Journal of Sound and Vibration (1997) **205**(5), 561–580

CONSTRAINED QUADRATIC PROGRAMMING, ACTIVE CONTROL OF ROTATING MASS IMBALANCE

D. W. Manchala

Xerox Corporation, El Segundo, California 90245, *U.S.A.*

A. B. Palazzolo

Texas A&M University, College Station, Texas 77843, *U.S.A.*

A. F. Kascak and G. T. Montague

U.S. Army at NASA Lewis, Cleveland, Ohio 44135-3191, *U.S.A.*

and

G. V. Brown

NASA Lewis Research Center, Cleveland, 44135-3191, *Ohio, U.S.A.*

(*Received* 14 *August* 1995, *and in final form* 13 *November* 1996)

Jet engines may experience severe vibration due to the sudden imbalance caused by blade failure. The current research investigates employment of piezoelectric actuators to suppress this using active vibration control. This requires identification of the source of the vibrations via an expert system, determination of the required phase angles and amplitudes for the correction forces, and application of the desired control signals to the piezoelectric actuators. Correction forces may exceed the physical limitations of the actuators; hence results of "constrained force" quadratic programming, least squares and multi-point correction algorithms will be compared. It is demonstrated that simply scaling down the least squares predicted correction forces to satisfy the actuator saturation constraints does not necessarily yield optimal reductions in vibration. In this paper test results are shown for sudden imbalance, and the computational time requirements and balancing effectiveness for the various approaches are compared.

1. INTRODUCTION

Vibration due to gradual or sudden (blade loss) imbalance may be detrimental to jet engine health due to internal rubs and/or fatigue failure. Balance correction by mass removal (grinding) or addition may be very time consuming or impossible in the case of blade loss. Hence, research is being performed to develop active control systems to correct imbalance related vibration while in operation. The high frequency actuator types typically employed are magnetic (see Tang and Palazzolo [1]), and piezoelectric and hydraulic (see Tang and Palazzolo [2]).

The influence coefficient method of balancing as applied to two-plane rotor balancing was first described by Thearle [3] in 1934. The fundamental procedure that Thearle used was to determine the rotor response at two locations by applying arbitrary trial weights at each of the two planes and then measuring the rotor response. By this procedure a set of four influence coefficients was determined in which the balance weights could be

0022–460X/97/350561 + 20 $25.00/0/sv961030

calculated in order that the amplitude be reduced to zero at the two measurement locations, at the given speed. The basic procedure of Thearle may be expanded into a three- or multi-plane balancing procedure in which the number of balancing planes is equal to the number of probe or vibration pick-up stations. This method is known as the exact point speed influence coefficient method and has been elaborated on by Tessarzik and Badgley [4].

The increasingly easy accessibility of large digital computers that perform complex arithmetic has made sophisticated balancing techniques more and more practical. The least squares method is one of these, which was proposed by Goodman [5] in 1964 and extended to include complex arithmetic by Lund [6] in 1971. It has been demonstrated that the application of the exact point influence coefficient method will satisfactorily balance the same number of measurement points (locations and speeds) as there are balance planes. However, the method will not necessarily ensure that the rotor will be adequately balanced at other locations and speeds. The procedure using a least squared error approach, as developed by Goodman and Lund, allows reduction of the residual (after balance) vibration at an arbitrary number of speeds and locations. Passive balancing of rotating shafts using non-linear techniques was performed by Woomer [7]. His research provided a means of limiting the size of the correction weights. This is especially important for active balancing in which the actuator and amplifier outputs are restricted by their thermal and electrical limitations. It is to be noted that simply scaling down the least squares predicted correction forces to satisfy the actuator saturation constraints does not necessarily yield optimal reductions in vibration. The mathematical theory on quadratic programming with bounded constraints is described by Gill [8, 9].

Higuchi [10] investigated an observer based active balancing sytem on a magnetic bearing supported rotor. Although this study considers both zero displacement and zero synchronous force approaches, the rotor speed is limited to 3600 rpm, and only steady state vibrations are corrected. Kanemitsu *et al*. [11] use a least squares balancing algorithm to obtain appropriate feedforward correction signals to suppress unbalance vibrations on a magnetic bearing supported rotor. Transient vibration suppression is not considered. Knopse [12] and Beale [13] also consider steady state, feedforward control for magnetic bearings utilizing an adaptive algorithm. Ku *et al*. [14] employ both zero displacement and zero synchronous force active balancing for steady state control using a rotordynamic model simulation. Higuchi [15] minimizes steady state housing vibration on a magnetic bearing test rig using synchronous current reduction based feedforward control. The current work extends these studies to include vibration due to sudden mass imbalance plus the application of a constrained quadratic approach to active balancing.

The influence coefficient balancing theory for uncoupled, coupled, least squares and constrained quadratic problems is described in section 3. The experimental set-up used for the blade loss detection and correction is described in section 4. Software issues, such as a layered architectural specification, the control flow diagram, etc., are described in section 5, the test results are described in section 6, and future work and conclusions are discussed in section 7.

2. INFLUENCE COEFFICIENT BALANCING

The method of balancing a flexible rotor may be roughly divided into two classifications: the modal method and the influence coefficient method [16]. The active balancing method employed here uses the influence coefficient method of balancing the rotor. For a discrete multi-mass rotor system, the total rotor response may be expressed as [17]

$$\{Z\} = [\alpha_u]\{e_u\} + [\alpha_r]\{\delta_r\} + [\alpha_m]\{\delta_m\} + \{\delta_o\}. \tag{1}$$

The total rotor response assumes that a linear system is composed of the responses due to mass unbalance eccentricity e_u at the various stations, shaft bow $\{\delta_r\}$ unbalance couples such as caused by skewed discs, $\{\delta_m\}$ and electrical or mechanical runout vectors $\{\delta_o\}$ which are constant and do not dynamically excite the system.

When a trial force $\{e_T\}$ in the form of a sinusoidal signal is applied to the rotor through piezoelectric actuators, the new rotor response is given by

$$\{Z_T\} = \{Z\} + [\alpha_u]\{e_T\}. \tag{2}$$

Neglecting the effects of δ_r, δ_m and δ_o, subtract the original vector Z and divide by the trial force to obtain the influence coefficients

$$[\alpha_u] = \frac{\{Z_T\} - \{Z\}}{\{\epsilon_T\}} \tag{3}$$

at any speed and between any response point and actuator location. Knowledge of the influence coefficients $[\alpha_u]$ allows determination of the correction forces which minimize the final response vector $\{Z\}$.

The four balancing approaches used here are the uncoupled, coupled, least squares and constrained quadratic forms of balancing. For the uncoupled and the coupled cases, two sensors and two actuators were used for the current tests. Let x_c, and y_c be the two correction forces for the x and y actuators, respectively, and let α_{x1} and α_{x2} be the influence coefficients for sensor 1 due to the application of force on actuator x only, and for sensor 2 due to the application of force on actuator x only, with no force being applied to the y actuators. Similarly, α_{y1} and α_{y2} are the influence coefficients for sensors 1 and 2 due the application of forces on the y actuator only, with no force applied to the x actuator. By requiring that the vibration is zeroed at the two sensors and at the single speed, the uncoupled correction forces become

$$x_c = \frac{-1{\cdot}0}{\alpha_{x1}} * s1, \qquad y_c = \frac{-1{\cdot}0}{\alpha_{y2}} * s2, \tag{4, 5}$$

where $s1$ and $s2$ are the vibration signals from sensors 1 and 2, respectively.

The correction forces for the coupled case are calculated from

$$x_c = \frac{-1{\cdot}0}{A} * (\alpha_{y2} * s1 - \alpha_{y1} * s2), \qquad y_c = \frac{-1{\cdot}0}{A} * (\alpha_{x1} * s2 - \alpha_{x2} * s1), \tag{6, 7}$$

where

$$A = \alpha_{x1} * \alpha_{y2} - \alpha_{x2} * \alpha_{y1}, \tag{8}$$

For the constrained quadratic programming approach, define

$$R = O + \alpha * X, \tag{9}$$

which is the residual vibration that occurs after applying the correction forces; O is the unbalanced system vibration vector, α is the influence coefficient matrix and X is the correction force vector. The quadratic function

$$f(X) = R^{\mathrm{T}}R = W_0 + c^{\mathrm{T}}X + X^{\mathrm{T}}DX \tag{10}$$

is minimized, subject to the constraints

$$\{l_R\} \leqslant \{X_R\} \leqslant \{u_R\}, \qquad \{l_I\} \leqslant \{X_I\} \leqslant \{u_I\}, \tag{11, 12}$$

where l and u are the lower and upper bounds, and the correction forces

$$\{X\} = \{X_R\} + \mathrm{i}\{X_I\} \tag{13}$$

contain both amplitude and phase information.

Comparison of equations (9) and (10) yields

$$W_0 = O^{\mathrm{T}}O, \qquad c^{\mathrm{T}} = 2 * (O_R^{\mathrm{T}} | O_I^{\mathrm{T}}) \begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix}, \tag{14, 15}$$

$$D = \begin{pmatrix} \alpha_R^{\mathrm{T}} & \alpha_I^{\mathrm{T}} \\ -\alpha_I^{\mathrm{T}} & \alpha_R^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix}, \qquad X = \begin{pmatrix} X_R^{\mathrm{T}} \\ X_I^{\mathrm{T}} \end{pmatrix}. \tag{16, 17}$$

Derivation of the above set of equations is given in Appendix A. These equations are solved using an iterative quadratic programming code with bounded constraints. Input to the code consists of the constraints l_R, u_R, l_I and u_I and the quantities W_o, c and D. Other techniques such as the gradient technique could also be used to solve the equations. The constrained quadratic programming program subsumes all other approaches. For instance, when the bounds are removed, i.e. by setting very high bounds, the constrained quadratic method is equivalent to least squares balancing. In addition, when the number of measurement points (number of speeds × number of sensors) are equal to the number of actuators, this is equivalent to the coupled approach. Finally, if each actuator is assumed to have influence only on a single, unique measurement point, constrained quadratic becomes the uncoupled approach. Validation of the quadratic programming code was performed using the E04N AF NAG Fortran library routine [18]. This routine is essentially identical to the subroutine SOL/QPSOL described by Gill *et al.* [19]. For using this subroutine, the problem is stated to be in the form

$$\textit{Minimize} \quad c^{\mathrm{T}}x + \tfrac{1}{2}x^{\mathrm{T}}Dx,$$

subject to the constraints

$$l \leqslant \begin{pmatrix} x \\ Ax \end{pmatrix} \leqslant u,$$

where c is a constant n-vector and D is a constant $n \times n$ Hessian (second partial derivative) symmetric matrix of the quadratic objective function, and A is an $m \times n$ dense matrix. Note that when the matrix D is a zero matrix, the resulting problem reduces to a linear programming problem.

A solution to the above quadratic programming problem proceeds in the form of trying to find a set of feasible solutions iteratively that minimizes the quadratic objective function. A subset of constraints defines the search direction at each iteration, moving a scalar step length in the defined direction. Typically, the next iteration is given by

$$x_{k+1} = x_k + \beta_k p_k,$$

where p_k is the n-dimensional search direction, and β_k is the scalar step length. Lagrange multipliers λ_k define the solution of the compatible overdetermined system,

$$C_k^{\mathrm{T}}\lambda_k = c + Hx_k,$$

when the projected gradient is zero at x_k. The vector $Z_k^{\mathrm{T}}(c + Hx_k)$ is called the projected gradient at x_k, where C_k denotes the matrix of coefficients of the bounds and general constraints in the subset of constraints, and $C_kZ_k = 0$. If the projected gradient is non-zero, the search direction p_k is defined as

$$p_k = Z_kp_z,$$

where p_z is obtained by solving the equations

$$Z_k^{\mathrm{T}}HZ_kp_z = -Z_k^{\mathrm{T}}(c + Hx_k).$$

Other details of the algorithm are described in Gill *et al*. [20].

The following example uses the above quadratic programming algorithm, and demonstrates that simply scaling down the least squares solution to satisfy the actuator constraints will not yield the optimal reduction in vibrations. Consider the experimentally measured influence coefficient matrix for a two-plane balance the two vibration sensors at two speeds, as shown below

$$\alpha = \begin{bmatrix} 8{\cdot}13,\ 341{\cdot}6^\circ & 7{\cdot}66,\ 342{\cdot}4^\circ \\ 1{\cdot}43,\ 265{\cdot}0^\circ & 1{\cdot}04,\ 265{\cdot}5^\circ \\ 9{\cdot}06,\ 162{\cdot}0^\circ & 8{\cdot}55,\ 224{\cdot}1^\circ \\ 3{\cdot}51,\ 255{\cdot}9^\circ & 2{\cdot}62,\ 257{\cdot}0^\circ \end{bmatrix},$$

in amplitude-angle format. The same written in real imaginary format is

$$\alpha = \begin{bmatrix} 7{\cdot}710 - \mathrm{i}2{\cdot}566 & 7{\cdot}301 - \mathrm{i}2{\cdot}316 \\ -0{\cdot}125 - \mathrm{i}1{\cdot}424 & -0{\cdot}081 - \mathrm{i}1{\cdot}036 \\ -8{\cdot}616 - \mathrm{i}2{\cdot}799 & -6{\cdot}139 - \mathrm{i}5{\cdot}950 \\ -0{\cdot}855 - \mathrm{i}3{\cdot}404 & -0{\cdot}589 - \mathrm{i}2{\cdot}552 \end{bmatrix}.$$

The original readings for two speeds and two probes were

$$O = \begin{bmatrix} 3{\cdot}07,\ 102{\cdot}4^\circ \\ 0{\cdot}42,\ \ 25{\cdot}88^\circ \\ 1{\cdot}51,\ 102{\cdot}4^\circ \\ 1{\cdot}05,\ 16{\cdot}94^\circ \end{bmatrix}$$

in amplitude-angle format. The same written in real-imaginary format is

$$O = \begin{bmatrix} -0{\cdot}659 + \mathrm{i}2{\cdot}998 \\ 0{\cdot}377 + \mathrm{i}0{\cdot}183 \\ -0{\cdot}324 + \mathrm{i}1{\cdot}474 \\ 1{\cdot}004 + \mathrm{i}0{\cdot}305 \end{bmatrix}$$

and the constraints on the correction forces were

$$l_R = l_I = \begin{pmatrix} -0{\cdot}25 \\ -0{\cdot}25 \end{pmatrix}, \qquad u_R = u_I = \begin{pmatrix} 0{\cdot}25 \\ 0{\cdot}25 \end{pmatrix}.$$

The above quantities are considered as input to the quadratic programming code and the least squares code (the same quadratic programming code could be used for least squares by setting the lower and upper bounds to the minimum and maximum values allowed on the computing machine, or by setting a provided variable in the E04N AF code). Upon running the code the least squares solution is

$$X = \begin{pmatrix} -0{\cdot}321815 - \mathrm{i}0{\cdot}324842 \\ 0{\cdot}557246 + \mathrm{i}0{\cdot}006832 \end{pmatrix}.$$

Note that the magnitude of three components exceeds the magnitude of the constraints; i.e. 0·25. Consider the least squares (LSQ) solution for the correction forces with each force component that exceeds its constraint value set equal to the constaint value given as

$$X = \begin{pmatrix} -0{\cdot}250000 - \mathrm{i}0{\cdot}250000 \\ 0{\cdot}250000 + \mathrm{i}0{\cdot}006832 \end{pmatrix}.$$

Designate the vibrations which result from applying the least squares solution above by set A. These vibrations are determined from equation (9). Calculation of the data set A applying equation (9) is shown in Appendix B.

The constraint quadratic programming (CQP) solution for the correction forces is

$$X = \begin{pmatrix} -0{\cdot}115406 - \mathrm{i}0{\cdot}193423 \\ 0{\cdot}250000 + \mathrm{i}0{\cdot}006831 \end{pmatrix}.$$

Designate the vibrations which result from applying these correction forces as set B. Calculations for determining the data set B applying equation (9) is shown in Appendix B. Comparing A and B yields the results shown in Table 1, where R_i is the amplitude of the residual vibration for reading i.

Obviously, the cost function of residual vibrations (ΣR_i^2) is smaller with the CQP method and the correction forces are smaller than the imposed limits. It is thus demonstrated that simply scaling down the least squares predicted correction forces to satisfy the actuator saturation constraints does not necessarily yield optimal reductions in vibration.

3. THE EXPERIMENTAL SET-UP

The experimental set-up for blade loss detection and control at the NASA Lewis Research Center is shown in Figures 1–3. The hardware set-up for blade loss detection and control is shown in Figure 1. Vibration data was acquired by an A/D card. A Fast Fourier Transform (FFT) of the first harmonic was performed on the acquired data to obtain the magnitude and the phase of the vibration signals in the X and Y directions for two planes. The two planes at which data was collected were the OD (outboard disk) and

Table 1

Comparison of CQP *and* LSQ *results*

Set	R_1	R_2	R_3	R_4	ΣR_i^2
A (LSQ)	1·82	0·31	1·74	0·765	7·021
B (CQP)	1·28	0·152	1·318	0·384	3·546

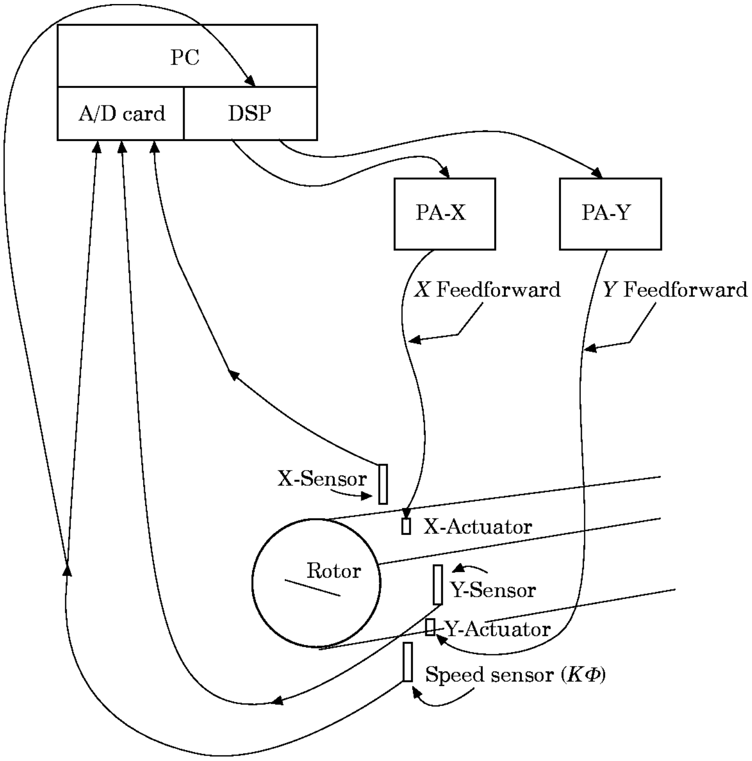

Figure 1. Experimental set-up for blade loss. PA-X, power amplifier for X-direction; PA-Y, in y-direction.

OB (outboard bearing) planes. Whenever a blade loss occurred, the high vibrations were detected by observing the sampled data, and the correction forces were calculated. These forces were downloaded using a digital signal processor (DSP). The DSP used to provide the feedforward control sine waves was an AT&T DSP-32C. The sampling rate of the DSP is 25 kHz/channel, and the processor itself runs at 25 MHz. The sinusoidal signals

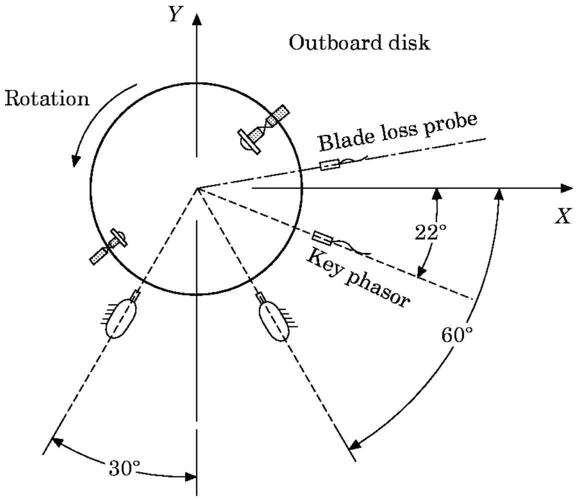

Figure 2. Sudden mass imbalance device at outboard disk location.

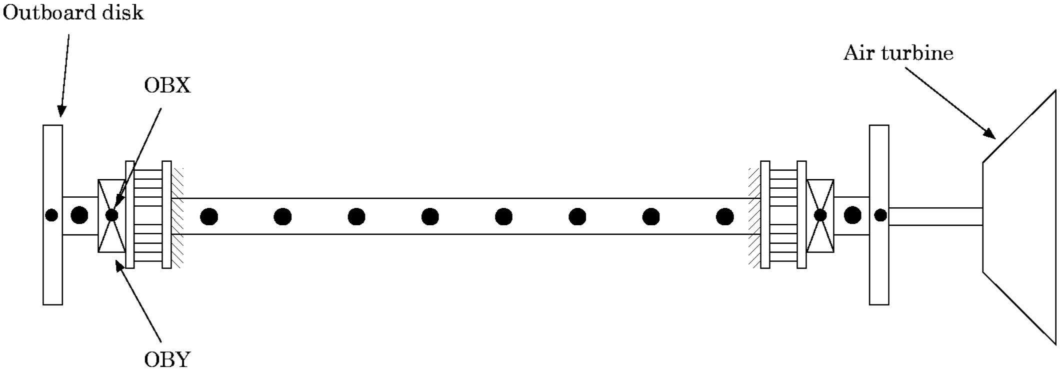

Figure 3. Horizontal view of the rotor showing outboard disk.

downloaded from the DSP were amplified using power amplifiers PA-X (for the X-direction) and PA-Y (for the Y-direction). The X-Actuator (for the X-direction) and the Y-Actuator (for the Y-direction) were used to provide the necessary forces.

In Figure 2 is shown the blade loss simulation performed by activating solenoid driven plungers that knock off rotating bolts at the outboard disk location. Piezoelectric actuators forces were transmitted to the rotating shaft via a flexibly mounted bearing housing which contains a ball bearing. The sensors (OBX, OBY) shown in Figure 3 were collocated with the actuators and were employed for the uncoupled, coupled and constrained quadratic balancing method, in addition to indicating the occurrence of sudden imbalance. Two additional sensors (ODX, ODY) were located on the outboard disk and were employed to obtain vibration data for the unconstrained quadratic balancing. The shaft was spun by an air turbine, connected to it by a flexible rubber coupling. The length of the shaft was 0·62 m and the diameter was 0·254 m. The outer bearing span was 0·48 m and the outboard disks were 0·152 m in diameter and 0·010 m thick. The weight of the outboard disk was 16 N. The bearing stiffnesses, including the piezoelectric actuator stiffness, were both $12{\cdot}0 \times 10^6$ N/m. The critical speeds occurred at 7200 rpm, and 13 600 rpm.

4. SOFTWARE ISSUES IN THE DEVELOPMENT OF THE REAL TIME SYSTEM

4.1. SOFTWARE ARCHITECTURE

Software architecture reflects the organizational structure of a system or component [21]. Background information on software architectures can be found in references [22], [23]. We present a layered hierarchy of object architecture for the real time system used in the testing for blade loss detection and correction.

The software system is organized into five layers, as shown in Figure 4. Layer 1 consists of the physical layer and corresponds to the actual physical devices used in the system. The sensors, actuators, keyboard, mouse, screen, file, etc. constitute the physical layer. The second layer is the system software layer and consists of vendor dependent device drivers for the hardware. These device drivers differ from one operating system to another. The remaining three layers are built on the existing two layers. Layer 3 is the severe software layer which provides services to the hardware devices. The data acquisition module acquires data from the sensors using the data-acquisition board. The routines for programming the board are located here. Feedback and feedforward modules send control signals to the actuators by using the routines provided by the lower level device driver routines. Feedforward signals are generated by the digital signal processor.

The fourth layer is the soft device layer that is the software counterpart to the hardware units in the physical layer. The sensor module contains various sensor class-objects. The Process Server contains the quadratic programming and other balance codes programmed in C + + language [24] using the object oriented paradigm; see Wegner [25] and Kreamer [26]. User interface and the knowledge base module form the fifth layer.

4.2. FLOWCHART DESCRIBING THE CONTROL ALGORITHM

A flowchart of the system is shown in Figure 5. There are two stages involved in testing the software. In the first stage, the dynamics of the rotating machinery are identified at a particular speed by injecting sinusoidal waveforms to the actuator. The second stage involves the actual monitoring of the rotating machinery. Before the dynamics of rotating machinery is identified, the rotor is balanced passively by adding correction weights using the techniques mentioned in references [4–6]. Two masses are cemented to the outboard disk, such that they lie in a straight line and on opposite sides of the axis of the rotor. Moreover, the weights of the masses are selected such that the overall rotor is still balanced in the plane of measurement. In other words, they are cemented in such a way that the product of one mass and its distance of attachment from the axis is equal to that of the product of the other mass and its distance of attachment from the axis of rotation. They are also cemented in such a way that when the solenoid driven plunger is activated the plunger gets in the way of the cemented bolt, and knocks it off. This mechanism is shown in Figure 2. The entire rotating system is housed in a strong case, so that no injury or damage results due to a flying bolt when it is knocked off. The step involved in the system identification of the rotor are given below.

(1) Slow roll is initially captured when the rotor is rotating at low speeds, say at speeds less than 1200 rpm. Fast Fourier Transformation (FFT) is performed on the samples, and the resulting amplitude and phase of the signal is stored as slow roll. Slow roll represents the false vibration sensed by the shaft displacement probes, due to surface scratches, out of roundness, etc.

(2) Vibration samples are collected when the rig is at a desired speed. The amplitude and phase of the signals from the different sensors are calculated by performing an FFT, and slow roll is subtracted to obtain the true vibration at the speed.

(3) An X-direction trial force is injected to the X actuator via a sine wave generated by the DSP. An FFT is performed on the captured vibration, slow roll is subtracted, and influence coefficients are calculated.

(4) Similar to the above, influence coefficients are then calculated for the Y trial force.

5. KB/UI layer	Knowledge base		User interface			
4. Soft device layer	Sensor	Actuator				
3. Server layer	DataAcq	FF ¦ FB	Keyboard server	Video server	File server	Process server
2. System software layer	A/D board device driver s/w	DSP ¦ PIID device driver s/w	Keyboard device driver	Screen/windows video s/w	File management	
		OPERATING SYSTEM SOFTWARE				
1. Physical layer	D/A board Sensors	DSP board Actuators	Keyboard	Video monitor	File/disk	CPU

Figure 4. Architecture layers of the system: FF, feedforward; FB, feed back; PIID, hybrid PID controller; A/D, analog to digital converter; DSP, digital signal procession; KB/UI, knowledge base/user interface.

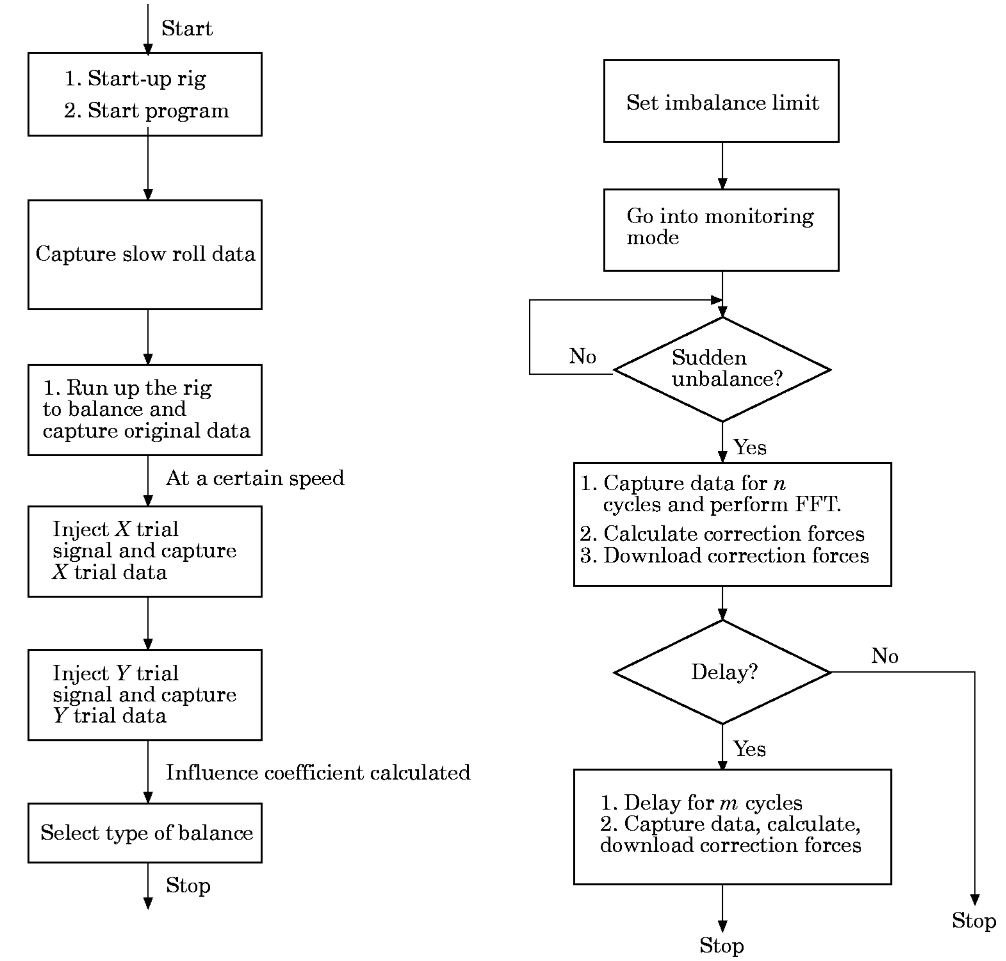
Start
1. Start-up rig
2. Start program
Capture slow roll data
1. Run up the rig to balance and capture original data
At a certain speed
Inject *X* trial signal and capture *X* trial data
Inject *Y* trial signal and capture *Y* trial data
Influence coefficient calculated
Select type of balance
Stop
Set imbalance limit
Go into monitoring mode
No
Sudden unbalance?
Yes
1. Capture data for *n* cycles and perform FFT.
2. Calculate correction forces
3. Download correction forces
Delay?
No
Yes
1. Delay for *m* cycles
2. Capture data, calculate, download correction forces
Stop
Stop

Figure 5. A flowchart for the sudden imbalance correction system: (a) system identification of the rotor; (b) rotor monitoring for sudden imbalance.

Once the dynamics of the system are identified at various speeds, a vibration limit for detecting a blade loss event is set. This limit is usually higher than the vibration occurring at critical speeds. This limit is set through the user interface when this quantity is entered in μm (micro meters) or mils of vibration amplitude displacement. The software system is then set into a monitoring mode. In the monitoring mode, the software system continually gathers vibration data samples, performs and FFT, and computes the amplitude and phase of the vibration signal from the various sensors, thereby continuously checking for sudden imbalance, which is detected by comparing current vibrations to the previously set limit value. Simulation of a blade loss is obtained by knocking off one of the bolts cemented to the outboard disk (OD) by manually triggering off the solenoid driven plungers while the rotor is rotating at a constant speed. After a blade is knocked off, vibration data for a preset number of cycles is captured, an FFT is performed on the captured data, and CQP is used to calculate correction forces. These forces are downloaded immediately. The software system has the provision to download a second set of forces after a specified delay.

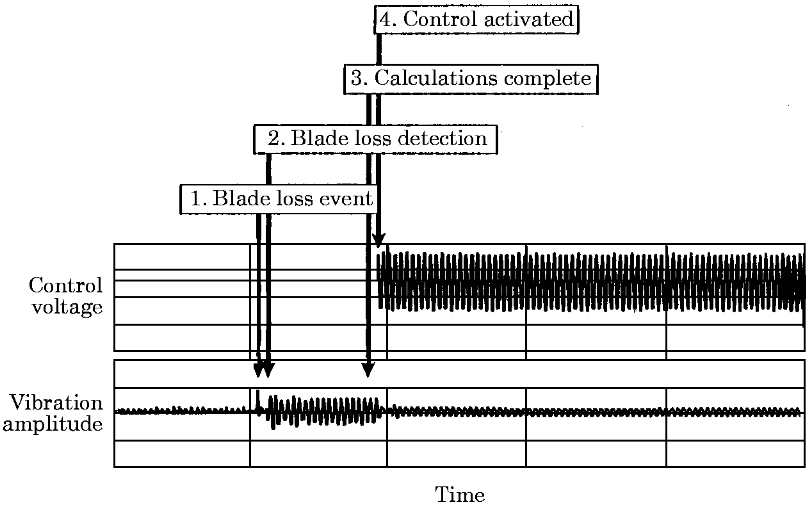

Figure 6. A generic blade loss event.

5. TEST RESULTS

A diagram to explain the generic blade loss event is shown in Figure 6. The diagram explains the exact moment in time when a sudden imbalance occurs, when it is detected by the computer, when the correction forces calculation are performed and when these forces are downloaded by the DSP.

Some test results for the sudden imbalance test are shown in Figures 7–12. Each of the figures show the feedforward control signals (FFX, FFY), and the vibration signals at the outboard bearing (OBX, OBY). The test results when no second balance is applied (i.e., there is no calculation of correction forces after the first download of correction forces) are shown in Figure 7. In this figure, the FFT is performed for 32 samples per cycle in determining the synchronous vibration. A sudden increase in vibrations is observed when a sudden imbalance occurs at the outboard disk, which results from the plunger impact.

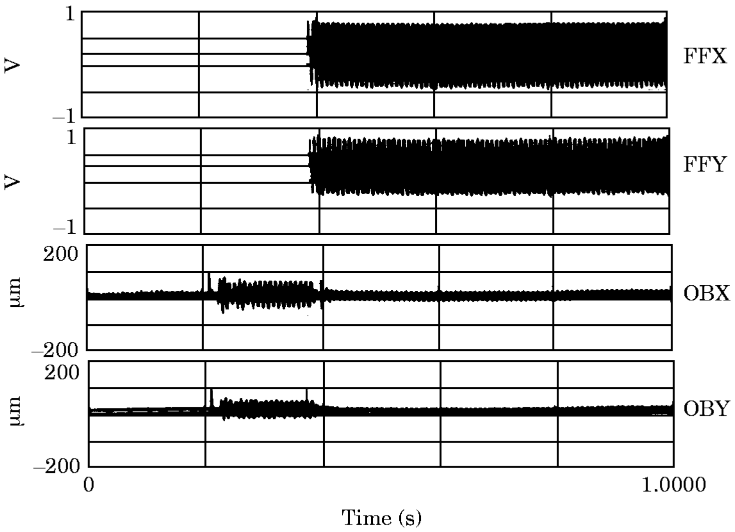

Figure 7. Blade loss event with 32 samples per cycle FFT, four cycles for first balance and no second balance.

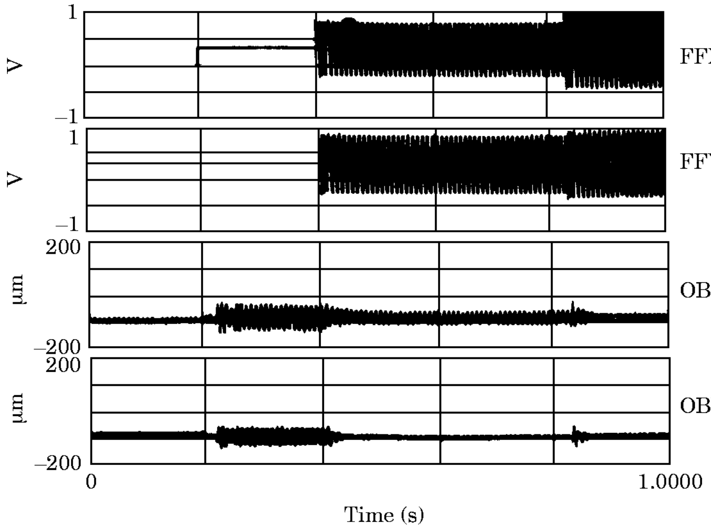

Figure 8. Blade loss event with 32 samples per cycle FFT, four cycles for first and second balance and a 20 cycle delay between first and second balance.

Feedforward correction signals are applied after the sudden imbalance is detected and the correct amount of amplitude and phase of the sinusoidal signals have been determined. The same experiment is performed in the second case (Figure 8), except that correction forces are calculated a second time, after a 20 cycle delay following the download of the first correction forces. A significant reduction is observed in OBX.

In Figure 9 are shown results when a lesser number of samples are taken per cycle (12 per cycle) to reduce the amount of computation time in calculating the correction forces. It can be seen that the correction forces are downloaded much earlier than before. Also shown in Figure 10 is a much quicker response when the number of cycles for which the data is taken is reduced from four to two and the number of samples per cycle is reduced to eight per

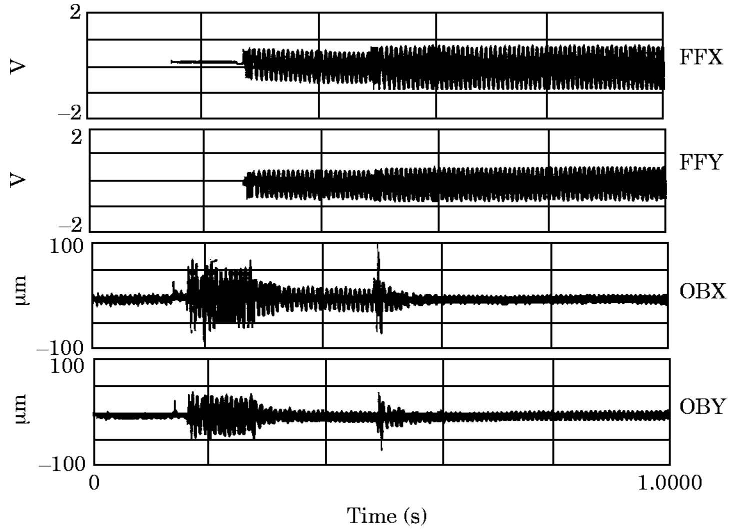

Figure 9. Blade loss event with 12 samples per cycle FFT, four cycles for first and second balance and a 20 cycle delay between first and second balance.

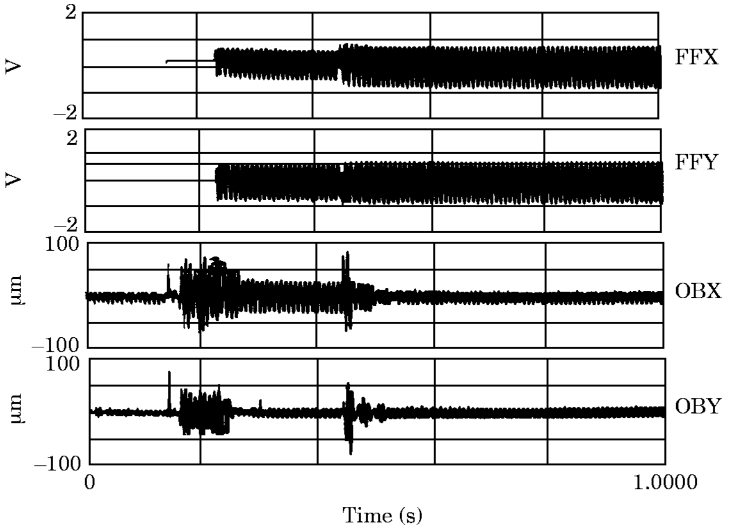

Figure 10. Blade loss event with eight samples per cycle FFT, two cycles for first and four cycles for second balance and a 20 cycle delay between first and second balance.

cycle. The response when the largest imbalance employed in the tests was suddenly applied are shown in Figure 12. Excellent control is maintained even though the imbalance force (21 lb) almost equals the total weight of the rotor.

Table 2 provides a summary of the test results performed with least squares balancing, varying the number of samples per cycle, the delay between the first and second balance and the number of cycles for data acquisition. The column defined as "cycles for first balance" indicates that the acquisition of data for sampling has been performed for the number of cycles shown. The next column, defined as "cycles between balance 1 and 2" indicates the number of cycles between the first download of correction forces and the second download of correction forces. A zero in this column indicates that balancing has

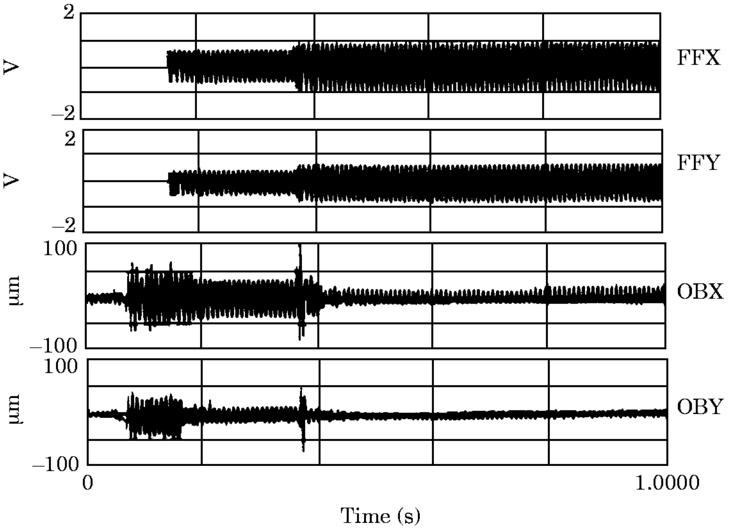

Figure 11. Blade loss event with 12 samples per cycle FFT, two cycles for first and four cycles for second balance and a 20 cycle delay between first and second balance.

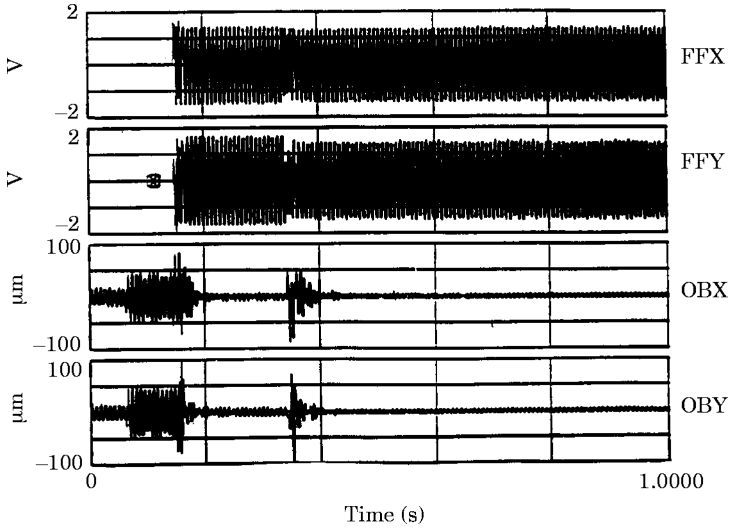

Figure 12. Vibration and control signals for high imbalance test: $F_{DYN} = 21{\cdot}0$ lbs, speed = 8000 rpm.

not been performed for the second time. The column "cycles for second balance" indicates the number of cycles of data acquisition for the second balance. The column "cycles for first download" indicates the number of cycles taken by the system to download the correction signals and make the change. The column "% change after first balance" indicates the amount of change observed in vibrations after the first set of correction forces were downloaded. The column "% change after second balance" indicates the amount of change observed in vibrations after the correction forces were downloaded for the second time. If a second download of correction forces was not done, that row entry shows a zero.

The constrained quadratic programming method was employed using four sensors and two actuators. In Figures 13 and 14 are shown the time histories of the vibrations at all four sensors for the constrained quadratic programming runs. The results of using the constrained quadratic programming techniques are summarized in Tables 3–5. In Table 3 a significant reduction in vibration is shown at all four sensors when applying the method with no constraints on force magnitudes. It can be seen from the last column of Table 3 that there is a 60% improvement in the sum of squares of vibration on all the four probes after application of the correction forces, compared to what was before the application of correction forces. Constraints somewhat smaller than some of the correction forces as shown in Table 4 were applied to obtain the data for Table 5. Although the vibration reduction was smaller than the previous one, it still yielded significant improvement. The

Table 2

Results of uncoupled balancing

Figure number	Samples per cycle	Cycles for first balance	Cycles between balance 1 and 2	Cycles for second balance	Cycles for first download	% change after first balance	% change after second balance
7	32	4	0	0	22	66	—
8	32	4	20	4	24	50	75
9	12	4	20	4	14	66	83
10	12	2	20	4	13	56	78
11	8	2	20	4	12	50	78

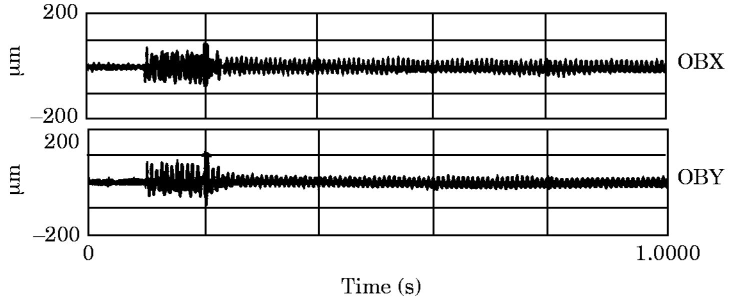

Figure 13. Balancing with constrained quadratic programming: eight samples per cycle, four cycles for first balance, no second balance, constrained case 1.

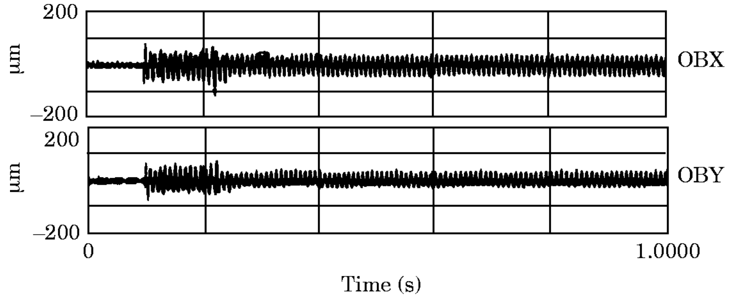

Figure 14. Balancing with constrained quadratic programming: eight samples per cycle, four cycles for first balance, no second balance, constrained case 2.

improvement here is about 47%. The constraints shown in Table 4 were halved to obtain the data shown in Table 5. The improvement observed is about 19%. Reducing the maximum permissible actuator forces is seen to lessen the effectiveness of the balance, although still yielding vibrations smaller than the unbalanced condition.

Figures 13 and 14 and Tables 3–5 confirm the ability of the system to correct the vibration within 15 cycles at 7000 rpm, even when employing the iterative constrained quadratic balancing method. The imbalance correction system employed a 50 MHz, Intel 80486 processor for the results in Figures 13 and 14, whereas a 33 MHz processor was used in the tests of Figures 7–11. Although for this reason a direct comparison of correction times between the methods could not be made, the results did confirm the ability quickly to employ quadratic programming.

Table 3

Test 1, *unconstrained case*: *correction force signal* (volts); FFX $= -2{\cdot}88 + i0{\cdot}78 = 2{\cdot}99$, $164{\cdot}66^\circ$, FFY $= -0{\cdot}51 - i2{\cdot}62 = 2{\cdot}67$, $259{\cdot}0^\circ$; *no constraints*; *samples per cycle* = 8; *cycles for data acquisition* = 4

Status	OBX (μm)	OBY (μm)	ODX (μm)	ODY (μm)	ΣA_i^2 (μm^2)
With control	8·66, 263·85°	26·75, 237·06°	111·0, 265·87°	121·72, 7·34°	27 909·1
Without control	9·88, 270·08°	45·0, 22·72°	185·67, 285·6°	183·08, 19·02°	70 021·2

Percentage decrease = 60%.

Table 4

Test 2, *constrained case* 1: *correction force signal*s (volts); FFX = −1·50 + i0·47 = 1·57, 162·5°, FFY = 0·55 − i1·5 = 1·57, 287·48°; *force signal constraints*; (volts); Max Re(FFX) = Max Re(FFY) = Max Im(FFX) = Max Im(FFY) = 1·5; Min Re(FFX) = Min Re(FFY) = Min Im(FFX) = Min Im(FFY) = −1·5; *samples per cycle* = 8; *cycles for data acquisition* = 4

Status	OBX (μm)	OBY (μm)	ODX (μm)	ODY (μm)	ΣA_i^2 (μm^2)
With control	9·93, 263·85°	16·256, 237·06°	155·73, 265·87°	153·92, 7·34°	48 284·7
Without control	10·92, 270·08°	53·09, 22·72°	208·33, 285·64°	209·86, 19·02°	90 312·9

Percentage decrease = 46·5%.

Table 5

Test 3, *constrained case* 2: *correction force signal*s (volts); FFX = −0·75 − i0·75 = 1·06, 225·0°, FFY = 0·0 − i0·75 = 0·75, 270·0°; *force signal constraints*; (volts); Max Re(FFX) = Max Re(FFY) = Max Im(FFX) = Max Im(FFY) = 0·75; Min Re(FFX) = Min Re(FFY) = Min Im(FFX) = Min Im(FFY) = −0·75; *samples per cycle* = 8; *cycles for data acquisition* = 4

Status	OBX (μm)	OBY (μm)	ODX (μm)	ODY (μm)	ΣA_i^2 (μm^2)
With control	10·16, 263·85°	34·04, 237·06°	186·94, 265·87°	191·01, 7·34°	72 678·6
Without control	10·67, 270·08°	52·07, 22·72°	209·30, 285·64°	208·56, 19·02°	90 093·6

Percentage decrease = 19·4%.

6. SUMMARY AND CONCLUSIONS

The results shown in this paper confirm the ability of the digitally based feedforward control expert system quickly to detect and correct vibration due to a sudden imbalance on rotating shafts. A quadratic programming balancing approach was selected in order to address the real limitations of actuators and power amplifiers for supplying desired correction forces to suppress sudden imbalance vibration. This balancing approach was able to produce significant reductions in vibrations within approximately ten shaft revolutions at 7000 rpm using a 50 MHz 486 processor.

Future work in this area includes further reduction in correction time and expansion of the expert system's knowledge base.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to NASA Lewis and the U.S. Army at NASA Lewis for funding this work. Appreciation is also extended to Jeff Legg of Texas A&M, Physics Department for programming the DSP.

REFERENCES

1. P. Tang, A. B. Palazzolo, G. Brown, E. DiRusso and A. F. Kascak 1993 in *International Gas Turbine and Aero Engine Congress and Exposition*, 61 *of* 91-*GT*-245, *Orlando*, *Florida*, 1–16. An electromechanical simulation method for active vibration control of a magnetic bearing supported rotor.
2. P. Tang, A. B. Palazzolo, A. F. Kascak, G. Montague and W. Li 1993 *Vibration and Control of Mechanical Systems* **61**, 223–241. Combined piezoelectric–hydraulic actuator based active vibration control for a rotor dynamic system.
3. E. L. Thearle 1934 *Transactions of the American Society of Mechanical Engineers* **56**, 745–753. Dynamic balancing of rotating machinery in the field.
4. J. M. Tessarzik, R. H. Badgley and W. J. Anderson 1972 *Transactions of the American Society of Mechanical Engineers*, *Journal of Engineering for Industry* **94**(1), 148–158. Flexible rotor balancing by the exact point speed influence coefficient method.
5. T. P. Goodman 1964 *Transactions of the American Society of Mechanical Engineers*, *Journal of Engineering for Industry* **86**(3), 273–279. A least squares method for computing balance corrections.
6. J. W. Lund and J. Tonnesen 1972 *Transactions of the American Society of Mechanical Engineers*, *Journal of Engineering for Industry* **94**(1), 233–242. Analysis and experiments on multi-plane balancing of a flexible rotor.
7. E. Woomer and W. Pilkey 1981 *Transactions of the American Society of Mechanical Engineers*, *Journal of Mechanical Design* **103**, 831–834. The balancing of rotating shafts by quadratic programming.
8. P. E. Gill, W. Murray and M. A. Saunders 1984 *ACM Transactions on Mathematical Software* **10**, 282–298. Procedures for optimization problems with a mixture of bounds and general linear constraints.
9. P. E. Gill, N. I. M. Gould, W. Murray, M. Saunders and M. H. Wright 1984 *Mathematical Programming* **30**, 176–195. A weighted Gram–Schmidt method for convex quadratic programming.
10. T. Higuchi, T. Mizuno and M. Tsukamoto 1990 in *Proceedings of the* 2nd *International Symposium on Magnetic Bearings*, *Tokyo*, *Japan*, 27–32, Digital control system for magnetic bearings with automatic balancing.
11. Y. Kanemitsu, M. Ohsawa and K. Watanabe 1990 in *Proceedings of the* 2nd *International Symposium on Magnetic Bearings*, *Tokyo*, *Japan*, 265–272. Real time balancing of a flexible rotor supported by magnetic bearing.
12. C. R. Knopse, R. W. Hope, S. J. Fedigan and R. D. Williams 1993 in *Proceedings of MAG* '93 *Magnetic Bearings*, *Magnetic Drives*, *and Dry Seals*, *Conference and Exhibition*, *Alexandria*, *Virginia*, 153–164. Adaptive on line rotor balancing using digital control.
13. S. Beale, B. Shafai, P. Larocca and E. Cusson 1992 in *Proceedings of the* 3rd *International Symposium on Magnetic Bearings*, *Alexandria*, *Virginia*, 601–611. Adaptive forced balancing for magnetic bearing systems.
14. C. P. R. Ku and H. M. Chen 1993 in *Proceedings of MAG* '93 *Magnetic Bearings*, *Magnetic Drives*, *and Dry Seals*, *Conference and Exhibition*, *Alexandria*, *Virginia*, 165–174. Optimum shaft balancing at a rotor bending critical speed with active magnetic bearings.
15. T. Higuchi, M. Otsuka and T. Mizuno 1992 in *Proceedings of the* 3rd *International Symposium on Magnetic Bearings*, *Alexandria*, *Virginia*, 571–579. Identification of rotor unbalance and reduction of housing vibration by periodic learning control in magnetic bearing.
16. A. B. Palazzolo 1977 *Master's thesis*, *University of Virginia*, *Mechanical Engineering Department*. Methods of modal and influence coefficient balancing with application to a three mass bowed rotor.
17. E. J. Gunter, L. E. Barrett and P. A. Allaire 1976 in *Proceedings of the Fifth Turbomachinery Symposium*, (*Texas A&M University*), Balancing of multimass flexible rotors.
18. N. A. Group, *NAG Fortran Library*. Downers Grove, Illinois.
19. P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright 1983 *Technical Report SOL* 83–7, *Department of Operations Research*, *Stanford University*. User's guide for sol/qpsol.
20. P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright 1982 *Technical Report SOL* 82–7, *Department of Operations Research*, *Stanford University*. The design and implementation of a quadratic programming algorithm.
21. L. Haikuan, J. Katwijk and A. M. Levy 1992 in *Proceedings of the* 4*th International Conference*

of Software Engineering and Knowledge Engineering, 170–177. The reuse of software design and software architecture.

22. R. Buhr and R. Casselman 1992 in *Conference on Object Oriented Programming Systems, Languages and Applications—ACM SIGPLAN*, 466–483. Architectures with pictures.
23. M. Shaw 1991 in *Proceedings of the 6th International Workshop on Software Specification and Design, Como, Italy*, 158–165. Heterogeneous design idioms for software architecture.
24. B. Stroustrup 1990 *The C++ Programming Language*. Wokingham, Berkshire: Addison-Wesley; second edition.
25. P. Wegner 1991 *OOPS Messenger* **1**(2), 7–87. Concepts and paradigms of object oriented programming.
26. F. T. Kraemer 1989 *Hewlett Packard Journal* **40**, 87–100. Product development using object oriented technology.

APPENDIX A: DERIVATION OF THE COEFFICIENTS FOR THE QUADRATIC FUNCTION

Consider equation (9), which can be represented in real–imaginary form as the following:

$$R = O + \alpha * X$$

$$R = R_R + \mathrm{i}R_I = O_R + \mathrm{i}O_I + (\alpha_R + \mathrm{i}\alpha_I)(X_R + \mathrm{i}X_I), \tag{A1}$$

which implies

$$R_R = O_R + \alpha_R X_R - \alpha_I X_I, \qquad R_I = O_I + \alpha_R X_I + \alpha_I X_R$$

and

$$R = \begin{pmatrix} R_R \\ R_I \end{pmatrix} = \begin{pmatrix} O_R \\ O_I \end{pmatrix} + \begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix} \begin{pmatrix} X_R \\ X_I \end{pmatrix},$$

Let

$$\begin{aligned} E = R^{\mathrm{T}}R = {} & (O_R^{\mathrm{T}}|O_I^{\mathrm{T}})\begin{pmatrix} O_R \\ O_I \end{pmatrix} + (O_R^{\mathrm{T}}|O_I^{\mathrm{T}})\begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix}\begin{pmatrix} X_R \\ X_I \end{pmatrix}, \\ & + (X_R^{\mathrm{T}}|X_I^{\mathrm{T}})\begin{pmatrix} \alpha_R^{\mathrm{T}} & \alpha_I^{\mathrm{T}} \\ -\alpha_I^{\mathrm{T}} & \alpha_R^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} O_R \\ O_I \end{pmatrix} \\ & + (X_R^{\mathrm{T}}|X_I^{\mathrm{T}})\begin{pmatrix} \alpha_R^{\mathrm{T}} & \alpha_I^{\mathrm{T}} \\ -\alpha_I^{\mathrm{T}} & \alpha_R^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix}\begin{pmatrix} X_R \\ X_I \end{pmatrix}. \end{aligned} \tag{A2}$$

But

$$E = f(X) = R^{\mathrm{T}}R = W_0 + c^{\mathrm{T}}X + X^{\mathrm{T}}DX. \tag{A3}$$

Comparing equation (10) with equation (A2) yields

$$W_0 = (O_R^{\mathrm{T}}|O_I^{\mathrm{T}})\begin{pmatrix} O_R \\ O_I \end{pmatrix}, \qquad c^{\mathrm{T}} = 2 * (O_R^{\mathrm{T}}|O_I^{\mathrm{T}})\begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix},$$

using the matrix multiplication property that

$$(AB)^{\mathrm{T}} = (B^{\mathrm{T}}A^{\mathrm{T}})$$

and recognizing that the second and third terms of equation (18) are symmetric matrices:

$$D = \begin{pmatrix} \alpha_R^{\mathrm{T}} & \alpha_I^{\mathrm{T}} \\ -\alpha_I^{\mathrm{T}} & \alpha_R^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \alpha_R & -\alpha_I \\ \alpha_I & \alpha_R \end{pmatrix}, \qquad X = \begin{pmatrix} X_R^{\mathrm{T}} \\ X_I^{\mathrm{T}} \end{pmatrix}.$$

APPENDIX B: CALCULATION OF THE RESIDUAL DATA SETS *A* AND *B*

Residual data set *A* is obtained by applying the LSQ data to equation (A1). These data are reproduced here for convenience:

$$R = O + \alpha * X,$$

where

$$O = \begin{bmatrix} -0\cdot 659 + \mathrm{i}2\cdot 998 \\ 0\cdot 377 + \mathrm{i}0\cdot 183 \\ -0\cdot 324 + \mathrm{i}1\cdot 474 \\ 1\cdot 004 + \mathrm{i}0\cdot 305 \end{bmatrix}, \qquad \alpha = \begin{bmatrix} 7\cdot 710 - \mathrm{i}2\cdot 566 & 7\cdot 301 - \mathrm{i}2\cdot 316 \\ -0\cdot 125 - \mathrm{i}1\cdot 424 & -0\cdot 081 - \mathrm{i}1\cdot 036 \\ -8\cdot 616 + \mathrm{i}2\cdot 799 & -6\cdot 139 + \mathrm{i}5\cdot 950 \\ -0\cdot 855 - \mathrm{i}3\cdot 404 & -0\cdot 589 - \mathrm{i}2\cdot 552 \end{bmatrix},$$

$$X = \begin{pmatrix} -0\cdot 250000 - \mathrm{i}0\cdot 250000 \\ 0\cdot 250000 + \mathrm{i}0\cdot 006832 \end{pmatrix}.$$

Applying the above values, we obtain the residual vector *R* for LSQ:

$$R = \begin{bmatrix} 1\cdot 82 \\ 0\cdot 31 \\ 1\cdot 74 \\ 0\cdot 765 \end{bmatrix},$$

Residual data set *B* is obtained by applying the CQP data to equation (A1). The values of *O* and α are the same as given above for calculating the residuals for the LSQ solution. The value of the correction forces *X*, given below for convenience, and the value of the calculated residual vector *R* for CPQ are

$$X = \begin{pmatrix} -0\cdot 115406 - \mathrm{i}0\cdot 193423 \\ 0\cdot 250000 + \mathrm{i}0\cdot 006831 \end{pmatrix}.$$

$$R = \begin{bmatrix} 1\cdot 28 \\ 0\cdot 152 \\ 1\cdot 318 \\ 0\cdot 384 \end{bmatrix},$$

The cost function is the sum of the squares of the residuals ($\Sigma\, R_i^2$). A summary of the residual sets *A* and *B* is shown in Table 1.

APPENDIX C: NOMENCLATURE

Z	total response of a rotor system	δ_m	unbalance couples due to skewed discs
e_u	unbalance eccentricity	δ_o	mechanical runout vectors
δ_r	shaft bow	e_T	runout

α_u	influence coefficients
x_c	X correction force
y_c	Y correction force
R	residual
O	current vibration
l_R	real part of the lower bounds
u_R	real part of the upper bounds
l_I	Imaginary part of the lower bound
u_I	Imaginary part of the upper bound