) - (Tss(ss)-Tss(ss-1))))/Lss) +...
            ((Tnut-Tnutshaft)*KnutShaft*AnsbContact);
      elseif ss>1 & ss<nss % normal screw FE
         Hss(ss) = (((K*AsSection) * ((Tss(ss+1)-Tss(ss)) ...
            - (Tss(ss)-Tss(ss-1))))/Lss) -...
            ((Hc+windage)*AsSurface*(Tss(ss)-Tamb));
      end
   end % end of screw shaft FE loop

   Tss(:) = Tss(:) + ((TI * Hss(:)) / (Mss*cSteel));

  % Update graph
   if xx~=1 & plottype==1
      set(h1,'ydata',Tss);
      % Nut
      set(h2,'xdata',[Pnut-0.069;Pnut-0.069]); set(h2,'ydata',[Tamb;Tnut]);
      set(h3,'xdata',[Pnut+0.077;Pnut+0.077]); set(h3,'ydata',[Tamb;Tnut]);
```

```
      set(h4,'xdata',[Pnut-0.069;Pnut+0.077]); set(h4,'ydata',[Tnut;Tnut]);
      % Front bearing
      set(h5,'ydata',[20;Tbrgo(1)]); set(h6,'ydata',[20;Tbrgo(1)]);
      set(h7,'ydata',[Tbrgo(1);Tbrgo(1)]);
      set(h11,'ydata',Tbrgi(1));
      % Rear bearing
      set(h8,'ydata',[20;Tbrgo(2)]); set(h9,'ydata',[20;Tbrgo(2)]);
      set(h10,'ydata',[Tbrgo(2);Tbrgo(2)]);
      set(h12,'ydata',Tbrgi(2));
      drawnow
   elseif xx==1
      waitbar((i/TI)/(TestTime/TI),barh);
   end

   t=t+1;
   warning off
   if (rectime>(waittime-TI) & rectime<(waittime+TI)) | i==0
      z1(recrows,:)=Tss;
      z2(recrows,:)=[Tnut Tbrgo(1) Tbrgo(2)];
      Total_exp=((sum(Tss(B1ss:B2ss)-Tamb))*ExpCoeff*Lss);
      Exp_change=Total_exp-expansion;
      expansion=Total_exp;
      F_change=Exp_change
   ////(1/(2*Kbrg(1)))+(1/(Kbrg(2)*2)))+((brgDis*1000000)/(AsSection*Esteel)));
      F_current=F_current-F_change;
      if brgArrange==2 | (brgArrange==1 & (Req_exp-Total_exp) <= 0);
         Kbrg=[0.54 0];
         Tbf=[0.4 0.2];   % rear bearing friction significantly reduced
      else
         Kbrg=[0.54 0.47];
         Tbf=[0.5 0.4];
      end

      if Kbrg(2) == 0; x2=0; else; x2=(F_current/2)/-Kbrg(2); end;
      x1=(F_current/2)/Kbrg(1);
      % Kdatum=(-x1*brgDis)/(x2-x1) %should be zero if no axial
      % Position y = 0.
      Th_exp(1)=(sum(Tss(ErrPos1ss(1):ErrPos1ss(2)-1)-Tamb))*ExpCoeff*Lss;
      E_err(1)=((((ErrPos1ss(2)-1)-ErrPos1ss(1))*Lss)*F_current*1000000)/...
         (AsSection*Esteel); % error due to strain (microns)

      % Position y = -460.
      Th_exp(2)=(sum(Tss(ErrPos2ss(1):ErrPos2ss(2)-1)-Tamb))*ExpCoeff*Lss;
      E_err(2)=((((ErrPos2ss(2)-1)-
ErrPos2ss(1))*Lss)*F_current*1000000)/(AsSection*Esteel);

      Total_err(recrows,1)=Th_exp(1)+E_err(1)+x1;
      Total_err(recrows,2)=Th_exp(2)+E_err(2)+x1;

      extra(recrows,1)=Total_err(recrows,1)-Total_err(recrows,2);
      extra(recrows,2)=(sum(Tss(ErrPos2ss(2)-1:ErrPos1ss(2)-1)-Tamb))
                       *ExpCoeff*Lss;

      % save image
      if storeimages==1
         z=datenum(2001,12,19,0,0,i);
         newtitle=datestr(z,13);
```

```
         title(newtitle);
         savename=sprintf('%s\\movie\\%s%.3d.jpg',froot,filename,recrows);
         print('-djpeg',savename);
      end

      time(recrows,1)=i;
      rectime=0;
      recrows=recrows+1;

   else
      rectime=rectime+TI;
   end
   warning on
   j=j+1;
end

if xx==1 & not(isempty(barh)); close(barh); end;
z3=Total_err-Total_err(1,1);
z4=extra;
%elapsedtime=etime(clock,t0)
%avetmp=sum(Tss(:))/nss
hold off
time=time/60;
figure
plot(time,[z3 z4]); xlabel('Time (min)'); ylabel('Error (um)');

legend('Error at 0','Error at y=-460','Difference in error','Expansion
between');
figure
plot(time,z2); title('Nut & bearings'); xlabel('Time (min)');
ylabel('Temperature (deg C)');legend('Nut','Front bearing','Rear bearing');
%figure
%plot(time,z4); legend('1','2','3','4','5');
%Title('Temperature of table elements')
%tilefigs
Total_exp
```

Appendix 7.4 **– Mini TK model "BS_COOL1.TK"**

Variable sheet

	Symbol	Value	Units calculation	Units display	Name
# 1	A	0.20894	1/m	1/m	
# 2	l	1.182	m	m	length
# 3	k	0.597	W/(m K)	W/(m K)	thermal conductivity
# 4	Nu	3.66			Nusselt number
# 5	Cp	4183	J/(kg K)	kJ/(kg K)	specific heat
# 6	rho	1000	kg/m^3	kg/m^3	density
# 7	d	10	m	mm	diameter of duct
# 8	vw	0.1	m/sec	m/sec	velocity of coolant
# 9	Tw_ave	11.48	K	deg C	average temperature of coolant
#10	Twi	10	K	deg C	initial temperature of coolant
#11	Tb	23	K	deg C	temperature of ballscrew metal
#12	H	8.1138	W/K	W/K	heat transfer rate per degree
#13	pi	3.14159265			pi
#14	dq_dt	-93.463	W	W	heat transfer rate
#15	dv_dt	0.4712	m^3/sec	l/min	coolant flow rate
#16	Re	802.14			Reynolds number
#17	nu	1.2467×10^{-6}	m^2/sec	m^2/sec	kinematic viscosity of the coolant
#18	nu0	1.788×10^{-6}	m^2/sec	m^2/sec	kinematic viscosity at 0 deg C
#19	nu20	1.006×10^{-6}	m^2/sec	m^2/sec	kinematic viscosity at 20 deg C
#20	nu40	0.658×10^{-6}	m^2/sec	m^2/sec	kinematic viscosity at 40 deg C

Unit sheet

	calculation	display		
# 1	m	mm	×1000	
# 2	m^3/sec	l/min	×60000	
# 3	K	deg C		-273
# 4	J/(kg K)	kJ/(kg K)	×0.001	

Rule sheet

```
# 1   pi=4.*atan(1.)
# 2   A=4.*k*Nu/(Cp*rho*d^2*vw)
# 3   Tw_ave=(Twi-Tb)/(A*l)*(1.-exp(-A*l))+Tb
# 4   H=pi*k*Nu*l
# 5   dq_dt=H*(Tw_ave-Tb)
# 6   dv_dt=pi/4*d^2*vw
# 7   Re=vw*d/nu
# 8   nu=nu20+(nu40-nu0)/40.*(Tw_ave-293.)
```

Appendix 8.1 - **Static deflection theory – zero pre-tension**

The ballscrew is considered as an elastic beam supported at each end by a bearing whose radial stiffness is K_i and whose tilt stiffness is Φ_i being one at the drive end of the screw and two at the non-drive end. The pre-tension T in the ballscrew is considered to be zero.

A8.1.1 **One element shaft**

Consider a beam of length L along the X axis.

Bending under moment M gives[**129**]:-

$$\frac{d^2y}{dx^2} = \frac{M}{EI} \tag{8.A1.1}$$

where EI is the flexural rigidity of the beam.

Considering the moment balance of an element of length δx under a shear force Q gives:-

$$M = Q.\delta x + M + \delta M = 0 \tag{8.A1.2}$$

whence in the limit

$$\frac{dM}{dx} = -Q \tag{8.A1.3}$$

and so

$$\frac{d^3y}{dx^3} = \frac{1}{EI}\frac{dM}{dx} = -\frac{Q}{EI} \tag{8.A1.4}$$

differentiating

$$\frac{d^4y}{dx^4} = -\frac{1}{EI}\frac{dQ}{dx} \tag{8.A1.5}$$

If the beam has a mass μ per unit length and is subject to a gravitational field of g the forces on the element give

$$Q + \mu g.\delta x = Q + \delta Q \tag{8.A1.6}$$

whence in the limit

$$\frac{dQ}{dx} = \mu g \tag{8.A1.7}$$

The deflected shape of the beam is therefore subject to the differential equation

$$\frac{d^4y}{dx^4} = \frac{1}{EI}\mu g \tag{8.A1.8}$$

The general solution is:-

$$y = -\frac{\mu g}{24EI}x^4 + \frac{A_1}{6}x^3 + \frac{A_2}{2}x^2 + A_3x + A_4 \tag{8.A1.9}$$

leading to

$$\frac{dy}{dx} = -\frac{\mu g}{6EI}x^3 + \frac{A_1}{2}x^2 + A_2 x + A_3 \tag{8.A1.10a}$$

$$\frac{d^2y}{dx^2} = -\frac{\mu g}{2EI}x^2 + A_1 x + A_2 \tag{8.A1.10b}$$

$$\frac{d^3y}{dx^3} = -\frac{\mu g}{EI}x + A_1 \tag{8.A1.10c}$$

$$\frac{d^4y}{dx^4} = -\frac{\mu g}{EI} \tag{8.A1.10d}$$

where A_i are constants determined by the boundary conditions.

Equation (8.A1.1) and equation (8.A1.10b) yield:-

$$\begin{aligned} M &= EI\frac{d^2y}{dx^2} \\ &= -\frac{\mu g}{2}x^2 + EI\,A_1 x + EI\,A_2 \end{aligned} \tag{8.A1.11}$$

and equation (8.A1.3) and equation (8.A1.10c) yield:-

$$\begin{aligned} Q &= -\frac{dM}{dx} \\ &= \mu g\, x - EI\, A_1 \end{aligned} \tag{8.A1.12}$$

Considering the support given by the bearings, at end $x = 0$,

$$\begin{aligned} K_1 y(0) &= Q_0 + Q(0) \\ K_1 A_4 &= Q_0 - EI\, A_1 \end{aligned} \tag{8.A1.13a}$$

and so

$$EI\, A_1 + K_1\, A_4 = Q_0 \tag{8.A1.13b}$$

and

$$\begin{aligned} \Phi_1 \frac{dy}{dx}(0) &= M_0 + M(0) \\ \Phi_1 A_3 &= M_0 + EI\, A_2 \end{aligned} \tag{8.A1.14a}$$

and so

$$-EI\, A_2 + \Phi_1 A_3 = M_0 \tag{8.A1.14b}$$

At end $x = L$,

$$\begin{aligned} Q(L) + K_2 y(L) &= Q_L \\ \mu g\, L - EI\, A_1 + K_2\left(-\frac{\mu g}{24EI}L^4 + \frac{A_1}{6}L^3 + \frac{A_2}{2}L^2 + A_3 L + A_4\right) &= Q_L \end{aligned} \tag{8.A1.15a}$$

and so

$$(-EI + K_2 \frac{L^3}{6})A_1 + K_2 \frac{L^2}{2} A_2 + K_2 L\, A_3 + K_2\, A_4 = Q_L - \mu g\, L + K_2 \frac{\mu g}{24EI} L^4 \quad \text{(8.A1.15b)}$$

and

$$\begin{aligned} M(L) + \Phi_2 \frac{dy}{dx}(L) &= M_L \\ -\frac{\mu g}{2} L^2 + EI\, A_1 L + EI\, A_2 + \Phi_2 (-\frac{\mu g}{6EI} L^3 + \frac{A_1}{2} L^2 + A_2 L + A_3) &= M_L \end{aligned} \quad \text{(8.A1.16a)}$$

and so

$$(EI\, L + \Phi_2 \frac{L^2}{2})A_1 + (EI + \Phi_2 L)A_2 + \Phi_2\, A_3 = M_L + \frac{\mu g}{2} L^2 + \Phi_2 \frac{\mu g}{6EI} L^3 \quad \text{(8.A1.16b)}$$

Combining equations (8.A1.48b), (8.60b), (8.61b) and (8.62b) gives

$$\begin{pmatrix} EI & 0 & 0 & K_1 \\ 0 & -EI & \Phi_1 & 0 \\ -EI + K_2 \frac{L^3}{6} & K_2 \frac{L^2}{2} & K_2 L & K_2 \\ -EI\, L + \Phi_2 \frac{L^2}{2} & EI + \Phi_2 L & \Phi_2 & 0 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = \begin{pmatrix} Q_0 \\ M_0 \\ Q_L - \mu g\, L + K_2 \frac{\mu g}{24EI} L^4 \\ M_L + \frac{\mu g}{2} L^2 + \Phi_2 \frac{\mu g}{6EI} L^3 \end{pmatrix} \quad \text{(8.A1.17)}$$

from which A_i can be calculated provided

$$\begin{vmatrix} EI & 0 & 0 & K_1 \\ 0 & -EI & \Phi_1 & 0 \\ -EI + K_2 \frac{L^3}{6} & K_2 \frac{L^2}{2} & K_2 L & K_2 \\ -EI\, L + \Phi_2 \frac{L^2}{2} & EI + \Phi_2 L & \Phi_2 & 0 \end{vmatrix} \neq 0 \quad \text{(8.A1.18).}$$

A8.1.2 **Four element shaft**

In practice the ballscrew carries the nut and two monitoring rings which, to an approximation at least, can be considered as giving rise to point loads. Also the portions of the screw which lie outside of the bearing span, for example half of the flexible coupling connecting the drive motor to the screw at the drive end, can be considered to be point loads acting at the bearing centres. Thus the ballscrew is considered as a beam in four sections, with loads (Q_0, M_0) at the non-drive end, loads (Q_i, M_i) at intermediate points with axial coordinates X_i and loads (Q_L, M_L) at the drive end.

In section 1, $0 \le x < X_1$ and

$$y = -\frac{\mu g}{24EI} x^4 + \frac{A_1}{6} x^3 + \frac{A_2}{2} x^2 + A_3 x + A_4 \qquad (8.A1.9)$$

leading to

$$\frac{dy}{dx} = -\frac{\mu g}{6EI} x^3 + \frac{A_1}{2} x^2 + A_2 x + A_3 \qquad (8.A1.10a)$$

$$\frac{d^2 y}{dx^2} = -\frac{\mu g}{2EI} x^2 + A_1 x + A_2 \qquad (8.A1.10b)$$

$$\frac{d^3 y}{dx^3} = -\frac{\mu g}{EI} x + A_1 \qquad (8.A1.10c)$$

and

$$M = -\frac{\mu g}{2} x^2 + EI\, A_1 x + EI\, A_2 \qquad (8.A1.11)$$

$$Q = \mu g\, x - EI\, A_1 \qquad (8.A1.12)$$

In section 2, $X_1 \le x < X_2$ and

$$y = -\frac{\mu g}{24EI} x^4 + \frac{A_5}{6} x^3 + \frac{A_6}{2} x^2 + A_7 x + A_8 \qquad (8.A1.19)$$

leading to

$$\frac{dy}{dx} = -\frac{\mu g}{6EI} x^3 + \frac{A_5}{2} x^2 + A_6 x + A_7 \qquad (8.A1.20a)$$

$$\frac{d^2 y}{dx^2} = -\frac{\mu g}{2EI} x^2 + A_5 x + A_6 \qquad (8.A1.20b)$$

$$\frac{d^3 y}{dx^3} = -\frac{\mu g}{EI} x + A_5 \qquad (8.A1.20c)$$

and

$$M = -\frac{\mu g}{2} x^2 + EI\, A_5 x + EI\, A_6 \qquad (8.A1.21)$$

$$Q = \mu g x - EI A_5 \qquad (8.A1.22)$$

In section 3, $X_2 \le x < X_3$ and

$$y = -\frac{\mu g}{24EI} x^4 + \frac{A_9}{6} x^3 + \frac{A_{10}}{2} x^2 + A_{11} x + A_{12} \qquad (8.A1.23)$$

leading to

$$\frac{dy}{dx} = -\frac{\mu g}{6EI} x^3 + \frac{A_9}{2} x^2 + A_{10} x + A_{11} \qquad (8.A1.24a)$$

$$\frac{d^2y}{dx^2} = -\frac{\mu g}{2EI}x^2 + A_9 x + A_{10} \tag{8.A1.24b}$$

$$\frac{d^3y}{dx^3} = -\frac{\mu g}{EI}x + A_9 \tag{8.A1.24c}$$

and

$$M = -\frac{\mu g}{2}x^2 + EI\,A_9 x + EI\,A_{10} \tag{8.A1.25}$$

$$Q = \mu g\, x - EI\,A_9 \tag{8.A1.26}$$

In section 4, $X_3 \le x \le L$ and

$$y = -\frac{\mu g}{24EI}x^4 + \frac{A_{13}}{6}x^3 + \frac{A_{14}}{2}x^2 + A_{15}x + A_{16} \tag{8.A1.27}$$

leading to

$$\frac{dy}{dx} = -\frac{\mu g}{6EI}x^3 + \frac{A_{13}}{2}x^2 + A_{14}x + A_{15} \tag{8.A1.28a}$$

$$\frac{d^2y}{dx^2} = -\frac{\mu g}{2EI}x^2 + A_{13}x + A_{14} \tag{8.A1.28b}$$

$$\frac{d^3y}{dx^3} = -\frac{\mu g}{EI}x + A_{13} \tag{8.A1.28c}$$

and

$$M = -\frac{\mu g}{2}x^2 + EI\,A_{13}x + EI\,A_{14} \tag{8.A1.29}$$

$$Q = \mu g\, x - EI\,A_{13} \tag{8.A1.30}$$

Considering the support given by the bearings, at the drive end $x = 0$,

$$\begin{aligned} K_1 y(0) &= Q_0 + Q(0) \\ K_1 A_4 &= Q_0 - EI\,A_1 \end{aligned} \tag{8.A1.13a}$$

and so

$$EI\,A_1 + K_1\,A_4 = Q_0 \tag{8.A1.13b}$$

and

$$\begin{aligned} \Phi_1 \frac{dy}{dx}(0) &= M_0 + M(0) \\ \Phi_1 A_3 &= M_0 + EI\,A_2 \end{aligned} \tag{8.A1.14a}$$

and so

$$-EI\,A_2 + \Phi_1 A_3 = M_0 \tag{8.A1.14b}$$

At $x = X_1$,

$$\begin{aligned} -Q(X_1)_- + Q(X_1)_+ &= Q_1 \\ -(\mu g X_1 - EI A_1) + \mu g X_1 - EI A_5 &= Q_1 \end{aligned} \tag{8.A1.31a}$$

and so

$$EI A_1 - EI A_5 = Q_1 \tag{8.A1.31b}$$

and

$$\begin{aligned} -M(X_1)_- + M(X_1)_+ &= M_1 \\ -(-\frac{\mu g}{2} X_1^2 + EI A_1 X_1 + EI A_2) - \frac{\mu g}{2} X_1^2 + EI A_5 X_1 + EI A_6 &= M_1 \end{aligned} \tag{8.A1.32a}$$

and so

$$-EI X_1 A_1 - EI A_2 + EI X_1 A_5 + EI A_6 = M_1 \tag{8.A1.32b}$$

Continuity gives

$$\begin{aligned} y(X_1) &= -\frac{\mu g}{24EI} X_1^4 + \frac{A_1}{6} X_1^3 + \frac{A_2}{2} X_1^2 + A_3 X_1 + A_4 \\ &= -\frac{\mu g}{24EI} X_1^4 + \frac{A_5}{6} X_1^3 + \frac{A_6}{2} X_1^2 + A_7 X_1 + A_8 \end{aligned} \tag{8.A1.33a}$$

and so

$$\frac{X_1^3}{6} A_1 + \frac{X_1^2}{2} A_2 + X_1 A_3 + A_4 - \frac{X_1^3}{6} A_5 - \frac{X_1^2}{2} A_6 - X_1 A_7 + A_8 = 0 \tag{8.A1.33b}$$

and

$$\begin{aligned} \frac{dy}{dx}(X_1) &= -\frac{\mu g}{6EI} X_1^3 + \frac{A_1}{2} X_1^2 + A_2 X_1 + A_3 \\ &= -\frac{\mu g}{6EI} X_1^3 + \frac{A_5}{2} X_1^2 + A_6 X_1 + A_7 \end{aligned} \tag{8.A1.34a}$$

and so

$$\frac{X_1^2}{2} A_1 + X_1 A_2 + A_3 - \frac{X_1^2}{2} A_5 + X_1 A_6 + A_7 = 0 \tag{8.A1.34b}$$

At $x = X_2$,

$$\begin{aligned} -Q(X_2)_- + Q(X_2)_+ &= Q_2 \\ -(\mu g X_2 - EI A_5) + \mu g X_2 - EI A_9 &= Q_2 \end{aligned} \tag{8.A1.35a}$$

and so

$$EI A_5 - EI A_9 = Q_2 \tag{8.A1.35b}$$

and

$$\begin{aligned} -M(X_2)_- + M(X_2)_+ &= M_2 \\ -(-\frac{\mu g}{2} X_2^2 + EI\, A_5 X_2 + EI\, A_6) - \frac{\mu g}{2} X_2^2 + EI\, A_9 X_2 + EI\, A_{10} &= M_2 \end{aligned} \tag{8.A1.36a}$$

and so

$$-EI\, X_2\, A_5 - EI\, A_6 + EI\, X_2\, A_9 + EI\, A_{10} = M_1 \tag{8.A1.36b}$$

Continuity gives

$$\begin{aligned} y(X_2) &= -\frac{\mu g}{24EI} X_2^4 + \frac{A_5}{6} X_2^3 + \frac{A_6}{2} X_2^2 + A_7 X_1 + A_8 \\ &= -\frac{\mu g}{24EI} X_2^4 + \frac{A_9}{6} X_2^3 + \frac{A_{10}}{2} X_2^2 + A_{11} X_2 + A_{12} \end{aligned} \tag{8.A1.37a}$$

and so

$$\frac{X_2^3}{6} A_5 + \frac{X_2^2}{2} A_6 + X_2\, A_7 + A_8 - \frac{X_2^3}{6} A_9 - \frac{X_2^2}{2} A_{10} - X_2\, A_{11} + A_{12} = 0 \tag{8.A1.37b}$$

and

$$\begin{aligned} \frac{dy}{dx}(X_2) &= -\frac{\mu g}{6EI} X_2^3 + \frac{A_5}{2} X_2^2 + A_6 X_2 + A_7 \\ &= -\frac{\mu g}{6EI} X_2^3 + \frac{A_9}{2} X_2^2 + A_{10} X_2 + A_{11} \end{aligned} \tag{8.A1.38a}$$

and so

$$\frac{X_2^2}{2} A_5 + X_2\, A_6 + A_7 - \frac{X_2^2}{2} A_9 + X_2\, A_{10} + A_{11} = 0 \tag{8.A1.38b}$$

At $x = X_3$,

$$\begin{aligned} -Q(X_3)_- + Q(X_3)_+ &= Q_3 \\ -(\mu g\, X_3 - EI\, A_9) + \mu g\, X_3 - EI\, A_{13} &= Q_1 \end{aligned} \tag{8.A1.39a}$$

and so

$$EI\, A_9 - EI\, A_{13} = Q_3 \tag{8.A1.39b}$$

and

$$\begin{aligned} -M(X_3)_- + M(X_3)_+ &= M_3 \\ -(-\frac{\mu g}{2} X_3^2 + EI\, A_9 X_3 + EI\, A_{10}) - \frac{\mu g}{2} X_3^2 + EI\, A_{13} X_3 + EI\, A_{14} &= M_3 \end{aligned} \tag{8.A1.40a}$$

and so

$$-EI\, X_3\, A_9 - EI\, A_{10} + EI\, X_3\, A_{13} + EI\, A_{14} = M_3 \tag{8.A1.40b}$$

Continuity gives

$$\begin{aligned} y(X_3) &= -\frac{\mu g}{24EI} X_3^4 + \frac{A_9}{6} X_3^3 + \frac{A_{10}}{2} X_3^2 + A_{11} X_3 + A_{12} \\ &= -\frac{\mu g}{24EI} X_3^4 + \frac{A_{13}}{6} X_3^3 + \frac{A_{14}}{2} X_3^2 + A_{15} X_3 + A_{16} \end{aligned} \tag{8.A1.41a}$$

and so

$$\frac{X_3^3}{6} A_9 + \frac{X_3^2}{2} A_{10} + X_3 A_{11} + A_{12} - \frac{X_3^3}{6} A_{13} - \frac{X_3^2}{2} A_{14} - X_3 A_{15} + A_{16} = 0 \tag{8.A1.41b}$$

and

$$\begin{aligned} \frac{dy}{dx}(X_3) &= -\frac{\mu g}{6EI} X_3^3 + \frac{A_9}{2} X_3^2 + A_{10} X_3 + A_{11} \\ &= -\frac{\mu g}{6EI} X_3^3 + \frac{A_{13}}{2} X_3^2 + A_{14} X_1 + A_{15} \end{aligned} \tag{8.A1.42a}$$

and so

$$\frac{X_3^2}{2} A_9 + X_3 A_{10} + A_{11} - \frac{X_3^2}{2} A_{13} + X_3 A_{14} + A_{15} = 0 \tag{8.A1.42b}$$

Considering the support given by the bearings, at the non-drive end $x = L$,

$$\begin{aligned} Q(L) + K_2 y(L) &= Q_L \\ \mu g L - EI A_{13} + K_2 (-\frac{\mu g}{24EI} L^4 + \frac{A_{13}}{6} L^3 + \frac{A_{14}}{2} L^2 + A_{15} L + A_{16}) &= Q_L \end{aligned} \tag{8.A1.43a}$$

and so

$$(-EI + \frac{L^3}{6}) A_{13} + K_2 \frac{L^2}{2} A_{14} + K_2 L A_{15} + K_2 A_{16} = Q_L - \mu g L + K_2 \frac{\mu g}{24EI} L^4 \tag{8.A1.43b}$$

and

$$\begin{aligned} M(L) + \Phi_2 \frac{dy}{dx}(L) &= M_L \\ -\frac{\mu g}{2} L^2 + EI A_{13} L + EI A_{14} + \Phi_2 (-\frac{\mu g}{6EI} L^3 + \frac{A_{13}}{2} L^2 + A_{14} L + A_{15}) &= M_L \end{aligned} \tag{8.A1.44a}$$

and so

$$(EI L + \Phi_2 \frac{L^2}{2}) A_{13} + (EI + \Phi_2 L) A_{14} + \Phi_2 A_{15} = M_L + \frac{\mu g}{2} L^2 + \Phi_2 \frac{\mu g}{6EI} L^3 \tag{8.A1.44b}$$

Equations (8.A1.13b), (8.31b) and (8.A1.31b)-(8.A1.44b) can be summarised in matrix form as:-

$$\begin{pmatrix} \mathbf{H_{11}} & \mathbf{H_{12}} & \mathbf{0} & \mathbf{0} \\ \mathbf{H_{21}} & \mathbf{H_{22}} & \mathbf{H_{23}} & \mathbf{0} \\ \mathbf{0} & \mathbf{H_{32}} & \mathbf{H_{33}} & \mathbf{H_{34}} \\ \mathbf{0} & \mathbf{0} & \mathbf{H_{43}} & \mathbf{H_{44}} \end{pmatrix} \begin{pmatrix} \mathbf{A_{1-4}} \\ \mathbf{A_{5-8}} \\ \mathbf{A_{9-12}} \\ \mathbf{A_{13-16}} \end{pmatrix} = \begin{pmatrix} \mathbf{V_1} \\ \mathbf{V_2} \\ \mathbf{V_3} \\ \mathbf{V_4} \end{pmatrix} \tag{8.A1.45}$$

where the terms $\mathbf{H_{ij}}$ are submatrices of quantities determined by the geometry of the ballscrew, $\mathbf{A_j}$ are subvectors of the 16 terms used to define the deflected shape of the ballscrew by the equation

$$y = -\frac{\mu g}{24EI}x^4 + \frac{A_{4(s-1)+1}}{6}x^3 + \frac{A_{4(s-1)+2}}{2}x^2 + A_{4(s-1)+3}x + A_{4(s-1)+4}, \text{for } s = 1\cdots 4 \quad (8.A1.46)$$

s being the screw section number, and $\mathbf{V_i}$ are subvectors of the forces vector given by:-

$$\mathbf{V_1} = \begin{pmatrix} Q_f \\ M_f \\ -Q_1 \\ -M_1 \end{pmatrix},\ \mathbf{V_2} = \begin{pmatrix} 0 \\ 0 \\ -Q_2 \\ -M_2 \end{pmatrix},\ \mathbf{V_3} = \begin{pmatrix} 0 \\ 0 \\ -Q_3 \\ -M_3 \end{pmatrix},\ \mathbf{V_4} = \begin{pmatrix} 0 \\ 0 \\ -Q_d - K_2\dfrac{\mu g l^4}{24EI} + \mu g l \\ -M_d - \Phi_2\dfrac{\mu g l^3}{6EI} - \mu g\dfrac{l^2}{2} \end{pmatrix} \quad (8.A1.47a\text{-}d)$$

The submatrices are as follows:-

$$H_{11} = \begin{pmatrix} EI & 0 & 0 & K_1 \\ 0 & -EI & \Phi_1 & 0 \\ EI & 0 & 0 & 0 \\ -EI\,X_1 & -EI & 0 & 0 \end{pmatrix} \quad (8.A1.48a)$$

$$H_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -EI & 0 & 0 & 0 \\ EI\,X_1 & EI & 0 & 0 \end{pmatrix} \quad (8.A1.48b)$$

$$H_{21} = \begin{pmatrix} \dfrac{X_1^2}{2} & X_1 & 1 & 0 \\ \dfrac{X_1^3}{6} & \dfrac{X_1^2}{2} & X_1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (8.A1.48c)$$

$$H_{22} = \begin{pmatrix} -\frac{X_1^2}{2} & -X_1 & -1 & 0 \\ -\frac{X_1^3}{6} & -\frac{X_1^2}{2} & -X_1 & -1 \\ EI & 0 & 0 & 0 \\ -EI\,X_2 & -EI & 0 & 0 \end{pmatrix} \tag{8.A1.48d}$$

$$H_{23} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -EI & 0 & 0 & 0 \\ EI\,X_2 & EI & 0 & 0 \end{pmatrix} \tag{8.A1.48e}$$

$$H_{32} = \begin{pmatrix} \frac{X_2^2}{2} & X_2 & 1 & 0 \\ \frac{X_2^3}{6} & \frac{X_2^2}{2} & X_2 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{8.A1.48f}$$

$$H_{33} = \begin{pmatrix} -\frac{X_2^2}{2} & -X_2 & -1 & 0 \\ -\frac{X_2^3}{6} & -\frac{X_2^2}{2} & -X_2 & -1 \\ EI & 0 & 0 & 0 \\ -EI\,X_3 & -EI & 0 & 0 \end{pmatrix} \tag{8.A1.48g}$$

$$H_{34} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -EI & 0 & 0 & 0 \\ EI\,X_3 & EI & 0 & 0 \end{pmatrix} \tag{8.A1.48h}$$

$$H_{43} = \begin{pmatrix} \frac{X_3^2}{2} & X_3 & 1 & 0 \\ \frac{X_3^3}{6} & \frac{X_3^2}{2} & X_3 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{8.A1.48i}$$

$$H_{44} = \begin{pmatrix} -\frac{X_3^2}{2} & -X_3 & -1 & 0 \\ -\frac{X_3^3}{6} & -\frac{X_3^2}{2} & -X_3 & -1 \\ -K_2\frac{l^3}{6} + EI & -K_2\frac{l^2}{2} & -K_2 l & -K_2 \\ -\Phi_2\frac{l^2}{2} - EI\,l & -\Phi_2 l - EI & -\Phi_2 & 0 \end{pmatrix} \quad (8.A1.48j).$$

For any particular geometrical configuration and loading condition equation (8.A1.45) can be solved to give a set $\{A_i\}$ which can be used to give a predicted set of $\{y_i\}$ to compare with the measured ones.

Appendix 8.2 - **Static deflection theory – non-zero pre-tension**

The ballscrew is considered as an elastic beam supported at each end by a bearing whose radial stiffness is K_i and whose tilt stiffness is Φ_i being one at the drive end of the screw and two at the non-drive end. The pre-tension T in the ballscrew is considered to be zero.

A8.2.1 **One element shaft**

Consider a beam of length L along the X axis, the moment balance of an element of length δx under a shear force Q and a tensile load T gives:-

$$T.\delta y + M = Q.\delta x + M + \delta M = 0 \tag{8.A2.1}$$

whence in the limit

$$\frac{dM}{dx} = -Q + T\frac{dy}{dx} \tag{8.A2.2}$$

and so

$$\frac{d^3y}{dx^3} = \frac{1}{EI}\frac{dM}{dx} = -\frac{Q}{EI} + \frac{T}{EI}\frac{dy}{dx} \tag{8.A2.3}$$

differentiating

$$\frac{d^4y}{dx^4} = -\frac{1}{EI}\frac{dQ}{dx} + \frac{T}{EI}\frac{d^2y}{dx^2} \tag{8.A2.4}$$

If the beam is subject to a gravitational field of g

$$\frac{dQ}{dx} = \mu g \tag{8.A1.7}$$

The deflected shape of the beam is therefore subject to the differential equation

$$\frac{d^4y}{dx^4} - \frac{T}{EI}\frac{d^2y}{dx^2} = \frac{1}{EI}\mu g \tag{8.A2.5}$$

The corresponding homogeneous equation is

$$\frac{d^4y}{dx^4} - \frac{T}{EI}\frac{d^2y}{dx^2} = 0 \tag{8.A2.6}$$

and the characteristic equation

$$\xi^4 - \frac{T}{EI}\xi^2 = 0 \tag{8.A2.6a}$$

has roots $\xi = 0$ (8.A2.6b) (two) and $\xi = \pm\sqrt{\frac{T}{EI}}$ (8.A2.6c) for the case of tensile loading ($T > 0$). For compressive loading ($T < 0$), $\xi = \pm\sqrt{\frac{T}{EI}}i$ (8.A2.6d)

Letting $\lambda = \sqrt{\frac{T}{EI}}$ (8.A2.7), the general solution for the tensile case is

$$y = A_1 + A_2 x + A_3 \cosh \lambda x + A_4 \sinh \lambda x \qquad (8.A2.8)$$

leading to

$$\frac{dy}{dx} = A_2 + \lambda A_3 \sinh \lambda x + \lambda A_4 \cosh \lambda x \qquad (8.A2.8a)$$

$$\frac{d^2 y}{dx^2} = \lambda^2 A_3 \cosh \lambda x + \lambda^2 A_4 \sinh \lambda x \qquad (8.A2.8b)$$

$$\frac{d^3 y}{dx^3} = \lambda^3 A_3 \sinh \lambda x + \lambda^3 A_4 \cosh \lambda x \qquad (8.A2.8c)$$

$$\frac{d^4 y}{dx^4} = \lambda^4 A_3 \cosh \lambda x + \lambda^4 A_4 \sinh \lambda x \qquad (8.A2.8d)$$

and letting $\lambda = \sqrt{\frac{-T}{EI}}$ (8.A2.9),for the compressive case is

$$y = B_1 + B_2 x + B_3 \cos \lambda x + B_4 \sin \lambda x \qquad (8.A2.10)$$

leading to

$$\frac{dy}{dx} = B_2 - \lambda B_3 \sin \lambda x + \lambda B_4 \cos \lambda x \qquad (8.A2.10a)$$

$$\frac{d^2 y}{dx^2} = -\lambda^2 B_3 \cos \lambda x - \lambda^2 B_4 \sin \lambda x \qquad (8.A2.10b)$$

$$\frac{d^3 y}{dx^3} = \lambda^3 B_3 \sin \lambda x - \lambda^3 B_4 \cos \lambda x \qquad (8.A2.10c)$$

$$\frac{d^4 y}{dx^4} = \lambda^4 B_3 \cos \lambda x + \lambda^4 B_4 \sin \lambda x \qquad (8.A2.10d)$$

where A_i and B_i are constants determined by the boundary conditions.

The solutions of the non-homogeneous equation (8.A2.5) are

$$y = A_1 + A_2 x - \frac{\mu g}{2T} x^2 + A_3 \cosh \lambda x + A_4 \sinh \lambda x \qquad (8.A2.11) \text{ tensile}$$

and

$$y = B_1 + B_2 x - \frac{\mu g}{2T} x^2 + B_3 \cos \lambda x + B_4 \sin \lambda x \qquad (8.A2.12) \text{ compressive}$$

Normally a ballscrew is set with pre-tension, that is T is positive. Therefore concentrating on the **tensile case**, differentiating equation (8.A2.11) gives

$$\frac{dy}{dx} = A_2 - \frac{\mu g}{T} x + \lambda A_3 \sinh \lambda x + \lambda A_4 \cosh \lambda x \qquad \text{(8.A2.13),}$$

equation (8.A1.1) and differentiating equation (8.A2.13) yields

$$\begin{aligned} M &= EI \frac{d^2 y}{dx^2} \\ &= -\frac{\mu EI\, g}{T} + \lambda^2 EIA_3 \cosh \lambda x + \lambda^2 EIA_4 \sinh \lambda x \\ &= -\frac{\mu EI\, g}{T} + TA_3 \cosh \lambda x + TA_4 \sinh \lambda x \end{aligned} \qquad \text{(8.A2.14)}$$

and equation (8.A1.3) and differentiating equation (8.A2.14) yields

$$\begin{aligned} Q &= -\frac{dM}{dx} \\ &= -\lambda TA_3 \sinh \lambda x - \lambda TA_4 \cosh \lambda x \end{aligned} \qquad \text{(8.A2.15)}$$

Considering the support given by the bearings, at end $x = 0$,

$$\begin{aligned} K_1 y(0) &= Q_0 + Q(0) \\ K_1(A_1 + A_3) &= Q_0 - \lambda TA_4 \end{aligned} \qquad \text{(8.A2.16a)}$$

and so

$$K_1 A_1 + K_1 A_3 + \lambda TA_4 = Q_0 \qquad \text{(8.A2.16b)}$$

and

$$\begin{aligned} \Phi_1 \frac{dy}{dx}(0) &= M_0 + M(0) \\ \Phi_1(A_2 + \lambda A_4) &= M_0 + \frac{\mu EI\, g}{T} + TA_3 \end{aligned} \qquad \text{(8.A2.17a)}$$

and so

$$\Phi_1 A_2 - TA_3 + \Phi_1 \lambda A_4 = M_0 + \frac{\mu EI\, g}{T} \qquad \text{(8.A2.17b)}$$

At end $x = L$,

$$Q(L) + K_2 y(L) = Q_L$$

$$-\lambda TA_3 \sinh \lambda L - \lambda TA_4 \cosh \lambda L + K_2(A_1 + A_2 L + \frac{\mu g}{2T} L^2 + A_3 \cosh \lambda L + A_4 \sinh \lambda L) = Q_L$$

(8.A2.18a)

and so

$$K_2 A_1 + K_2 LA_2 + (K_2 \cosh \lambda L - \lambda T \sinh \lambda L)A_3 + (K_2 \sinh \lambda L - \lambda T \cosh \lambda L)A_4 = Q_L - K_2 \frac{\mu g}{2T} L^2$$

(8.A2.18b)

and

$$M(L) + \Phi_2 \frac{dy}{dx}(L) = M_L$$

$$-\frac{\mu EI\, g}{T} + TA_3 \cosh \lambda L + TA_4 \sinh \lambda L + \Phi_2 (A_2 - \frac{\mu g}{T} L + \lambda A_3 \sinh \lambda L + \lambda A_4 \cosh \lambda L) = M_L$$

(8.A2.19a)

and so

$$\Phi_2 A_2 + (\Phi_2 \lambda \sinh \lambda L + T \cosh \lambda L) A_3 + (\Phi_2 \lambda \cosh \lambda L + T \sinh \lambda L) A_4 = M_L + \frac{\mu g}{T}(EI + \Phi_2 L)$$

(8.A2.19b)

Combining equations (8.A2.16b), (8.A2.17b), (8.A2.18b) and (8.A2.19b) gives

$$\begin{pmatrix} K_1 & 0 & K_1 & \lambda T \\ 0 & \Phi_1 & -T & \Phi_1 \lambda \\ K_2 & K_2 L & K_2 \cosh \lambda L - \lambda T \sinh \lambda L & K_2 \sinh \lambda L - \lambda T \cosh \lambda L \\ 0 & \Phi_2 & \Phi_2 \lambda \sinh \lambda L + T \cosh \lambda L & \Phi_2 \lambda \cosh \lambda L + T \sinh \lambda L \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{pmatrix} = \begin{pmatrix} Q_0 \\ M_0 - \frac{\mu EI\, g}{T} \\ Q_L - K_2 \frac{\mu g}{2T} L^2 \\ M_L + \frac{\mu g}{T}(EI + \Phi_2 L) \end{pmatrix}$$

(8.A2.20)

from which A_i can be calculated provided

$$\begin{vmatrix} K_1 & 0 & K_1 & \lambda T \\ 0 & \Phi_1 & -T & \Phi_1 \lambda \\ K_2 & K_2 L & (K_2 \cosh \lambda L - \lambda T \sinh \lambda L) & (K_2 \sinh \lambda L - \lambda T \cosh \lambda L) \\ 0 & \Phi_2 & (\Phi_2 \lambda \sinh \lambda L + T \cosh \lambda L) & (\Phi_2 \lambda \cosh \lambda L + T \sinh \lambda L) \end{vmatrix} \neq 0 \quad \text{(8.A2.21).}$$

A8.2.2 **Four element shaft**

As in the case of no pre-tension, the beam is considered in four sections.

In section 1, $0 \le x < X_1$ and

$$y = A_1 + A_2 x + A_3 \cosh \lambda x + A_4 \sinh \lambda x \qquad \text{(8.A2.22)}$$

leading to

$$\frac{dy}{dx} = A_2 + \lambda A_3 \sinh \lambda x + \lambda A_4 \cosh \lambda x \qquad \text{(8.A2.22a)}$$

$$\frac{d^2 y}{dx^2} = \lambda^2 A_3 \cosh \lambda x + \lambda^2 A_4 \sinh \lambda x \qquad \text{(8.A2.22b)}$$

$$\frac{d^3 y}{dx^3} = \lambda^3 A_3 \sinh \lambda x + \lambda^3 A_4 \cosh \lambda x \qquad \text{(8.A2.22c)}$$

$$\frac{d^4y}{dx^4} = \lambda^4 A_3 \cosh \lambda x + \lambda^4 A_4 \sinh \lambda x \qquad (8.A2.22d)$$

and

$$M = -\frac{\mu EI\, g}{T} + TA_3 \cosh \lambda x + TA_4 \sinh \lambda x \qquad (8.A2.23)$$

$$Q = -\lambda TA_3 \sinh \lambda x - \lambda TA_4 \cosh \lambda x \qquad (8.A2.24)$$

In section 2, $X_1 \le x < X_2$ and

$$y = A_5 + A_6 x + A_7 \cosh \lambda x + A_8 \sinh \lambda x \qquad (8.A2.25)$$

leading to

$$\frac{dy}{dx} = A_6 + \lambda A_7 \sinh \lambda x + \lambda A_8 \cosh \lambda x \qquad (8.A2.25a)$$

$$\frac{d^2y}{dx^2} = \lambda^2 A_7 \cosh \lambda x + \lambda^2 A_8 \sinh \lambda x \qquad (8.A2.25b)$$

$$\frac{d^3y}{dx^3} = \lambda^3 A_7 \sinh \lambda x + \lambda^3 A_8 \cosh \lambda x \qquad (8.A2.25c)$$

$$\frac{d^4y}{dx^4} = \lambda^4 A_7 \cosh \lambda x + \lambda^4 A_8 \sinh \lambda x \qquad (8.A2.25d)$$

and

$$M = -\frac{\mu EI\, g}{T} + TA_7 \cosh \lambda x + TA_8 \sinh \lambda x \qquad (8.A2.26)$$

$$Q = -\lambda TA_7 \sinh \lambda x - \lambda TA_8 \cosh \lambda x \qquad (8.A2.27)$$

In section 3, $X_2 \le x < X_3$ and

$$y = A_9 + A_{10} x + A_{11} \cosh \lambda x + A_{12} \sinh \lambda x \qquad (8.A2.28)$$

leading to

$$\frac{dy}{dx} = A_{10} + \lambda A_{11} \sinh \lambda x + \lambda A_{12} \cosh \lambda x \qquad (8.A2.28a)$$

$$\frac{d^2y}{dx^2} = \lambda^2 A_{11} \cosh \lambda x + \lambda^2 A_{12} \sinh \lambda x \qquad (8.A2.28b)$$

$$\frac{d^3y}{dx^3} = \lambda^3 A_{11} \sinh \lambda x + \lambda^3 A_{12} \cosh \lambda x \qquad (8.A2.28c)$$

$$\frac{d^4y}{dx^4} = \lambda^4 A_{11} \cosh \lambda x + \lambda^4 A_{12} \sinh \lambda x \qquad (8.A2.28d)$$

and

$$M = -\frac{\mu EI\, g}{T} + TA_{11}\cosh\lambda x + TA_{12}\sinh\lambda x \qquad (8.A2.29)$$

$$Q = -\lambda TA_{11}\sinh\lambda x - \lambda TA_{12}\cosh\lambda x \qquad (8.A2.30)$$

In section 4, $X_3 \le x \le L$ and

$$y = A_{13} + A_{14}x + A_{15}\cosh\lambda x + A_{16}\sinh\lambda x \qquad (8.A2.31)$$

leading to

$$\frac{dy}{dx} = A_{14} + \lambda A_{15}\sinh\lambda x + \lambda A_{16}\cosh\lambda x \qquad (8.A2.31a)$$

$$\frac{d^2y}{dx^2} = \lambda^2 A_{15}\cosh\lambda x + \lambda^2 A_{16}\sinh\lambda x \qquad (8.A2.31b)$$

$$\frac{d^3y}{dx^3} = \lambda^3 A_{15}\sinh\lambda x + \lambda^3 A_{16}\cosh\lambda x \qquad (8.A2.31c)$$

$$\frac{d^4y}{dx^4} = \lambda^4 A_{15}\cosh\lambda x + \lambda^4 A_{16}\sinh\lambda x \qquad (8.A2.31d)$$

and

$$M = -\frac{\mu EI\, g}{T} + TA_{15}\cosh\lambda x + TA_{16}\sinh\lambda x \qquad (8.A2.32)$$

$$Q = -\lambda TA_{15}\sinh\lambda x - \lambda TA_{16}\cosh\lambda x \qquad (8.A2.33)$$

Considering the support given by the bearings, at the drive end $x = 0$,

$$\begin{aligned} 0 &= Q_0 - K_1 y(0) - \lambda TA_4 \\ &= Q_0 - K_1(A_1 + A_3) - \lambda TA_4 \end{aligned} \qquad (8.A2.34a)$$

and so

$$K_1A_1 + K_1A_3 - \lambda TA_4 = Q_0 \qquad (8.A2.34b)$$

and

$$\begin{aligned} 0 &= M_0 - \Phi_1\frac{dy}{dx}(0) + EI\frac{d^2y}{dx^2}(0) \\ &= M_0 - \Phi_1(A_2 + \lambda A_4) - \frac{\mu EI\, g}{T} + TA_3 \end{aligned} \qquad (8.A2.35a)$$

and so

$$\Phi_1A_2 - TA_3 + \Phi_1\lambda A_4 = M_0 - \frac{\mu EI\, g}{T} \qquad (8.A2.35b)$$

At $x = X_1$,

$$\begin{aligned} Q(X_1)_+ &= Q_1 - \lambda TA_7 \sinh \lambda X_1 - \lambda TA_8 \cosh \lambda X_1 \\ &= -\lambda TA_3 \sinh \lambda X_1 - \lambda TA_4 \cosh \lambda X_1 \end{aligned} \quad (8.A2.36a)$$

and so

$$-\lambda T \sinh \lambda X_1 A_3 - \lambda T \cosh \lambda X_1 A_4 + \lambda T \sinh \lambda X_1 A_7 + \lambda T \cosh \lambda X_1 A_8 = Q_1 \quad (8.A2.36b)$$

and

$$\begin{aligned} M(X_1)_+ &= M_1 - \frac{\mu EI\ g}{T} + TA_7 \cosh \lambda X_1 + TA_8 \sinh \lambda X_1 \\ &= -\frac{\mu EI\ g}{T} + TA_3 \cosh \lambda X_1 + TA_4 \sinh \lambda X_1 \end{aligned} \quad (8.A2.37a)$$

and so

$$T \cosh \lambda X_1 A_3 + T \sinh \lambda X_1 A_4 - T \cosh \lambda X_1 A_7 - T \sinh \lambda X_1 A_8 = M_1 \quad (8.A2.37b)$$

Continuity gives

$$\begin{aligned} y(X_1) &= A_1 + A_2 X_1 + A_3 \cosh \lambda X_1 + A_4 \sinh \lambda X_1 + e(X_1) \\ &= A_5 + A_6 X_1 + A_7 \cosh \lambda X_1 + A_8 \sinh \lambda X_1 + e(X_1) \end{aligned} \quad (8.A2.38a)$$

and so

$$A_1 + X_1 A_2 + \cosh \lambda X_1 A_3 + \sinh \lambda X_1 A_4 - A_5 - X_1 A_6 - \cosh \lambda X_1 A_7 - \sinh \lambda X_1 A_8 = 0 \quad (8.A2.38b)$$

and

$$\begin{aligned} \frac{dy}{dx}(X_1) &= A_2 + \lambda A_3 \sinh \lambda X_1 + \lambda A_4 \cosh \lambda X_1 + \frac{de}{dx}(X_1) \\ &= A_6 + \lambda A_7 \sinh \lambda X_1 + \lambda A_8 \cosh \lambda X_1 + \frac{de}{dx}(X_1) \end{aligned} \quad (8.A2.39a)$$

and so

$$A_2 + \lambda \sinh \lambda X_1 A_3 + \lambda \cosh \lambda X_1 A_4 - A_6 - \lambda \sinh \lambda X_1 A_7 - \lambda \cosh \lambda X_1 A_8 = 0 \quad (8.A2.39b)$$

At $x = X_2$,

$$\begin{aligned} Q(X_2)_+ &= Q_2 - \lambda TA_{11} \sinh \lambda X_2 - \lambda TA_{12} \cosh \lambda X_2 \\ &= -\lambda TA_7 \sinh \lambda X_2 - \lambda TA_8 \cosh \lambda X_2 \end{aligned} \quad (8.A2.40a)$$

and so

$$-\lambda T \sinh \lambda X_2 A_7 - \lambda T \cosh \lambda X_2 A_8 + \lambda T \sinh \lambda X_2 A_{11} + \lambda T \cosh \lambda X_2 A_{12} = Q_2 \quad (8.A2.40b)$$

and

$$\begin{aligned} M(X_2)_+ &= M_2 - \frac{\mu EI\ g}{T} + TA_{11} \cosh \lambda X_2 + TA_{12} \sinh \lambda X_2 \\ &= -\frac{\mu EI\ g}{T} + TA_7 \cosh \lambda X_2 + TA_8 \sinh \lambda X_2 \end{aligned} \quad (8.A2.41a)$$

and so

$$T\cosh\lambda X_2\,A_7 + T\sinh\lambda X_2\,A_8 - T\cosh\lambda X_2\,A_{11} - T\sinh\lambda X_2\,A_{12} = M_2 \quad \text{(8.A2.41b)}$$

Continuity gives

$$\begin{aligned} y(X_2) &= A_5 + A_6 X_2 + A_7\cosh\lambda X_2 + A_8\sinh\lambda X_2 + e(X_2) \\ &= A_9 + A_{10} X_2 + A_{11}\cosh\lambda X_2 + A_{12}\sinh\lambda X_2 + e(X_2) \end{aligned} \quad \text{(8.A2.42a)}$$

and so

$$A_5 + X_2 A_6 + \cosh\lambda X_2\,A_7 + \sinh\lambda X_2\,A_8 - A_9 - X_2 A_{10} - \cosh\lambda X_2\,A_{11} - \sinh\lambda X_2\,A_{12} = 0 \quad \text{(8.A2.42b)}$$

and

$$\begin{aligned} \frac{dy}{dx}(X_2) &= A_6 + \lambda A_7\sinh\lambda X_2 + \lambda A_8\cosh\lambda X_2 + \frac{de}{dx}(X_2) \\ &= A_{10} + \lambda A_{11}\sinh\lambda X_2 + \lambda A_{12}\cosh\lambda X_2 + \frac{de}{dx}(X_2) \end{aligned} \quad \text{(8.A2.43a)}$$

and so

$$A_6 + \lambda\sinh\lambda X_2\,A_7 + \lambda\cosh\lambda X_2\,A_8 - A_{10} - \lambda\sinh\lambda X_2\,A_{11} - \lambda\cosh\lambda X_2\,A_{12} = 0 \quad \text{(8.A2.43b)}$$

At $x = X_3$,

$$\begin{aligned} Q(X_3)_+ &= Q_3 - \lambda T A_{15}\sinh\lambda X_3 - \lambda T A_{16}\cosh\lambda X_3 \\ &= -\lambda T A_{11}\sinh\lambda X_3 - \lambda T A_{12}\cosh\lambda X_3 \end{aligned} \quad \text{(8.A2.44a)}$$

and so

$$-\lambda T\sinh\lambda X_3\,A_{11} - \lambda T\cosh\lambda X_3\,A_{12} + \lambda T\sinh\lambda X_3\,A_{15} + \lambda T\cosh\lambda X_3\,A_{16} = Q_3 \quad \text{(8.A2.44b)}$$

and

$$\begin{aligned} M(X_3)_+ &= M_3 - \frac{\mu\,EI\,g}{T} + TA_{15}\cosh\lambda X_3 + TA_{16}\sinh\lambda X_3 \\ &= -\frac{\mu\,EI\,g}{T} + TA_{11}\cosh\lambda X_3 + TA_{12}\sinh\lambda X_3 \end{aligned} \quad \text{(8.A2.45a)}$$

and so

$$T\cosh\lambda X_3\,A_{11} + T\sinh\lambda X_3\,A_{12} - T\cosh\lambda X_3\,A_{15} - T\sinh\lambda X_3\,A_{16} = M_3 \quad \text{(8.A2.45b)}$$

Continuity gives

$$\begin{aligned} y(X_3) &= A_9 + A_{10} X_3 + A_{11}\cosh\lambda X_3 + A_{12}\sinh\lambda X_3 + e(X_3) \\ &= A_{13} + A_{14} X_3 + A_{15}\cosh\lambda X_3 + A_{16}\sinh\lambda X_3 + e(X_3) \end{aligned} \quad \text{(8.A2.46a)}$$

and so

$$A_9 + X_3 A_{10} + \cosh \lambda X_3 A_{11} + \sinh \lambda X_3 A_{12} - A_{13} - X_3 A_{14} - \cosh \lambda X_3 A_{15} - \sinh \lambda X_3 A_{16} = 0$$

(8.A2.46b)

and

$$\begin{aligned} \frac{dy}{dx}(X_9) &= A_{10} + \lambda A_{11} \sinh \lambda X_3 + \lambda A_{12} \cosh \lambda X_1 + \frac{de}{dx}(X_3) \\ &= A_{14} + \lambda A_{15} \sinh \lambda X_3 + \lambda A_{16} \cosh \lambda X_3 + \frac{de}{dx}(X_3) \end{aligned}$$

(8.A2.47a)

and so

$$A_{10} + \lambda \sinh \lambda X_3 A_{11} + \lambda \cosh \lambda X_3 A_{12} - A_{14} - \lambda \sinh \lambda X_3 A_{15} - \lambda \cosh \lambda X_3 A_{16} = 0$$

(8.A2.47b)

Considering the support given by the bearings, at the non-drive end $x = L$,

$$\begin{aligned} -\lambda T A_{15} \sinh \lambda L - \lambda T A_{16} \cosh \lambda L &= Q_L - K_2 y(L) \\ &= Q_L - K_2 (A_{13} + A_{14} L - \frac{\mu g}{2T} L^2 + A_{15} \cosh \lambda L + A_{16} \sinh \lambda L) \end{aligned}$$

(8.A2.48a)

and so

$$\begin{aligned} K_2 A_{13} + K_2 L A_{14} + (K_2 \cosh \lambda L - \lambda T \sinh \lambda L) A_{15} + (K_2 \sinh \lambda L - \lambda T \cosh \lambda L) A_{16} \\ = Q_L + K_2 \frac{\mu g}{2T} L^2 \end{aligned}$$

(8.A2.48b)

and

$$EI \frac{d^2 y}{dx^2}(L) = M_L - \Phi_2 \frac{dy}{dx}(L)$$

$$-\frac{\mu EI\, g}{T} + T A_{15} \cosh \lambda L + T A_{16} \sinh \lambda L = M_L - \Phi_2 (A_{14} - \frac{\mu g}{T} L + \lambda A_{15} \sinh \lambda L + \lambda A_{16} \cosh \lambda L)$$

(8.A2.49a)

and so

$$\begin{aligned} \Phi_2 A_{14} + (\Phi_2 \lambda \sinh \lambda L + T \cosh \lambda L) A_{15} + (\Phi_2 \lambda \cosh \lambda L + T \sinh \lambda L) A_{16} \\ = M_L + \Phi_2 \frac{\mu g}{T} L + \frac{\mu EI\, g}{T} \end{aligned}$$

(8.A2.49b)

Equations (8.A2.34b)-(8.A2.49b) can be summarised in matrix form as:-

$$\begin{pmatrix} \mathbf{H_{11}} & \mathbf{H_{12}} & \mathbf{0} & \mathbf{0} \\ \mathbf{H_{21}} & \mathbf{H_{22}} & \mathbf{H_{23}} & \mathbf{0} \\ \mathbf{0} & \mathbf{H_{32}} & \mathbf{H_{33}} & \mathbf{H_{34}} \\ \mathbf{0} & \mathbf{0} & \mathbf{H_{43}} & \mathbf{H_{44}} \end{pmatrix} \begin{pmatrix} \mathbf{A_{1-4}} \\ \mathbf{A_{5-8}} \\ \mathbf{A_{9-12}} \\ \mathbf{A_{13-16}} \end{pmatrix} = \begin{pmatrix} \mathbf{V_1} \\ \mathbf{V_2} \\ \mathbf{V_3} \\ \mathbf{V_4} \end{pmatrix} \tag{8.A1.45}$$

where, as in Appendix 8.1, the terms $\mathbf{H_{ij}}$ are submatrices of quantities determined by the geometry of the ballscrew, $\mathbf{A_j}$ are subvectors of the 16 terms used to define the deflected shape of the ballscrew by the equation

$$y = A_{4(s-1)+1} + A_{4(s-1)+2}x - \frac{\mu g}{2T}x^2 + A_{4(s-1)+3}\cosh\lambda x + A_{4(s-1)+4}\sinh\lambda x, \text{for } s = 1\cdots 4 \tag{8.A2.50}$$

s being the screw section number, and $\mathbf{V_i}$ are subvectors of the forces vector given by:-

$$\mathbf{V_1} = \begin{pmatrix} Q_0 \\ M_0 - \dfrac{\mu EI\, g}{T} \\ Q_1 \\ M_1 \end{pmatrix},\ \mathbf{V_2} = \begin{pmatrix} 0 \\ 0 \\ Q_2 \\ M_2 \end{pmatrix},\ \mathbf{V_3} = \begin{pmatrix} 0 \\ 0 \\ Q_3 \\ M_3 \end{pmatrix},\ \mathbf{V_4} = \begin{pmatrix} 0 \\ 0 \\ Q_L + K_2\dfrac{\mu g}{2T}L^2 \\ M_L + \Phi_2\dfrac{\mu g}{T}L + \dfrac{\mu EI\, g}{T} \end{pmatrix} \tag{8.A2.51a-d}$$

The submatrices are as follows:-

$$H_{11} = \begin{pmatrix} K_1 & 0 & K_1 & -\lambda T \\ 0 & \Phi_1 & -T & \Phi_1\lambda \\ 0 & 0 & -\lambda T\sinh\lambda X_1 & -\lambda T\cosh\lambda X_1 \\ 0 & 0 & T\cosh\lambda X_1 & T\sinh\lambda X_1 \end{pmatrix} \tag{8.A2.52a}$$

$$H_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda T\sinh\lambda X_1 & \lambda T\cosh\lambda X_1 \\ 0 & 0 & -T\cosh\lambda X_1 & -T\sinh\lambda X_1 \end{pmatrix} \tag{8.A2.52b}$$

$$H_{21} = \begin{pmatrix} 1 & X_1 & \cosh\lambda X_1 & \sinh\lambda X_1 \\ 0 & 1 & \lambda\sinh\lambda X_1 & \lambda\cosh\lambda X_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{8.A2.52c}$$

$$H_{22}=\begin{pmatrix}-1 & -X_1 & -\cosh\lambda X_1 & -\sinh\lambda X_1\\ 0 & -1 & -\lambda\sinh\lambda X_1 & -\lambda\cosh\lambda X_1\\ 0 & 0 & -\lambda T\sinh\lambda X_2 & -\lambda T\cosh\lambda X_2\\ 0 & 0 & T\cosh\lambda X_2 & T\sinh\lambda X_2\end{pmatrix} \tag{8.A2.52d}$$

$$H_{23}=\begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & \lambda T\sinh\lambda X_2 & \lambda T\cosh\lambda X_2\\ 0 & 0 & -T\cosh\lambda X_2 & -T\sinh\lambda X_2\end{pmatrix} \tag{8.A2.52e}$$

$$H_{32}=\begin{pmatrix}1 & X_2 & \cosh\lambda X_2 & \sinh\lambda X_2\\ 0 & 1 & \lambda\sinh\lambda X_2 & \lambda\cosh\lambda X_2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix} \tag{8.A2.52f}$$

$$H_{33}=\begin{pmatrix}-1 & -X_2 & -\cosh\lambda X_2 & -\sinh\lambda X_2\\ 0 & -1 & -\lambda\sinh\lambda X_2 & -\lambda\cosh\lambda X_2\\ 0 & 0 & -\lambda T\sinh\lambda X_3 & -\lambda T\cosh\lambda X_3\\ 0 & 0 & T\cosh\lambda X_3 & T\sinh\lambda X_3\end{pmatrix} \tag{8.A2.52g}$$

$$H_{34}=\begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & \lambda T\sinh\lambda X_3 & \lambda T\cosh\lambda X_3\\ 0 & 0 & -T\cosh\lambda X_3 & -T\sinh\lambda X_3\end{pmatrix} \tag{8.A2.52h}$$

$$H_{43}=\begin{pmatrix}1 & X_3 & \cosh\lambda X_3 & \sinh\lambda X_3\\ 0 & 1 & \lambda\sinh\lambda X_3 & \lambda\cosh\lambda X_3\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{pmatrix} \tag{8.A2.52i}$$

$$H_{44}=\begin{pmatrix}-1 & -X_3 & -\cosh\lambda X_3 & -\sinh\lambda X_3\\ 0 & -1 & -\lambda\sinh\lambda X_3 & -\lambda\cosh\lambda X_3\\ K_2 & K_2L & K_2\cosh\lambda L-\lambda T\sinh\lambda L & K_2\sinh\lambda L-\lambda T\cosh\lambda L\\ 0 & \Phi_2 & \Phi_2\lambda\sinh\lambda L+T\cosh\lambda L & \Phi_2\lambda\cosh\lambda L+T\sinh\lambda L\end{pmatrix}$$

(8.A2.52j).

For any particular geometrical configuration and loading condition equation (8.A1.45) can be solved to give a set $\{A_i\}$ which can be used to give a predicted set of $\{y_i\}$ to compare with the measured ones.

Appendix 8.3 – **Dynamic behaviour of a shaft under centrifugal forces**

Consider a beam of length *L* along the X axis. It has already been established that:-

$$\frac{d^2y}{dx^2} = \frac{M}{EI} \tag{8.A1.1}$$

and that:-

$$\frac{d^3y}{dx^3} = \frac{1}{EI}\frac{dM}{dx} = -\frac{Q}{EI} \tag{8.A1.4).}$$

Let us assume that the beam, together with its Y and Z axes, is spinning about its X axis at ω radians per second and that after a sufficient length of time it settles down to a deflected shape *y(x)* determined by a balance between centrifugal forces and the elastic properties of the beam. In this case, considering the force balance of an element of length δx under the centrifugal forces acting on a beam of linear density μ gives:-

$$-Q + \mu\,\delta x\, y\omega^2 + Q + \delta Q = 0 \tag{8.A3.1}$$

whence in the limit

$$\frac{dQ}{dx} = -\mu\omega^2 y \tag{8.A3.2}$$

and so

$$\frac{d^4y}{dx^4} = -\frac{1}{EI}\frac{dQ}{dx} = \frac{\mu\omega^2}{EI}y \tag{8.A3.3}$$

This can be written as

$$\frac{d^4y}{dx^4} - k^4 y = 0 \tag{8.A3.4}$$

where

$$k = \sqrt[4]{\frac{\mu\omega^2}{EI}} \tag{8.A3.5}$$

For a solution let $y = Ae^{\alpha x}$ (8.A3.6)

whence $\frac{dy}{dx} = A\alpha\, e^{\alpha x}$ (8.A3.7)

$$\frac{d^2y}{dx^2} = A\alpha^2\, e^{\alpha x} \tag{8.A3.8}$$

$$\frac{d^3y}{dx^3} = A\alpha^3\, e^{\alpha x} \tag{8.A3.9}$$

$$\frac{d^4y}{dx^4} = A\alpha^4\ e^{\alpha x} \tag{8.A3.10}$$

Putting (8.A3.6) and (8.A3.10) in (8.A3.4) gives:-

$$\begin{aligned} A\alpha^4 e^{\alpha x} - k^4 A e^{\alpha x} &= 0 \\ (\alpha^4 - k^4) A e^{\alpha x} &= 0 \end{aligned} \tag{8.A3.11}$$

For the beam to be "spinning" as opposed to "stationary" $\omega \neq 0$ hence $0 < k^4$ (8.A3.12). One solution to this equation is that $A = 0$ (8.A3.13a). The other is that $\alpha^4 = k^4$ (8.A3.13b), whence a general solution is of the form

$$y = A_1 e^{kx} + A_2 e^{-kx} + A_3 e^{ikx} + A_4 e^{-ikx} \tag{8.A3.14a}$$

or the form

$$y = B_1 \cosh kx + B_2 \sinh kx + B_3 \cos kx + B_4 \sin kx \tag{8.A3.14b}$$

The two solutions are related by

$$\begin{aligned} A_1 &= \frac{B_1 + B_2}{2} & A_2 &= \frac{B_1 - B_2}{2} \\ A_3 &= \frac{B_3 - iB_4}{2} & A_4 &= \frac{B_3 + iB_4}{2} \end{aligned} \tag{8.A3.15}$$

Differentiating gives the slope

$$\frac{dy}{dx} = B_1 k \sinh kx + B_2 k \cosh kx - B_3 k \sin kx + B_4 k \cos kx \tag{8.A3.16a}$$

and the curvature

$$\frac{d^2y}{dx^2} = B_1 k^2 \cosh kx + B_2 k^2 \sinh kx - B_3 k^2 \cos kx - B_4 k^2 \sin kx \tag{8.A3.16b}$$

and the 3rd derivative

$$\frac{d^3y}{dx^3} = B_1 k^3 \sinh kx + B_2 k^3 \cosh kx + B_3 k^3 \sin kx - B_4 k^3 \cos kx \tag{8.A3.16c}$$

Consider the beam simply supported at both ends. Then

$$\begin{aligned} 0 &= B_1 + B_3 \\ 0 &= B_1 \cosh kL + B_2 \sinh kL + B_3 \cos kL + B_4 \sin kL \\ 0 &= B_1 - B_3 \\ 0 &= B_1 \cosh kL + B_2 \sinh kL - B_3 \cos kL - B_4 \sin kL \end{aligned} \tag{8.A3.17}$$

which gives $B_1 = 0$ (8.A3.18a), $B_2 \sinh kL = 0$ (8.A3.18b), $B_3 = 0$ (8.A3.18c) and $B_4 \sin kL = 0$ (8.A3.18d). $0 < \sinh kL$ and so $B_2 = 0$ (8.A3.19). $B_4 \neq 0$ (8.A3.20a) only

when $\sin kL = 0$ (8.A3.20b), i.e. $kL = n\pi$ (8.A3.20c) where *n* is a non-zero integer. In other words, a non-zero solution for *y* is only possible when

$$\omega^2 = n^4\pi^4 \frac{EI}{\mu L^4} \tag{8.A3.21}$$

that is at a set of defined speeds.

Consider the beam built in at both ends. Then

$$\begin{aligned} 0 &= B_1 + B_3 \\ 0 &= B_1 \cosh kL + B_2 \sinh kL + B_3 \cos kL + B_4 \sin kL \\ 0 &= B_2 + B_4 \\ 0 &= B_1 \sinh kL + B_2 \cosh kL - B_3 \sin kL + B_4 \cos kL \end{aligned} \tag{8.A3.22}$$

from which can be deduced

$$\begin{aligned} B_3 &= -B_1 \\ B_4 &= -B_2 \\ 0 &= B_1 \cosh kL + B_2 \sinh kL - B_1 \cos kL - B_2 \sin kL \\ 0 &= B_1 \sinh kL + B_2 \cosh kL + B_1 \sin kL - B_2 \cos kL \end{aligned} \tag{8.A3.23}$$

which leads to

$$\begin{aligned} 0 = {} & (B_1 - B_2)\cosh kL - (B_1 - B_2)\sinh kL \\ & - (B_1 - B_2)\cos kL - (B_1 + B_2)\sin kL \end{aligned} \tag{8.A3.24a}$$

$$\begin{aligned} 0 = {} & (B_1 + B_2)\cosh kL + (B_1 + B_2)\sinh kL \\ & - (B_1 + B_2)\cos kL + (B_1 - B_2)\sin kL \end{aligned} \tag{8.A3.24b}$$

or, in matrix form

$$\begin{pmatrix} -\sin kL & \cosh kL - \sinh kL - \cos kL \\ \cosh kL + \sinh kL - \cos kL & \sin kL \end{pmatrix} \begin{pmatrix} B_1 + B_2 \\ B_1 - B_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{8.A3.25}$$

The determinant of the matrix in (8.A3.25) is given by

$$\begin{aligned} \det &= (-\sin kL) \times (\sin kL) - (\cosh kL - \sinh kL - \cos kL) \times (\cosh kL + \sinh kL - \cos kL) \\ &= -2 \times (1 - \cosh kL \cos kL) \end{aligned} \tag{8.A3.26}$$

Since $1 \le \cosh kL$ (8.A3.27a) and $-1 \le \cos kL \le 1$ (8.A3.27b), $\det = 0$ only when $kL = 0$ (8.A3.28). This does not hold for non-zero length beam spinning at a non-zero speed. It therefore follows that $\begin{pmatrix} B_1 + B_2 \\ B_1 - B_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ (8.A3.29a), $B_1 = 0$ (8.A3.29b) and $B_2 = 0$ (8.A3.29c). Therefore, if the shaft is built in there is no stable "skipping" mode in the case of

a perfectly balanced shaft. This in turn leads to the conclusion that any transverse vibration of the ballscrew is likely to be caused by external forces.

Appendix 8.4 – **Data extraction and results – Fixed nut, all positions**

Sample of data extraction – Position 1, Speed 3

B = without excitation weights

O = with excitation weights

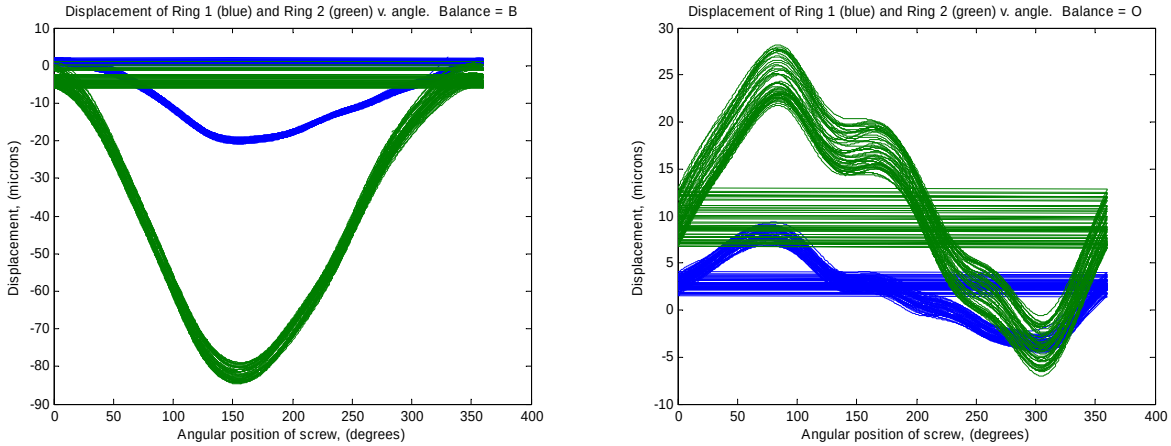


Figure A8.4.1 – *Raw data*

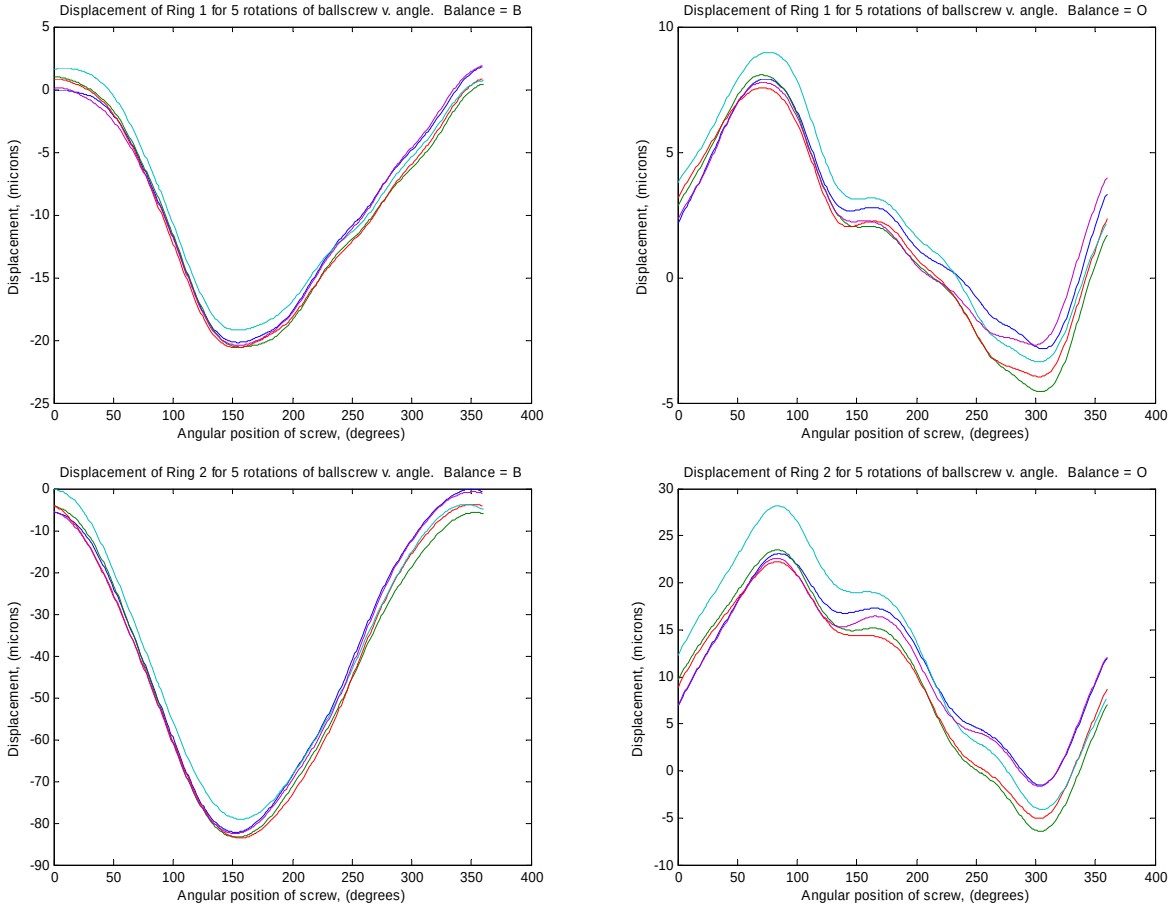


Figure A8.4.2 – *Development of vibration response over time*

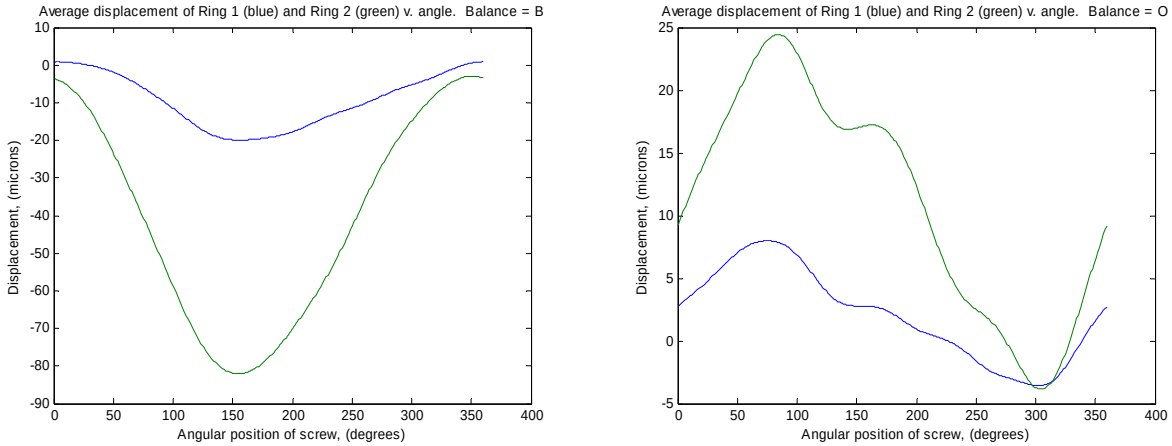


Figure A8.4.3 – *Average vibration response*

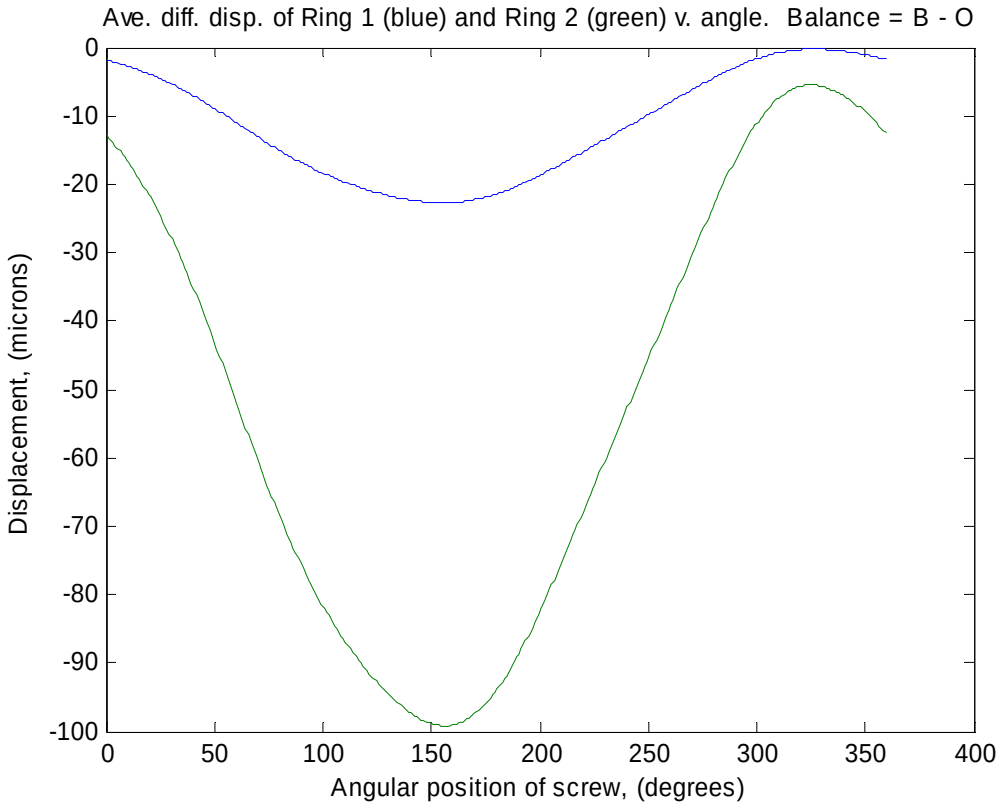


Figure A8.4.4 – *Effect of excitation weights*

Table A8.4.1 - Extracted mean vibration levels - **Position 1**

Speed	1	2	3	4	5	6
top	0.1	0.1	0.1	0.1	0.1	0.1
tail	0.9	0.9	0.9	0.9	0.9	0.9
Ring 1						
a0/2	4.1910	11.0321	9.5450	1.7009	-12.9013	15.9004
amp1	8.4869	4.7371	9.6341	15.2111	21.6647	13.3961
	157.2717	-16.3621	-16.3200	-16.2648	-19.3235	7.9638
amp2	1.5249	0.0316	0.0457	0.1064	0.5636	2.3433
	-26.8151	158.9314	-176.5315	141.8884	147.7374	39.3129
amp3	0.3078	0.0044	0.0214	0.0313	0.2976	0.1699
	56.3450	34.4782	-108.6215	0.8089	130.0082	-6.4777
amp4	0.2975	0.0055	0.5778	0.0089	0.0244	0.3834
	147.4322	163.3038	7.7661	112.7908	167.5528	-82.4145
amp5	0.0502	0.0030	0.0049	0.0019	0.0375	0.0247
	163.9259	21.1011	-44.7590	44.9435	166.0433	-20.6929
Ring 2						
a0/2	-13.4319	51.5370	51.0596	-12.3176	-68.5804	83.8246
amp1	50.3418	26.1541	53.2913	85.2851	118.1917	116.7272
	120.9508	-16.6694	-16.7699	-16.6387	-18.9519	-3.9027
amp2	0.9494	0.1300	0.2875	0.7007	2.6500	2.7581
	48.4523	159.8836	154.0191	129.1613	146.2441	54.3890
amp3	0.5778	0.0418	0.0873	0.1449	1.5813	0.3232
	5.7237	97.4000	-71.1011	22.2837	127.0766	140.4804
amp4	0.1629	0.0148	2.8912	0.0072	0.1102	0.0822
	-46.9183	94.6104	9.7228	108.7139	-168.1067	-98.0059
amp5	0.0327	0.0128	0.0164	0.0246	0.0352	0.0367
	131.0733	162.0896	61.5383	136.2864	-170.1634	99.6144

A tabulation of the results of the Fourier analysis of the extracted mean data

"top" is the proportion of the raw data at which extraction starts, and "tail" is the proportion when analysis ends.

a0/2 is the "DC" shift (µm), amp1 – amp5 are the first to fifth order amplitudes (µm), and the smaller figures under the amplitude rows are the phase angles (deg)

As can be seen, the first order dominates showing that the out-of-balance load was the most significant forcing term in the tests.

Table A8.4.2 - Extracted mean vibration levels - **Position 2**

Speed	1	2	3	4	5	6
top	0.1	0.1	0.1	0.1	0.1	0.1
tail	0.9	0.9	0.9	0.9	0.9	0.9
Ring 1						
a0/2		-7.8777	-11.3733	-16.0938	4.2525	-15.6233
amp1		5.3840	11.1827	17.5892	23.7577	30.6115
		-27.8972	-27.8917	-27.8089	-28.9150	-28.8707
amp2		0.0249	0.0534	0.1041	0.5279	0.0626
		-148.3967	133.3716	156.1584	153.2548	100.4677
amp3		0.0105	0.0171	0.0412	0.6152	0.0864
		112.9156	-157.7967	-25.9301	136.0491	-63.0190
amp4		0.0094	0.2231	0.0160	0.0350	0.0070
		-145.9144	147.6797	65.4653	-101.7995	152.7793
amp5		0.0014	0.0041	0.0070	0.0658	0.0128
		136.2639	125.2678	80.3816	-107.3668	-48.3818
Ring 2						
a0/2		-28.9305	-52.1879	-75.7360	45.0290	-31.5496
amp1		22.3293	46.5257	76.5703	103.9727	134.5700
		-28.2859	-28.5908	-27.7233	-29.0885	-28.9540
amp2		0.0489	0.2413	0.2665	1.6703	0.4714
		-174.2340	-133.8620	153.3977	155.8023	90.7148
amp3		0.0884	0.0645	0.1322	2.8759	0.3624
		175.8745	-63.7732	-25.7134	138.4327	-17.0432
amp4		0.0122	1.1012	0.0334	0.1224	0.1576
		117.7808	149.0840	68.4780	-89.1936	143.4308
amp5		0.0123	0.0246	0.0176	0.1204	0.0152
		68.8007	78.2687	110.0702	-114.3815	-108.5945

A tabulation of the results of the Fourier analysis of the extracted mean data

Table A8.4.3 - Extracted mean vibration levels - **Position 3**

Speed	1	2	3	4	5	6
top	0.1	0.1	0.1	0.1	0.1	0.1
tail	0.9	0.9	0.9	0.9	0.9	0.9
Ring 1						
a0/2	0.6683	-11.2527	10.9007	-20.2107	3.0424	12.0696
amp1	6.0360	5.2846	10.1077	15.9946	20.3326	26.5002
	-140.9955	0.5147	1.7527	1.6080	4.8812	0.7903
amp2	2.9351	0.0045	0.0828	0.1184	1.1159	0.2121
	-82.6578	28.2197	-136.8319	-178.0826	54.6717	-177.1406
amp3	0.3459	0.0074	0.0218	0.0303	1.7981	0.0520
	62.8207	88.3890	37.9255	-72.2728	167.7665	-123.9314
amp4	0.0499	0.0085	0.0357	0.0178	0.3130	0.0055
	31.9631	134.5015	-157.6599	70.7459	-106.5213	-54.7289
amp5	0.1069	0.0075	0.0064	0.0036	0.1932	0.0062
	-70.5120	7.0924	-108.1948	-126.5379	8.1737	131.2005
Ring 2						
a0/2	-3.5100	-4.8060	14.0010	-46.8298	3.1540	38.4407
amp1	10.7551	12.4898	24.1299	38.4050	45.9750	64.2591
	161.6336	0.3830	2.0334	1.6357	4.2401	0.6459
amp2	3.1828	0.1141	0.0677	0.1626	1.9125	0.4005
	-39.8200	84.1261	131.4923	162.3007	108.2741	171.7765
amp3	0.5407	0.0295	0.0238	0.0879	4.7377	0.0355
	43.6550	-55.1943	36.4149	-29.1239	163.8056	172.1146
amp4	0.0080	0.0108	0.0812	0.0315	0.2434	0.0804
	71.5728	-85.8703	-133.4760	-111.4519	-63.5669	-105.5698
amp5	0.0782	0.0120	0.0145	0.0168	0.2071	0.0193
	-79.6779	-114.7211	-135.9262	-107.9982	-10.0262	-99.5421

A tabulation of the results of the Fourier analysis of the extracted mean data

Appendix 8.5 – **Data extraction and results – Moving nut, all positions**

Sample of data extraction – Position 1, Speed 5

B = without excitation weights O = with excitation weights

top = 0.1000, tail = 0.9000, fname = 'mp1s5'
[a1_0,a1,b1,amp1,phas1,a2_0,a2,b2,amp2,phas2,trav1,trav2,angl]=geoff5(fname,top,tail);

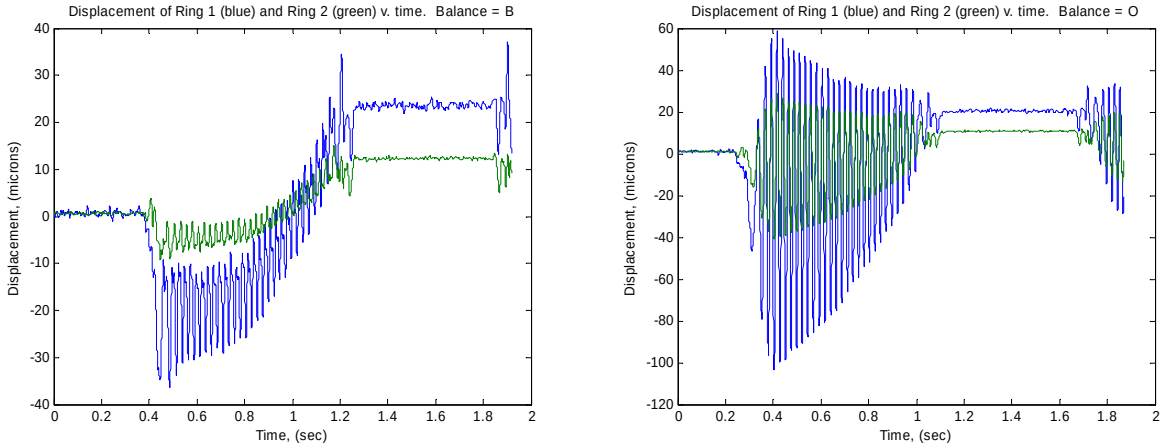


Figure A8.5.1 – *Raw data as a function of time*

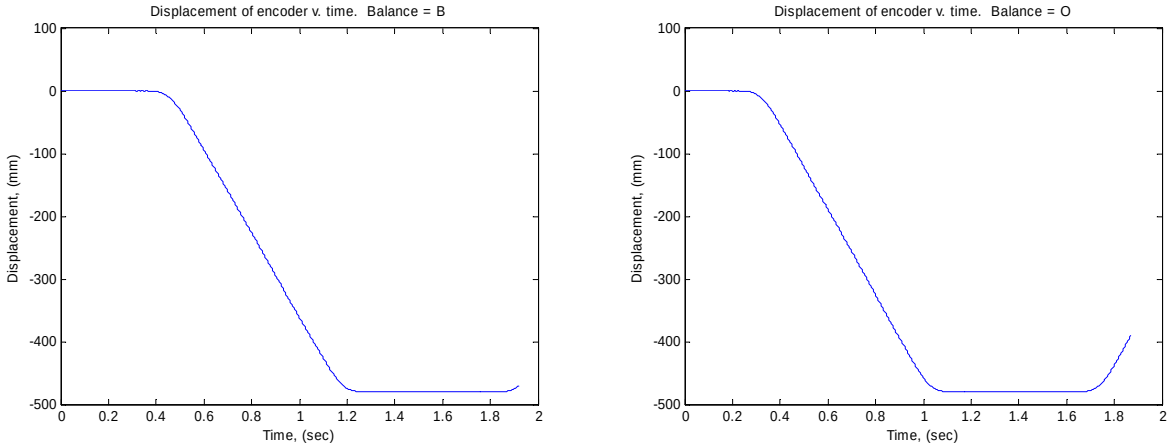


Figure A8.5.2 – *Position of nut in controller co-ordinates*

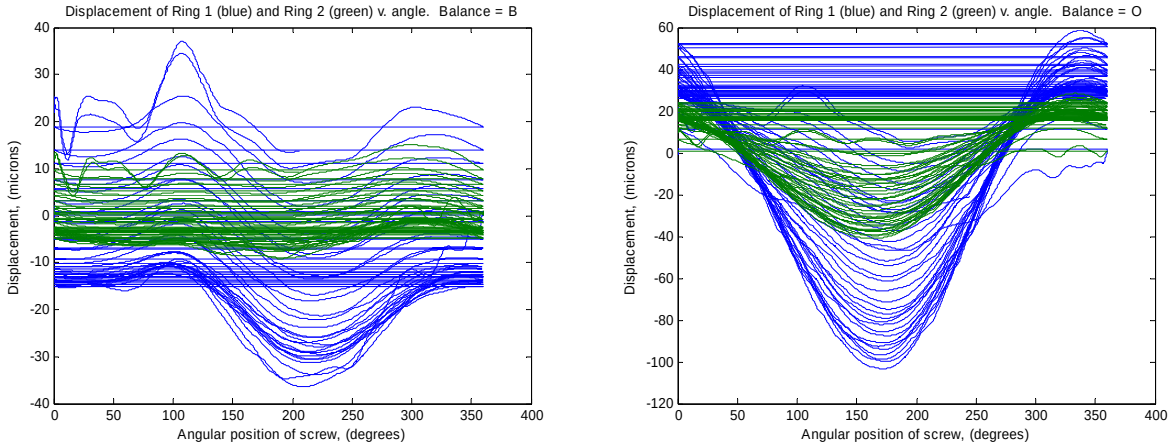


Figure A8.5.3 – *Raw data as a function of angular position*

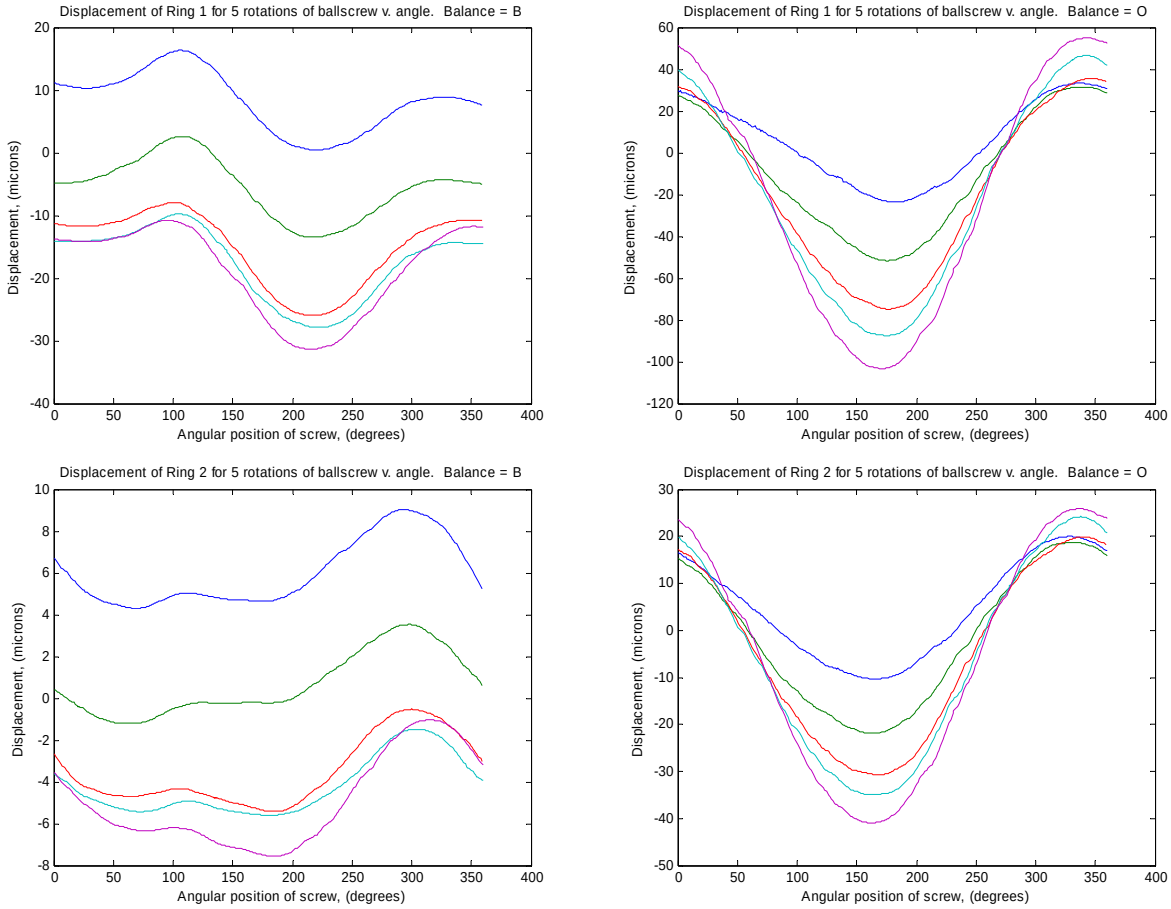


Figure A8.5.4 – *Development of vibration response over time*

Difference, balance = B – O

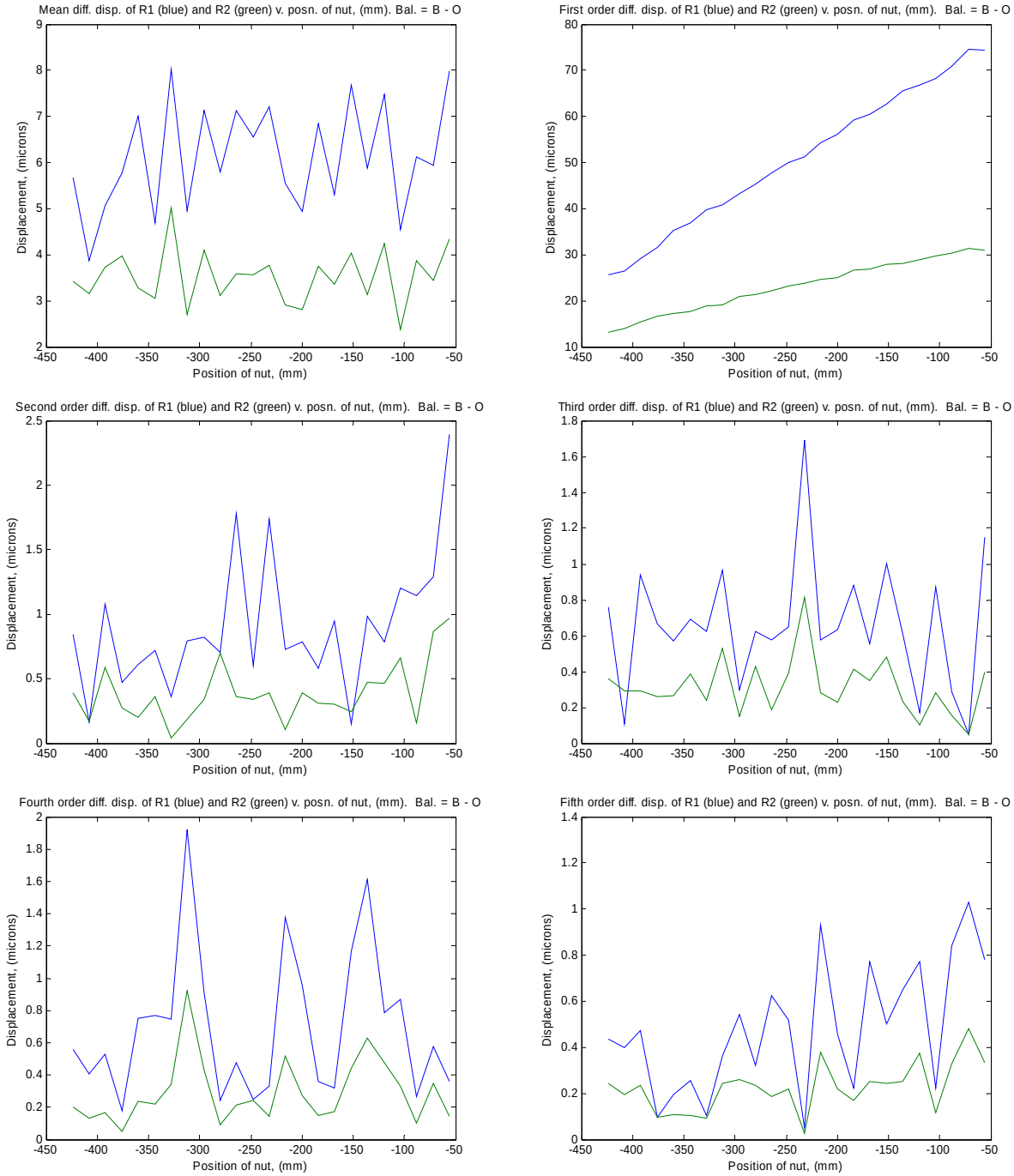


Figure A8.5.5 – *Effect of excitation weights*

Table A8.5.1 – **Raw data - Moving mass - Position 1**

File	Pos'n	Speed, (m/min)	State	Date	Start time	End time	Sample time, (sec)
MP1S1B.PEC	1	2 666	bal	9:6:2004	16: 7:28.88	16: 7:53.98	2.5006
MP1S1O.pec	1	2 666	OOB	9:6:2004	17:30:47.49	17:31:12.59	2.5006
MP1S2B.PEC	1	25 298	bal	10:6:2004	9:19:52.87	9:19:55.61	0.26353
MP1S2O.pec	1	25 298	OOB	10:6:2004	9: 8:54. 9	9: 8:56.78	0.26353
MP1S3B.PEC	1	30 984	bal	10:6:2004	9:36:13.89	9:36:16.14	0.21516
MP1S3O.pec	1	30 984	OOB	10:6:2004	9:47:22.66	9:47:24.92	0.21516
MP1S4B.PEC	1	35 777	bal	10:6:2004	10:37:27.86	10:37:29.78	0.18634
MP1S4O.pec	1	35 777	OOB	10:6:2004	10:29:54. 1	10:29:55.93	0.18634
MP1S5B.PEC	1	40 000	bal	9:6:2004	15:33:13. 8	15:33:15. 5	0.16667
MP1S5BR .PEC	1	40 000	bal rep	10:6:2004	10:44:39. 8	10:44:41. 5	0.16667
MP1S5O.PEC	1	40 000	OOB	10:6:2004	10:58:35.76	10:58:37.79	0.16667
MP1S6B.PEC	1	37 947	bal	10:6:2004	11:17:11.84	11:17:13.71	0.17568
MP1S6O.PEC	1	37 947	OOB	10:6:2004	11:10:31.16	11:10:33. 3	0.17568

Table A8.5.2 - **Raw data - Moving mass - Position 2**

File	Pos'n	Speed, (m/min)	State	Date	Start time	End time	Sample time, (sec)
MP2S1B.PEC	2	2 666	bal	10:6:2004	14:46:12.71	14:46:37.81	2.5006
MP2S1O.PEC	2	2 666	OOB	10:6:2004	14:32:37. 1	14:33: 2.11	2.5006
MP2S2B.pec	2	25 298	bal	10:6:2004	14:55:20.15	14:55:22.89	0.26353
MP2S2O.PEC	2	25 298	OOB	10:6:2004	15: 2: 3.85	15: 2: 6.60	0.26353
MP2S3B.PEC	2	30 984	bal	10:6:2004	15:14:26.88	15:14:29.13	0.21516
MP2S3O.PEC	2	30 984	OOB	10:6:2004	15: 9: 2.27	15: 9: 4.52	0.21516
MP2S4B.PEC	2	35 777	bal	10:6:2004	15:22: 5.46	15:22: 7.43	0.18634
MP2S4O.PEC	2	35 777	OOB	10:6:2004	15:29:12.39	15:29:14.31	0.18634
MP2S5B.PEC	2	40 000	bal	10:6:2004	15:48:54.39	15:48:56.37	0.16667
MP2S5O.PEC	2	40 000	OOB	10:6:2004	16: 4:24.66	16: 4:26.64	0.16667
MP2S6B.PEC	2	37 947	bal	10:6:2004	15:41:44.10	15:41:45.97	0.17568
MP2S6O.PEC	2	37 947	OOB	10:6:2004	15:35:41.21	15:35:43. 8	0.17568

Table A8.5.3 - **Raw data - Moving mass - Position 3**

File	Pos'n	Speed, (m/min)	State	Date	Start time	End time	Sample time, (sec)
MP3S1B.PEC	3	2 666	bal	11:6:2004	10:59:49.14	11: 0:14.24	2.5006
MP3S1O.PEC	3	2 666	OOB	11:6:2004	11: 6:49.43	11: 7:14.47	2.5006
MP3S2B.PEC	3	25 298	bal	11:6:2004	10:53:48.22	10:53:50.97	0.26353
MP3S2O.PEC	3	25 298	OOB	11:6:2004	10:48:36.52	10:48:39.27	0.26353
MP3S3B.PEC	3	30 984	bal	11:6:2004	10:35:15.65	10:35:17.90	0.21516
MP3S3O.PEC	3	30 984	OOB	11:6:2004	10:41:51.77	10:41:54. 3	0.21516
MP3S4B.PEC	3	35 777	bal	11:6:2004	10:28:57.71	10:28:59.69	0.18634
MP3S4O.PEC	3	35 777	OOB	11:6:2004	10:22:25.76	10:22:27.74	0.18634
MP3S5B.PEC	3	40 000	bal	10:6:2004	16:47:15.78	16:47:17.76	0.16667
MP3S5O.PEC	3	40 000	OOB	10:6:2004	16:39:35. 6	16:39:37. 4	0.16667
MP3S6B.PEC	3	37 947	bal	11:6:2004	10: 4:20.65	10: 4:22.52	0.17568
MP3S6O.PEC	3	37 947	OOB	11:6:2004	10:15:51.12	10:15:52.99	0.17568

Appendix 8.6 - **Linear Guide rig – Data**

A8.6.1 **Drive motor data (including controller parameters)**

Name plate details:-

SIEMENS	3 ~ Permanent Magnet Motor	CE
MADE IN GERMANY	1FT 6082 – 1AFY1 – 1AG1	
	Nr.E L2904930 04 011 EN 60034	
Mn = 10.3 Nm	3000 /min Ui(eff.) = 240 V Y	
(M = 11.7 Nm	1500 /min Ui(eff.) = 120 V Y)	
M_0 = 10.4 / 13 Nm	$I_{0(eff.)}$ = 8.2 / 10.65 A 60 / 100 k	
IMB5 IP 65	Th.CL.F n_{max} : 4400 /min	KTY 84
OPTICAL ENCODER	B01 2048 S/R	51

The resulting data is:-

16–digit motor Order No. 1FT 6082 – 1AFY1 – 1AG1
Serial number E L2904930 04 011 EN 60034
Rated torque for S1 duty = 10.3 Nm at 3000 rpm rated speed,
with an induced phase–to–phase, motor voltage of 240 V.
(Additional operating point (for 230V drive converter input voltage)
= 11.7 Nm at 1500 rpm rated speed,
with an induced phase–to–phase, motor voltage of 120 V.)
Stall torque 10.4/13 Nm at a stall current of 8.2/10.65 A with a 60/100 K
winding temperature rise
Type of construction = IMB5
Degree of protection = IP 65
Thermal protection = CL.F

Siemens AG, “Simodrive AC motors for feed and main spindle drives - Planning guide 01 98 edition”, Manufacturer’s documentation, 6SN1197-0AA20, SIMODRIVE 611 (PJ), 1997

Sheet 1FT6/3-10:-

In addition to the data on the name plate,

Torsional rotational inertia (without brake) = 30×10^{-4} kg m^2
Winding resistance = 1.46 Ω
Three-phase inductance = 13.7×10^{-3} H

This data was up-dated by Veimar Casteneda in an e-mail dated Wednesday 19th July 2007 at 10:05 hr to:-

Winding resistance = 0.68 Ω
Three-phase inductance = 6.2×10^{-3} H

The motor obeys the equation (c.f. Appendix 9.1 and [**142**])

$$L\frac{d}{dt}(i_q) = e_{q\,ref} - R\,i_q - \lambda\omega_e$$

where

L = the winding inductance, (= 6.2×10^{-3} H)
i_q = the nett motor current, (A)
t = time, (sec)
$e_{q\,ref}$ = the electric potential given by the current controller, (V)
R = the winding resistance, (= 0.68 Ω)
λ = the flux, (= 0.7639 V sec/rad)
ω_e = the electrical velocity, (rad/sec).

The current controller voltage is given by

$$e_{q\,ref} = K_{cp} \times (i_{q\,ref} - i_q)$$

where

K_{cp} = the proportional gain current control, (= 50 V/A)
$i_{q\,ref}$ = the output current of the velocity controller, (A)

and the electrical velocity is

$$\omega_e = p\omega_m$$

where

p = is the number pairs of poles in the motor (= 4)
ω_m = the speed of the motor, (rad/sec)

Re-arranging the differential equation gives

$$i_q = \frac{e_{q\,ref}}{R} - \frac{L}{R}\frac{di_q}{dt} - \frac{\lambda}{R}\omega_e$$

Substituting for $i_{q\,ref}$ and ω_e gives

$$i_q = \frac{K_{cp}(i_{q\,ref} - i_q)}{R} - \frac{L}{R}\frac{di_q}{dt} - \frac{\lambda p}{R}\omega_m$$

Re-arranging

$$\frac{R + K_{cp}}{R} i_q = \frac{K_{cp}}{R} i_{q\,ref} - \frac{L}{R}\frac{di_q}{dt} - \frac{\lambda p}{R}\omega_m$$

In finite difference terms

$$\frac{R + K_{cp}}{R} i_q = \frac{K_{cp}}{R} i_{q\,ref} - \frac{L}{R}\frac{(i_q - i_{q\,save})}{\delta t} - \frac{\lambda p}{R}\omega_m$$

where δt = the loop time interval, (sec).

Re-arranging

$$\frac{R + K_{cp} + L/\delta t}{R} i_q = \frac{K_{cp}}{R} i_{q\,ref} + \frac{L}{R.\delta t} i_{q\,save} - \frac{\lambda p}{R}\omega_m$$

or

$$i_q = \frac{K_{cp} i_{q\,ref} + \frac{L}{\delta t} i_{q\,save} - \lambda p \omega_m}{R + K_{cp} + L/\delta t}$$

finally

$$i_q = \frac{AA\, i_{q\,ref} + BB\, i_{q\,save} - CC\, \omega_m}{denominator}$$

where $AA = \dfrac{K_{cp}}{denominator}$, $BB = \dfrac{L/\delta t}{denominator}$, $CC = \dfrac{\lambda p}{denominator}$ and $denominator = R + K_{cp} + L/\delta t$.

These equations are coded into subroutines "init_controller" and "controller". Including definitions this is coded by:-

```
static float J_mot=0.0030F;          // drive motor inertia, (kg m^2)
static double K_v=4.*1000./60.;      // position controller constant = 4 (m/min)/mm, (1/sec)
static double K_k1=1./double(pitch); // constant, (1/m)
static double K_vff=K_k1;            // gain - velocity feed forward, (1/m)
static double K_vel;                 // constant =1./(2.*pi), (rev/rad)
static double K_aff=0.039;           // gain - acceleration feed forward, (A sec^2/rev)
static double K_p=26.;               // proportional gain - velocity control, (A sec/rev)
static double K_i=6300.;             // integral gain - velocity control, (A/rev)
static double LA=0.0062;             // motor armature inductance, (H)
static double RA=0.68;               // motor armature resistance, (ohm)
static double K_cp=50.;              // proportional gain - current control, (V/A)
static double lamda=0.7639;          // flux, (V sec/rad)
static signed int N_poles=4;         // number of pairs of motor poles
denominator=RA+K_cp+LA/dt;
AA=K_cp/denominator;
BB=(LA/dt)/denominator;
CC=lamda*double(N_poles)/denominator;
iq_save=0.;                          // motor armature current (previous filter cycle), (A)
iq=AA*iq_ref + BB*iq_hold - CC*wm_s; // motor drive current
iq_hold=iq;
static double Kt=1.28F;              // motor torque time constant, (N m/A)
```

A8.6.2 **Motor mechanical losses**

A term is included in the model to account for viscous drag in the motor bearings. The "mechanical time constant" included in the motor data sheets provided by the supplier (AL_S/1-8) is to do with a theoretical ramp function, rather than giving an indication of damping in the motor. It is therefore estimated as follows:-

Rated torque = 10.30 N m
Rated speed = 3000 rpm
Assume viscous drag is 2% of rated torque at rated speed, then

$$c_{brg\ m} = \frac{0.02 \times 10.30}{3000/60 \times 2\pi} = 0.656 \times 10^{-3} \text{ N sec/rad}$$

It is coded by:-

```
static float c_brg_m=0.656e-3F;      // equivalent damping coefficient of bearings, (N m sec/rad)
```

Further an allowance for Coulomb friction in the motor of $0.01 \times$ the rated motor torque is included (Mode 1). This is coded by:-

```
static double rEnD[M3][M9];          // rate of energy dissipation, (W)
static float T_mot_max=42.F;         // maximum motor torque, (N m)
static float n_mot_max=4700.F;       // maximum rated motor speed, (rpm)
static float mu_brg=0.01F;           // nominal coefficient of friction of bearings
signed int mode=0;                   // friction mode of operation
                                     // 0 = "T = A + B*w"
```

```
                                        // 1 = friction torque based on loading considerations
double dt1;                             // time step, inner loop, (sec)
rEnD[j+1][1]+=fric_brg(mu_brg,0.F,T_mot_max,1,double(n_mot_max)*2.*pi/60.,j,dt1,mode);
```

Thus it is assumed that at its rated speed, mechanical losses account for some 3% of rated power.

An alternative option is to use measured data if available (Mode 0). For the linear guide rig as a whole:-

```
// "Mode 0" bearing friction model
//  constant term       speed-dependent
    Abrg[0]=0.5453;     Bbrg[0]=0.001208;     // motor bearing
    Abrg[1]=0.5;        Bbrg[1]=0.;           // ballscrew bearing 1
    Abrg[2]=0.4;        Bbrg[2]=0.;           // ballscrew bearing 2
//   (N m)              (N m sec/rad)
```

This data is based on work by Dr Simon Fletcher, (see Chapter 7).

In addition to the data for the motor and controller, the motor shaft is taken into account by the following data:-

```
static float J_ms=0.0000075F;     // drive motor shaft inertia, (kg m^2)
static float k_ms=5466.F;         // torsional stiffness of motor shaft, (N m/rad)
static float c_ms=0.01F;          // torsional damping coefficient of motor shaft, (N m sec/rad)
```

A8.6.3 **Coupling**

Information from [**142**]:-

Torsional stiffness – static	4200 N m/rad
Torsional stiffness – dynamic	10100 N m/rad
Axial stiffness	3500×10^3 N/m
Torsional inertia of coupling half	200.3×10^{-6} kg m^2
Mass of coupling half	0.253 kg

From measurement / observation:-

Material	aluminium
Outside diameter	65 mm
Inside diameter	28 mm (ballscrew drawing)
Length to start of “dog teeth”	35 mm
Length to end of “dog teeth”	50 mm

Hence the mass can be estimated to be:-

$$m_{P1} = \frac{\pi}{4} \times (0.065^2 - 0.028^2) \times (0.035 + 0.050) / 2 \times 2700 = 0.31012 \text{ kg}$$

the rotational inertia:-

$$J_{P1} = \frac{\pi}{32} \times (0.065^4 - 0.028^4) \times (0.035 + 0.050) / 2 \times 2700 = 194.2 \times 10^{-6} \text{ kg m}^2$$

and the transverse inertia:-

$$J_{t\,P1} = 0.31012 \times \left(\frac{0.065^2 + 0.028^2}{16} + \frac{((0.035 + 0.050)/2)^2}{12} \right) = 143.8 \times 10^{-6} \text{ kg m}^2 .$$

These estimates model the coupling half as a hollow cylinder and it is therefore not surprising that the estimated mass is greater than the figures from [**142**]. Although the estimated torsional inertia is very close to the value from [**142**], it is surprising that it is slightly lower.

The driving half of the coupling is modelled using the following data:-

```
static float J_dr1=194.2e-6F;        // driving pulley, coupling half etc. inertia, (kg m^2)
```

and the coupling characteristics are modelled as follows:-

```
static float k_dr=10100.F;           // torsional stiffness of drive referred to driving shaft, (N m/rad)
static float c_dr=0.797F;            // torsional damping coefficient of drive referred to driving shaft,
                                             (N m sec/rad)
static long double u=1.L;            // drive ratio
```

The driven half of the coupling is modelled along with the ballscrew.

A8.6.4 **Ballscrew data**

This is given on PGM / THK Group drawing PSM9137.1, issue B, dated 22.6.99. It is included in the C program as follows:-

```
static float m_dr2=0.310F;           // driven pulley, coupling half etc. mass, (kg)
static float J_dr2=194.2e-6F;        // driven pulley, coupling half etc. rotational inertia, (kg m^2)
static float Jt_dr2=143.8e-6F;       // driven pulley, coupling half etc. transverse inertia, (kg m^2)
(static float d_bsO=40.0e-3F;        // outer diameter of ballscrew, (m) – PGM / THK data)
static float d_bsO=37.5e-3F;         // outer diameter of ballscrew, (m) – see Ch 8, §8.1.3, p 151
static float d_bsI=10.0e-3F;         // inner diameter of ballscrew, (m)
static float dp_bs=42.0e-3F;         // pitch diameter of ballscrew, (m)
static float d_trace=0.045F;         // diameter of bearing thrust race, (m)
static float l_bs1=0.109F;           // length of ballscrew pulley to bearing, (m)
static float l_bs2=1.183F;           // length of ballscrew between bearings, (m)
static float l_bs3=0.052F;           // length of ballscrew bearing to end, (m)
static signed int Ns=1;              // number of starts of ballscrew grooves
static float pitch=16.0e-3F;         // ballscrew pitch, (m)
static signed int SignB=+1;          // hand of helix,   -1 = left hand screw
                                     //                  +1 = right hand screw
```

The ballscrew shaft is made of steel. The material properties are assumed to be:-

```
static float E=207.0e9F;             // Young's modulus, (N/m^2)
static float G=80.0e9F;              // shear modulus, (N/m^2)
static float rho=7860.F;             // density, (kg/m^3)
```

The stiffness of the ballscrew shaft elements is calculated as explained in Chapters 5 and 6. In addition the damping properties of the shaft are modelled by including viscous damping elements whose coefficients of damping are assumed to be a multiple of the respective spring stiffness. Light material damping is achieved by assuming a damping ratio as follows:-

```
static float drs=0.00002F;           // damping ratio - shaft
```

The ballscrew nut axial stiffness for a 40 mm diameter 16 mm pitch ballscrew ranges from 540 N/µm to 1500 N/µm, see “LM system ball screws”, catalog. no.200-2BE, THK Co. Ltd., p 177. The values used in the version of LG_rig.cpp dated 08/12/07 are:-

```
static float k_ax_nut=1.2e9F;        // axial stiffness of nut, (N/m)
static float ky_nut=1.2e9F;          // radial stiffness of nut - "y" direction, (N/m)
static float kz_nut=1.2e9F;          // radial stiffness of nut - "z" direction, (N/m)
static float kOy_nut=0.7e5F;         // tilt stiffness of nut - "Oy" direction, (N m/rad)
static float kOz_nut=0.7e5F;         // tilt stiffness of nut - "Oz" direction, (N m/rad)
```

In addition the damping properties of the ballscrew nut are modelled by including viscous damping elements whose coefficients of damping are assumed to be a multiple of the respective spring stiffness. The damping ratio assumed is as follows:-

```
static float drn=0.0005F;          // damping ratio – nut.
```

The friction in the nut is represented by treating the ballscrew as an equivalent lead screw using a coefficient of friction as follows:-

```
static float mu_scr=0.019F;        // nominal coefficient of friction of screw nut.
```

The following measured data (Mode 0) is also available:-

```
static double Absnf=1.44;          // constant term in "Mode 0" ballscrew friction model, (N m)
static double Bbsnf=0.;            // speed-dependent term in "Mode 0" ballscrew friction model,
 (N m sec/rad)
```

The monitoring rings, fitted to the ballscrew as targets for the non-contacting displacement transducers, which also carry the out-of-balance weights, have the following properties:-

```
static float m_Mring=1.029F;       // mass of monitoring ring, (kg)
static float Jtor_Mring=0.001289F; // torsional inertia of monitoring ring, (kg m^2)
static float Jtil_Mring=0.000701F; // tilt inertia of monitoring ring, (kg m^2)
static float rOoB=40.e-3F;         // radius of out-of-balance weight, (m)
```

A8.6.5 **Ballscrew support bearings**

A pair of axial angular contact ball bearings is fitted close to each end of the ballscrew. These are:-

type	ZKLF 3080.2RS
mass	0.78 kg
inner diameter	30 mm
outer diameter	80 mm
length	28 mm
bearing frictional torque	0.5 N m
axial rigidity	850 N/μm
tilting rigidity	300 N m/mrad

Source – “Bearings for screw drives, Axial angular contact ball bearings, Needle roller/axial cylindrical roller bearings”, INA Publication ZAE, Sach-Nr.002-045-869 Δ, pp 40-41

See also
http://medias.ina.de/medias/de!hp.ec.br.pr/ZKLF...2RS*ZKLF%203080.2RS;aELuDt9h4_-7

The stiffness of the ballscrew bearings is modelled in the version of LG_rig.cpp dated 08/12/07 using the following values:-

```
static float k_ax_1=850.e6F;       // axial stiffness of bearings, (N/m)
static float k_ax_2=850.e6F;
static float ky_rad_1=1275.e6F;    // radial stiffness of bearings - "y" direction, (N/m)
static float ky_rad_2=1275.e6F;
static float kz_rad_1=1275.e6F;    // radial stiffness of bearings - "z" direction, (N/m)
static float kz_rad_2=1275.e6F;
static float ky_tilt_1=160.e3F;    // tilt stiffness of bearings - "Oy" direction, (N m/rad)
static float ky_tilt_2=160.e3F;    // Note!!  Too much bearing tilt stiffness crashes solution method
static float kz_tilt_1=160.e3F;    // tilt stiffness of bearings - "Oz" direction, (N m/rad)
static float kz_tilt_2=160.e3F;
```

In addition the damping properties of the bearings are modelled by including viscous damping elements whose coefficients of damping are assumed to be a multiple of the respective spring stiffness. A damping ratio is included in the C program, a typical value (09/12/07) is:-

```
static float drb=0.0005F;          // damping ratio - bearings
```

Terms are included in the model to account for viscous drag in the ballscrew bearings. They were estimated in a similar manner to the corresponding term for the motor bearings and are:-

```
static float c_brg_1=0.668e-3F;      // equivalent damping coefficient of bearings, (N m sec/rad)
static float c_brg_2=0.668e-3F;
```

When using measured data if available (Mode 0):-

```
static float c_brg_1=0.F;            // equivalent damping coefficient of bearings, (N m sec/rad)
static float c_brg_2=0.F;            // Note:  c_brg_i=0.F when comparing with "bs_model.m" on 08/12/07.
```

Further an allowance for Coulomb friction in the ballscrew bearings is included (Mode 1). In addition to the code set out in Section A8.6.2 the following two lines are included for the ballscrew bearings:-

```
rEnD[j+1][1]+=fric_brg(mu_brg,0.F,T_mot_max,1,double(n_mot_max)*2.*pi/60.,j,dt1,0,mode);
        // - ballscrew bearing 1
rEnD[j+1][3]+=fric_brg(mu_brg,float(fabs(k_ax_1*float(d[MDR+NFS*B1+1][j]))),float(u)*T_mot_max,
        MDR+NFS*B1+4,-v_max/bs_rat,j,dt1,1,mode);
        // - ballscrew bearing 2
```

A8.6.6 **The co-ordinate system**

The **origin** of the coordinate system is at the motor end of the ballscrew.

The **X axis** is horizontal along the axis of the un-deflected ballscrew in the direction of the free end.

The **Y axis** is horizontal, parallel to the W4 corridor and towards the small offices at the end of the W4/03 workshop.

The **Z axis** is vertically upwards.

The **nodes** are numbered from zero upwards starting at the motor end of the ballscrew. In these examples the ballscrew has 25 elements, the drive end bearing being at Node 1 and the non-drive end bearing being at Node 24. The x co-ordinates of the various nodes are thus:-

Node	x co-ordinate, (m)	Remarks		Node	x co-ordinate, (m)	Remarks
0	0			13	0.728217	
1	0.111	B1		14	0.779652	
2	0.162435			15	0.831087	
3	0.21387			16	0.882522	
4	0.265304			17	0.933957	
5	0.316739			18	0.985391	
6	0.368174			19	1.03683	
7	0.419609			20	1.08826	
8	0.471043			21	1.1397	
9	0.522478			22	1.19113	
10	0.573913			23	1.24257	
11	0.625348			24	1.294	B2
12	0.676783			25	1.346	

A8.6.7 **Other data**

The linear dimensions of the model are based on the ballscrew drawing and bearing data as follows:-

Overall length = 1346 mm, (drawing)

Distance from drive end to centre of drive end bearing = 125 mm, (drawing, the bearing buts up to the ballscrew portion) – 28 mm, (bearing data, bearing length)/2 = 111 mm

Distance from drive end of ballscrew portion to estimated centre of mass of coupling half = 28 mm, (bearing data) + 60 mm, (measured from rig, drive end of drive end bearing to non-drive end of coupling half) + ((35+50)/2)/2 mm, (measured from rig) = 109.25 mm

Distance from centre of non-drive end bearing to end of ballscrew = 136 mm, (drawing, non-drive end to the ballscrew portion) – 98 mm, (measured from rig, non-drive end of non-drive end bearing to end of ballscrew portion) + 28 mm, (bearing data)/2 = 52 mm.

Therefore the lengths of the elements are as follows:-

Element 1	*l_bs1*	0.109 m
Elements 2 to 24	*l_bs2*/23	1.183/23=0.051435 m
Element 25	*l_bs3*	0.052 m

From data used in the thermal model (Appendix 7.1):-

```
Ltbl=1.755;        % length of table
AtblSection=0.026; % area of table cross section (m^2)
```

Therefore

Weight of “table” = 1.722 × 0.026 × 7860 = 352 kg

The load moved by the ballscrew, often referred to as the “table”, is in fact a saddle in the case of the linear guide rig. Its properties are modelled as follows:-

```
static float m_tab=352.F;          // mass of saddle/table, (kg)
static float Jy_tab=0.546F;        // inertia of saddle/table about "y" axis, (kg m^2)
static float Jz_tab=120.F;         // inertia of saddle/table about "z" axis, (kg m^2)
static float z_tab=125.e-3F;       // height of saddle/table above ballscrew, (m)
static float z_slid=75.e-3F;       // height of saddle/table above slideway, (m)
static float z_lenc=155.e-3F;      // height of saddle/table above linear encoder, (m), (not used)
static float mu_tab=0.0341F;       // coefficient of friction between saddle and slideways.
```

The stiffness of the linear guides on which the saddle is mounted is represented by a set of springs with values:-

```
static float ky_slide=1.e9F;       // stiffness of slides supporting nut - "y" direction, (N/m)
static float kz_slide=1.e9F;       // stiffness of slides supporting nut - "z" direction, (N/m)
static float kOy_slide=10.4e6F;    // stiffness of slides supporting nut - "Oy" direction, (N m/rad)
static float kOz_slide=10.4e6F;    // stiffness of slides supporting nut - "Oz" direction, (N m/rad)
```

In addition the damping properties of the slideways are modelled by including viscous damping elements whose coefficients of damping are assumed to be a multiple of the respective spring stiffness. The damping ratio assumed is:-

```
static float drsl=0.0005F;    // damping ratio - slide
```

The following data is for the version of the C program used for comparison with the data generated by "bs_model.m" on 08/12/07.

G:\PhD_archive_191207\C_drive_1\C_progs\Test_VCpp\LG_rig.cpp, 08/12/07 19:34
- Parameters used

```
// Basic solution parameters
        #define MDR 2           // Number of degrees of freedom in motor drive (in addition to ballscrew)
        #define SMAX 50         // Maximum number of elements in ballscrew
        #define NFS 6           // Number of degrees of freedom per ballscrew node
        #define BWS 12          // Bandwidth of ballscrew
        #define NMAX 0          // Maximum number of elements in ballscrew nut, (0 = 1 node)
        #define NFN 5           // Number of degrees of freedom per nut node
        #define BWN 5           // Bandwidth of nut
        #define KMAX 42         // Maximum number of iterations
        #define NBLOCK 1000     // Maximum number of steps in time history block
        #define SWRIT 100       // Number of time history steps between screen writes
        #define TINY 1.0e-20    // Used in subroutine "ludcmp"
        #define NSOURCE 8       // Number of heat sources
        #define NAVEMAX 250     // Number of terms in "moving average" filter

//% Common data block
//% ===============
//% Fundamental constants
        static float g=9.80665F;    // acceleration due to gravity, (m/sec^2)
        static double pi;           // pi

//% Dimensional data etc.
//% Motor
//% --------
        static float i_mot_max=41.0F; // peak motor current, (A)

//% Ballscrew
//% ------------
        static signed int ij=2;     // correction for "jointed track" phenomenon, 0 = none

//%
//% Controller constants
        static double a_max=6.;     // maximum acceleration, (m/sec^2).
        static double G_in=2.e-3;   // input gain for white noise, (m/V)

//%
//% Input variables
        signed int S=25;            // number of elements in ballscrew
        signed int N=0;             // number of elements in ballscrew nut, (0 = 1 node)
        signed int B1=1;            // node number at Bearing 1, (node number of first node in ballscrew = 0)
        signed int B2=24;           // node number at Bearing 2
        double xs=1079.e-3F;        // "x" coordinate of start point, (m)
        double xf=xs-460.e-3F;      // "x" coordinate of end of deceleration, (m)
        double v_max=20./60.;       // maximum nut velocity, (m/sec)
        double t1_=0.;              // time of start of acceleration, (sec)
        double tacc=1./9.;          // acceleration time, (sec)
        double tdec=1./9.;          // deceleration time, (sec)
        float blash=0.F;            // backlash, (m)
```

```
float PT=14.e-6F;           // pre-tension, (m/m)
float xMR1=l_bs1+133.e-3F; // "x" coordinate of monitoring ring 1
float xMR2=l_bs1+276.e-3F; // "x" coordinate of monitoring ring 2
mOoB1=10.e-3F;              // mass of out-of-balance weight 1, (kg) **
phOoB1=0.F;                 // phase of out-of-balance weight 1, (deg)
mOoB2=10.e-3F;              // mass of out-of-balance weight 2, (kg) **
phOoB2=0.F;                 // phase of out-of-balance weight 2, (deg)
float dt_in=50.e-6F;        // input time step, (sec)
signed int i_filt=0;        // low-pass Butterworth filter between controller velocity loop and power
                                amplifier
                            //      0 = no filter, 1 = 200 Hz, 2 = 400 Hz, 3 = 2000 Hz
                            //      -1 = take moving average over 3 msec
signed int timp=0;          // mode of input for demand data
                            //      0 = trapezoidal demand
                            //      1 = white noise mode
signed int comp=0;
double V_in=0.6;            // input signal - mode 1, (V)
double rms=2.0069;          // standard deviation of random signal on file
double t_int=2.e-3;         // sampling time - mode 1, (sec)
signed int N_int=0;         // number of main loops in time t_int
plst=3.0e-3;                // time step – controller position loop, (sec)
vlst=0.6e-3;                // time step – controller velocity loop, (sec)
ilst=0.2e-3;                // time step – controller current loop, (sec)
// Use zero to trigger end of out-of-balance excitations
iOoB[1]=0;                  OoBa[1]=0.L;                    OoBp[1]=0.;
```

//%

****** These levels of out-of-balance were used for comparison with "bs_model.m" which deals with situations where the screw is not put out of balance deliberately. The values used where the ballscrew is put out of balance to generate measurable levels of vibration are typically:-

```
mOoB1=0.F;          // mass of out-of-balance weight 1, (kg)
mOoB2=124.e-3F;     // mass of out-of-balance weight 2, (kg).
```

Appendix 9.1 - **Heidenhain controller theory**

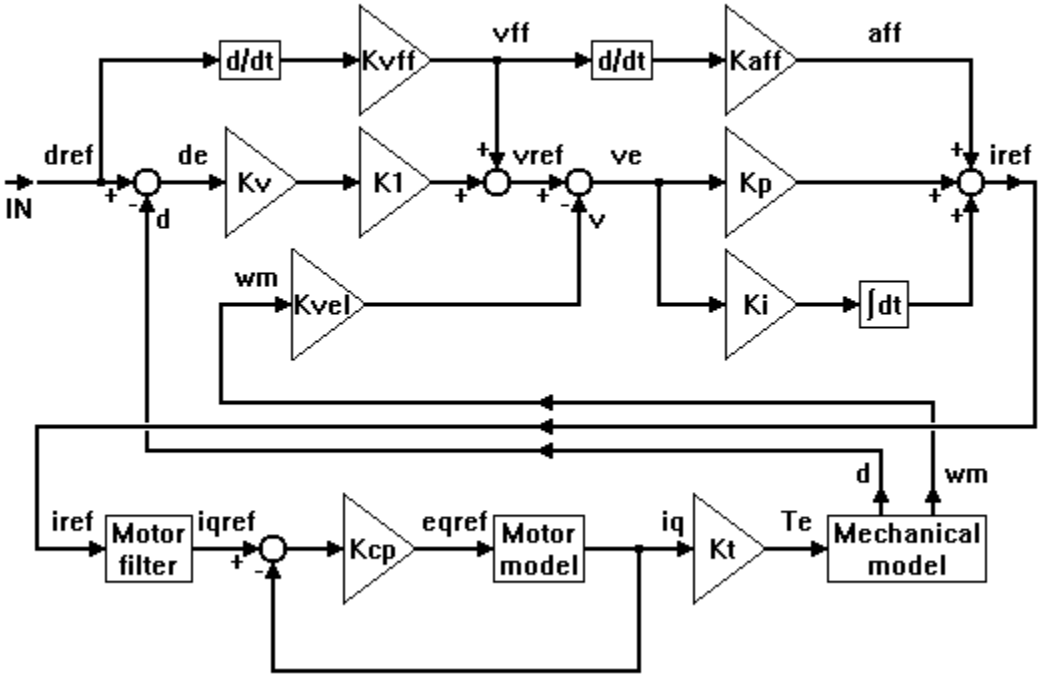


Figure A9.1.1 *A block diagram of the Heidenhain controller*

Consider a mass *m* mounted on a ballscrew which is being driven by a Heidenhain controller, (see Figure A9.1.1). This controller has a proportional constant K_p and an integral constant K_i. There is an acceleration feed-forward term which is in effect $K_{vff} \times K_{aff}$. The reference current feeding into the motor controller is defined by the equation:-

$$i_{ref} = K_p v_e + K_i \int v_e .dt + K_{vff} K_{aff} \ddot{D}$$

where

i_{ref} = the motor controller reference current, (A)

K_p = the PI controller proportional gain, (A sec/rev)

v_e = the velocity error, (rev/sec)

K_i = the PI controller integral gain, (A/rev)

t = time, (sec)

K_{vff} = the velocity fast forward gain, (1/m)

D = the demand position of the mass, (m)

The velocity error is defined as the difference between the reference velocity and the speed of the ballscrew nut as determined at the motor, viz:-

$$v_e = v_{ref} - K_{vel}\,\omega$$

where

v_{ref} = the reference velocity, (rev/sec)

K_{vel} = is a constant to convert from rad/sec to rev/sec, (= 1/(2π))

ω = motor speed, (rad/sec)

The reference velocity is determined from the position error and the velocity demand as follows:-

$$v_{ref} = K_v K_1 \times (D - x) + K_{vff}\dot{D}$$

where

$K_v\,K_1$ = a pair of constants which together define a component of the velocity demand in terms of the position error, (rev/(m sec))

x = ballscrew nut movement (sensed at motor), (m)

whence

$$\begin{aligned} v_e &= K_v K_1 \times (D - x) + K_{vff}\dot{D} - K_{vel}\,\omega \\ &= K_v K_1 \times (D - x) + K_{vff}\dot{D} - \frac{2\pi K_{vel}}{N_s\,p}\dot{x} \end{aligned}$$

and

$$\begin{aligned} \int v_e.dt &= K_v K_1 \times (\int D.dt - \int x.dt) + K_{vff}\int \dot{D}.dt - \frac{2\pi K_{vel}}{N_s\,p}\int \dot{x}.dt \\ &= K_v K_1 \times (\int D.dt - \int x.dt) + K_{vff}\int \frac{dD}{dt}.dt - \frac{2\pi K_{vel}}{N_s\,p}\int \frac{dx}{dt}.dt \\ &= K_v K_1 \times (\int D.dt - \int x.dt) + K_{vff}\int dD - \frac{2\pi K_{vel}}{N_s\,p}\int dx \\ &= K_v K_1 \times (\int D.dt - \int x.dt) + K_{vff}\,D - \frac{2\pi K_{vel}}{N_s\,p}\,x \end{aligned}$$

Current feed-back gives rise to:-

$$e_{q\,ref} = K_{cp} \times (i_{q\,rev} - i_q)$$

where

$e_{q\,ref}$ = the motor applied voltage, (V)

K_{cp} = the motor controller electrical constant, (V/A)

i_q = motor current, (A)

The electrical behaviour of the motor is given by:-

$$L\frac{di_q}{dt} = e_{q\,ref} - Ri_q - \frac{2\pi N_p \lambda}{N_s p}\dot{x}$$

where

L = the motor inductance, (H=V sec/A)

R = the motor resistance, (Ω=V/A)

N_p = the number of motor pole pairs

λ = the motor back e.m.f. constant, (V sec/rad)

N_s = the number of starts on the ballscrew

p = the ballscrew pitch, (rad)

The torque applied by the electrical windings to the motor rotor is given by:-

$$F = K_t \times i_q$$

where

F = electrical torque, (N m)

K_t = motor mechanical constant, (N m/A)

The equation of motion for the motor rotor (and any load rigidly attached to it) in terms of motion of the ballscrew nut is:-

$$m\ddot{x} + c\dot{x} = \frac{2\pi}{N_s p}F + F_A$$

where

m = the motor and ballscrew inertia referred to linear motion plus mass of load, (kg)

c = viscous damping on mass movement, (N sec/m)

F_A = applied force, (N)

Substituting for the motor voltage gives the electrical differential equation as:-

$$L\frac{di_q}{dt} = K_{cp} \times (i_{q\,ref} - i_q) - R\,i_q - \frac{2\pi N_p \lambda}{N_s p}\dot{x}$$

$$L\frac{di_q}{dt} + (K_{cp} + R) \times i_q + \frac{2\pi N_p \lambda}{N_s p}\dot{x} = K_{cp} i_{q\,ref}$$

By eliminating the torque, the mechanical equation of motion can be rendered in electrical terms as:-

$$i_q = \frac{N_s pm}{2\pi K_t}\ddot{x} + \frac{N_s pc}{2\pi K_t}\dot{x} - \frac{N_s p}{2\pi K_t}F_A$$

and

$$\frac{di_q}{dt} = \frac{N_s pm}{2\pi K_t}\dddot{x} + \frac{N_s pc}{2\pi K_t}\ddot{x}$$

Substituting for i_m in the electrical differential equation gives:-

$$L\times(\frac{N_s pm}{2\pi K_t}\dddot{x} + \frac{N_s pc}{2\pi K_t}\ddot{x}) + (K_{cp}+R)\times(\frac{N_s pm}{2\pi K_t}\ddot{x} + \frac{N_s pc}{2\pi K_t}\dot{x} - \frac{N_s p}{2\pi K_t}F_A) + \frac{2\pi N_p \lambda}{N_s p}\dot{x} = K_{cp} i_{q\,ref}$$

$$\frac{Lm}{K_t}\dddot{x} + (\frac{Lc}{K_t} + \frac{K_{cp}m}{K_t} + \frac{Rm}{K_t})\ddot{x} + (\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p \lambda)\dot{x} = K_{cp}\frac{2\pi}{N_s p} i_{q\,ref} + \frac{K_{cp}+R}{K_t}F_A$$

In the absence of a filter $i_{qref} = i_{ref}$. Substituting for i_{ref} gives:-

$$\frac{Lm}{K_t}\dddot{x} + (\frac{Lc}{K_t} + \frac{K_{cp}m}{K_t} + \frac{Rm}{K_t})\ddot{x} + (\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p \lambda)\dot{x}$$
$$= K_{cp}\frac{2\pi}{N_s p}(K_p v_e + K_i \int v_e.dt + K_{vff}K_{aff}\ddot{D}) + \frac{K_{cp}+R}{K_t}F_A$$

Substituting for v_e and $\int v_e.dt$:-

$$\frac{Lm}{K_t}\dddot{x} + (\frac{Lc}{K_t} + \frac{K_{cp}m}{K_t} + \frac{Rm}{K_t})\ddot{x} + (\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p \lambda)\dot{x}$$
$$= \frac{2\pi K_{cp}K_p}{N_s p}(K_v K_1 \times (D - x) + K_{vff}\dot{D} - \frac{2\pi K_{vel}}{N_s p}\dot{x})$$
$$+ \frac{2\pi K_{cp}K_i}{N_s p}(K_v K_1 \times (\int D.dt - \int x.dt) + K_{vff}D - \frac{2\pi K_{vel}}{N_s p}x)$$
$$+ \frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p}\ddot{D} + \frac{K_{cp}+R}{K_t}F_A$$

$$\frac{Lm}{K_t}\dddot{x}+(\frac{Lc}{K_t}+\frac{K_{cp}m}{K_t}+\frac{Rm}{K_t})\ddot{x}+(\frac{K_{cp}c}{K_t}+\frac{Rc}{K_t}+(\frac{2\pi}{N_s p})^2 N_p\lambda+K_{cp}K_pK_{vel}(\frac{2\pi}{N_s p})^2)\dot{x}$$
$$+(\frac{2\pi K_{cp}K_pK_vK_1}{N_s p}+K_{cp}K_iK_{vel}(\frac{2\pi}{N_s p})^2)x+\frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times\int x.dt$$
$$=\frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\int D.dt+(\frac{2\pi K_{cp}K_iK_{vff}}{N_s p}+\frac{2\pi K_{cp}K_pK_vK_1}{N_s p})D+\frac{2\pi K_{cp}K_pK_{vff}}{N_s p}\dot{D}$$
$$+\frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p}\ddot{D}+\frac{K_{cp}+R}{K_t}\times F_A$$

This is a fourth order differential equation in $\int x.dt=\theta$, say, which can be put in a simpler form as:-

$$a\dot{\theta}^{\times 4}+b\dddot{\theta}+c\ddot{\theta}+d\dot{\theta}+e\theta=A\dot{D}+BD+C\int D.dt+EF_A$$

The homogeneous equation has either a solution of the form:-

$$\theta'=\alpha e^{aat}+\beta e^{bbt}+\gamma e^{cct}+\delta e^{ddt}$$

or a solution of the form:-

$$\theta'=\alpha e^{aat}+\beta e^{bbt}+\gamma e^{(cc+idd)t}+\delta e^{(cc-idd)t}$$

or a solution of the form:-

$$\theta'=\alpha e^{(aa+ibb)t}+\beta e^{(aa-ibb)t}+\gamma e^{(cc+idd)t}+\gamma e^{(cc+idd)t}$$

where *aa*, *bb*, *cc* and *dd* are solutions of the equation $ax^4+bx^3+cx^2+dx+e=0$ and α, β, γ and δ are determined by the boundary conditions. The displacement is given by:-

$$x'=\dot{\theta}'=aa\alpha e^{aat}+bb\beta e^{bbt}+cc\gamma e^{cct}+dd\delta e^{ddt},$$

or similar. If the system is to be stable and the roots of $ax^4+bx^3+cx^2+dx+e=0$ are all real, they also must all be negative. If two or four of the roots form complex conjugate pairs, the real components must be negative. This must be the case for a system that works, so let us assume that after a while the transient activity that the homogeneous equation represents fades away, and let us consider three cases of demand.

Let us consider the response of the system to three different types of demand.

1. ***A constant position***

Let $D = P$, a constant, and let $\int D.dt = Pt + A$, where A is an arbitrary constant of integration. Then $\dot{D} = 0$ and $\ddot{D} = 0$. Let us try as a solution $x = K$, a constant and let $\int x.dt = Kt + \alpha$ where α is a constant. Then $\dot{x} = 0$, $\ddot{x} = 0$, $\dddot{x} = 0$ and:-

$$\begin{aligned}
&\frac{Lm}{K_t}\times 0 + (\frac{Lc}{K_t} + \frac{K_{cp}m}{K_t} + \frac{Rm}{K_t})\times 0 + (\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p \lambda + K_{cp}K_p K_{vel}(\frac{2\pi}{N_s p})^2)\times 0 \\
&\quad + (\frac{2\pi K_{cp}K_p K_v K_1}{N_s p} + K_{cp}K_i K_{vel}(\frac{2\pi}{N_s p})^2)\times K + \frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(Kt+\alpha) \\
&= \frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(Pt+A) + (\frac{2\pi K_{cp}K_i K_{vff}}{N_s p} + \frac{2\pi K_{cp}K_p K_v K_1}{N_s p})\times P + \frac{2\pi K_{cp}K_p K_{vff}}{N_s p}\times 0 \\
&\quad + \frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p}\times 0 + \frac{K_{cp}+R}{K_t}\times F_A
\end{aligned}$$

$$\begin{aligned}
&(\frac{2\pi K_{cp}K_p K_v K_1}{N_s p} + K_{cp}K_i K_{vel}(\frac{2\pi}{N_s p})^2)\times K + \frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(Kt+\alpha) \\
&= \frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(Pt+A) + (\frac{2\pi K_{cp}K_i K_{vff}}{N_s p} + \frac{2\pi K_{cp}K_p K_v K_1}{N_s p})\times P + \frac{K_{cp}+R}{K_t}\times F_A
\end{aligned}$$

Comparing coefficients of t suggests:-

$$K = P$$

Now $K_{vff} = \frac{1}{N_s p}$ and $K_{vel} = \frac{1}{2\pi}$, and so:-

$$\alpha = A + \frac{\frac{1}{N_s p} - \frac{1}{2\pi}\times(\frac{2\pi}{N_s p})}{K_v K_1}\times P + \frac{N_s p\times(K_{cp}+R)}{2\pi K_{cp}K_i K_v K_1 K_t}\times F_A = A + \frac{N_s p\times(K_{cp}+R)}{2\pi K_{cp}K_i K_v K_1 K_t}\times F_A$$

which gives the general solution for x as:-

$$x = D + x'.$$

and the general solution for $\int x.dt$ as:-

$$\int x.dt = \int D.dt + \frac{N_s p\times(K_{cp}+R)}{2\pi K_{cp}K_i K_v K_1 K_t}\times F_A + \theta'.$$

2. ***A constant velocity***

Let $D = at + b$ where a and b are constants and let $\int D.dt = a\frac{t^2}{2} + bt + G$, where G is an arbitrary constant of integration. Then $\dot{D} = a$ and $\ddot{D} = 0$. Let us try as a solution $x = \alpha t + \beta$, where α and β are constants, and let $\int x.dt = \alpha\frac{t^2}{2} + \beta t + \gamma$ where γ is a constant. Then $\dot{x} = \alpha$, $\ddot{x} = 0$, $\dddot{x} = 0$ and:-

$$\begin{aligned}
&\frac{Lm}{K_t}\times 0 + (\frac{Lc}{K_t} + \frac{K_{cp}m}{K_t} + \frac{Rm}{K_t})\times 0 + (\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p\lambda + K_{cp}K_pK_{vel}(\frac{2\pi}{N_s p})^2)\times\alpha \\
&\quad + (\frac{2\pi K_{cp}K_pK_vK_1}{N_s p} + K_{cp}K_iK_{vel}(\frac{2\pi}{N_s p})^2)\times(\alpha t + \beta) + \frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times(\alpha\frac{t^2}{2} + \beta t + \gamma) \\
&= \frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times(a\frac{t^2}{2} + bt + G) + (\frac{2\pi K_{cp}K_iK_{vff}}{N_s p} + \frac{2\pi K_{cp}K_pK_vK_1}{N_s p})\times(at + b) \\
&\quad + \frac{2\pi K_{cp}K_pK_{vff}}{N_s p}\times a + \frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p}\times 0 + \frac{K_{cp} + R}{K_t}\times F_A
\end{aligned}$$

$$\begin{aligned}
&(\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p\lambda + K_{cp}K_pK_{vel}(\frac{2\pi}{N_s p})^2)\times\alpha \\
&\quad + (\frac{2\pi K_{cp}K_pK_vK_1}{N_s p} + K_{cp}K_iK_{vel}(\frac{2\pi}{N_s p})^2)\times(\alpha t + \beta) + \frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times(\alpha\frac{t^2}{2} + \beta t + \gamma) \\
&= \frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times(a\frac{t^2}{2} + bt + G) + (\frac{2\pi K_{cp}K_iK_{vff}}{N_s p} + \frac{2\pi K_{cp}K_pK_vK_1}{N_s p})\times(at + b) \\
&\quad + \frac{2\pi K_{cp}K_pK_{vff}}{N_s p}\times a + \frac{K_{cp} + R}{K_t}\times F_A
\end{aligned}$$

Comparing coefficients of t^2 gives $\alpha = a$ and:-

$$\begin{aligned}
&(\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p\lambda)\times a + (\frac{2\pi K_{cp}K_pK_vK_1}{N_s p} + \frac{2\pi K_{cp}K_i}{(N_s p)^2})\times\beta + \frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times(\beta t + \gamma) \\
&= \frac{2\pi K_{cp}K_iK_vK_1}{N_s p}\times(bt + G) + (\frac{2\pi K_{cp}K_pK_vK_1}{N_s p} + \frac{2\pi K_{cp}K_i}{(N_s p)^2})\times b + \frac{K_{cp} + R}{K_t}\times F_A
\end{aligned}$$

Comparing coefficients of t gives $\beta = b$ and:-

$$\gamma = G - \frac{\frac{N_s pK_{cp}c}{K_t} + \frac{N_s pRc}{K_t} + \frac{(2\pi)^2 N_p}{N_s p}\lambda}{2\pi K_{cp}K_iK_vK_1}\times a + \frac{N_s p\times(K_{cp} + R)}{2\pi K_{cp}K_iK_vK_1K_t}\times F_A$$

This gives the general solution for *x* as:-

$$x = D + x' .$$

and the general solution for $\int x.dt$ as:-

$$\int x.dt = \int D.dt - \frac{\frac{N_s pK_{cp}c}{K_t} + \frac{N_s pRc}{K_t} + \frac{(2\pi)^2 N_p}{N_s p}\lambda}{2\pi K_{cp}K_i K_v K_1} \times \dot{D} + \frac{N_s p \times (K_{cp} + R)}{2\pi K_{cp}K_i K_v K_1 K_t} \times F_A + \theta' .$$

3. ***A constant acceleration***

Let $D = s + ut + \frac{1}{2} f t^2$ where *s*, *u* and *f* are constants and let $\int D.dt = f\frac{t^3}{6} + u\frac{t^2}{2} + st + G$, where *G* is an arbitrary constant of integration. Then $\dot{D} = u + ft$ and $\ddot{D} = f$. Let us try as a solution $x = \sigma + \upsilon t + \frac{1}{2}\phi t^2$, where σ, υ and ϕ are constants, and let $\int x.dt = \phi\frac{t^3}{6} + \upsilon\frac{t^2}{2} + \sigma t + \gamma$ where γ is a constant. Then $\dot{x} = \upsilon + \phi t$, $\ddot{x} = \phi$, $\dddot{x} = 0$ and:-

$$\begin{aligned}
&\frac{Lm}{K_t} \times 0 + (\frac{Lc}{K_t} + \frac{K_{cp}m}{K_t} + \frac{Rm}{K_t}) \times \phi + (\frac{K_{cp}c}{K_t} + \frac{Rc}{K_t} + (\frac{2\pi}{N_s p})^2 N_p \lambda + K_{cp}K_p K_{vel}(\frac{2\pi}{N_s p})^2) \times (\upsilon + \phi t) \\
&\quad + (\frac{2\pi K_{cp}K_p K_v K_1}{N_s p} + K_{cp}K_i K_{vel}(\frac{2\pi}{N_s p})^2) \times (\sigma + \upsilon t + \frac{1}{2}\phi t^2) \\
&\quad + \frac{2\pi K_{cp}K_i K_v K_1}{N_s p} \times (\phi\frac{t^3}{6} + \upsilon\frac{t^2}{2} + \sigma t + \gamma) \\
&= \frac{2\pi K_{cp}K_i K_v K_1}{N_s p} \times (f\frac{t^3}{6} + u\frac{t^2}{2} + st + G) \\
&\quad + (\frac{2\pi K_{cp}K_i K_{vff}}{N_s p} + \frac{2\pi K_{cp}K_p K_v K_1}{N_s p}) \times (s + ut + \frac{1}{2} f t^2) + \frac{2\pi K_{cp}K_p K_{vff}}{N_s p} \times (u + ft) \\
&\quad + \frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p} \times f + \frac{K_{cp} + R}{K_t} \times F_A
\end{aligned}$$

Comparing coefficients of t^3 gives $\phi = f$ and:-

$$(\frac{Lc}{K_t}+\frac{K_{cp}m}{K_t}+\frac{Rm}{K_t})\times f+(\frac{K_{cp}c}{K_t}+\frac{Rc}{K_t}+(\frac{2\pi}{N_s p})^2 N_p\lambda+K_{cp}K_p K_{vel}(\frac{2\pi}{N_s p})^2)\times(\upsilon+ft)$$
$$+(\frac{2\pi K_{cp}K_p K_v K_1}{N_s p}+K_{cp}K_i K_{vel}(\frac{2\pi}{N_s p})^2)\times(\sigma+\upsilon t)+\frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(\upsilon\frac{t^2}{2}+\sigma t+\gamma)$$
$$=\frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(u\frac{t^2}{2}+st+G)+(\frac{2\pi K_{cp}K_i K_{vff}}{N_s p}+\frac{2\pi K_{cp}K_p K_v K_1}{N_s p})\times(s+ut)$$
$$+\frac{2\pi K_{cp}K_p K_{vff}}{N_s p}\times(u+ft)+\frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p}\times f+\frac{K_{cp}+R}{K_t}\times F_A$$

Comparing coefficients of t^2 gives $\upsilon = u$ and:-

$$(\frac{Lc}{K_t}+\frac{K_{cp}m}{K_t}+\frac{Rm}{K_t})\times f+(\frac{K_{cp}c}{K_t}+\frac{Rc}{K_t}+(\frac{2\pi}{N_s p})^2 N_p\lambda)\times(u+ft)$$
$$+(\frac{2\pi K_{cp}K_p K_v K_1}{N_s p}+K_{cp}K_i K_{vel}(\frac{2\pi}{N_s p})^2)\times(\sigma+ut)+\frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(\sigma t+\gamma)$$
$$=\frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times(st+G)+(\frac{2\pi K_{cp}K_i K_{vff}}{N_s p}+\frac{2\pi K_{cp}K_p K_v K_1}{N_s p})\times(s+ut)$$
$$+\frac{2\pi K_{cp}K_{vff}K_{aff}}{N_s p}\times f+\frac{K_{cp}+R}{K_t}\times F_A$$

Comparing coefficients of t gives:-

$$\frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times\sigma=\frac{2\pi K_{cp}K_i K_v K_1}{N_s p}\times s-(\frac{K_{cp}c}{K_t}+\frac{Rc}{K_t}+(\frac{2\pi}{N_s p})^2 N_p\lambda)\times f$$

or:-

$$\sigma=s-\frac{\frac{N_s pK_{cp}c}{K_t}+\frac{N_s pRc}{K_t}+\frac{(2\pi)^2}{N_s p}N_p\lambda}{2\pi K_{cp}K_i K_v K_1}\times f$$

and:-

$$\gamma=G-\frac{\frac{N_s pLc}{K_t}+\frac{N_s pK_{cp}m}{K_t}+\frac{N_s pRm}{K_t}}{2\pi K_{cp}K_i K_v K_1}\times f$$
$$+(\frac{K_p}{K_i}+\frac{1}{N_s pK_v K_1})\times\frac{\frac{N_s pK_{cp}c}{K_t}+\frac{N_s pRc}{K_t}+\frac{(2\pi)^2}{N_s p}N_p\lambda}{2\pi K_{cp}K_i K_v K_1}\times f+\frac{K_{vff}K_{aff}}{K_i K_v K_1}\times f$$
$$-\frac{\frac{N_s pK_{cp}c}{K_t}+\frac{N_s pRc}{K_t}+\frac{(2\pi)^2}{N_s p}N_p\lambda}{2\pi K_{cp}K_i K_v K_1}\times u+\frac{N_s p\times(K_{cp}+R)}{2\pi K_{cp}K_i K_v K_1 K_t}\times F_A$$

This gives the general solution for x as:-

$$x = D - \frac{\frac{N_s pK_{cp}c}{K_t} + \frac{N_s pRc}{K_t} + \frac{(2\pi)^2}{N_s p} N_p \lambda}{2\pi K_{cp} K_i K_v K_1} \times \ddot{D} + x' .$$

and the general solution for $\int x.dt$ as:-

$$\begin{aligned}\int x.dt = \int D.dt &- \frac{\frac{N_s pK_{cp}c}{K_t} + \frac{N_s pRc}{K_t} + \frac{(2\pi)^2}{N_s p} N_p \lambda}{2\pi K_{cp} K_i K_v K_1} \times \dot{D} \\ &- \frac{\frac{N_s pLc}{K_t} + \frac{N_s pK_{cp}m}{K_t} + \frac{N_s pRm}{K_t}}{2\pi K_{cp} K_i K_v K_1} \times \ddot{D} \\ &+ (\frac{K_p}{K_i} + \frac{1}{N_s pK_v K_1}) \times \frac{\frac{N_s pK_{cp}c}{K_t} + \frac{N_s pRc}{K_t} + \frac{(2\pi)^2}{N_s p} N_p \lambda}{2\pi K_{cp} K_i K_v K_1} \times \ddot{D} + \frac{K_{vff} K_{aff}}{K_i K_v K_1} \times \ddot{D} \\ &+ \frac{N_s p \times (K_{cp} + R)}{2\pi K_{cp} K_i K_v K_1 K_t} \times F_A + x'\end{aligned} .$$

Comments

Let us consider the behaviour of the controller with a constant position demand and an external applied force. Position is maintained independent of force, which is counter-intuitive and suggests an infinite controller stiffness. Let us consider the situation more carefully. The displacement response, nett of transients, is given by:-

$$\begin{aligned} x &= \frac{d}{dt} \int x.dt \\ &= \frac{d}{dt} \left(\int D.dt + \frac{N_s p \times (K_{cp} + R)}{2\pi K_{cp} K_i K_v K_1 K_t} \times F_A \right). \\ &= D + \frac{N_s p \times (K_{cp} + R)}{2\pi K_{cp} K_i K_v K_1 K_t} \times \frac{dF_A}{dt} \end{aligned}$$

This gives rise to the concept of **controller jerk stiffness** k_{conj} defined by:-

$$k_{conj} = \frac{2\pi K_{cp} K_i K_v K_1 K_t}{N_s p \times (K_{cp} + R)}$$

Thus, when an external force is applied to the system, there is a response depending on the rate at which the force is being applied, but when a steady force is achieved, the system comes back to true position. This analysis neglects transient behaviour predicted by the complementary

equation. In practice, such behaviour may mask completely this phenomenon, but at least the apparent "infinite stiffness" has been explained. In terms of controller jerk stiffness, the position time integral is given by:-

$$\int x.dt = \int D.dt + \frac{F_A}{k_{conj}} + \theta'$$

When a constant velocity is required, after transients have died away, a true value is given both for position and velocity. Rendered in terms of controller jerk stiffness, the position time integral is given by:-

$$\int x.dt = \int D.dt - (\frac{c}{k_{conj}} + \frac{2\pi N_p \lambda}{N_s p K_{cp} K_i K_v K_1}) \times \dot{D} + \frac{F_A}{k_{conj}} + \theta'$$

In addition to the error due to external force, there is an error term dependent on the mechanical damping term and the jerk stiffness of the controller, and another term dependent on the back e.m.f. of the motor. Both of these error terms are proportional to the velocity demand.

When the demand is a constant acceleration, the response settles down to the true value for velocity, but a "lagging" error arises for the position. After transients, the position is given by:-

$$x = D - (\frac{c}{k_{conj}} + \frac{2\pi N_p \lambda}{N_s p K_{cp} K_i K_v K_1}) \times \ddot{D} + x'$$

Thus, not surprisingly, the position error contains the same terms as those in the time integral of position, but they are proportional to acceleration demand. The position time integral is given by:-

$$\int x.dt = \int D.dt - (\frac{c}{k_{conj}} + \frac{2\pi N_p \lambda}{N_s p K_{cp} K_i K_v K_1}) \times \dot{D} - (\frac{m}{k_{conj}} + \frac{N_s p L c}{2\pi K_{cp} K_i K_v K_1 K_t}) \times \ddot{D}$$
$$+ (\frac{K_p}{K_i} + \frac{1}{N_s p K_v K_1}) \times (\frac{c}{k_{conj}} + \frac{2\pi N_p \lambda}{N_s p K_{cp} K_i K_v K_1}) \times \ddot{D} + \frac{K_{vff} K_{aff}}{K_i K_v K_1} \times \dddot{D} + \frac{F_A}{k_{conj}} + x'$$

This includes a "lagging term", which includes an internal force term due to accelerating the inertia and a term proportional to the motor inductance times the load's viscous drag coefficient, and a more complex "leading term" which is dependent on the system's mechanical damping, the motor's back e.m.f. and the ratio of the proportional and integral terms of the controller.

"Quantisation"

It is common for a controller to operate in part in a step-wise manner, that is, the demand and feedback is sampled once in a defined time interval. In fact, the controller used here as an example up-dates its demand and velocity signal every 3 msec, its velocity feedback every 0.6 msec, and its current feedback every 0.2 msec.

What effect does this have on errors? Consider some demand function $D(t)$. Let it be sampled every τ seconds, that is to say, for a set of sample times $\{0, \tau, 2\tau, \ldots, n\tau, \ldots\}$, where n is an integer, $t_n = n \times \tau$ and the sampled demand function $D_s(t) = D(t_n)$ for $t_n \le t < t_{n+1}$. Over every time interval the sampled demand varies from being "up-to-date" to being τ old, therefore, to a close approximation, at least for mathematically "well-behaved" functions, we can write:-

$$D_s(t) = D(t - \tau/2) + P(\tau, t)$$

where $P(\tau, t)$ is a periodic function of time with a repeat time of τ. (Here the meaning of P is not as precise as the usual definition of a periodic function, that is that $P(t) = P(t + i\tau)$ for any integer i. Rather that the values of P go up and down over a succession of time intervals τ in a broadly similar manner with a mean value over any period which is close to zero. The amplitude can be expected to change over a time scale larger than τ.)

Leaving aside the fluctuations implied by P, the longer term demand as it affects the controller can be approximated as D_m given by:-

$$D_m(t) = D(t - \tau/2)$$

To get a feel for the effects of quantisation let us examine the three cases considered before.

1. ***A constant position.*** Here $D = P$, a constant, which implies $D_m(t) = P$ and so no extra errors are to be expected.

2. ***A constant velocity.*** Here $D = at + b$ where a and b are constants. This means that:-

$$D_m(t) = a \times (t - \frac{\tau}{2}) + b = D(t) - \dot{D}(t)\frac{\tau}{2}.$$

Therefore the demand, averaged over the sample time, is reduced by $\dot{D}\tau/2$. Since the controller settles down to true position under constant velocity demand, a follower error of $\dot{D}\tau/2$ can be expected.

3. ***A constant acceleration.*** Here $D = s + ut + \frac{1}{2} f t^2$ where *s*, *u* and *f* are constants. The average sampled demand is then close to:-

$$\begin{aligned} D_m(t) &= s + u \times (t - \frac{\tau}{2}) + \frac{1}{2} f \times (t - \frac{\tau}{2})^2 \\ &= s + ut + \frac{1}{2} f t^2 - u\frac{\tau}{2} - f t \frac{\tau}{2} + \frac{1}{2} f (\frac{\tau}{2})^2 . \\ &= D(t) - \dot{D}(t)\frac{\tau}{2} + \frac{1}{2}\ddot{D}(t)(\frac{\tau}{2})^2 \end{aligned}$$

Therefore, in addition to the error arising through constant acceleration which has already been evaluated, a lagging error of $\dot{D}\tau/2$ can be expected together with a much smaller leading error of $\ddot{D}\tau^2/4$.

In addition to these "steady state" errors, errors fluctuating at the sampling rate of τ will occur. The amplitude of these is expected to increase broadly with velocity demand. However, the effect of a periodic "forcing term" like $P(\tau, t)$ will stimulate the transient motion predicted by the homogeneous equation. The amplitude of the response will be dependent in part on whether the solution of the homogenous equation involves oscillatory behaviour, and if so, how close the sample rate $1/\tau$ is to the controller/mechanical system natural frequencies, and how much "damping" is involved.

An example of the effect of sampling on controller response is given in Figures A9.1.2 and A9.1.3.

Effect of sampling in the position loop on a controller with a "velocity feed forward" feature

Ramp up at continuous acceleration with continuous variables

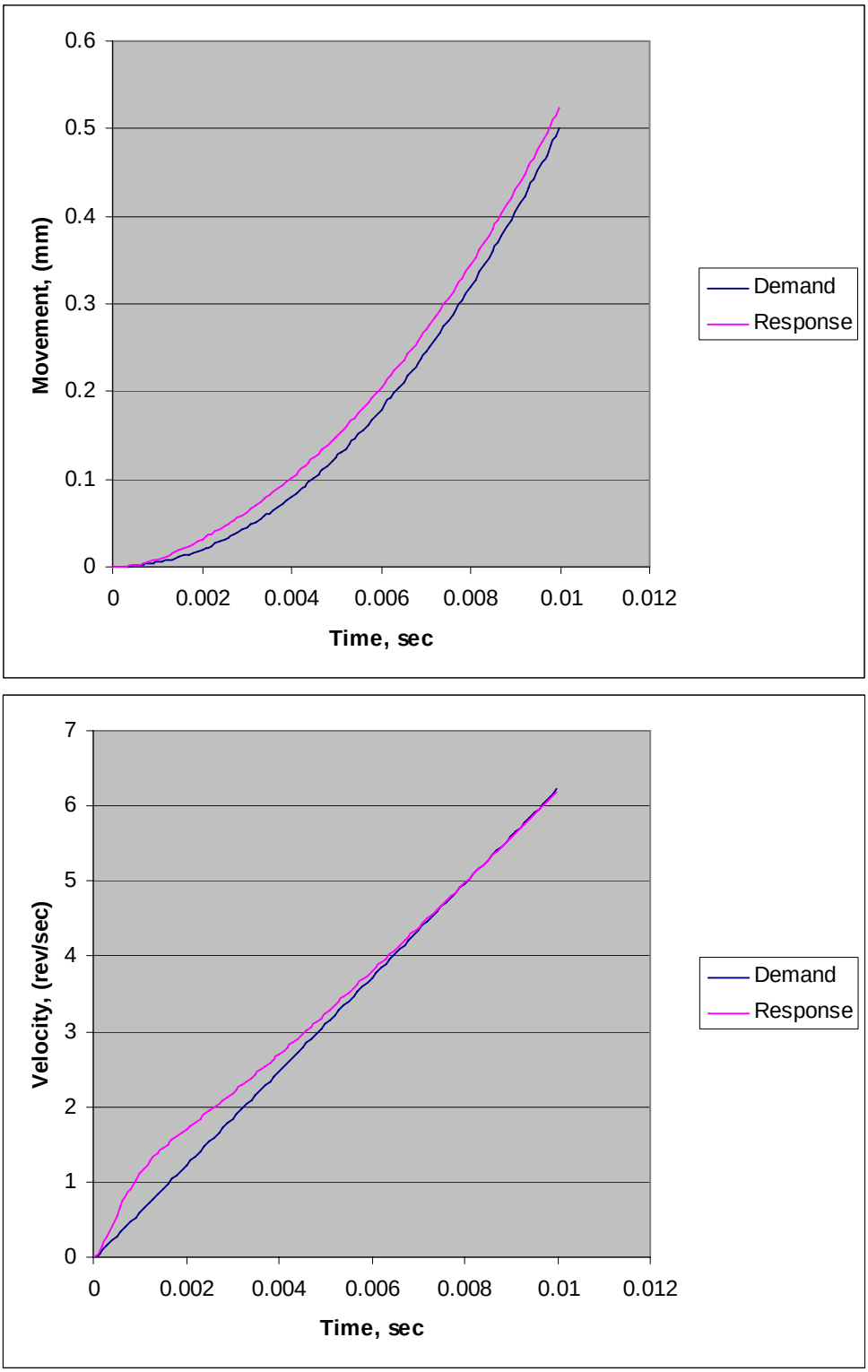
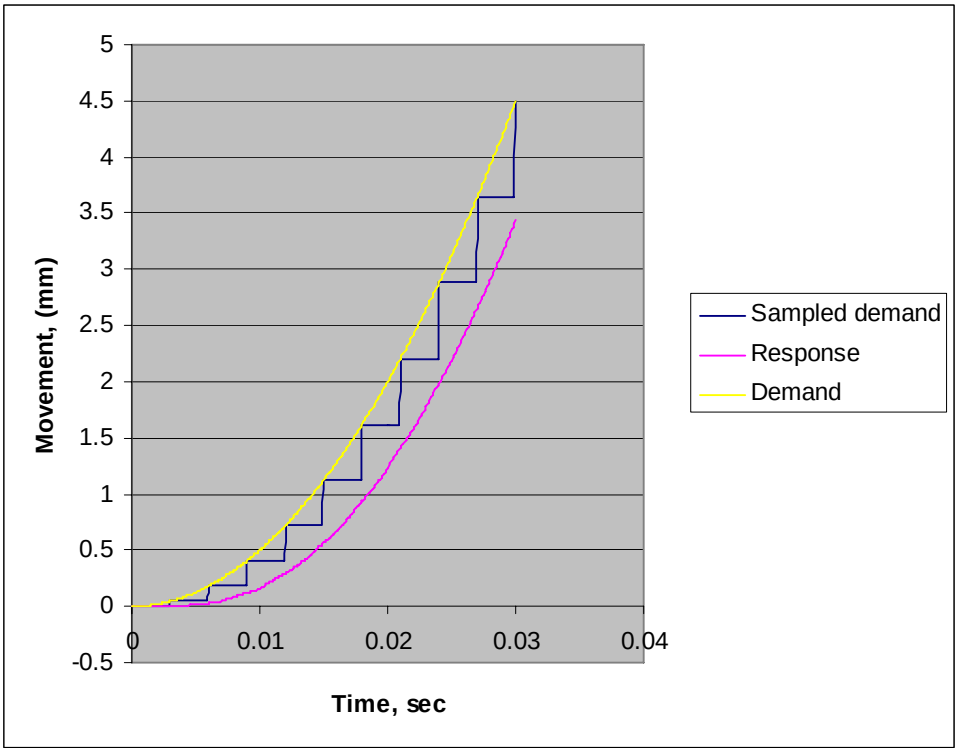


Figure A9.1.2 *Controller behaviour with continuous variables*

Ramp up at continuous acceleration sampled position loop



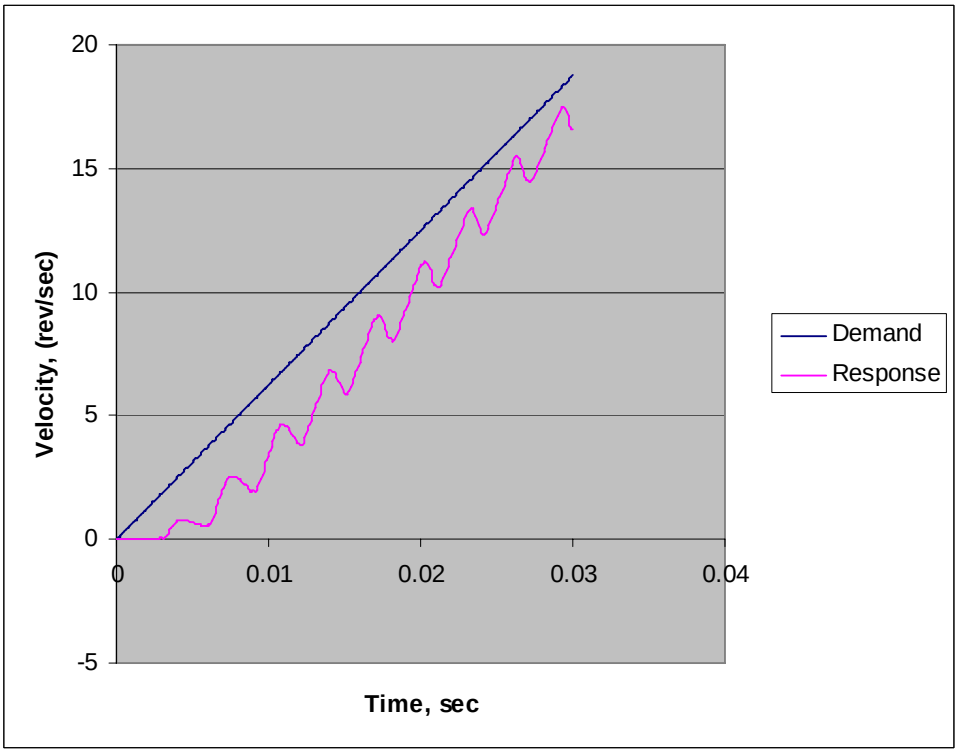


Figure A9.1.3 *Controller behaviour with a 3 msec time step in the velocity loop*

Note the change from a smooth response in front of demand (which can be tuned to match it), to a response which is behind demand and which fluctuates at the sample rate.