ISSN 0140 3818
SSSU 45

A TEST RIG FOR THE INVESTIGATION OF SHIP PROPELLER/RUDDER INTERACTIONS

by S.R. Turnock

Ship Science Report No. 45

October 1990

ISSN 0140 3818
SSSU 45

UNIVERSITY
OF
SOUTHAMPTON

DEPARTMENT OF SHIP SCIENCE

FACULTY OF ENGINEERING

AND APPLIED SCIENCE

A TEST RIG FOR THE INVESTIGATION OF SHIP PROPELLER/RUDDER INTERACTIONS

by S.R. Turnock

Ship Science Report No. 45

October 1990

A TEST RIG FOR THE INVESTIGATION OF SHIP PROPELLER/RUDDER INTERACTIONS

by

S.R.Turnock

UNIVERSITY OF SOUTHAMPTON

SHIP SCIENCE REPORT No. 45

October 1990

SUMMARY

This report details the design and installation of a wind tunnel rig for the investigation of Ship Propeller and Rudder interactions. The overall rig is designed for use in either of the two low-speed wind tunnels at the University of Southampton. The rig consists of two separate units which allow model Ship rudders and propellers to be either tested independently or as a whole. The independence of the two units allows parametric studies to be made of rudder/propeller interactions while at the same time providing a facility for testing the freestream characteristics of both rudders and propellers.

The rudder rig incorporates a six-component strain gauge dynamometer to measure forces and moments. Both semi-balanced skeg and all-movable rudders can be tested. The rudder models are pressure tapped to give spanwise and chordwise pressure distributions.

The model propeller is driven from a 30 kW A.C. synchronous electric motor via a belt drive. The propeller support assembly is mounted to a tubular steel framework beneath the wind tunnel working section. The distance of the propeller axis of rotation from the wind-tunnel floor can be varied between 600 and 900mm. An in-line strain gauge dynamometer is used to measure the propeller Torque and Thrust. A variable-frequency electric motor-controller allows the propeller revolutions to be varied between 50 and 3000 RPM. The rig is designed to take a range of propellers with diameters between 700 and 1000mm. For the first application a four-bladed propeller with a diameter of 800mm was used.

TABLE OF CONTENTS

Appendices

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

A propeller upstream of a Ship's rudder influences its performance. The rig described in this report is designed to help quantify that influence. The propeller accelerates the flow over the rudder and alters the effective incidence of the local flow. Quasi-steady pressure and force measurements allow parametric relationships to be constructed between propeller loading characteristics and rudder performance. The rig is designed to provide the necessary data and to allow the parametric studies to be made for use in ship performance and manouvering studies.

The rig uses air as a working fluid rather than water. This eases significantly the measurement of data and the test procedures. Propeller cavitation cannot be investigated, but this is not considered a significant factor in rudder/propeller interactions.

Primarily designed for use in the 11' x 8' working section closed-return tunnel at the University of Southampton the rig can also be used in the smaller 7' x 5' wind tunnel. Appropriate scaling of Rudder Chord Reynolds number and propeller performance allows full-scale similarity to be obtained. This allows the results to be directly applied to full-scale Ship Propeller/Rudder arrangements.

The report is divided into five sections. The remainder of Section 1 details the main design requirements of the rig. Sections 2 and 3 detail the design and manufacture of the Propeller and Rudder rigs respectively. The Data Acquisition system is described in Section 4 and the Installation and Commissioning of the rig in Section 5.

1.1 Overall Design Requirements

The research into rudder/propeller interactions required a test rig capable of measuring forces, moments and pressures on a rudder mounted downstream of a propeller of known loading characteristics.

The rig is required to operate primarily in a 11' x 8' low-speed wind tunnel as well as in a 7' x 5' working section tunnel. An investigation of standard ship rudder/propeller configurations provided the overall range of geometries required for a parametric study of the interaction.

An investigation was made of the wind tunnel speed U, propeller revolutions N, physical size of the rudder (c, s) propeller diameter D and number of blades k necessary to achieve satisfactory scaling without exceeding design load limitations. Table 1 gives the required operating conditions. This information was used as the basis for the design of the rig.

2 PROPELLER RIG

For the first series of tests, a satisfactory range of thrust loading values for a four-bladed propeller was obtained for a maximum revolution speed of 3,000 RPM and a diameter of 0.8 m. The design of the actual propeller, based on the Wageningen B4-40 series, is given in references [1,2]. A variable pitch propeller design was used to extend the available range of propeller loadings.

The safety and integrity of the system had to be guaranteed as the high speed of revolution and large mass of the propeller blade would cause significant damage if failure occurred. The supporting structure for the propeller and its associated drive system had to be capable of :-

- Allowing a range of propellers with diameters between 700 and 1000mm to be tested.

- Alteration of the distance of propeller axis of rotation from the tunnel floor in the range of 0.6 to 0.9 m.

-Yawing the whole rig relative to the direction of the wind tunnel flow.

- Acquisition of accurate data on the performance of the Propeller. A minimum requirement being the measurement of Hub Revolutions, Torque and Thrust. A consideration should be made for the possible inclusion of blade surface pressure measurements at a later date via the mounting of a scanivalve in the propeller hub.

Further constraints were that:

-The propeller drive system had to provide sufficient Torque to enable all points within the operational envelope of the propeller to be tested.

-The propeller support structure should have minimal impact on the flowfield upstream of the propeller and rudder.

-The propeller rig should be compact to enable the rig to be easily manoeuvred both for setting up purposes and for the parametric studies.

--The Propeller rig had to be designed, built and commissioned to a twelve month timescale.

-The cost of Wind Tunnel time is high and therefore the time for installing and removing the completed rig from the wind tunnel should be minimised.

-The rig should be capable of being used in both of Southampton University's low-speed wind tunnels.

-The propeller rig should be independent of the rudder rig to allow freestream tests of propeller in both ahead and astern flows.

Within these constraints the rig had to be as powerful and flexible in use as possible. An important consideration was the use of the rig for tests outside the scope of the research contract. Considerable savings of time and effort would be made by including these possibilities at the initial design stage.

2.1 Design

The parametric study gave design values for the maximum power of the propeller (30kW), its overall diameter (800mm), R.P.M. (3000) and range of heights of the propeller axis above the tunnel floor (600-900mm). The propeller rig was designed and manufactured over a 12 month period. Once the preliminary design scheme for the overall rig had been detailed, final design work was carried out in three stages. These were:

1) Propeller Drive Shaft, Instrumentation and associated framework.

2) Support structure to locate the propeller relative to the wind tunnel working section floor.

3) Structure to support electric motor and propeller structure from underneath tunnel floor, and final design of the propeller hub and blades.

All the rig manufacture, apart from the propeller blades and fairing, was carried out in the joint Aeronautics/ I.S.V.R./ Ship Science workshops at Southampton University.

Figure 1 shows an overall view of the final propeller rig with the three phases of work labelled. In the following sections the design of the various components is given in detail. The general philosophy of the design was to provide a simple and robust structure which could be installed rapidly and which would provide accurate information on the Propeller performance within the allowed budget.

In Table 2 the time taken for the design, manufacture, and completion of each phase is shown. Considerable delays occurred during the Autumn and Winter of 1989 due to the lack of availability of technicians on account of the demands of Undergraduate Project work.

2.2 Instrumentation Framework

The propeller's thrust and torque have to be measured as close to the actual propeller as possible. It was decided to use an in-line strain gauge dynamometer to measure the thrust and torque developed. Figure 2 shows the design of the dynamometer used. The manufacture and calibration of the dynamometer are detailed in [3].

The in-line-dynamometer requires 8 wires to be connected to it (4 each for the torque and thrust bridge circuits). Brushes and slip-rings are the standard way of making such rotary connections. It was decided that to allow for future use of a rotary pressure transducer (Scanivalve) in the propeller hub a 20-way slipring assembly was to be used as this would provide enough spare rotary connections.

The model propeller and hub need to be rigidly supported with a minimum of disturbance to the propeller inflow. Standard propeller/hub configurations have a hub diameter to propeller diameter ratio in the range of 0.2 - 0.25. For a propeller diameter of 800mm this gave a maximum hub diameter of 200mm. As the propeller was to have variable pitch a hub diameter of 180mm was decided upon to allow for fairing of the hub into the root shape of the propeller blades.

This constraint of propeller hub diameter meant that the propeller driveshaft bearings, associated instrumentation, structural framework and a fairing all had to be contained within an overall diameter of 180mm. The framework supporting the structure in the wind tunnel working section had to be as rigid as possible as any vibration in the system would adversely affect the modelling of the flow produced by the propeller and the measurement of propeller forces.

Due to the mass of the electric motor and associated structure the propeller rig framework has to be supported from the floor of the wind tunnel building. A means of transmitting the power from the motor to drive the propeller has to be provided. The construction of a custom-built 30 kW electric motor with a maximum 180mm diameter mounted in the wind tunnel would be prohibitive both in cost and lead time for manufacture.

The connection of the framework to the wind tunnel floor had to be as far upstream from the propeller plane of rotation as possible while at the same time minimising the unsupported length to the propeller. An overall length of 1m from propeller plane of rotation to the framework was decided on.

Two viable options for the instrumentation framework were : a 180mm diameter steel tube with access panels removed; and a square cross-sectioned welded structure. It was decided that the use of the circular tube would require the removal of too much area to maintain rigidity. This resulted in the design of a fabricated framework using 4 lengths of 40mm X 40mm angle section welded to four 1" thick blocks spaced along the length. By having the blocks of sufficient thickness a boxed structure with both lateral and cross-sectional stiffness was produced. The need for access to the propeller shaft and fittings could be accommodated without detracting from the overall stiffness of the framework. As an additional safeguard, 1/4" plate was welded along the two sides of the structure.

The position of the dynamometer determines the layout of the framework by dividing the power drive shaft into two halves which need individually supporting by pairs of bearing races. The shaft(Propeller shaft) between the dynamometer and the propeller has to have a minimal effect on the forces measured by the dynamometer. The solution used was to have two roller bearing races which have a nominal zero-axial resistance and minimal rolling resistance. The bearings on the other shaft (pulley shaft) have to absorb axial thrust in both directions while locking the overall shaft and dynamometer assembly in position. The final design of the power transmission (see next section) has a belt and pulley arrangement which requires bearings capable of taking radial loads. For the pulley shaft the arrangement used was two angular contact bearings mounted back to back. This acts as a bidirectional thrust bearing and locates the shaft. A plain ball race supports the other end of the shaft and takes half of the radial loading due to the belt tension. The four bearing units were to be each mounted in one of the frameworks blocks. Although this would entail a removal of a large amount of material from the individual blocks the strength would be maintained through the structure of the bearings and their mounts.

It was decided not to use a flexible coupling between the dynamometer and the pulley shaft as this might have an adverse affect on the performance of the dynamometer. Instead it was decided to use a horizontal boring machine to in-line bore all four bearing blocks on a single axis.

Figure 3 shows the overall layout of the instrumentation framework. The bearing blocks were first machined to within 1/4" of their final diameter. The angle, blocks and pads were welded together and heat treated to remove any residual stresses prior to the machining operation to final bore and face the bearing blocks.

The dynamometer is attached to both the propeller and pulley shafts by 4mm shear pins. In the event of failure of the propeller the shear pins between the dynamometer and the pulley shaft will fail first preventing the hub and propeller shaft pulling out from the framework. The diameter of the propeller shaft was determined by the constraint on the size of the bearing races and was manufactured from high tensile EN16T steel. This steel has an UTS greater than that caused by the out of balance forces due to the shedding of a single blade (worst case failure - outline stress calculations shown in Appendix 8).

2.3 Propeller Drive System

In the original research contract it was envisaged that the propeller would be driven by a combination of solid shafting and bevel gears from a 3-phase AC Synchronous Electric Motor. The electric motor would be mounted underneath the wind tunnel floor. The attraction of the proposal was that the diameter of the vertical drive shaft needed to carry 30 kW would have caused minimal disturbance to the flowfield upstream of the propeller. It was known that bevel gears available for the klimited space would be working close to their maximum limits in power and revs. Analysis of the structural rigidity required to support the propeller with minimal possible vibration showed that a single column would not suffice, as the necessary supporting tube diameter would disturb the propeller inflow to a large extent. It was recognised that the structural members used to support the propeller, shafting, instrumentation and supporting framework rather than the the drive shaft diameter alone determined the disturbance to the flowfield; this allowed the use of high-power toothed timing belts to be considered. It was found that a cheaper and simpler transmission system could be designed which would be able to transmit 30 kW of power, at 3,000 r.p.m. with a maximum pulley diameter within the allowable width of 140mm. Two support tubes were used for the framework as this further reduced the tube diameters and provided additional vibrational stiffness. This allowed the position of the drive belt to be moved further away from the propeller plane of rotation. The aerofoil fairing around the drive belt could, thereby, have a smaller thickness/ chord ratio while still having its trailing edge one propeller radius upstream of the propeller plane of rotation.

The standard frame size of the electric motor needed to provide 30 kW at 3,000 r.p.m. has a mass of at least 250 Kg. Therefore, for reasons of static stability and to reduce the overall mass, the motor had to be placed as close as practical to the wind tunnel building floor . The high power transmitted through small diameter pulleys restricted the length of drive belt, so a lay-shaft and secondary drive belt were required between the motor and the propeller drive shaft. An advantage of this arrangement was that by the use of different lengths of secondary belts the rig could be simply adapted for use in both of the wind tunnels.

It was decided that rather than the use of idler pulleys for tensioning the drive belts a more reliable scheme would be the physical movement of the various frame structures to directly tension the belts. This would also allow the system to be driven in forward and reverse without the need for any adjustment.

Details of the toothed timing belts, pulley diameters and Taper-Lock bushes are given in Appendix 2.

2.4 Framework

2.4.1 Above wind Tunnel Floor

As described already the instrumentation framework is supported on two tubes protruding into the wind tunnel working section between which the drive belt runs. The structure which supports the instrumentation framework is shown in Figure 4. The three pads on the instrumentation framework are used to bolt onto a plate welded across the top of the two support tubes and a shaped angle bracket provides additional lateral rigidity. The top plate is profiled to stay within the overall 180mm diameter of the instrumentation framework and has a flame cut clearance hole to allow the drive belt to pass through.

At the lower end , each tube is securely bolted into a flange welded onto a 500*600 mm 1/2" thick steel plate. This plate has lateral and longitudinal stiffeners welded across its underside and incorporates two slotted plates used for holding the lay-shaft assembly as shown in Figure 5. The vertical position of the lay-shaft can be adjusted up and down so that the belt to the propeller drive shaft can be set to the correct tension. A clearance hole between the two flanges on the main plate was flame cut for the drive belt.

The lay-shaft assembly contains two angular contact ball races facing each other to locate the shaft in position. The pulleys are secured to the shaft by taper-lock bushes without a key so that in the event of the transmission system jamming the system will fail on the lay-shaft rather than on the propeller drive shaft. the bearings were mounted in the same way as those in the Instrumentation framework.

2.4.2 Below wind Tunnel Floor

Slots on the side of the two support tubes for the instrumentation framework provide a means of locking the propeller rig to the wind tunnel floor. However, the main mass of the rig is supported from the concrete floor of the wind tunnel building underneath the working section. The overall structure has to support approximately 600Kg including the

250kg of electric motor. There are restrictions in the access to the small 7' x 5' wind tunnel so that the components of the framework had to be made to fit within these. It was decided to split the framework into three main frames(Top, Middle and Motor) as shown in Figure 6 for the 11' x 8' Wind Tunnel . By not including the Top frame and using a shorter motor belt, the 7' x 5' wind tunnel configuration as shown in Figure 7 is achieved.

The Top, Middle and Above Wind Tunnel Floor assembly are attached together using an arrangement of bolts through six pairs of machined pads. The four slots at the bottom of the Middle frame allow the motor belt to be tensioned. The whole mass of the rig is taken on four 20mm diameter lengths of studding. Four slotted plates on the sides of the Middle frame allow the Middle frame to be rigidly clamped to the Motor frame once the Motor belt has been correctly tensioned. The long length of slot is necessary to allow for the use of the rig with two different lengths of motor belt.

The Motor frame is designed to fit a standard 200L frame size 3-phase A.C. electric motor although there is clearance for a 250L frame size. The four leg assemblies bolt onto the sides of the Motor frame to provide a wide base area. The motor frame has a several series of bolt holes to take the leg assemblies which, thereby, allow adjustment in the overall height of the rig and also for altering the distance between the centre-line of the propeller and the tunnel floor. The whole rig rests on four adjustable feet which are used to level the rotational axis of the propeller and for fine adjustment of the propeller height. For transportation purposes, the Top, Middle, Motor, and Leg assemblies are separated and the Motor frame is provided with four castors.

All three main frames were manufactured from 50*50mm 5mm thick wall stock steel tube. The lengths of tube were welded together using gusset plates for extra rigidity. Standard components were used to reduce manufacturing time. The four leg assemblies were fabricated from steel channel.

2.5 Propeller Hub and Blades

The detailed design of the propeller blade split moulds and manufacture of the blades in hybrid carbon/glass fibre composite are given in references [1,2]. A four bladed propeller was used for the first phase of testing. Each blade was manufactured using the same split female mould to produce four identical blades.

Following experience gained on the use of such blades at Southampton University each composite blade has at the root, an aluminium collar which locates into a hub split along the propeller plane of rotation as shown in Figure 8. The split hub is used to clamp the blades in position when they have been set to the desired pitch. For ease of machining and to reduce the rotational inertia of the system the split hub was designed in aluminium alloy.

Figure 8 shows the split hub design. Four shear pins take the thrust loading and a key resisting torque are used to attach the hub to the propeller shaft. Precision machining on a horizontal boring machine was required to ensure all four blades were located at the same radial position relative to the centre of the hub. During boring of the four holes and faces to locate the blade collars a thin paper gasket was used to ensure a positive clamping action when the two halves of the hub are bolted together. The recess at the rear of the hub was provided to allow space for the provision of masses required after dynamic balancing of the hub and blades. Sufficient space is available at the front of the hub for the use of a scanning pressure transducer for measurement of blade surface pressures.

2.6 Fairing and Hub Cone

To minimise the disturbance to the flow due to the instrumentation frame and support tubes a fairing was required. For the instrumentation framework a cylindrical cowling with a diameter of 180mm equal to that of the propeller hub provides a smooth flow onto the propeller blade root sections. A nose cone on the upstream end of the cowling and another cone attached to the hub provides an overall aerodynamic shape.

A NACA 63035 [4] section with a chord of 550mm and maximum thickness of 148mm at 35% of the chord from the leading edge provides a fair shape over the support tubes and belt while also allowing enough space for displacement of the belt due to vibration. The aerofoil profile was merged into the cylindrical cowling and nose cone to give an overall smooth shape. The trailing edge of the aerofoil profile was stretched from an overall chord of 550mm to 750mm to enclose the angel bracket between the support tube and instrumentation frame.

Figure 9 shows the overall shape of the fairing in relation to the underlying structure. The cylindrical cowling and rear section of the aerofoil were manufactured from thin plywood. The nose and hub cones, and the aerofoil nose were made from Glass Reinforced Plastic (G.R.P.)

Appendix 1 provides accurate dimensions of the overall fairing and cones in a form suitable for defining their surface shape for the purpose of numerical modelling.

2.7 Propeller Revolution Control System

In the original research contract proposal an existing variable frequency 3-phase A.C. electric motor control system was to be used to allow exact propeller revolutions control. After a detailed appraisal of this method of control it was found that a non-standard 3-phase synchronous motor (frame size 250L) would have to be used and also that the rig would then be restricted to use only in the 11' x 8' wind-tunnel as the other tunnel did not possess such a facility.

Information was obtained about a new design of digitally controlled frequency inverter system which allows very precise control of revolutions. Although the electric control system was expensive it allowed the use of a standard A.C. motor (frame size 200L) which was significantly cheaper than the 'special' motor. This motor would be considerably lighter and the electric control system could be used anywhere with a suitable 3-phase supply.

Figure 10 illustrates the layout of the electric motor control system . Details of the motor and control system are given in Appendix 2 and reference [5]. The frequency inverter, and associated components were mounted within a dexion frame on castors. A remote control box was manufactured for use in the wind tunnel control area. Initially only a small number of the possible control functions of the inverter were implemented.

These were: a facility for forward/reverse driving of the propeller shaft; and direct control of the inverter frequency supply and hence propeller revolutions. The digital nature of the system allows a wide range of control parameters to be set. A particularly useful feature is the ability to control the ramp acceleration and deceleration times of the system which permits emergency stops to be carried out in a controlled manner.

3 RUDDER RIG

The rudder rig has been modified from an existing one designed for freestream semi-balanced skeg rudder tests conducted by Molland [6,7,8]. For this investigation larger rudders than the original series were to be used and the higher forces generated had to be kept within the design loading of the rig based primarily on the permitted stress levels in the six-component strain gauge dynamometer. This places restrictions on the possible tunnel wind speeds if onset of stall is to be investigated.

The rudder has to be supported clear of the wind-tunnel working section to isolate the dynamometer from tunnel vibration. The gap between the bottom of the rudder and tunnel floor has to be kept to a minimum to prevent spurious gap flow effects.

The required data is the Lift and Drag Forces acting on the rudder/skeg combination and three Moments about the x, y and z axes to allow the centre of pressure and rudder turning torque to be determined.

The existing rig already allowed the following:

-Testing of both all-moveable and semi-balanced skeg rudders with and without taper.

-Surface pressure measurements on the rudder to allow local two-dimensional lift coefficients to be determined and hence the spanwise lift distribution across the rudder.

-The angle of attack of the rudder can be increased until stalled flow occurs.

-The skeg and rudder can be set at different angles relative to each other.

3.1 Design

The rudder rig is based on that developed for measuring the free-stream performance of semi-balanced skeg rudders (Molland[6,7,8]). This rig was originally for use in the 7' x 5' Wind Tunnel. The only modification required to the rig was the provision of an extension piece to compensate for the additional height beneath the tunnel working section in the 11' x 8' wind tunnel. This dimension was carefully surveyed and an allowance made for possible variation in the level of the building floor beneath the tunnel. A simple fabricated extension was inserted between the cast steel pedestal and the rudder dynamometer using an identical pattern of bolt holes for attachment, see figure 11.

Figure 12 shows the rudder rig for both wind tunnel configurations. The six-component strain gauge dynamometer sits directly underneath the wind tunnel floor. The rudder stand is independent of the wind tunnel working section and for further support can be braced to the structure of the wind tunnel building.

The series of rudders used in the original freestream work were of too small a chord for the rudder/propeller investigation. Therefore, a new series of rudders were designed for these tests. So far two rudders have been manufactured the details of which are described in the following sections.

3.2 Instrumentation

The investigation of pressure distribution across the rudder surface required the provision of pressure tubes at numerous points on the rudder surface. The method used to achieve this is based on that used in the freestream tests.

Figure 13 illustrates the technique. Thin grooves were routed in a spanwise direction for all the required chordwise pressure locations. A plastic tube with diameter equal to that of the groove was glued inside the groove ensuring that the tube was always below the surface of the rudder. The surface of the rudder is then made good with filler. For all the required pressure locations a fine 1mm diameter hole is drilled through the filler into the plastic tube. One end of each tube is sealed and the other is taken out of the tunnel and connected to a port on a Scanivalve. For taking pressure measurements all holes bar one on a particular tube are sealed with clear adhesive tape.

3.3 Semi-Balanced Skeg Rudder - RAP 1.

Figure 14 shows the Semi-Balanced Skeg rudder manufactured. Its dimensions are those of the original Rudder No 2 [8] scaled up to have a mean chord of 667mm and total span of 1m. The rudder has a taper ratio of 0.8 and a section profile of NACA0020.

The rudder can be used either as a skeg or all-movable rudder. Two flanged steel tubes are used to attach the rudder and skeg to the dynamometer. The rudder was manufactured out of a solid assembly of Jelutong planks. On completion considerable interior wood was removed to keep the mass as low as possible without sacrificing structural strength.

The numbering and location of the individual pressure tubes are given in Appendix 3.

3.4 All-Movable Rudder - RAP 2 and RAP3

The All-Movable rudder has a rectangular planform with a constant chord of 667mm and a span of 1m. An add-on piece allows the span of the rudder to be increased to 1.2m giving a third rudder (RAP3). The rudder is attached to the dynamometer by a single flanged tube. The section profile was a NACA0020 and the rudder is a scaled version of the original Rudder No. 3 [8] without sweep. Figure 15 shows the all-movable rudder with extension.

The simple shape of the rudder allowed a lighter construction. Thin planks of mahogany were glued across 1" ply formers and then shaped to the desired profile. Four tubes were used for each chordwise pressure position so that a complete spanwise distribution could be achieved in a quarter of the time than that for the skeg rudder. The labelling and location of the tubes are given in Appendix 4.

4 DATA ACQUISITION SYSTEM

The large number of individual data readings required the use of an automated system for data acquisition. The system used is based on that developed for measuring the performance of 2-D aerofoil sections (Turnock[9]). A schematic of the data acquisition system is shown in figure 16.

Data is recorded from the six-component strain-gauge dynamometer which measures the forces and moments acting on the rudder. The propeller's torque and thrust are measured using another strain gauge dynamometer. The propeller revolutions are measured indirectly via a voltage proportional to frequency generated by the Electric Motor Controller. Pressure measurements over the surface of the rudder are obtained using a compressed air stepping Scanivalve which expose four differential pressure transducers to one of 36 input ports each. This allows pressure data to be measured from a maximum of 144 individual pressure tubes.

The measurement component of the system is an accurate digital voltmeter connected to one of 15 input channels. the voltmeter and input channels are controlled by software running on a RM Nimbus PC via an IEEE connection. The stepping of the scanivalve is also controlled via an RS232c system connected to the PC. The computer provides data storage and backup through the use of twin floppy disk drives. For the initial test setup used, only two of the four pressure transducers were used.

4.1 System Hardware

The data acquisition hardware used for the Rudder and propeller rig was mainly existing equipment with well documented performance. The individual system components are described below and the overall layout for data flow is given in figure 16. Plug-socket connections and wiring boxes were used to allow rapid wiring of the system and to minimise problems with bad connections.

4.1.1 Six-Component Strain-Gauge Rudder Dynamometer

The design, construction and calibration of this dynamometer is described by Molland [6]. Designed originally for freestream semi-balanced skeg rudder tests no modifications were required for use. A new plug was fitted to the wiring to facilitate connection to the system. Only five force/moment components were used. These were Lift (L), Drag (D), and the z, x and y Moments (Mz, Mx and My). The strain gauge bridge circuits used Fylde units to provide a bridge voltage supply and balance. In use the zero wind-off voltages and supply voltages were measured regularly.

4.1.2 Thrust-Torque Propeller Dynamometer:

This dynamometer was designed and constructed specifically for the Propeller rig. The design and calibration of this dynamometer, manufactured in the University of Southampton workshops, is detailed in Molland[3]. Design maximum loadings are 110 Nm torque and 700N Thrust. The rotational nature of the dynamometer means that a stationary null-loading voltage cannot be measured. Instead, a null-loading calibration procedure has been developed. This uses the fact that propeller thrust and torque at low speed are proportional to the revolutions squared. With the wind speed at zero the dynamometer bridge voltages are measured for three different revolutions, typically 300, 600 and 1000 RPM. These voltages are plotted against the square of the revolutions and a least-squares fit applied to determine the effective null-loading voltage. This procedure is quick to carry out and was repeated regularly to check for possible drift in the estimated null-loading voltage.

4.1.3 Fylde Bridge-Voltage Units.

These units were purchased by the Department for general use with strain-gauge bridge circuits. They provide an excellent stable supply voltage with negligible drift. The units provide a facility for compensating for the offset voltage. It is possible to vary both supply voltage in the range 1-15v and either to use the unit without amplification of the bridge balance voltage or with amplification in the range 1-100. For these tests an accurate digital micro-voltmeter was available so no amplification was used. This ensures that potential non-linear amplification problems and the calibration of the amplification factor is not needed.

4.1.4 Minate -Solartron Digital Voltmeter (7061)

This is part of the 11' x 8' Wind Tunnel equipment. Voltages can be measured to an accuracy of 0.1 microvolt which provide good resolution for strain-gauge bridge circuits with designed outputs in the range 0-5,000 microVolts. The voltmeter is controlled via an IEEE parallel cable from any IEEE compatible device. The voltmeter can be connected to up to 16 different automatic 16-channel switches which can allow up to 255 different voltages to be accessed. The performance and operation of the voltmeter is detailed in [10].

4.1.5 Minate Solartron Digital switch (7010)

In these tests a single automatic switch unit was connected to the digital voltmeter. This provides up to 15 input channels. The channels used and their connections are shown in Appendix 5. The switching time between channels is of the order of 0.5 second which is adequate for the quasi-steady state readings from the dynamometers and scanivalve.

4.1.6 Scanivalve Controller

For these tests a gang of four 36 port rotary differential pressure transducers (Scanivalve) was used. A compressed air supply is used to provide power to step the scanivalves around the 36 input ports. The reference pressure for all four transducers was taken from the tunnel wall static pressure in the entrance to the tunnel working section. The stepping of the scanivalve is driven by a custom built unit. An 8 bit parallel cable was used to provide the signal to either step the scanivalves one port or to return the scanivalve to the Home position. A four bit code is used (see Appendix 6) to monitor the position of the scanivalve input port. The 8 bit parallel cable was connected via the BBC interface on a RM Nimbus Parallel Piconet box[11] and RS232c cable to the COM1 port of RM Nimbus PC.

4.1.7 RM Nimbus PC

The Nimbus PC is used in Ship Science Department as the main means of controlling data acquisition systems and substantial experience has been gained with such systems. The configuration of the system used for these tests consisted of a dual floppy disk RM Nimbus PC fitted with a GPIB interface board. The GPIB board[12] provided the IEEE connection necessary to control the Minate digital voltmeter and automatic rotary switch. The COM1 port was used to control the scanivalve controller and a printer connected to the PRN port. The dual floppy disks allowed two copies of data files to be made and provides a back-up facility. The control software used is described in the next section.

4.2 Acquisition Software

The software used for acquiring data and controlling the devices connected to the Nimbus PC is based on that developed for the 2-D aerofoil tests (Turnock[9]). Executable programs were developed using the PROPASCAL dialect of Pascal as wide variety of procedures were available for driving the devices connected to the PC.

To ease subsequent data analysis a flexible menu driven system was used to allow the operator to define which particular subset of data was to be recorded. A graphical screen display was used to allow monitoring of the data acquired and to produce pressure plots on the screen. A single data file was created for each data recording session which contained all the information necessary to analyse the data. This data was stored in ASCII format and allowed a readable hardcopy to be made immediately after a data test. A setup data file (SETUP.RAP) stores variable system parameters, calibration constants for the dynamometer and pressure transducers, and other information relevant to the individual test. This data file is read in at the beginning of each test and provides the unique test number. At the end of each test run it is rewritten, including any changes.

The software was developed as individual units which could be tested independently. Figure 17 shows a chart of the software units and their function. The GRAPHSEG and MINSEG units are described in Turnock[9] and DISPSEG in Hyde et al.[13]. The other units and overall program are described in the following sections. Listings of SCANRAP.PAS, RAPSEG.PAS and INITRAP.PAS are given as Appendix 7.

4.2.1 SCANRAP

This segment contains procedures used for driving the stepping of the scanivalve and registering which port it is measuring. Commands to step and home the scanivalve are sent across to an 8 bit parallel port (Byte 64 and 128 respectively). The position of the scanivalve is recorded on the other 6 bits and the values of those bits have to be decoded to give a number between 0 and 35.

4.2.2 RAPSEG

This contains high-level procedures for carrying out specific tasks by using the general functions and procedures defined in GRAPHSEG, SCANRAP, MINSEG and DISPSEG. The main procedures are for:

1: Writing and reading acquired data to the floppy disks

2: Displaying information on the screen either in a graphical or text format.

3: Processing data acquired from the Minate and converting it into actual values for display purposes.

4: Controlling the Scanivalve.

4.2.3 INITRAP

The program INITRAP is used to actually run the data acquisition system using a series of menu displays for the user to determine the required option to carry out. The main options are series of high-level procedures defined in the segment RAPSEG. The user is allowed to check the performance of both the minate system and the stepping of the scanivalve independent of a particular test.

5 INSTALLATION AND COMMISSIONING

5.1 General

This section contains guidance notes for the successful installation of the overall propeller and rudder rig in the 11' x 8' Wind Tunnel. As yet (October 1990) the rig has not been used in the 7' x 5' Wind Tunnel. however, the overall process should be similar to that of installing the rig in the 11' x 8 ' Wind Tunnel.

The large amount of equipment associated with the test rig and Data Acquisition system has to be moved from storage prior to testing in the wind tunnel and removed once the test programme has been concluded. This operation is complicated by the poor approach on the access road to the 11' x 8' Wind Tunnel. A '1-ton' flat-bed lorry has been used to transport the equipment with a fork-lift truck used to load the Motor frame and Middle frame of the Propeller Rig. An awkward item to move is the cast-steel pedestal for the Rudder rig and were possible this has been left at the Wind Tunnel building. The overall operation to move the equipment takes about 1 hour with four people required at each end for loading/unloading.

The floor of the working section of both Wind Tunnels consists of blockboard panels. These 4' x 4' panels are through bolted to the lateral cross-beams of the tunnel. In the 11' x 8' tunnel the two components of the rig fit into the space of two adjacent panels. For these tests, two pairs of 4' x 2' panels were used. A slot was cut in the propeller rig panels to provide clearance for the propeller support tubes and drive belt. The slot also allows room for longitudinal movement of the propeller rig relative to the (fixed) rudder rig for the parametric study of relative propeller/rudder separation. The rudder rig panels had a clearance hole cut for the main rudder support tube and the pressure tubes to pass through. At the upstream end an arc slot was removed to allow movement of the skeg support tube for angles between -35 and 35 relative to the inflow.

The rudder and propeller can be installed independently of each other. For convenience it is best to assemble the propeller rig framework first unless the test programme requires otherwise. Installation of the three main components of the overall rig is described in the following sections.

5.2 Propeller Rig

So far the Propeller rig has been sub-divided into just three sections: Motor frame, Middle frame and leg Assemblies; Top Frame and Above Wind Tunnel Floor assembly,; and Instrumentation Framework. With two 4' x 4' panels removed from the working section floor the motor frame is moved into its approximate position and a fork-lift used to lift the Top Frame into position. The two units are then bolted together. The Instrumentation Framework is then placed onto the matching pads on the Above Wind Tunnel Floor assembly. Care is taken to ensure that the Drive Belt is fed down through the clearance holes. The Instrumentation Framework is securely bolted down and the propeller rig is ready to be placed in its final position. The vertical position and levelling of the Propeller axis of rotation is an iterative process. dependent on the process of belt tensioning. as a first stage the four adjustable feet are lowered and the weight of the rig taken off the four castors.

5.2.1 Belt Tensioning

The correct tensioning of the drive and motor belt is the most complex part of the rig installation. To aid this process a special jig was designed and is shown as Figure 18. This consists of a steel plate which bolts across the bottom of the two slotted plates which contain the lay-shaft assembly. Four lengths of studding are fixed into the lay-shaft assembly and pass through a nut arrangement in the bottom plate. the nut arrangement allows the lay-shaft assembly to be move both up and down relative to the bottom plate.

Correct tensioning of the Motor and Drive belt consists of the following stages:

1) With the Middle frame lowered relative to the Motor frame, fit both belts onto the lay-shaft assembly pulleys and the Motor belt onto the Motor pulley. Insert bolts through slots into lay-shaft assembly.

2) Bolt tensioning jig onto bottom of slotted plates.

3) Raise Middle Frame relative to the Motor frame until there is some tension in Motor belt.

4) Use belt tensioner to tension Drive belt ensuring lay-shaft and pulley-shaft are parallel (ie. belt does not bear up hard against either end of pulleys when the shafts are rotated).

5) Adjust Middle Frame to tension Motor Belt and to ensure its level running.

6) Finally, tighten all bolts between lay-shaft assemble and slotted plates and between Middle Frame and Motor Frame.

Phase 3, 4 and 5 will probably have to be repeated to achieve the final configuration.

5.5.5 Final Setting Up

Figures 19, 20 and 21 show various photographs of the Propeller rig both above and below the working section floor. Once the belts are correctly tensioned final levelling and horizontal positioning of the rig is done using the adjustable feet. By gentle tapping of these feet the whole rig can be 'walked' to the required position. a spirit-level on the propeller shaft is used to determine the level of the rig. The propeller fairing is slid into position and then taped down. The power cable from Motor control system has to be connected by an Electrician to the Motor. The power cable can be simply plugged into the wall socket. The remote control lead is fed up into the wind tunnel control room and connected to the motor control box.

5.3 Rudder Rig

The installation of the rudder rig is an straightforward exercise. Figure 22 shows various photographs of the rudder rig. The rudder extension piece, original extension piece and dynamometer are simply bolted onto the cast-steel pedestal. As shown in the photographs the overall rig is braced to the wind tunnel building frame and clamped to the floor. Packing and shims are used to adjust the level of the rig. The scanivalve unit is clamped to the side of the rudder extension piece. A compressed air line is connected to the unit and two tube runs to the working section pitot-static probe to provide reference pressures. Installation of the rig is detailed in Molland [6].

5.4 Data Acquisition System

AS the whole system is independent of the wind tunnel wiring circuits, installation is a matter of setting up the instrumentation in a convenient location. Wiring up of the equipment is done using specially made cable and plug arrangements to minimise the risk of wrong connections.

5.5 Commissioning

The Propeller and Rudder rig was successfully commissioned in march 1990. The first trials of the propeller drive system were carried out in the George Edwards Laboratory, without the Propeller and hub assembly. It was demonstrated that the belt drive system worked and could safely run up to the maximum speed of 3000 R.P.M.

To date (October 1990) two series of two weeks of wind tunnel testing have been carried out. The first series in March 1990 used the RAP1 Skeg Rudder and validated all the various components of the rig. the second series in August 1990 tested mainly the RAP2 and RAP3 rudders and carried out a detailed parametric study.

The propeller rig has been operated, without problem, at the extremes of its operational envelope. Some belt resonance occurs at specific speeds but this imposes no major limitations on its use.

CONCLUSION

A rig has been developed which allows the influence of a Propeller upstream of a rudder to be accurately investigated. An automated data acquisition system has been developed which allows forces, moments and surface pressures on the rudder to be measured and provides information about the thrust and torque developed on the propeller. The rig can be separated to carry out freestream tests on both rudders and propellers.

The rig allows parametric studies to be made of rudder/propeller interactions which will provide the basis for developing theoretical and empirical formulations for use in ship manouvering calculations.

The rig has been developed so that it can be used in both of low-speed wind tunnels at the University of Southampton. The installation time for the whole rig has been kept to a minimum to allow maximum use to be made of expensive wind tunnel facilities.

ACKNOWLEDGEMENTS

The work described in this report covers part of a research project funded by the S.E.R.C./M.O.D. through the Marine Technology Directorate Ltd. under research grant Ref No GR/E/65289.

REFERENCES

1. Molland, A.F.,& Turnock, S.R.," The Design and Construction of Propeller Blades in Composite Materials for a Wind Tunnel Model". University of Southampton, Ship Science Report 41, __ 1990.

2. Turnock, S.R.," Computer aided Design and Numerically Controlled Manufacture of a Split mould for a Composite Model Ship Propeller", University of Southampton, Ship Science Report 42, __ 1990.

3. Molland, A.F., & Turnock, S.R.," The Design, Construction and Calibration of a Thrust and Torque Dynamometer for a Wind Tunnel Propeller Model", University of Southampton, Ship Science Report 44, __ 1990.

4. Abbott,I.H. & Von Doenhoff, A.E.," Theory of Wing Sections", Dover Publications. 1958.

5. _,"General Purpose AC Adjustable Drives Fuji Inverters: FRENIC5000G5", Fuji Electric.

6. Molland, A.F." The Design, Construction and Calibration of a five-component strain gauge wind tunnel dynamometer " , University of Southampton, Ship Science Report 1/77, Nov. 1976.

7. Molland, A.F., " The freestream Characteristics of a Semi-Balanced Ship Skeg Rudder", University of Southampton, Ship Science Report 3/77, May 1977.

8. Molland, A.F., " Further freestream Characteristics of a Semi-Balanced Ship Skeg Rudder", University of Southampton, Ship Science Report 2/78, August 1978.

9. Turnock, S.R., " Two-Dimensional Wind Tunnel Tests of SSS Series 1 Structurally Efficient Aerofoils for Wind Turbines", University of Southampton, Ship Science Report 38, June 1989.

10. _,"Minate Solartron 7061 Digital Voltmeter Technical Manual: Part 3", Schlumberger Electronics.

11. _,"Piconet Parallel Interface Module: How to use it", PN15283, Research Machines Ltd. 1985.

12 _, "The GPIB Interface", PN15049, Research Machines Ltd., 1986.

13. Cardases, M.H., Hyde, K.M. & Makinson, I.D.," Horizontal Planar Motion Mechanism", Southampton University Four Year Course in Engineering, Design project 1987.

Appendix 1 Dimensions of Propeller Support Structure

	NACA 64-021 (MODIFIED TO 24% thickness/chord)								INTERSECTION LINE		
	Y	X									
		550	575	600	640	680	720	750	X	Y	Z
a	0	0									87.0
b	10.6	2.75									86.4
c	12.7	4.1									87.1
d	16.2	6.9									87.3
e	22.4	13.8									86.2
F	31.3	27.5									83.9
g	38.0	41.3									81.3
h	43.4	55.0									78.8
i	52.0	82.5		AS FOR CHORD = 550							73.4
j	58.4	110.0									68.5
k	62.9	137.5									64.3
l	65.9	165.0							AS FOR CHORD = 750		61.3
m	67.3	192.5									59.8
n	67.0	220.0									60.2
o	64.4	247.5									62.9
p	60.4	275									66.8
q	55.2	302.5	305	307.5	311.5	315.5	319.5	322.5			71.0
r	49.3	330	335	340.0	348	356	364	370			75.3
s	42.7	357.5	365	372.5	384.5	396.5	408.5	417.5			79.2
t	35.6	385	395	405	42:	437	351	465			82.7
u	28.3	412.5	425	437.5	457.5	477.5	497.5	512.5			85.4
v	21.1	440	455	470	494	518	542	560			87.5
w	14.2	467.5	485	502.5	530.5	558.5	586.5	607.5			88.9
x	8.0	495	515	535	567	599	631	655			89.6
y	2.9	522.5	545	567.5	603.5	639.5	675.5	702.5			89.95
z	0.0	550	575	600	640	680	720	750			90.0

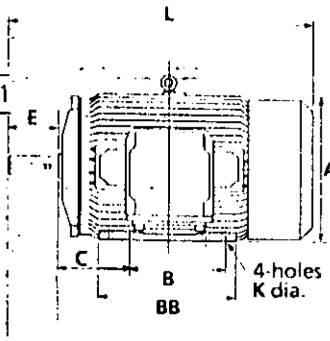
NOSE CONE	
X	Z
50	90.0
30	89.8
10	88.2
-10	85.0
-30	81.0
-50	75.0
-70	67.0
-90	55.5
-110	38.5
-130	0

HUB	CONE
X	Y
0	90.0
15	89.8
30	88.2
45	85.0
60	81.0
75	75.0
90	67.0
105	55.5
120	38.5
135	0.0

Appendix 2 Details of Proprietary Items

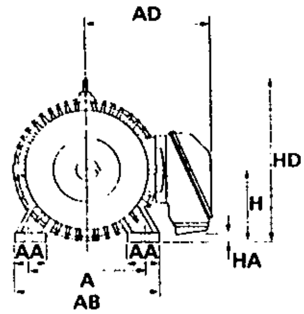
GEC 30 KW ELECTRIC MOTOR - FRAME SIZE 200L

Frame Size	Poles	ALL DIMENSIONS IN MILLIMETRES											
		A	AA	AB	AC	AD	B	BA	BB	C	D		
											Nom	Tolerance	
D200L	2-8	318	90	390	412	346	305	—	365	133	55	+0·030	+0·011



Frame Size	Poles	ALL DIMENSIONS IN MM		Coupling Bore	
		K	L	Nom	Tol
D200L	2-8	19	776	54·975	+0·030

RECOMMENDED BORE OF COUPLING OR PULLEY (mm)						
Pulley Bore		Keyway Width		Keyseat Depth		Nett Mass
Nom	Tol	Nom	Tol	Nom	Tol	kg
55·000	+0·030	15·979	+0·043	59·3	+0·2	280



FENNERS PULLEYS, BELTS AND TAPER LOCK BUSHES

8mm Pitch and 85mm Wide HTD

Drive Belt:

1040-8M-85

Motor Belt:

7' x 5 ' Configuration: 1280-8M-85
11' x 8' Configuration: 2600-8M-85

Drive Belt Pulleys:
40-8-85 (043M0040) 40 teeth, Pitch Dia =101mm
Motor Belt Pulleys:
64-8-85 (043M0064) 64 teeth, Pitch Dia =163mm

Taper Lock bushes to fit 35mm and 55mm shafts respectively

D.J. MOULDINGS 20-WAY SLIP RING

Part No: 150-2-20 Internal Diameter = 30mm
External Diameter = 48mm
Length = 110mm
20 Way Silver Rings and Carbon Brushes

Appendix 3 Pressure Tube Numbering and Location for Semi-Balanced Skeg Rudder -RAP1

Bundle	Tube No.	ColourScani	Chord %	
A	1	Orange	T2 0	0
"	2	Green	T2 1	2.5
"	3	Blue	T2 2	5
"	4	Yellow	T2 3	10
"	5	Red	T2 4	20
"	6	Black	T2 5	30
"	7	Orange	T2 6	40
D	8	Orange	T2 24	50
"	9	Black	T2 23	60
"	10	Red	T2 22	70
"	11	Yellow	T2 21	80
"	12	Blue	T2 20	90
"	13	Green	T2 19	95
E	14	White	T2 30	95
"	17	Black	T2 29	90
"	18	Red	T2 28	80
"	19	Yellow	T2 27	70
"	20	Blue	T2 26	60
"	21	Green	T2 25	50
B	23	Green	T2 7	40
"	24	Blue	T2 8	30
"	25	Yellow	T2 9	20
"	26	Red	T2 10	10
"	27	Black	T2 11	5
"	28	Orange	T2 12	2.5
F	30	Green	T2 31	35
"	31	Yellow	T2 32	32
"	32	Red	T2 33	30
"	33	Black	T2 34	28
"	34	White	T2 35	26
C	36	Orange	T2 18	25
"	37	Black	T2 17	26
"	38	Red	T2 16	28
"	39	Yellow	T2 15	30
"	40	Blue	T2 14	32
"	41	Green	T2 13	35
G	42	White	T3 0	0
"	44	Yellow	T3 1	2.5
"	45	Red	T3 2	5
"	46	Green	T3 3	10
"	47	White	T3 4	15
"	48	Yellow	T3 5	20
"	49	Red	T3 6	25
"	50	Green	T3 7	30
"	54	Green	T3 14	30
H	55	Orange	T3 13	25
"	56	Yellow	T3 12	20
"	57	Green	T3 11	15
"	58	Red	T3 10	10
"	59	Yellow	T3 9	5
"	60	White	T3 8	2.5

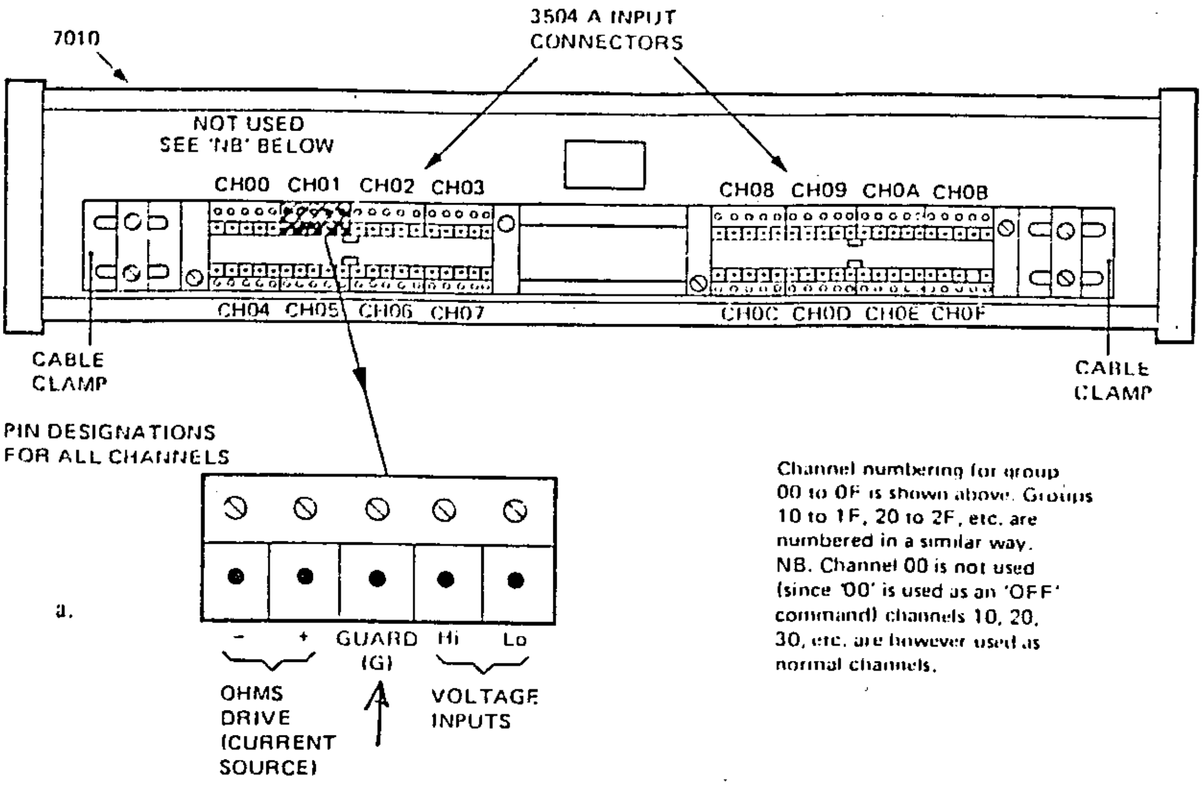
Appendix 4 Pressure Tube Numbering and Location for
All-Movable Rudders -RAP2 and RAP3

Scani Valve Connections

Port	Scani 1	Scani 2	Scani 3	Scani 4
0	Pd-Ps	Pd-Ps	Pd-Ps	Pd-Ps
1	Ps	Ps	Ps	Ps
2	A1	B1	C1	D1
3	A2	B2	C2	D2
4	A3	B3	C3	D3
5	A4	B4	C4	D4
6	A5	B5	C5	D5
7	A6	B6	C6	D6
8	A7	B7	C7	D7
9	A8	B8	C8	D8
10	A9	B9	C9	D9
11	A10	B10	C10	D10
12	A11	B11	C11	D11
13	A12	B12	C12	D12
14	A13	B13	C13	D13
15	A14	B14	C14	D14
16	A15	B15	C15	D15
17	A16	B16	C16	D16
18	A17	B17	C17	D17
19	A18	B18	C18	D18
20	A19	B19	C19	D19
21	A20	B20	C20	D20
22	A21	B21	C21	D21
23	A22	B22	C22	D22
24	A23	B23	C23	D23
25	A24	B24	C24	D24
26	A25	B25	C25	D25
27	E1	E8	E15	E22
28	E2	E9	E16	E23
29	E3	E10	E17	E24
30	E4	E11	E18	E25
31	E5	E12	E19	E25
32	E6	E13	E20	E25
33	E7	E14	E21	E25

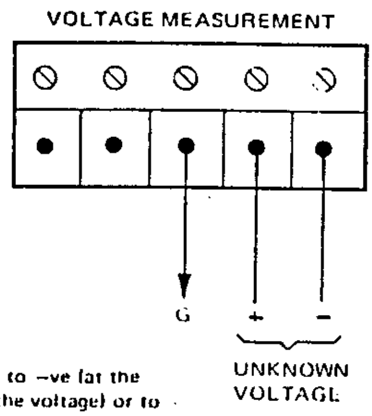
Pd = Tunnel Dynamic Pressure
Ps = Tunnel Static Pressure
(Used as reference for all 4 scani's)

Appendix 5 Minate Solartron 7010 Connections



a.

Channel numbering for group 00 to 0F is shown above. Groups 10 to 1F, 20 to 2F, etc. are numbered in a similar way.
NB. Channel 00 is not used (since '00' is used as an 'OFF' command) channels 10, 20, 30, etc. are however used as normal channels.



b.

Connect G to -ve (at the source of the voltage) or to the source of common mode voltage.

Channel	Connection Series A	Connection Series B
01	RPM	RPM
02	Thrust	Thrust
03	Torque	Torque
04	Lift	Lift
05	Drag	Drag
06	Mz	Mz
07	Mx	Mx
08	My	My
09	Scani T2	Scani T1
0A	Scani T3	Scani T2
0B	V thrust	Scani T3
0C	V torque	Scani T4
0D	V Lift	V Thrust
0E	V Drag	V Torque
0F	V Mz	V Lift

Appendix 6 Scanivalve Output Code

ScaniValve Port	6 bit Output Code					
	32	16	8	4	2	1
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	1	0	0	0	0
11	0	1	0	0	0	1
12	0	1	0	0	1	0
13	0	1	0	0	1	1
14	0	1	0	1	0	0
15	0	1	0	1	0	1
16	0	1	0	1	1	0
17	0	1	0	1	1	1
18	0	1	1	0	0	0
19	0	1	1	0	0	1
20	1	0	0	0	0	0
21	1	0	0	0	0	1
22	1	0	0	0	1	0
23	1	0	0	0	1	1
24	1	0	0	1	0	0
25	1	0	0	1	0	1
26	1	0	0	1	1	0
27	1	0	0	1	1	1
28	1	0	1	0	0	0
29	1	0	1	0	0	1
30	1	1	0	0	0	0
31	1	1	0	0	0	1
32	1	1	0	0	1	0
33	1	1	0	0	1	1
34	1	1	0	1	0	0
35	1	1	0	1	0	1

Input Codes: Bit 8 (128) HOME Scani to Port 0
Bit 7 (64) STEP Scani to next port

Appendix 7 Program Listing

7.1 SCANRAP.PAS

7.2 RAPSEG.PAS

7.3 INITRAP.PAS

Appendix 7.1 SCANRAP.PAS

```
SEGMENT SCANRAP;

CONST

  Home_That_Scani=128;
  Step_That_Scani=64;

PROCEDURE OPENSCANREAD

(var fpic:text);

  var nameit:string;

begin
    nameit:='LPT1';
    assign(fpic,nameit);
    reset(fpic);
end;

PROCEDURE OPENSCANWRITE

(var fpic:text);

  begin
    assign(fpic,'LPT1');
    rewrite(fpic);
end;

PROCEDURE CloseScan
(var fpic:text);

  begin
    close(fpic);
end;

PROCEDURE WriteValue
(value:integer);

  var fpic:text;

  begin
    OpenScanWrite(fpic);
    Write(fpic,chr(value));
    CloseScan(fpic);
end;

PROCEDURE ReadValue
(var value:integer);

  var fpic:text;
     pic:char;

  begin
    OpenScanRead(fpic);
    Read(fpic,pic);
    CloseScan(fpic);
```

```
    value:=ord(pic);
end;

PROCEDURE DecodeScani
(var portnumber:integer);

  var
    targ,value,
    bit5,bit6:integer;

 begin
    ReadValue(value);
    if value>31 then begin
      bit6:=1;
      value:=value-32;
    end else bit6:=0;
    if value>15 then begin
      bit5:=10;
      value:=value-16;
    end else bit5:=0;
    targ:=bit6+bit5;
    case targ of
     0:portnumber:=value;
     1:portnumber:=20+value;
    10:portnumber:=10+value;
    11:portnumber:=30+value;
    end;


end;

PROCEDURE HomeScani
(var portat:integer);

  var portly:integer;

  begin
    DecodeScani(portly);
    DecodeScani(portly);
    if portly=0 then
      portat:=1
    else begin
     WriteValue(128);
     repeat
       DecodeScani(portly);
     until portly=0;
     portat:=1;
    end;
end;

PROCEDURE StepScani
(var portat:integer);

  var portly:integer;

  begin
    writevalue(64);
    repeat
```

```
     DecodeScani(portly);
     writeln('  At Scani
     Port = ',portly:8);
    until portly=portat;
    portat:=portat+1;
end;
{===========================}
BEGIN
END.
```

Appendix 7.2 RAPSEG.PAS

```
SEGMENT NEWRAP;

{ This segment contains routines used for controlling the
Minate Solartron digital voltmeter and 16 channel switch.
It is part of the EXPRAP.pas suite of programs for use in
testing of rudder and propeller interactions in the
Southampton University 11'* 8' wind tunnel.

     Written by: S. Turnock
          Date : 23/2/90}

CONST

   nints  = 10;
   mreals = 20;
   tnp =    30;
   config = 0;
   TestConditions=1;
   supplyvoltage=2;
   WindOff=3;
   pressures=4;
   QT=5;
   RPM=6;
   RudderDyno=7;
   EndTestRun=8;
   comment=9;
   {indicator}
   ShowVoltage=1;
   ShowSD=2;
   ShowValue=3;
   ShowCorrect=4;
   ShowSupply=5;

TYPE

   nthym = record
            loc,num:integer;
           end;

   channels = Array[1..15]
              of real;

   scanichan = array[1..60]
               of real;

   header = record
              title:string;
         nint,mreal:integer;
        ints:array[1..nints]
             of integer;
      reals:array[1..mreals]
            of real;
    descrip:array[1..tnp]
             of string;
            end;
```

```
   ChannelData = record
       name:array[1..15]
            of string;
       ConvConst:channels;
       nobs:array[2..7] of
            nthym;
            end;
    TestCond = record
       valts:array[1..3] of
             real;
       nvlts:array[1..3] of
             string;
       lims:array[1..3,1..2]
            of real;
       end;

  SysConst = record
        bools:array[1..4] of
              boolean;
         nms:array[1..4] of
             string;
            end;

 arr5real = array[1..5,1..5]
            of real;
 arr2real = array[1..2,1..2]
           of real;

 ScaniData = record
NoRud,NoSkeg,NoSpan:integer;
span,chord:array[1..10] of
           real;
  chords:array [1..10] of
         scanichan;
      end;

      SetExpUp = record
         headings:header;
         chans:ChannelData;
         Scnst:SysConst;
         RudMat:arr5real;
         QTmat:arr2real;
         Scan:ScaniData;
       end;

    ExpData  = record
  L_value,L_sd,
  T_value,C_value,woff:
              channels;
    TC:Testcond;
S_value,S_sd,S_press:
                scanichan;
              end;

      INT = -32768..32767;    {Graphics output type}
          PLOTDATA = ARRAY[1..60] of REAL;
{+++++++++++++++++++++++++++}
```

```
FUNCTION Cstat:boolean;
               EXTERNAL;

     {Dispseg Externals   written by K.M.Hyde}

PROCEDURE AnyKeyToCont
              EXTERNAL;
PROCEDURE PrintReal
       (Text:String;
X:Real;F,D:Integer);
             EXTERNAL;
FUNCTIONInputInt
(Text:String;Min,Max:Integer):Integer;
             EXTERNAL;
FUNCTION InputReal
(Text:String;Min,Max:Real;
f,d:integer) :Real;
             External;
FUNCTION Question
(Text:String) : Boolean;
             EXTERNAL;

     {  MinSeg Externals }

PROCEDURE StringToReal
(Value : String; VAR Result :Real);
            EXTERNAL;

PROCEDURE WriteSolar
(message: string);
            EXTERNAL;

PROCEDURE ReadSolar
(VAR answer :string);
            EXTERNAL;

{     Graphseg Externals }
PROCEDURE PlotReal
( Message : String;X:Real;
F,D:Integer ;
Xcoord,Ycoord : Integer);
           EXTERNAL;
PROCEDURE DrawBox
(Imax,Imin,Jmax,Jmin : INT);
           EXTERNAL;
PROCEDURE PlotText
(x,y : INT; s: STRING);
            EXTERNAL;
PROCEDURE PlotPressure
(Xdata,Ydata:PlotData;
Xmax,Xmin,Xorig,
Ymax,Ymin,Yorig:real;
Imin,Jmin,Imax,Jmax,
NoRud,NoSkeg:INT);
           EXTERNAL;

{       Scani externals }
PROCEDURE HomeScani
```

```
(var portat:integer);
           EXTERNAL;
PROCEDURE StepScani
(var portat:integer);
           EXTERNAL;

{    Internal Procedures }

PROCEDURE ReadChannel
(start,number,times:integer;
 VAR values,sds:channels);

  var
    i,j,test:integer;
    mess,instring:string;
    vault:real;
  storeit:array[1..10,1..15]
          of real;

 begin

   For i:=1 to 15 do
    values[i]:=0.0;
    CASE start of
     1:mess:='C01';
     2:mess:='C02';
     3:mess:='C03';
     4:mess:='C04';
     5:mess:='C05';
     6:mess:='C06';
     7:mess:='C07';
     8:mess:='C08';
     9:mess:='C09';
     10:mess:='C0A';
     11:mess:='C0B';
     12:mess:='C0C';
     13:mess:='C0D';
     14:mess:='C0E';
     15:mess:='C0F';
   end;
   If number >1 then begin
    test:=start+number-1;
    CASE test of
      2:mess:=
        concat(mess,'>02');
      3:mess:=
        concat(mess,'>03');
      4:mess:=
        concat(mess,'>04');
      5:mess:=
        concat(mess,'>05');
      6:mess:=
        concat(mess,'>06');
      7:mess:=
        concat(mess,'>07');
      8:mess:=
        concat(mess,'>08');
      9:mess:=
```

```
        concat(mess,'>09');
      10:mess:=
         concat(mess,'>0A');
      11:mess:=
         concat(mess,'>0B');
      12:mess:=
         concat(mess,'>0C');
      13:mess:=
         concat(mess,'>0D');
      14:mess:=
         concat(mess,'>0E');
      15:mess:=
         concat(mess,'>0F');
   end;
  end;
   For j:=1 to times do
    begin
     writesolar(mess);
     For i:=1 to number do
       begin
        readsolar(instring);
        StringToReal(
            instring,vault);
        storeit[j,i]:=vault;
        values[start+i-1]:=
        values[start+i-1]+
              (vault/times);
     end;
   end;
   { find s.d. }
   For i:=1 to number do
        sds[i]:=0.0;
   For j:=1 to times do
     begin
     For i:=1 to number do
       begin
         sds[i]:=sds[i]+
         ((storeit[j,i]*
         storeit[j,i])/
         times);
     end;
   end;
   For i:=1 to number do
     sds[i]:=sqrt(
            abs(sds[i]-
       (values[start+i-1]*
       values[start+i-1])));

end;

PROCEDURE WriteChannel(
start,number,times:integer;
values:channels);

  var i:integer;

  begin
    writeln;
```

```
    writeln('         MINATE
  CHANNEL VOLTAGE VALUES ');
    writeln;
    For i:=1 to number do
      begin
       write(' CHANNEL ');
       write((start+i-1):2);
     write('  Voltage is ');
     writeln(values[start+i-
      1]:15:6);
    end;
    writeln;
    write('   AVERAGE OF ');
    write(times:5);
    writeln(' READINGS');
    writeln;
    writeln('  press return
             to continue ');
    readln;
end;

{+++++++++++++++++++++++++++}
PROCEDURE WritePressures
(number,times:integer;
var values:scanichan);

  var i:integer;

  begin
    writeln;
    writeln('     SCANI-
    VALVE PRESSURE VOLTAGE
    VALUES ');
    writeln;
    For i:=1 to number do
     begin
     write(' PORT ');
     write((i-1):6);
     write('  Voltage is ');
     writeln(values[i]:15:6);
    end;
    writeln;
    write(' AVERAGE OF   ');
    write(times:5);
    writeln(' READINGS');
    writeln;
    writeln('  press return
    to continue ');
    readln;
end;

{+++++++++++++++++++++++++++}

PROCEDURE FileChannelData
(var fname:text;
choice,times:integer;
var E_dat:ExpData;
var stup:SetExpUp);
```

```
var
  spud,i,start,
  number:integer;
   nameit:string;
begin
  case choice of
  6:  nameit:='   RPM
             channel';
  5:  nameit:='  Torque-
      Thrust channels';
  2:  nameit:='  supply
      volts channels';
  3:  nameit:=' windoff
      channels';
  7:  nameit:=' Rudder-
      dynamometer
      channels';
  4:  nameit:='
      Scanivalve';
   end;
 if choice<>pressures then
   begin
     start:=
stup.chans.nobs[choice].loc;
     number:=
stup.chans.nobs[choice].num;
   writeln(fname,number:5,
'  Number of channels');
   writeln(fname,times:5,'
  Number of readings ');
   writeln(fname,' Channel
    Voltage    S.D.   '
   ,nameit,'  Corrected');
  For i:=1 to number do
    begin
    write(fname,
         (start+i-1):4);
   write(fname,
   E_dat.L_value[start+i-
              1]:15:6);
   write(fname,E_dat.L_sd[start+i-1]:15:6);
   write(fname,E_dat.T_value[start+i-1]:15:6);
   writeln(fname,E_dat.C_value[start+i-1]:15:6);
  end;
 end else begin
  start:=9;
  number:=stup.scan.norud + stup.scan.noskeg;
  spud:=stup.headings.ints[8];
  writeln(fname,stup.scan.norud:5,stup.scan.noskeg:5,'
                         Number of Rudder and Skeg');
  writeln(fname,times:5,'  Number of readings ');
  writeln(fname,stup.scan.chord[spud]:15:6,'   Rudder
                                        Chord (mm)');
  writeln(fname,stup.scan.span[spud]:15:6,'   Spanwise
                                      position(mm)');
  writeln(fname,'  Port      Voltage    S.D.    ',
                                nameit,'  position');
```

```
    For i:=1 to number do begin
     write(fname,(i-1):4);
     write(fname,E_dat.S_value[i]:15:6);
     write(fname,E_dat.S_sd[i]:15:6);
     write(fname,E_dat.S_press[i]:15:6);
     write(fname,stup.scan.chords[spud,i]:15:6);
     if i<=stup.scan.norud then writeln(fname,' RUDDER')
     else writeln(fname,' SKEG');
    end;
   end;
end;

{+++++++++++++++++++++++++++++++++++++++++++++++++++++++}

PROCEDURE ReadHeader(var fname:text;
                      var ExpH:Header);
  var i:integer;
  begin
    Readln(fname,ExpH.title);
    Readln(fname,ExpH.nint,ExpH.mreal);
    For i:=1 to (ExpH.Nint+ExpH.mreal) do begin
     if i<=ExpH.Nint then read(fname,ExpH.ints[i])
     else read(fname,ExpH.reals[i]);
     readln(fname,ExpH.descrip[i]);
    end;

end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}

PROCEDURE WriteHeader(var fname:text;
                      ExpH:Header);
  var i:integer;
  begin
    Writeln(fname,ExpH.title);
    Writeln(fname,ExpH.nint:8,ExpH.mreal:8);
    For i:=1 to (ExpH.Nint+ExpH.mreal) do begin
     if i<=ExpH.Nint then write(fname,ExpH.ints[i]:8)
     else write(fname,ExpH.reals[i]:12:4);
     writeln(fname,ExpH.descrip[i]);
    end;

end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE PrintHeader(var ExpH:Header);
  var i:integer;
  begin
    writeln;
    Writeln(ExpH.title);
    For i:=1 to (ExpH.Nint+ExpH.mreal) do begin
     write(' RECORD ');
     write(i:4);
     if i<=ExpH.Nint then write(ExpH.ints[i]:8)
     else write(ExpH.reals[i]:12:4);
     writeln(ExpH.descrip[i]);
    end;

end;
```

```
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE WriteChannelData(var fname:text;
                           var ChD:ChannelData);
  var i:integer;
  begin
    For i:=1 to 15 do begin
     write(fname,ChD.convconst[i]:15:6);
     writeln(fname,ChD.name[i]);
    end;
    For i:=2 to 7 do begin
     write(fname,Chd.nobs[i].loc:5,Chd.nobs[i].num:5);
     case i of
        6:  writeln(fname,'   RPM channel');
        5:  writeln(fname,'  Torque-Thrust channels');
        2:  writeln(fname,'  supply volts channels');
        3:  writeln(fname,' windoff channels');
        7:  writeln(fname,' Rudder-dynamometer
                          channels');
        4:  writeln(fname,'  Scanivalve channel');
     end;
    end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE PrintChannelData(var ChD:ChannelData);
  var i:integer;
  begin
    writeln;
    For i:=1 to 15 do begin
     write('  RECORD ');
     write(i:4);
     write(ChD.convconst[i]:15:6);
     writeln(ChD.name[i]);

    end;
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE ReadChannelData(var fname:text;
                          var ChD:ChannelData);
  var i:integer;
  begin
    For i:=1 to 15 do begin
     read(fname,ChD.convconst[i]);
     read(fname,ChD.name[i]);
    end;
    for i:=2 to 7 do begin
       readln(fname,ChD.nobs[i].loc,Chd.nobs[i].num);
    end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE ReadRudMat(var fname:text;var RM:arr5real);
  var i:integer;
  begin
    readln(fname);
    readln(fname);
   For i:=1 to 5 do begin
   readln(fname,rm[i,1],rm[i,2],rm[i,3],rm[i,4],rm[i,5]);
```

```
    end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE WriteRudMat(var fname:text;
                              RM:arr5real);
  var i:integer;
  begin
    writeln(fname,' Rudder Dynamometer Interaction
                   matrix');
    writeln(fname,'  Lift Drag Mz  Mx  My');
    For i:=1 to 5 do begin
     writeln(fname,rm[i,1]:15:8,rm[i,2]:15:8,rm[i,3]:15:8
            , rm[i,4]:15:8,rm[i,5]:15:8);
    end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE WriteScaniData(var fname:text;
                             var S_D:ScaniData);
  var i,j:integer;
  begin
    writeln(fname,'  SCANI-VALVE PRESSURE DISTRIBUTION');
    writeln(fname,S_D.NoSpan:8,S_D.NoRud:8,S_D.NoSkeg:8,'
            Number of spanwise stations ,rudder,skeg');
    For i:=1 to S_D.NoSpan do begin
     writeln(fname,S_D.span[i]:10:2,S_D.chord[i]:10:2,'
                              Span (mm)    Chord(mm)');
     For j:=1 to S_D.NoRud do begin
       writeln(fname,S_D.chords[i,j]:10:4,'Rudder
                                   Percentage Chord ');
     end;
     For j:=1 to S_D.NoSkeg do begin
       writeln(fname,S_D.chords[i,(S_D.NoRud+j)]:10:4,
                         'Skeg Percentage Chord ');
     end;
    end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE ReadScaniData(var fname:text;
                            var S_D:ScaniData);
  var i,j:integer;
  begin
    readln(fname);
    readln(fname,S_D.NoSpan,S_D.NoRud,S_D.NoSkeg);
    For i:=1 to S_D.NoSpan do begin
     readln(fname,S_D.span[i],S_D.chord[i]);
     For j:=1 to (S_D.NoRud+S_D.NoSkeg) do begin
       readln(fname,S_D.chords[i,j]);
     end;
    end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE ReadQTMat(var fname:text;
                        var Qm:arr2real);
  var i:integer;
  begin
    readln(fname);
    readln(fname);
    For i:=1 to 2 do begin
```

```
    readln(fname,qm[i,1],qm[i,2]);
   end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE WriteQTMat(var fname:text;
                         Qm:arr2real);
  var i:integer;
  begin
   writeln(fname,' Torque-Thrust Interaction Matrix ');
   writeln(fname,'  Thrust    Torque ');
   For i:=1 to 2 do begin
    writeln(fname,qm[i,1]:15:8,qm[i,2]:15:8);
   end;
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE ReadSystemConstants(var fname:text;
                              var sconst:SysConst);
  var i,typ:integer;
  begin
   For i:=1 to 4 do begin
    read(fname,typ);
    readln(fname,sconst.nms[i]);
    if typ=0 then sconst.bools[i]:=FALSE
    else sconst.bools[i]:=TRUE;
   end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE WriteSystemConstants(var fname:text;
                               var sconst:SysConst);
  var i:integer;
  begin
   For i:=1 to 4 do begin
    if sconst.bools[i]=TRUE then write(fname,'   1 ')
    else write(fname,'   0 ');
    writeln(fname,sconst.nms[i]);
   end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE PrintSystemConstants(sconst:SysConst);
  var i:integer;
  begin
   For i:=1 to 4 do begin
    if sconst.bools[i]=TRUE then
    write('  TRUE ') else write('   FALSE');
    writeln(sconst.nms[i]);
   end;
end;
{++++++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE ReadSetUp(var SetUpdata:SetExpUp);

  var fname:text;
  begin
   assign(fname,'setup.rap');
   reset(fname);
   readheader(fname,SetUpData.headings);
   readchanneldata(fname,SetUpdata.chans);
   readsystemconstants(fname,SetUpData.scnst);
```

```
      readRudMat(fname,SetUpData.rudmat);
      readQTMat(fname,SetUpData.QTmat);
      readScaniData(fname,SetUpData.scan);
      close(fname);
  end;
  {++++++++++++++++++++++++++++++++++++++++++++++++++++++}
  PROCEDURE WriteSetUp(var SetUpdata:SetExpUp);

    var fname:text;
    begin
      assign(fname,'setup.rap');
      rewrite(fname);
      writeheader(fname,SetUpData.headings);
      writechanneldata(fname,SetUpdata.chans);
      writesystemconstants(fname,SetUpData.scnst);
      writeRudMat(fname,SetUpData.rudmat);
      writeQTMat(fname,SetUpData.QTmat);
      writeScaniData(fname,SetUpdata.scan);
      close(fname);
  end;
  {++++++++++++++++++++++++++++++++++++++++++++++++++++++}
  PROCEDURE ChangeSetUp(var SetUpData:SetExpUp);
    var mark,nreal,mint:integer;
       blop:string;
    begin
      nreal:=SetUPData.headings.mreal;
      mint:=SetUpData.headings.nint;
      repeat
       page;
       writeln('    Alter Experimental SetUp Data ');
       writeln;
       PrintHeader(SetUpData.headings);
       writeln;
       mark:=InputInt('  To change a value enter record
                number or 0 if correct',0,(mint+nreal));
       if mark<>0 then begin
        writeln;

        if mark>mint then begin
           blop:=concat(' Enter new value of ',
                     SetUpData.headings.descrip[mark]);
          SetUpData.headings.reals[mark]:=
                    InputReal(blop,-10000.0,10000.0,15,2);
          end else begin
          blop:=concat(' Enter new value of ',
                SetUpData.headings.descrip[mark]);
          SetUpData.headings.ints[mark]:=
                    InputInt(blop,-10000,10000);
          end;
       end;
      until mark=0;

      repeat
       page;
       writeln('    Alter Channel SetUp Data ');
       writeln;
       PrintChannelData(SetUpData.chans);
       writeln;
```

```
        mark:=InputInt('  To change a value enter
                  record number or 0 if correct',0,15);
        if mark<>0 then begin
           writeln;
           blop:=concat(' Enter new value of ',
                        SetUpData.chans.name[mark]);
           SetUpData.chans.convconst[mark]:=
               InputReal(blop,-10000.0,10000.0,15,6)
        end;
       until mark=0;
       repeat
        page;
        writeln('    Alter System Constants SetUp Data.');
        writeln;
        PrintSystemConstants(SetUpData.scnst);
        writeln;
        mark:=InputInt('  To change a value enter record
                         number or 0 if correct',0,4);
        if mark<>0 then begin
           writeln;
           SetUpData.scnst.bools[mark]:=
           Question(concat(' Is ',
                  setupdata.scnst.nms[mark],' True '));
        end;
       until mark=0;

   end;
   {+++++++++++++++++++++++++++++++++++++++++++++++++++++}
   PROCEDURE EnterTestCondition(var T_cond:TestCond);
     var i:integer;
     begin
       T_cond.nvlts[1]:=
            '  tunnel atmospheric pressure (mm Hg) ';
       T_cond.nvlts[2]:=
           '  tunnel atmospheric temperature (deg Celsius)';
       T_cond.nvlts[3]:=
           '  tunnel dynamic pressure (mm H2O)';
       T_cond.lims[1,1]:=700.0;
       T_cond.lims[1,2]:=800.0;
       T_cond.lims[2,1]:=0.0;
       T_cond.lims[2,2]:=30.0;
       T_cond.lims[3,1]:=0.0;
       T_cond.lims[3,2]:=10000.0;
       For i:=1 to 3 do begin
        T_cond.valts[i]:=
            InputReal(T_cond.nvlts[i],T_cond.lims[i,1],
                      T_cond.lims[i,2],15,6);
       end;
   end;

   {+++++++++++++++++++++++++++++++++++++++++++++++++++++}
   PROCEDURE WriteTestCondition(var fname:text;
                                   T_cond:Testcond);
     var i:integer;
     begin
       For i:=1 to 3 do begin
        write(fname,T_cond.valts[i]:15:6);
        writeln(fname,T_cond.nvlts[i]);
```

```
    end;

  end;
  {++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
  PROCEDURE OpenTestFile(var TestNo:integer;
                             var fname:text);
    var tit:string;
       sze:integer;
    begin
      TestNo:=TestNo+1;
      str(testno,tit);
      sze:=length(tit);
      case sze of
       1:tit:=concat('b:A_Exp00',tit,'.rap');
       2:tit:=concat('b:A_Exp0',tit,'.rap');
       3:tit:=concat('b:A_Exp',tit,'.rap');
       OTHERWISE;
      end;
      assign(fname,tit);
      rewrite(fname);
  end;

  PROCEDURE OpenPrinter(var fname:text;
                        var stup:SetExpUp;
                        var printeron:boolean);
    begin
      printeron:=stup.scnst.bools[1];
      if printeron=true then begin
       assign(fname,'COMA');
       rewrite(fname);
      end;
  end;
  {++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
  Procedure CloseTestFile(var fname:text);
    begin
      writeln(fname,EndTestRun:5,'   End of File');
      close(fname);
  end;

  procedure WriteCommentFile(var f1,f2:text;
                                  printeron:boolean);
    var
      commie:string;
    begin
      writeln(f1,comment:5,'    comment');
      writeln;
      writeln('    Enter comment,
                   press return to complete');
      readln(commie);
      writeln(f1,commie);
      if printeron=true then begin
       writeln(f2,comment:5,'    comment');
       writeln(f2,commie);
      end;

  end;

  {++++++++++++++++++++++++++++++++++++++++++++++++++++++++}
```

```
PROCEDURE WriteTestFile(var fname:text;
                        choice,times:integer;
                        var stup:SetExpUp;
                        var expup:ExpData);
  begin
    writeln(fname,choice:5,'    MARKER');
    Case choice of
     config:WriteHeader(fname,stup.headings);
     TestConditions:WriteTestCondition(fname,expup.TC);
     supplyvoltage:
          FileChannelData(fname,choice,times,expup,stup);
     WindOff:
         FileChannelData(fname,choice,times,expup,stup);
     pressures:
         FileChannelData(fname,choice,times,expup,stup);
     QT:FileChannelData(fname,choice,times,expup,stup);
     RPM:FileChannelData(fname,choice,times,expup,stup);
     RudderDyno:
         FileChannelData(fname,choice,times,expup,stup);
     EndTestRun:CloseTestFile(fname);
    end;
end;
{===========================================================}
PROCEDURE ReadDataFromMinate(choice,times:integer;
                             var stup:setexpup;
                             var E_dat:ExpData);
  var total,i,start,number,portat:integer;
  begin

   if choice<>pressures then begin
    start:=stup.chans.nobs[choice].loc;
    number:=stup.chans.nobs[choice].num;

ReadChannel(start,number,times,e_dat.L_value,e_dat.L_sd);
    if choice=windoff then begin
     for i:=start to (start+number-1) do
       e_dat.woff[i]:=e_dat.L_value[i];
    end;
   end else begin
     if stup.scan.norud>stup.scan.noskeg then
           total:=stup.scan.norud
     else
           total:=stup.scan.noskeg;
     if stup.scnst.bools[4]=true then
       HomeScani(portat)
     else begin
        writeln('  When scani has been homed press key to
        continue');
        anykeytocont;
     end;
     For i:=1 to total do begin
      ReadChannel(stup.chans.nobs[pressures].loc,
                  stup.chans.nobs[pressures].num,
                  times,e_dat.L_value,e_dat.L_sd);
      if i<=stup.scan.NoRud then begin
         e_dat.s_value[portat]:=e_dat.l_value[9];
         e_dat.s_sd[portat]:=e_dat.L_sd[9];
      end;
```

```
      if i<=stup.scan.NoSkeg then begin
         e_dat.s_value[stup.scan.NoRud+portat]:=
                                   e_dat.l_value[10];
        e_dat.s_sd[stup.scan.norud+portat]:=
                                   e_dat.L_sd[10];
      end;
      if i<>total then begin
        if stup.scnst.bools[4]=true then
                StepScani(portat)
        else begin
         writeln(' When scani has been stepped press key
                    to continue');
          anykeytocont;
        end ;
      end;
     end;
   end;
end;
{==========================================================}
PROCEDURE ConvertDataFromMinate(choice,times:integer;
                                   var stup:setexpup;
                                   var E_dat:ExpData);
  const million=1000000.0;
  var indy,i,ind,start,number:integer;
     V_corr:real;
  begin
   if choice<>pressures then begin
    start:=stup.chans.nobs[choice].loc;
    number:=stup.chans.nobs[choice].num;
    For i:=1 to number do begin
      ind:=start+i-1;

      if (choice=QT) or (choice=RudderDyno) then begin
        indy:=ind+9;
        if indy>15 then indy:=15;
       { converts voltages based on Wind-Off values }
        V_corr:=stup.chans.convconst[indy]/
                                   E_dat.woff[indy];
     end else V_corr:=1.0;
      if choice<>supplyvoltage then
         E_dat.T_value[ind]:=million*
            (E_dat.L_value[ind]-E_dat.woff[ind])*V_corr
                    /stup.chans.convconst[ind];
    end;
   end else begin
     For i:=1 to stup.scan.norud do begin
      E_dat.S_press[i]:=(E_dat.S_value[i]-E_dat.woff[9])*
                         stup.chans.convconst[9];
     end;
     For i:=1 to stup.scan.noskeg do begin
      E_dat.S_press[i+stup.scan.norud]:=
         (E_dat.S_value[i+stup.scan.norud]
          -E_dat.woff[10])*stup.chans.convconst[10];
     end;
   end;
end;
{==========================================================}
PROCEDURE DynoCalibration(choice,times:integer;
```

```
                                  var stup:setexpup;
                                 var E_dat:ExpData);
          var indy,i,ind,start,number,j:integer;
                                      valt:real;
  begin
    start:=stup.chans.nobs[choice].loc;
    number:=stup.chans.nobs[choice].num;
    For i:=1 to number do begin
      ind:=start+i-1;
     E_dat.C_value[ind]:=0.0;
      For j:=1 to number do begin
       if choice=rudderdyno then valt:=stup.rudmat[i,j]
       else valt:=stup.qtmat[i,j];
       E_dat.C_value[ind]:=
          E_dat.C_value[ind]+valt*E_dat.T_value[ind+j-1];
     end;
    end;
end;

Procedure CorrectMinateData(choice,times:integer;
                                    var stup:setexpup;
                                   var E_dat:ExpData);
  begin
    CASE choice of
     RudderDyno:DynoCalibration(choice,times,stup,E_dat);
     QT:DynoCalibration(choice,times,stup,E_dat);
     Otherwise;
    end;
end;
{=====================================================}
PROCEDURE ChangeTimes(var times:integer);
  var blop:boolean;
  begin
    writeln('  Current number of times minate averages
               voltage reading = ',times:5);
    blop:=Question(' Do you wish to change this value ');
    if blop=true then times:=InputInt('Enter new value
                                              ',1,10);
end;
{=====================================================}
PROCEDURE PlotThings(typ,indicator:integer;
                          var stup:setexpup;
                         var e_dat:expdata);
  var message:string;
     realy:array[1..7] of real;
     lidd,liddsv,struy,a,b,i:integer;
  BEGIN
    struy:=stup.chans.nobs[typ].loc;
    For i:=1 to stup.chans.nobs[typ].num do begin;
     lidd:=struy+i-1;
     if lidd+9>15 then liddsv:=6 else liddsv:=lidd+9;
     case indicator of
     ShowVoltage:begin
                 realy[i]:=e_dat.L_value[lidd];
                 a:=10;
                 b:=6;
               end;
     ShowSupply:begin
```

```
                a:=10;
                b:=3;
                realy[i]:=e_dat.L_value[liddsv];
               end;
     ShowSD:begin
             realy[i]:=e_dat.L_SD[lidd];
             a:=10;
             b:=6;
           end;
     ShowValue:begin
               a:=10;
               b:=2;
               realy[i]:=e_dat.T_value[lidd];
               end;
     ShowCorrect:begin
                 realy[i]:=e_dat.C_value[lidd];
                 a:=10;
                 b:=2;
               end;
      Otherwise;
     end;
    end;
    Case typ of
     QT:begin
         PlotReal('  T: ',realy[1],a,b,140,232);
         PlotReal('  Q: ',realy[2],a,b,270,232);
       end;
     RudderDyno:begin
               PlotReal('  L: ',realy[1],a,b,420,190);
               PlotReal('  D: ',realy[2],a,b,420,160);
               PlotReal(' Mz: ',realy[3],a,b,420,130);
               PlotReal(' Mx: ',realy[4],a,b,420,100);
               PlotReal(' My: ',realy[5],a,b,420,70);
              end;
     RPM:PlotReal('RPM: ',realy[1],a,b,10,232);
     SupplyVoltage:begin
         PlotReal('  T: ',realy[1],10,3,140,232);
         PlotReal('  Q: ',realy[2],10,3,270,232);
         PlotReal('  L: ',realy[3],10,3,420,190);
         PlotReal('  D: ',realy[4],10,3,420,160);
         PlotReal(' Mz: ',realy[5],10,3,420,130);
         PlotReal(' Mx: ',realy[6],10,3,420,100);
         PlotReal(' My: ',realy[7],10,3,420,70);

                end;
     Otherwise;
    end;

end;
{====================================}
PROCEDURE PlotThis(indicator:integer;
                    var stup:setexpup;
                   var e_dat:expdata);
  var DataSize:int;
     Xmax,Xmin,Ymax,Ymin:real;
     Xdata,Ydata:plotData;
     i:integer;
  begin
```

```
    DataSize:=stup.scan.NoRud+stup.scan.NoSkeg;
    Xmax:=0.0;
    Ymax:=0.0;
    Xmin:=0.0;
    Ymin:=0.0;
    For i:=1 to DataSize do begin
     Xdata[i]:=stup.scan.chords[stup.headings.ints[8],i];
    If Xdata[i]>Xmax then Xmax:=Xdata[i] else
      if Xdata[i]<Xmin then Xmin:=Xdata[i];
    end;
    For i:=1 to DataSize do begin
     case indicator of
         ShowVoltage:Ydata[i]:=e_dat.S_Value[i];
         ShowSD:Ydata[i]:=e_dat.S_SD[i];
         ShowValue:Ydata[i]:=e_dat.S_press[i];
         ShowCorrect:Ydata[i]:=e_dat.S_press[i];
     end;
     If Ydata[i]>Ymax then Ymax:=Ydata[i] else
       if Ydata[i]<Ymin then Ymin:=Ydata[i];
    end;
    PlotPressure(Xdata,Ydata,Xmax,Xmin,0.0,
            Ymax,Ymin,0.0,5,50,395,220,stup.scan.NoRud,
            stup.scan.NoSkeg);
end;
{======================================}
PROCEDURE DrawDisplay(realy:real);
  begin

    DrawBox(130,5,229,244);
    DrawBox(260,135,229,244);
    DrawBox(390,265,229,244);
    DrawBox(630,410,220,70);
    DrawBox(395,5,220,50);
    DrawBox(630,410,229,244);
    PlotReal(' RUN No : ',realy,6,0,415,232);
    PlotText(410,55,'  RUDDER & PROPELLER TESTS');
    PlotText(415,205,'  RUDDER DYNAMOMETER');
    write(chr(27),'[23;3;25;79~B');
    { define text scrolling window}
    write(chr(27),'[30;43m');
    {change to black on white}
    page;
end;
{=========================================================}
PROCEDURE AcquireData(var fout,fprn:text;
                      typ,times,indicator:integer;
                      printeron,graphicson:boolean;
                      var stup:setExpUp;
                      var e_dat:ExpData);
  begin
    ReadDataFromMinate(typ,times,stup,E_dat);
    ConvertDataFromMinate(typ,times,stup,E_dat);
    CorrectMinateData(typ,times,stup,E_dat);
    WriteTestFile(fout,typ,times,stup,e_dat);
    if printeron=true then
              WriteTestFile(fprn,typ,times,stup,e_dat);
    if graphicson=false then begin
     if typ<>pressures then
```

```
     WriteChannel(stup.chans.nobs[typ].loc,
                  stup.chans.nobs[typ].num,
                  times,e_dat.L_value)
    else begin
     WritePressures((stup.scan.NoRud+stup.scan.NoSkeg),
                                     times,e_dat.S_value);
    end;
   end else begin
    case typ of
       supplyvoltage:PlotThings(supplyvoltage,
                                showsupply,stup,e_dat);
        pressures:PlotThis(indicator,stup,e_dat);
       QT: PlotThings(QT,indicator,stup,e_dat);
        RPM: PlotThings(RPM,indicator,stup,e_dat);
       RudderDyno:PlotThings(RudderDyno,indicator,
                                 stup,e_dat);
       Otherwise;
    end;
   end;
end;

{++++++++++++++++++++++++++++++++++++++++++++++++++}
PROCEDURE MonitorProp(graphicson:boolean;var
stup:setExpUp;var e_dat:ExpData);
  begin
    writeln('  Press key when desired RPM achieved ');
    repeat
      ReadDataFromMinate(RPM,1,stup,E_dat);
      ConvertDataFromMinate(RPM,1,stup,E_dat);
     ReadDataFromMinate(QT,1,stup,E_dat);
      ConvertDataFromMinate(QT,1,stup,E_dat);
     if graphicson=true then begin
         PlotThings(QT,showvalue,stup,e_dat);
         PlotThings(RPM,showvalue,stup,e_dat);
     end else begin
       WriteChannel(1,3,1,e_dat.L_value);
     end;
    until cstat=true;
end;
{                    End of Procedure}
BEGIN
END.
```

Appendix 7.3 INITRAP.PAS

```
PROGRAM INITRAP;

{  Sets up and allows various program components to be
individually tested prior to use. Allows creation of test
data files

Written by : S. Turnock
Date: 1/8/88

Modified version for Rudder & Propeller tests:  19/2/90
}
CONST
     pi=3.141595428;

     maxsize = 100;
       doubmaxsize = 200;
     {   MSDOS filenames }
     rawout = 'RAW';
     refout = 'REF';

   nints  = 10;
   mreals = 20;
   tnp =    30;
   config = 0;       TestConditions=1;
   supplyvoltage=2;  WindOff=3;
   pressures=4;      QT=5;
   RPM=6;            RudderDyno=7;
   EndTestRun=8;     Comment=9;
   {indicator}
   ShowVoltage=1;
   ShowSD=2;
   ShowValue=3;
   ShowCorrect=4;
   ShowSupply=5;

{=========================================================
==}
{    type declarations}
TYPE
      nthym = record
             loc,num:integer;
            end;
     channels = Array[1..15] of real;
     scanichan = array[1..60] of real;
     header = record
               title:string;
               nint,mreal:integer;
               ints:array[1..nints] of integer;
               reals:array[1..mreals] of real;
               descrip:array[1..tnp] of string;
             end;
      ChannelData = record
                   name:array[1..15] of string;
                   ConvConst:channels;
                   nobs:array[2..7] of nthym;
                 end;
```

```
     TestCond = record
                 valts:array[1..3] of real;
                 nvlts:array[1..3] of string;
                 lims:array[1..3,1..2] of real;
                end;
      SysConst = record
                 bools:array[1..3] of boolean;
                 nms:array[1..3] of string;
                end;
     arr5real = array[1..5,1..5] of real;
     arr2real = array[1..2,1..2] of real;
     ScaniData = record
                  NoRud,NoSkeg,NoSpan:integer;
                  span,chord:array[1..10] of real;
                  chords:array [1..10] of scanichan;
                end;
      SetExpUp = record
                headings:header;
                chans:ChannelData;
                Scnst:SysConst;
                RudMat:arr5real;
                QTmat:arr2real;
                Scan:ScaniData;
               end;
     ExpData  = record

L_value,L_sd,T_value,C_value,woff:channels;
                 TC:Testcond;
                 S_value,S_sd,S_press:scanichan;
                end;


    DWORD = 0..maxint;          {graphics output type}
     INT  = -32768..32767;      {Graphics output type}
     INT0 = 0..65535;           {Graphics output type}
     BYTE = 0..255;               {Graphics Output type}
     INT1 = -128..127;          {1401 : 8 bit ADC/DAC data}
     INT2 = -32768..32767;      {1401 : 16 bit ADC/DAC
data}
     WORD = 0..65535;           {1401 : PSTH family array
type}
       RECT   = RECORD
                xl,yl,xr,yr : INT;
             END;

     BAL =  Array[1..6] of real;
     Scan = Array[1..4] of real;


{=========================================================
==}
{    common declarations}
COMMON cin,cout:TEXT;    {1401 files. Must be common for
Proload}

{=========================================================
==}
{    variable declarations}
```

```
VAR
  reply,try:boolean;
  value,indicator,choice,choose,times:Integer;
  sds,values:channels;
  stup:SetExpUp;
  finch,printeron,graphicson:boolean;
  E_dat:expdata;
  fout,fprn:text;
  realy:real;
{    external procedure / function declarations}


     {Dispseg Externals   written by K.M.Hyde}

PROCEDURE AnyKeyToCont;                              EXTERNAL;
PROCEDURE PrintReal(Text:String;
                    X:Real;F,D:Integer);EXTERNAL;
FUNCTION InputInt(Text:String;
                  Min,Max:Integer): Integer;EXTERNAL;
FUNCTION InputReal(Text:String;Min,Max:Real
                   ;f,d:integer) :Real;External;
FUNCTION Question(Text:String) : Boolean;EXTERNAL;

     {  MinSeg Externals }

PROCEDURE StringToReal(Value : String;
                  VAR Result :Real);EXTERNAL;

PROCEDURE WriteSolar(message: string);EXTERNAL;

PROCEDURE ReadSolar(VAR answer : string);EXTERNAL;

PROCEDURE BALANCEd(strain,calibrate : BAL; SuppVolt :
Real;
                               VAR balcoeff :
BAL);EXTERNAL;
PROCEDURE InitMinate(typ:boolean);EXTERNAL;
PROCEDURE Minate( VAR  balance : BAL;scani : SCAN;
                         suppvolt : real);EXTERNAL;
PROCEDURE Calibrated(VAR MeanBal,SDBal : BAL;
                      MeanScan,SDScan : SCAN);EXTERNAL;
PROCEDURE DirContMin;EXTERNAL;
PROCEDURE MinatePrint(Balance : Bal; Scani :
SCAN);EXTERNAL;

     {graphseg externals }
PROCEDURE StartGraph;EXTERNAL;
PROCEDURE CloseUp;EXTERNAL;

     {  RAPSEG Externals }

PROCEDURE ChangeSetUp(var SetUpData:SetExpUp);External;
PROCEDURE ReadSetUp(var SetUpdata:SetExpUp);EXTERNAL;
PROCEDURE WriteSetUp(SetUpdata:SetExpUp);EXTERNAL;
PROCEDURE ReadChannel(start,number,times:integer;
                  VAR values,sds:channels);EXTERNAL;
PROCEDURE WriteChannel(start,number,times:integer;
```

```
                        values:channels);EXTERNAL;
PROCEDURE WriteTestFile(var
fname:text;choice,times:integer;

stup:SetExpUp;expup:ExpData);EXTERNAL;
PROCEDURE ReadDataFromMinate(choice,times:integer;
            stup:setexpup; var E_dat:ExpData);EXTERNAL;
PROCEDURE ConvertDataFromMinate(choice,times:integer;
            stup:setexpup;var E_dat:ExpData);EXTERNAL;
PROCEDURE EnterTestCondition(var
T_cond:TestCond);EXTERNAL;
PROCEDURE OpenTestFile(var TestNo:integer;
                       var fname:text);EXTERNAL;
PROCEDURE OpenPrinter(var fname:text;
              var
stup:SetExpUp;printeron:boolean);EXTERNAL;
Procedure CloseTestFile(var fname:text);EXTERNAL;
PROCEDURE ChangeTimes(var times:integer);EXTERNAL;
PROCEDURE AcquireData(var fout,fprn:text;
               typ,times,indicator:integer;
               printeron,graphicson:boolean;
               var
stup:setExpUp;vare_dat:ExpData);EXTERNAL;
PROCEDURE DrawDisplay(realy:real);EXTERNAL;
PROCEDURE MonitorProp(graphicson:boolean;
                        var stup:setExpUp;
                       var e_dat:ExpData);EXTERNAL;
procedure WriteCommentFile(var f1,f2:text;
                          printeron:boolean);EXTERNAL;

{     SCAN RAP procedures }
PROCEDURE WriteValue(value:integer);EXTERNAL;
PROCEDURE ReadValue(var value:integer);EXTERNAL;

{*******************************************************
**

               MAIN PROGRAM

*******************************************************
**}
BEGIN
     try := false;
     repeat
     page;
     writeln;
     writeln('      INITIALISATION      MENU');
     writeln;
     writeln;
     writeln('   Enter Number for required option');
     writeln;
     writeln;
     writeln('            RAP   Test Data Input
(1)');
     writeln;
     writeln('            Minate Operation
(2)');
     writeln;
```

```
     writeln('              Experiment testing
(3)');
     writeln;
     writeln('              ScaniValve Testing
(4)');
     writeln;
     writeln('              To Quit Program
(5)');
     writeln;
     Choice := InputInt('   Enter Choice ?',1,5);
     CASE Choice of
          1: begin
               ReadSetUp(StUp);
               ChangeSetUp(StUp);
               WriteSetUp(StUp);
             end;
          2: begin
               reply := Question(' Is Minate powered up
');

               If reply = true then begin
                         ReadSetUp(StUp);
                         InitMinate(stup.scnst.bools[2]);
                         ReadChannel(1,15,1,values,sds);
                         WriteChannel(1,15,1,values);
                         AnyKeyToCont;
                         end;
            end;
          3: begin
             times:=1;
             indicator:=1;
             graphicson:=TRUE;
             ReadSetUp(StUp);
             ChangeSetUp(StUp);
             EnterTestCondition(E_dat.TC);
             OpenTestFile(stup.headings.ints[1],fout);
             OpenPrinter(fprn,stup,printeron);
             WriteTestFile(fout,config,1,stup,e_dat);
             WriteTestFile(fout,testconditions,
                                        1,stup,e_dat);
             if printeron=true then begin
               WriteTestFile(fprn,config,1,stup,e_dat);
               WriteTestFile(fprn,testconditions,
                                          1,stup,e_dat);
             end;
             InitMinate(stup.scnst.bools[2]);
             ChangeTimes(times);

ReadDataFromMinate(windoff,times,stup,E_dat);

WriteTestFile(fout,windoff,times,stup,e_dat);
             if printeron=true then

WriteTestFile(fprn,windoff,times,stup,e_dat);
             ReadDataFromMinate(supplyvoltage,times,
                                   stup,E_dat);
             WriteTestFile(fout,supplyvoltage,times,
                                   stup,e_dat);
             if printeron=true then
```

```
            WriteTestFile(fprn,supplyvoltage,times,
                             stup,e_dat);
            if graphicson=true then begin
              startgraph;
              realy:=1.0*stup.headings.ints[1];
              DrawDisplay(realy);
            end;
            MonitorProp(graphicson,stup,e_dat);
              finch:=False;
            { tunnelwindup; }
            repeat
              if graphicson=FALSE then begin
               page;
               writeln;
               writeln('   AVAILABLE OPTIONS ');
               writeln;
               writeln('   Finish Test Run     (0) ');
               writeln('   Supply Voltage      (1) ');
               writeln('   Rudder Dynamometer  (2) ');
               writeln('   Torque/Thrust Dyno  (3) ');
               writeln('   RPM                 (4) ');
               writeln('   Scani Pressures     (5) ');
               writeln('   Change Display      (6) ');;
               writeln('   Printer On/Off      (7) ');
               writeln('   Change No of times  (8) ');
               writeln('   Graphics On/Off     (9) ');
               writeln('   comment             (10)');
               writeln;
              end else begin
               realy:=1.0*stup.headings.ints[1];
               DrawDisplay(realy);
               writeln('  Finish(0)   Supply(1) Rudder(2)
                       Q-T(3)       RPM(4)');
               writeln('   Scani(5) Display(6)  Print(7)
                      *Times(8) Graphics(9)
Comment(10)');
              end;
              choose:=
                 InputInt('  Enter desired option
',0,10);
              case choose of
               0:begin
                   if graphicson=true then closeup;
                   WriteTestFile(fout,EndTestRun,1,
                                    stup,e_dat);
                   if printeron=true then
                     WriteTestFile(fprn,EndTestRun,1,
                                    stup,e_dat);
                     finch:=Question('   Have you
                                completed tunnel
run');
                     WriteSetUp(StUp);
                     if finch=false then begin
                        ReadSetUp(StUp);
                        ChangeSetUp(StUp);
                        EnterTestCondition(E_dat.TC);

OpenTestFile(stup.headings.ints[1],
```

```
fout);

OpenPrinter(fprn,stup,printeron);
                         WriteTestFile(fout,config,1,

stup,e_dat);

WriteTestFile(fout,testconditions,

1,stup,e_dat);
                        if printeron=true then begin
                          WriteTestFile(fprn,config,1,
                                     stup,e_dat);

WriteTestFile(fprn,testconditions,
                                   1,stup,e_dat);
                        end;
                        ChangeTimes(times);
                        WriteTestFile(fout,windoff,times,
                                      stup,e_dat);
                        if printeron=true then

WriteTestFile(fprn,windoff,times,
                                      stup,e_dat);
                        ReadDataFromMinate(supplyvoltage,
                                     times,stup,E_dat);
                        WriteTestFile(fout,supplyvoltage,
                                     times,stup,e_dat);
                        if printeron=true then

WriteTestFile(fprn,supplyvoltage,

times,stup,e_dat);
                        if graphicson=true then begin
                           startgraph;

realy:=1.0*stup.headings.ints[1];
                           DrawDisplay(realy);
                        end;

MonitorProp(graphicson,stup,e_dat);
                    end else begin
                        if graphicson=true then begin
                           startgraph;

realy:=1.0*stup.headings.ints[1];
                           DrawDisplay(realy);
                        end;

MonitorProp(graphicson,stup,e_dat);
                        if graphicson=true then closeup;
                    end;
                  end;
                1:AcquireData(fout,fprn,supplyvoltage,
                              times,indicator,printeron,
                              graphicson,stup,e_dat);
```

```
                2:AcquireData(fout,fprn,RudderDyno,
                              times,indicator,
                              printeron,graphicson,
                              stup,e_dat);
                3:AcquireData(fout,fprn,QT,times,
                              indicator,printeron,
                              graphicson,stup,e_dat);

4:AcquireData(fout,fprn,RPM,times,indicator,
                              printeron,graphicson,
                              stup,e_dat);
                5:AcquireData(fout,fprn,pressures,times,
                              indicator,printeron,
                              graphicson,stup,e_dat);
                6:begin
                    write(' Type of display is ');
                    write(indicator:2);
                    writeln(' at the moment');
                    writeln(' Choice of (1) Voltage (2)
S.D.
                             (3) Value (4) Corrected
Value');
                    indicator:=
                        InputInt('  Enter new choice
',1,4);
                  end;
                7:if printeron=true then
printeron:=FALSE;
                8:changetimes(times);
                9:begin
                    if graphicson=true then begin
                      graphicson:=false;
                      closeup;
                    end else begin
                      graphicson:=True;
                      startgraph;
                    end;
                  end;
                10:WriteCommentFile(fout,fprn,printeron);
                  end;
               until finch=true;
             { tunnelwinddown; }
           end;
         4:begin
           repeat
            finch:=Question(' Write Value to Scan');
            if finch=true then begin
              value:=InputInt(' Enter output value
',0,255);
              WriteValue(value);
            end else begin
              readvalue(value);
              writeln('   read value of ',
                      value:8,' from port');
            end;
            finch:=Question('  Finished testing');
           until finch=true;
          end;
```

```
          5: begin
               try := true;
             end;
      end;
    until try = true;
END.
```

Appendix 8 Propeller Shaft Stess Calculation

Possible Propeller Shaft Materials:

	UTS MN/m^2	Yield Stress MN/m^2
Stainless Steel T304	587	242
EN16T Steel	850-1000	800

Propeller Performance Loading:

Max. Torque Q = 110 Nm   Max. Thrust T = 750 N

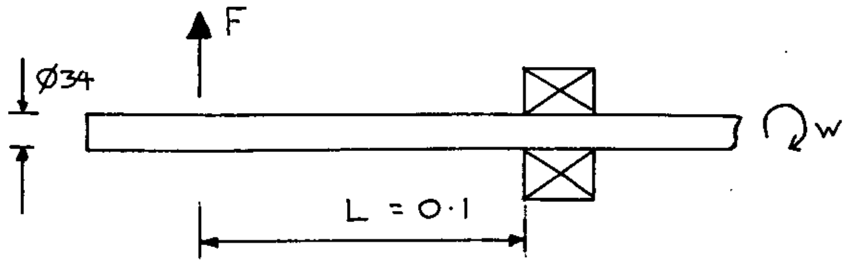
$$\text{Shear Stress} = \tau = \frac{Q \cdot R}{J} = \frac{2 \cdot Q}{Pi \cdot R^3} = 16.2 \ MN/m^2$$

$$\text{Tensile/Compressive Stress} = \frac{T}{Pi \ R^2} = 0.83 \ MN/m^2$$

Both Materials O.K.

Worst Case Loading:

One propeller blade fails. Loading equivalent to the case of a single blade attached to the hub.



m = mass of blade = 2kg r = centre of mass radius = 0.28m
w = rad/sec = (3,000 * 2 * Pi/60)

$$F = \frac{m \ r \ w^2}{2} = 27600N$$

$$\text{Bending Stress} \ \sigma = \frac{F \cdot L \cdot R}{2 \cdot I} = 358 \ MN/m^2$$

$$\text{ShearStress} \ \tau = \frac{F}{Pi \cdot R^2} = 30.4 \ MN/m^2$$

$$\text{Principal Stress} \ \sigma = 374 \ MN/m^2$$

> Yield Stress of Stainless T304 so use EN16T

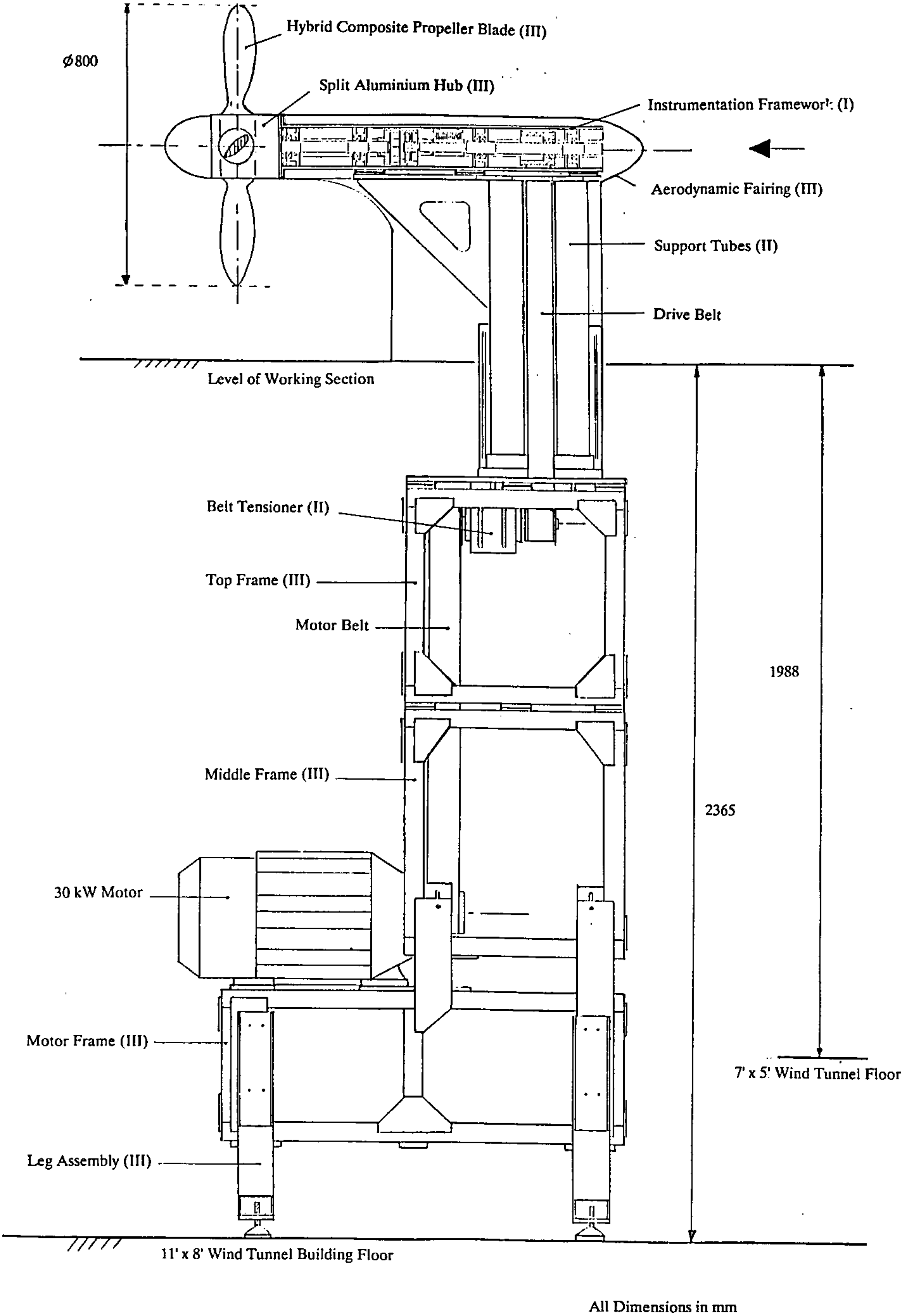
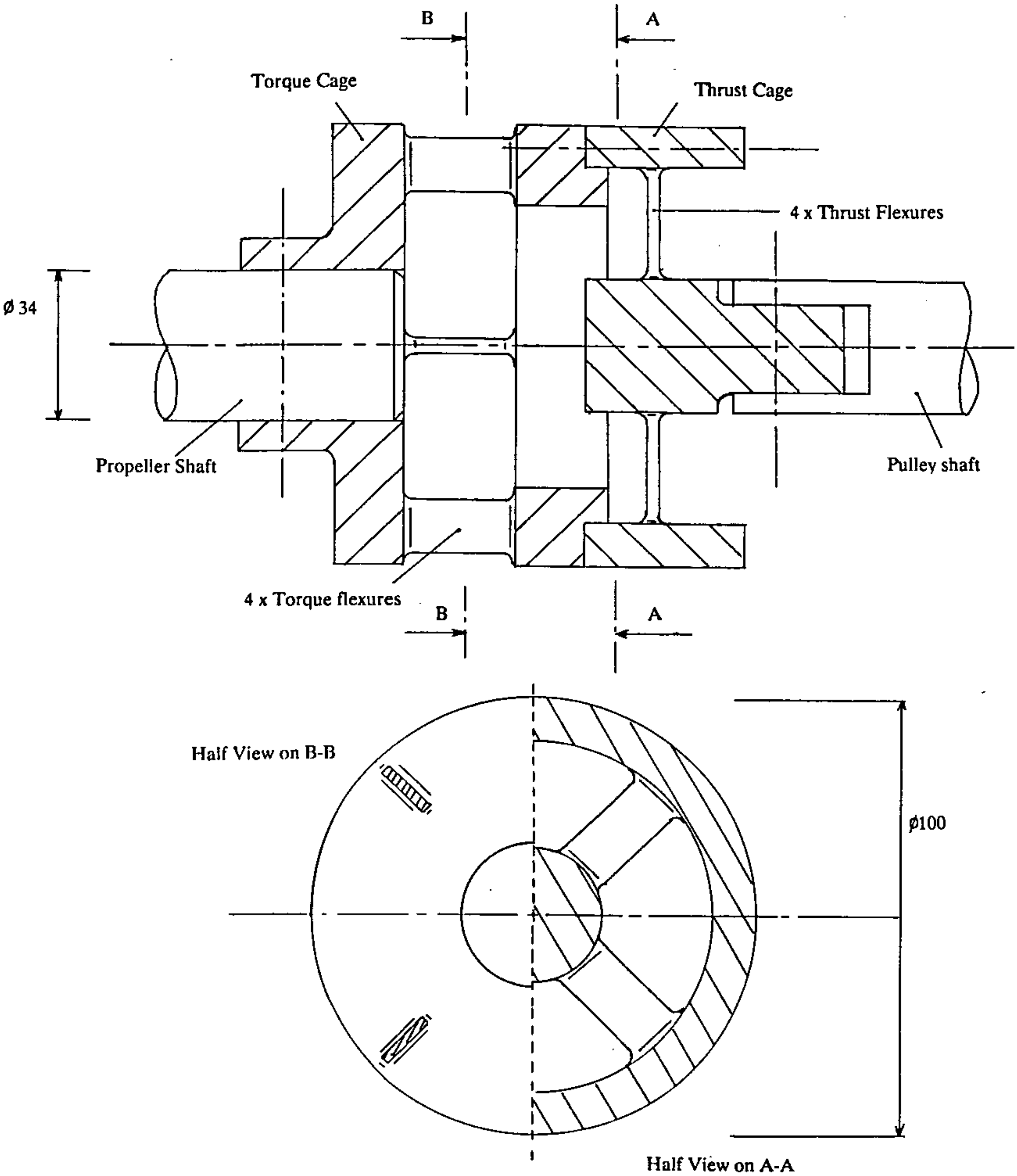


Figure 1 Side View of Propeller Rig



All Dimensions in mm

Figure 2 Torque-Thrust Dynamometer

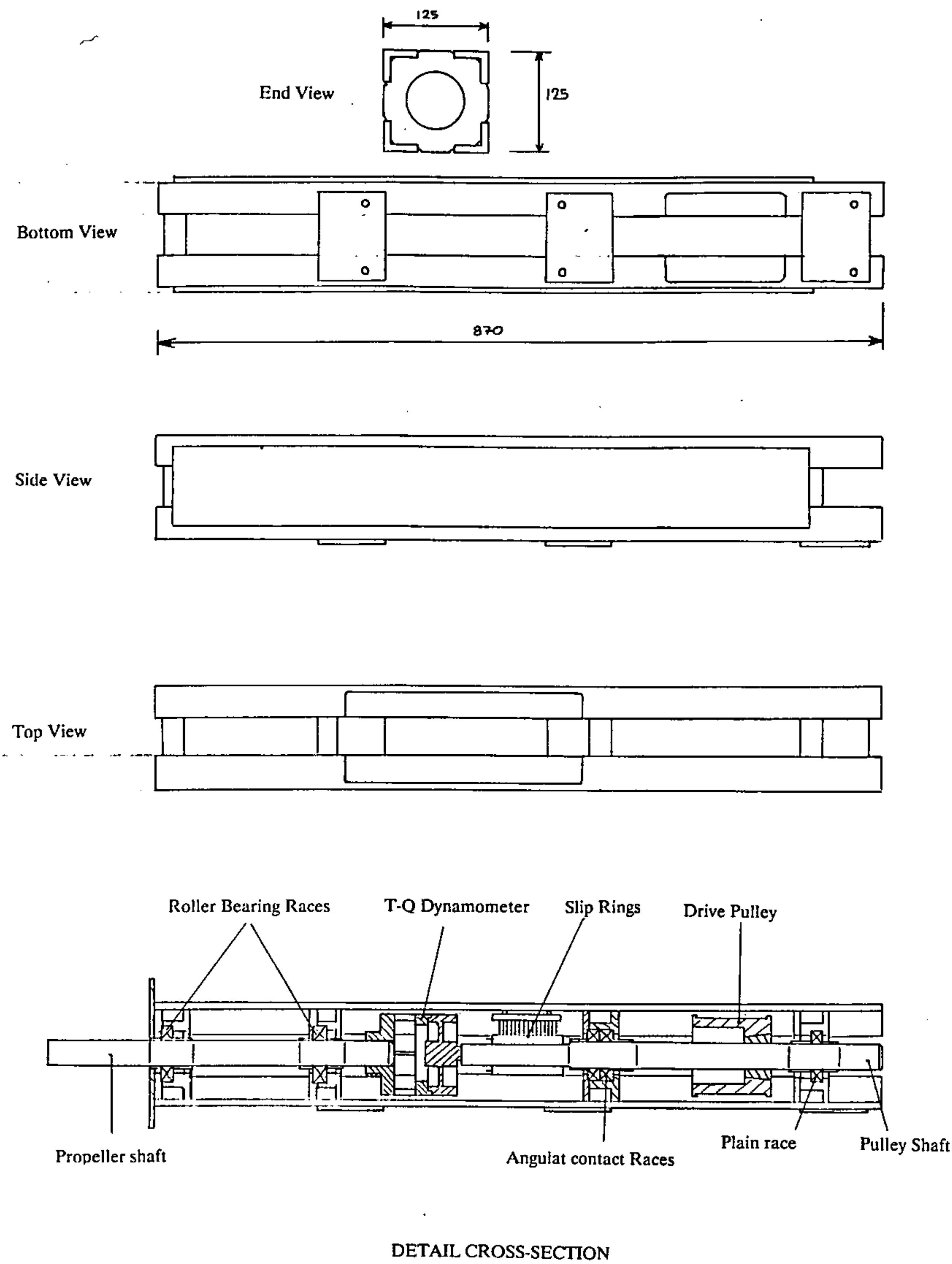


Figure 3 Propeller rig instrumentation Framework

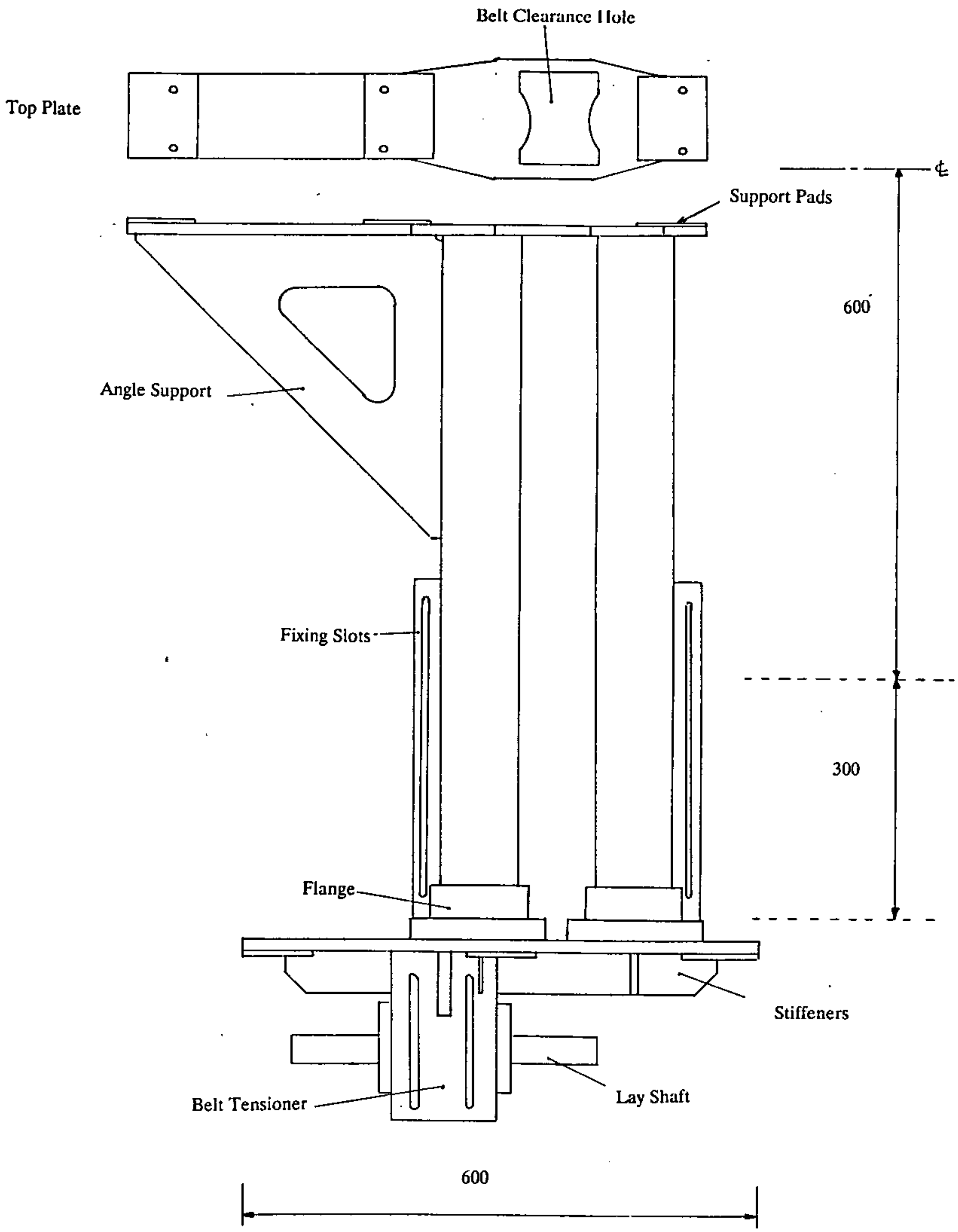


Figure 4 Above Wind Tunnel Floor Support Structure

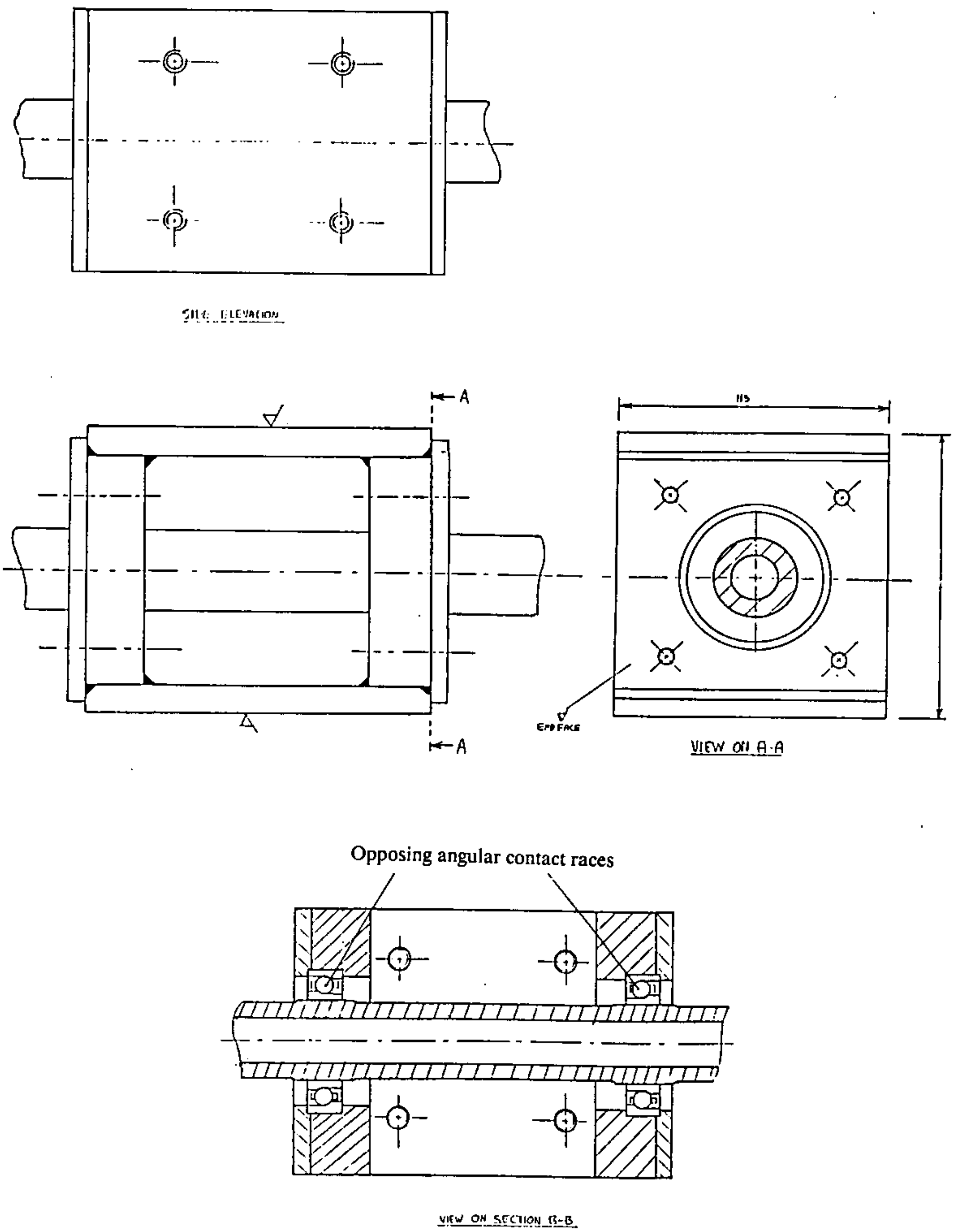


Figure 5 Lay shaft assembly arrangement

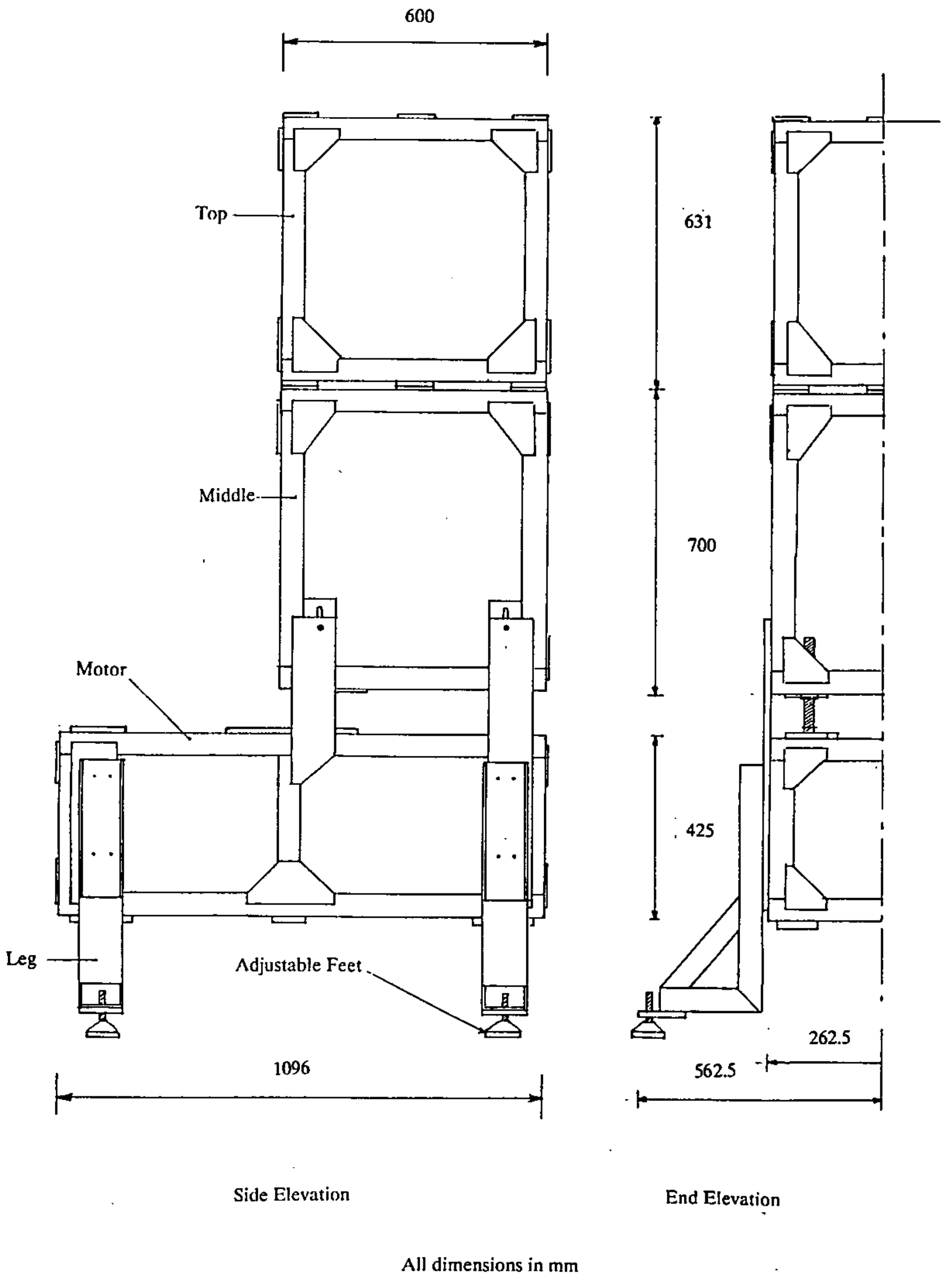


Figure 6 Below Wind Tunnel Floor Framework for use in 11' x 8' Wind Tunnel

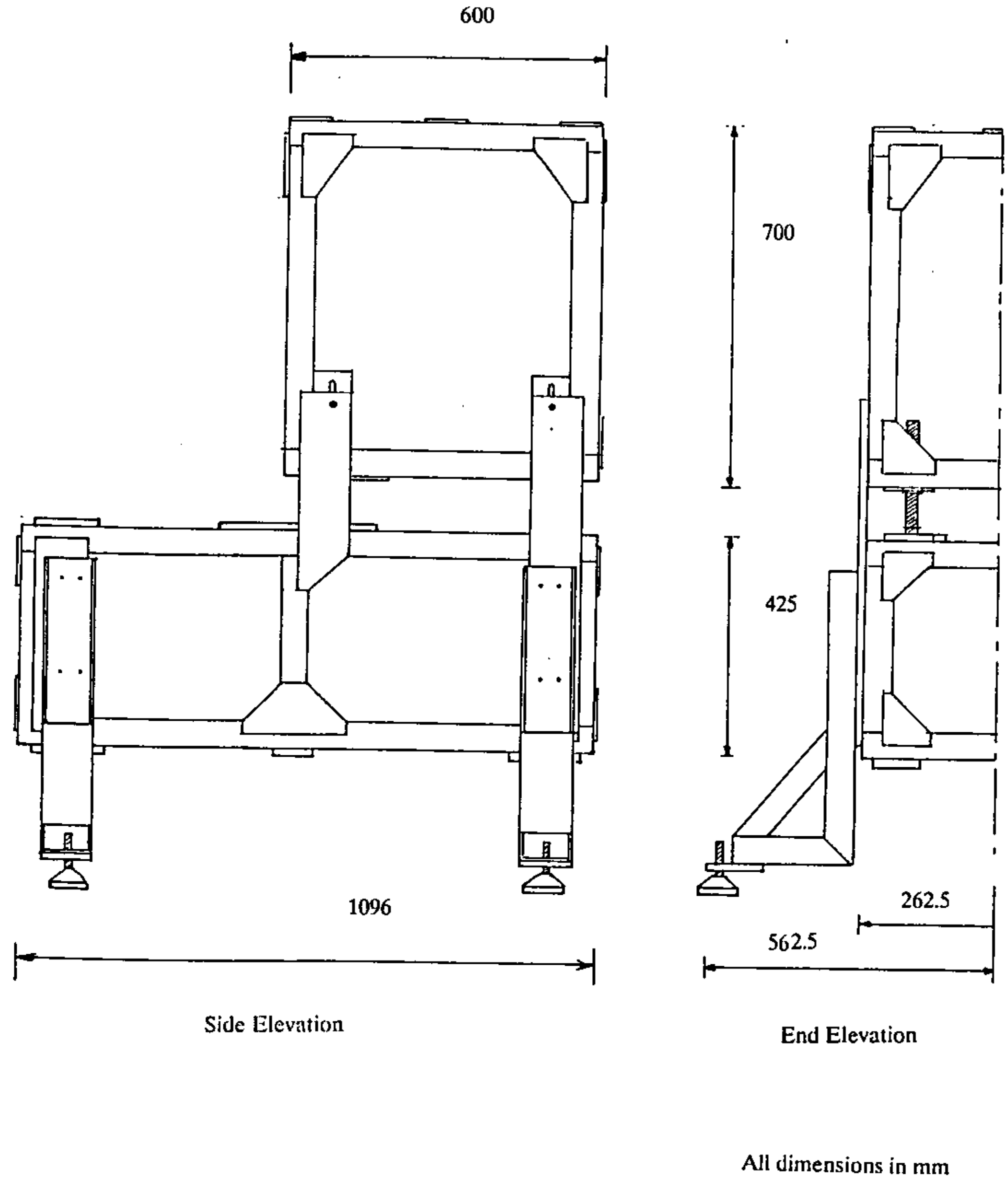
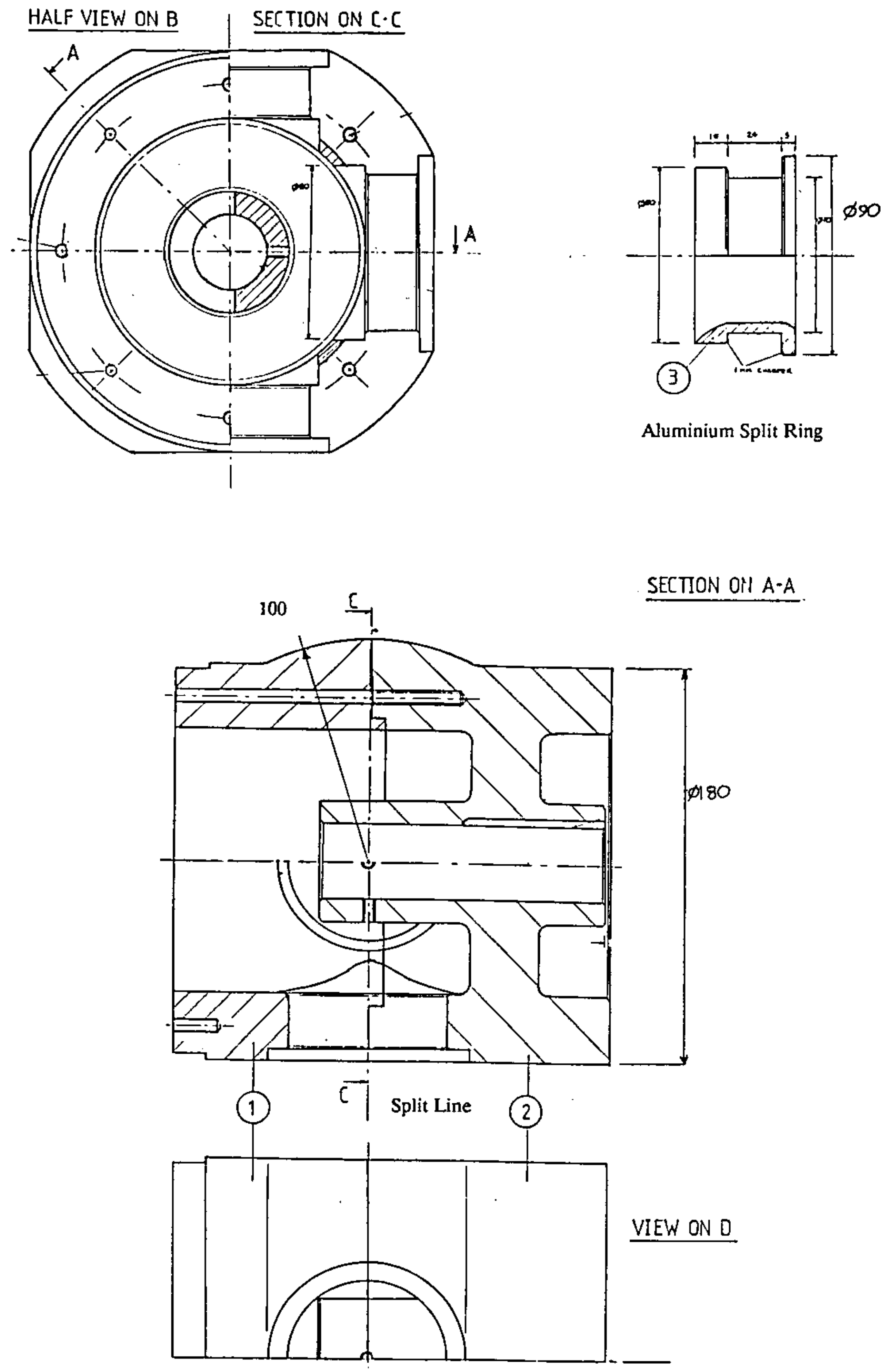
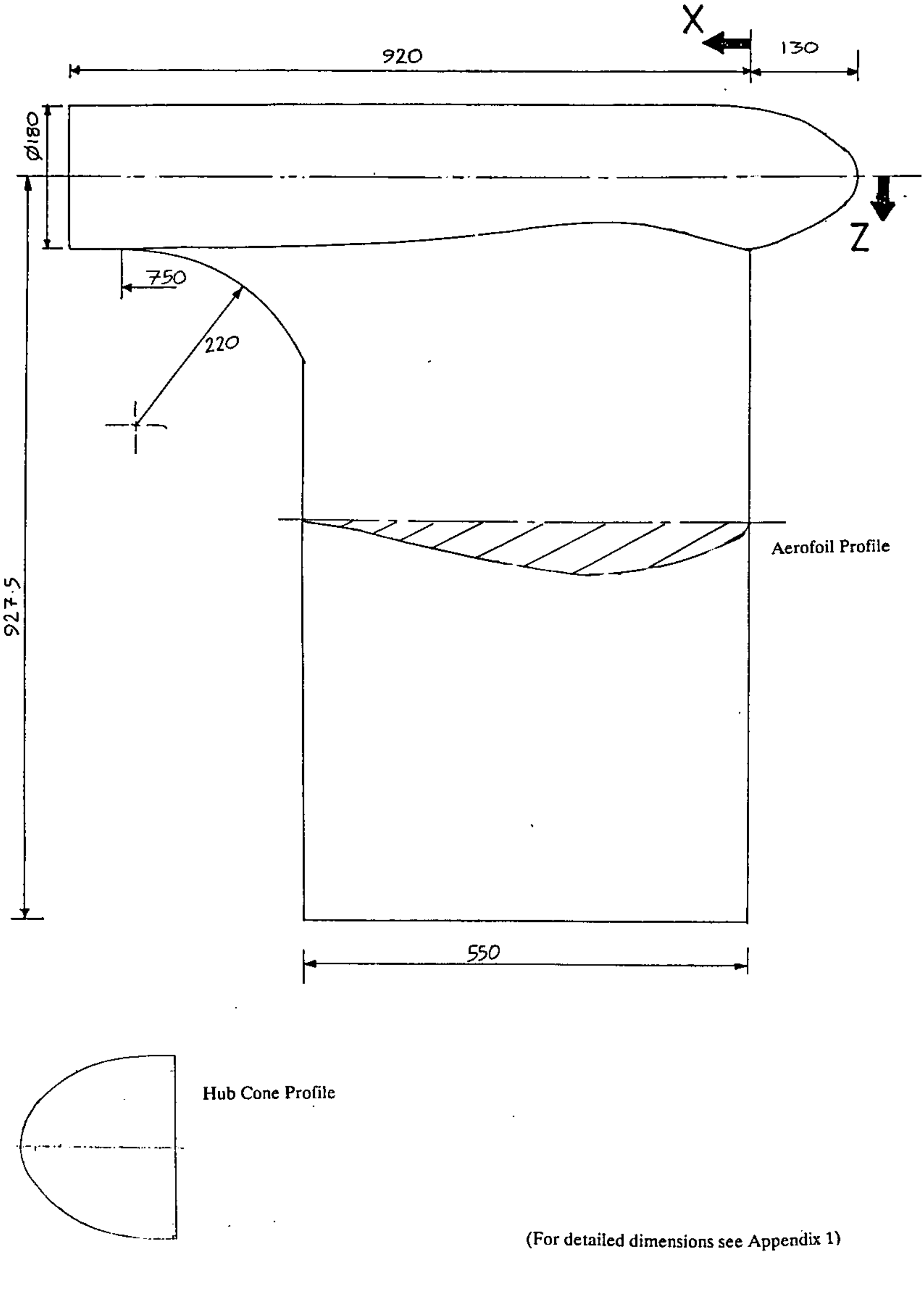


Figure 7 Below Wind Tunnel Floor Framework for use in 7' x 5' Wind Tunnel



All dimensions in mm

Figure 8 Propeller Hub and Split Ring



(For detailed dimensions see Appendix 1)

Figure 9 Fairing Around Above-Wind-Tunnel-Floor and Instrumentation Framework

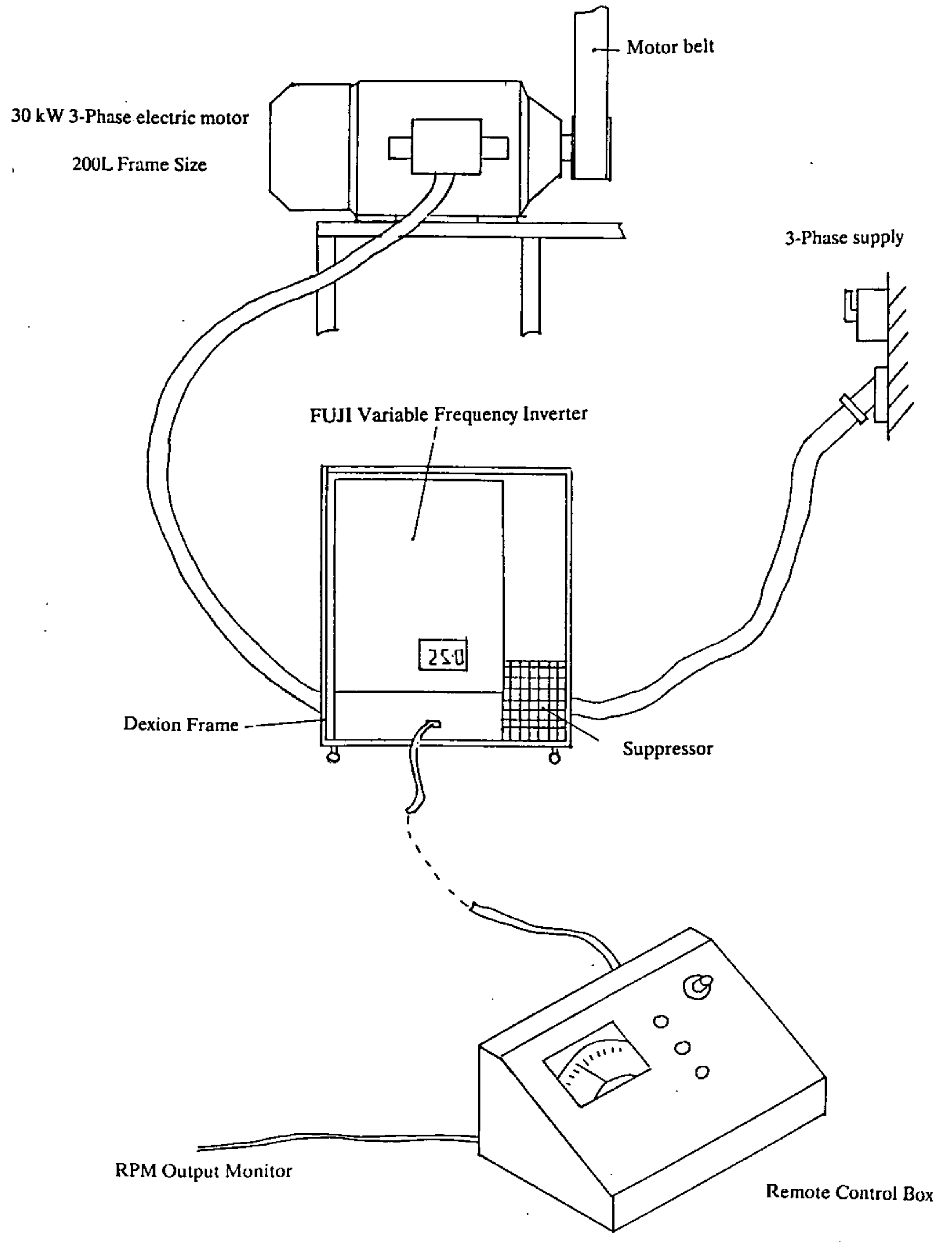


Figure 10 Electric Motor Control System

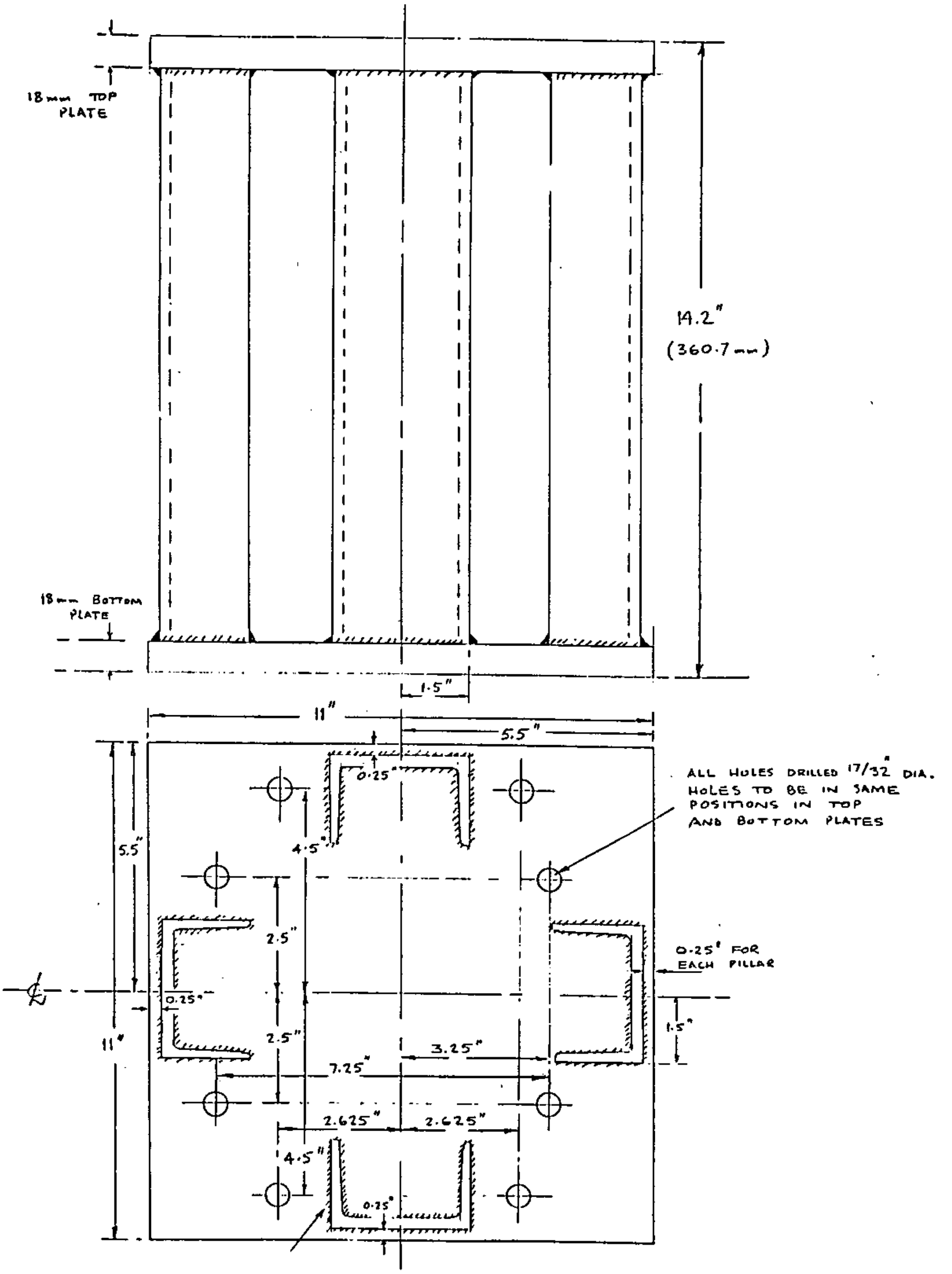


Figure 11 Rudder Extension Piece

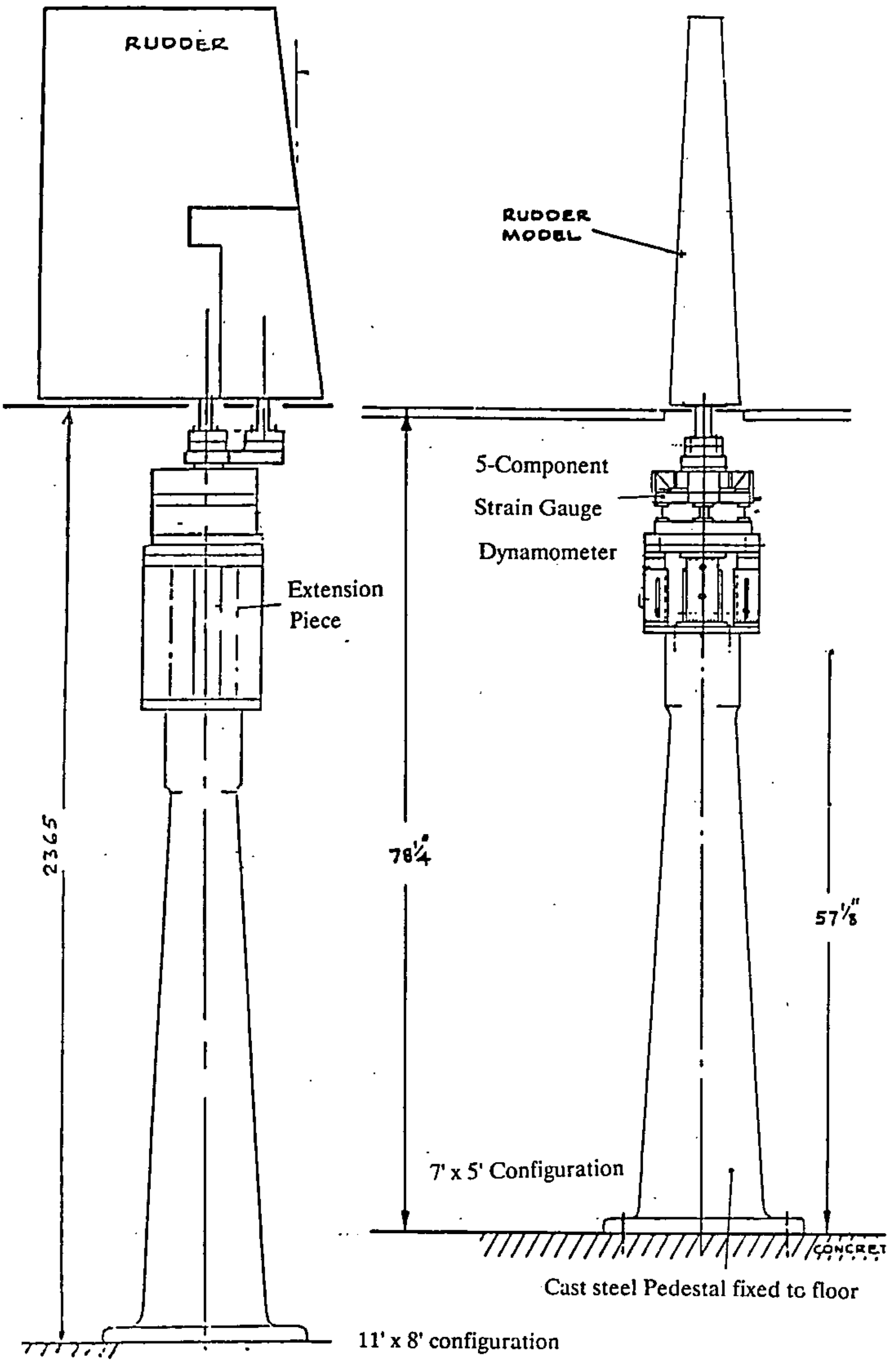


Figure 12 Rudder Rig Arrangement for both Wind Tunnels

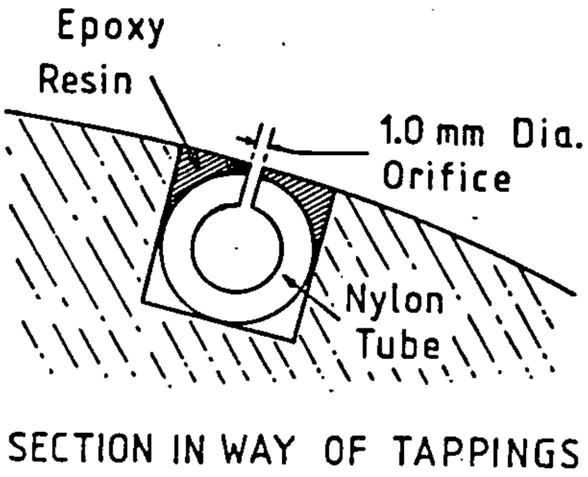


Figure 13 Pressure Tube Arrangement

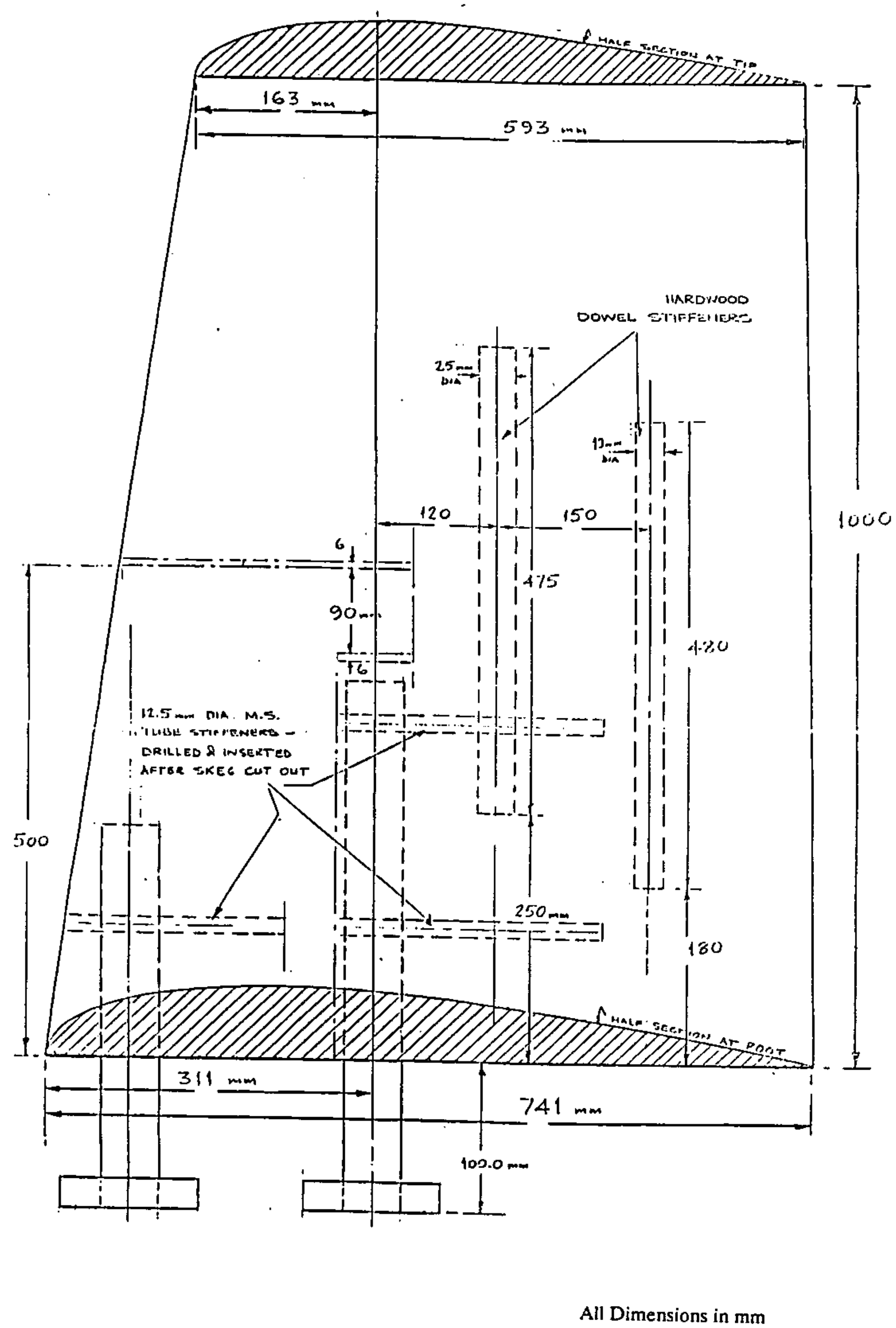


Figure 14 Semi-Balanced Skeg Rudder - RAP1

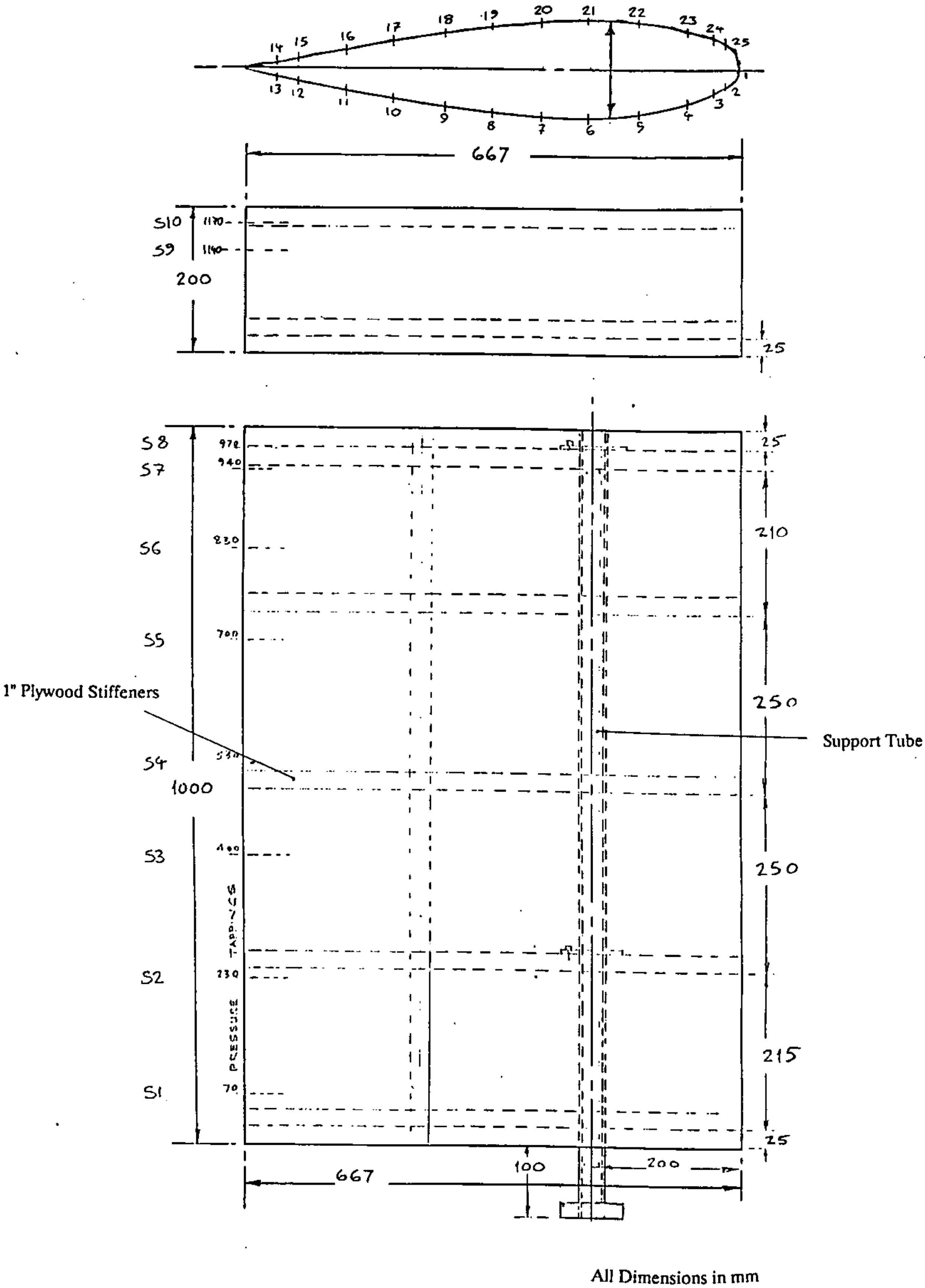


Figure 15 All-Movable Rudders - RAP2 and RAP3

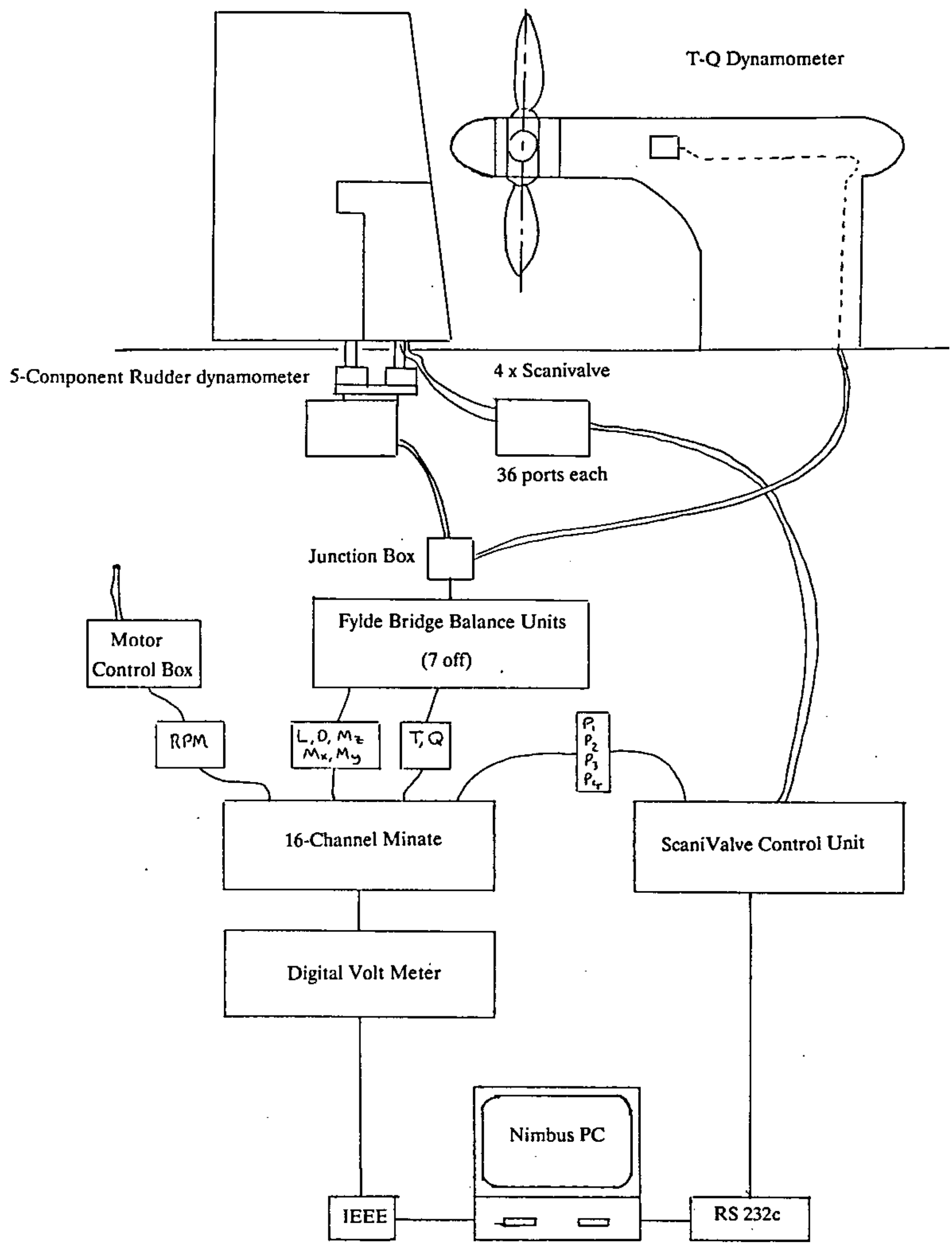


Figure 16 Schematic of Data Acquisition System

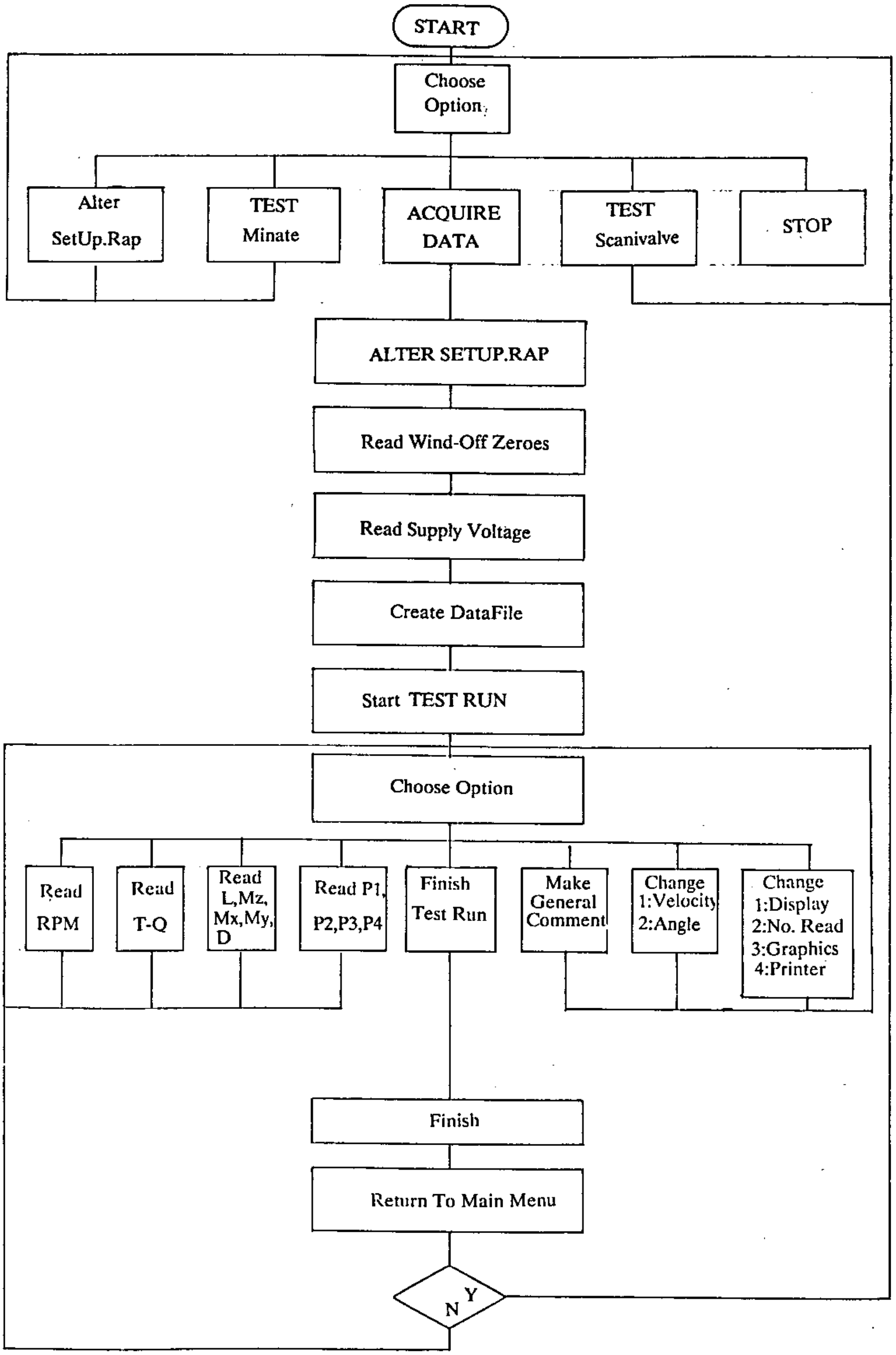
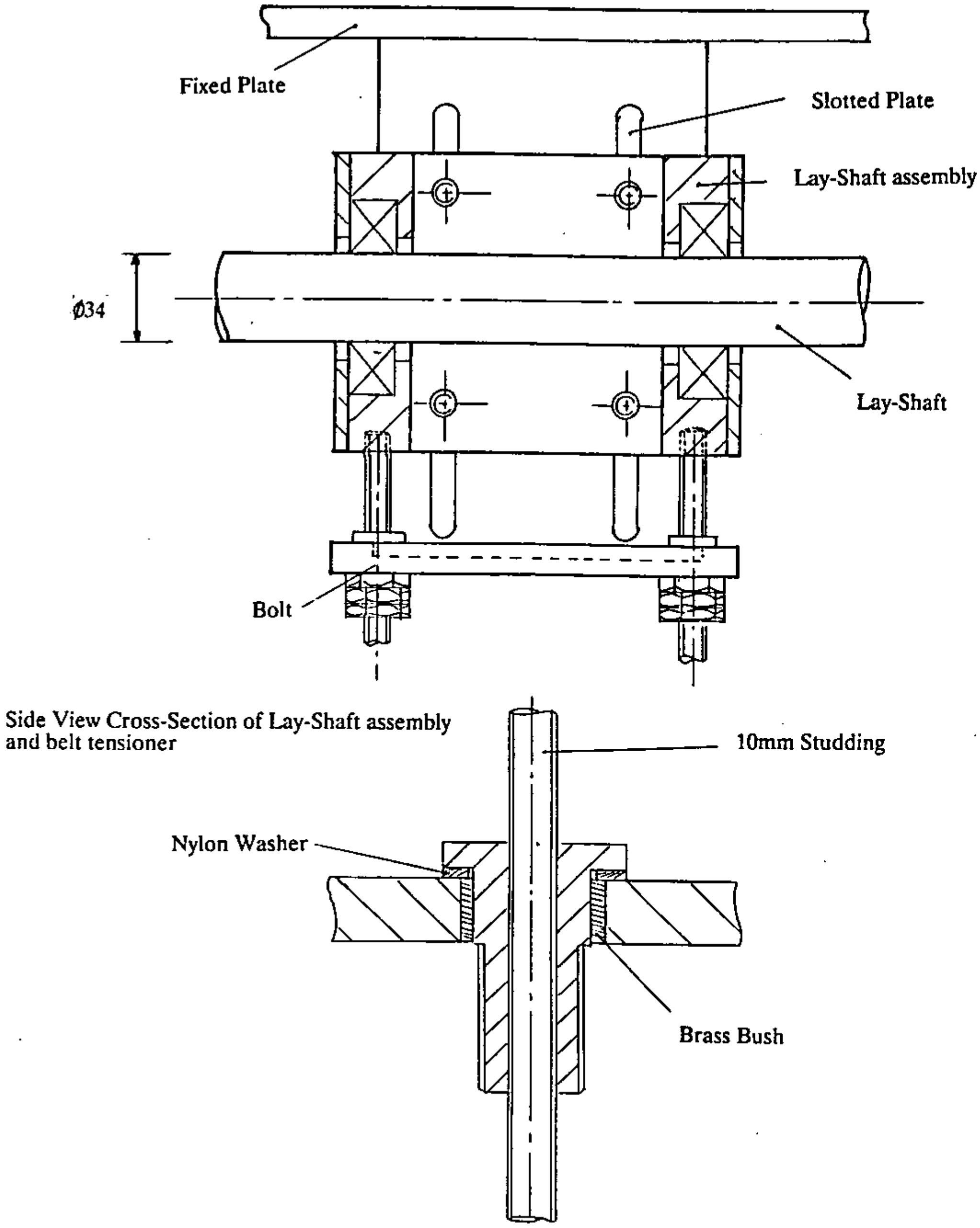


Figure 17 Flow Chart of Data Acquisition Software



CROSS-SECTION DETAIL OF BOLT

Figure 18 Belt Tensioner

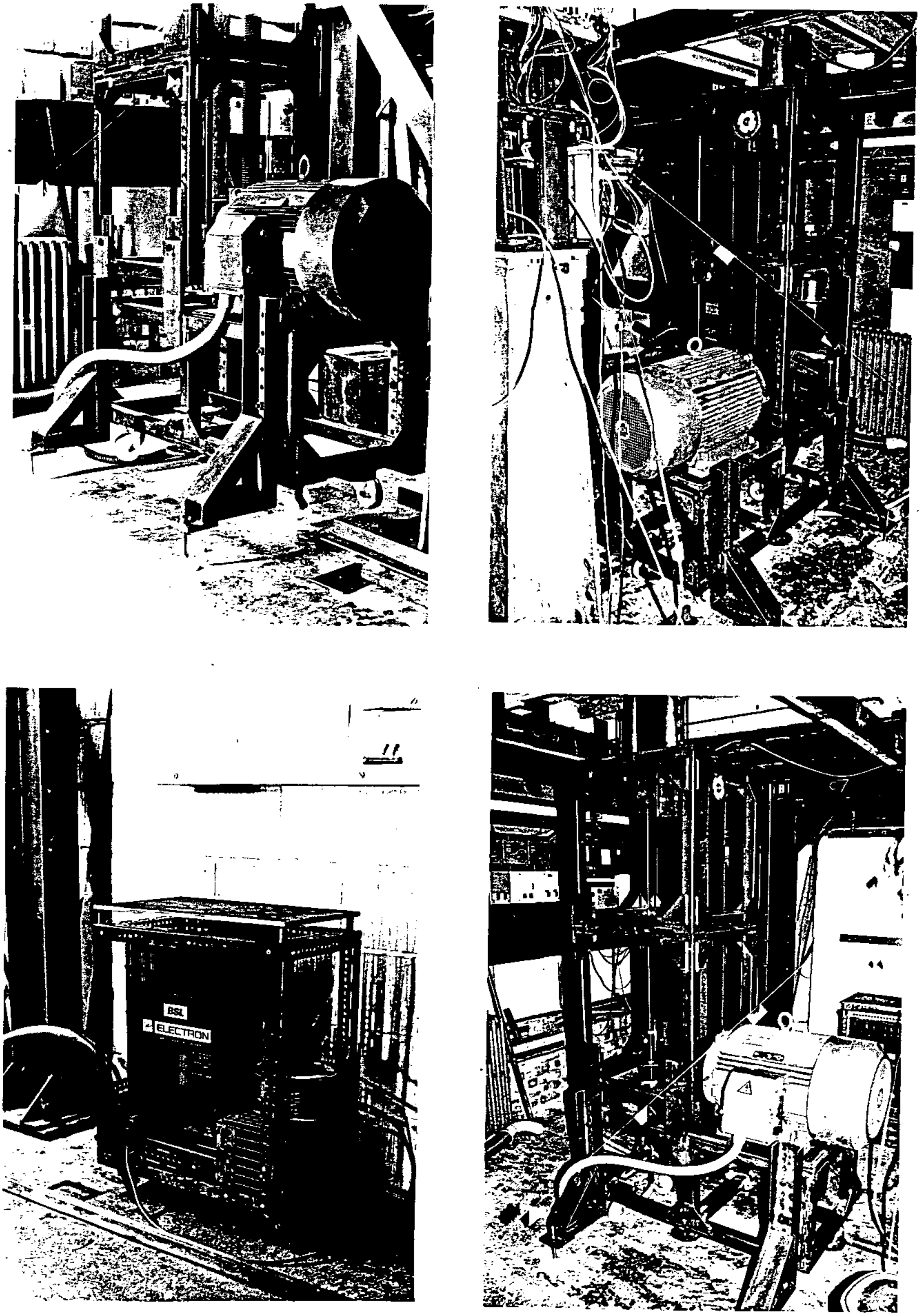
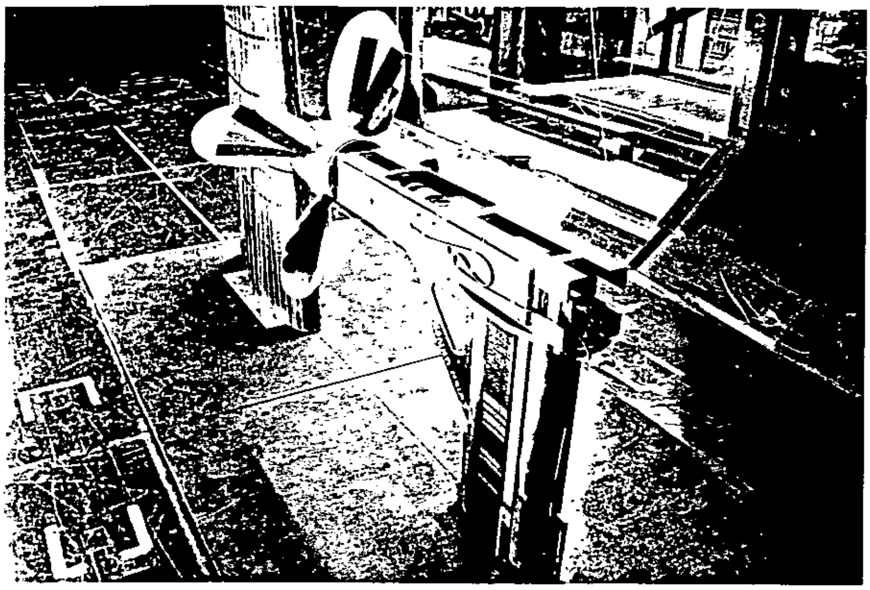


Figure 19 Photographs of Propeller Rig Below Wind Tunnel Floor



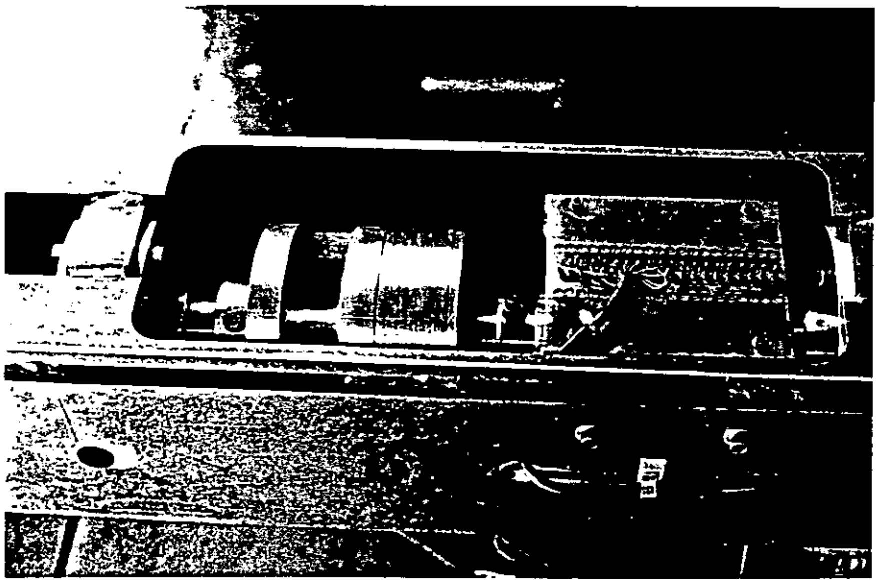


Figure 20 Photographs of Propeller Rig Instrumentation Framework

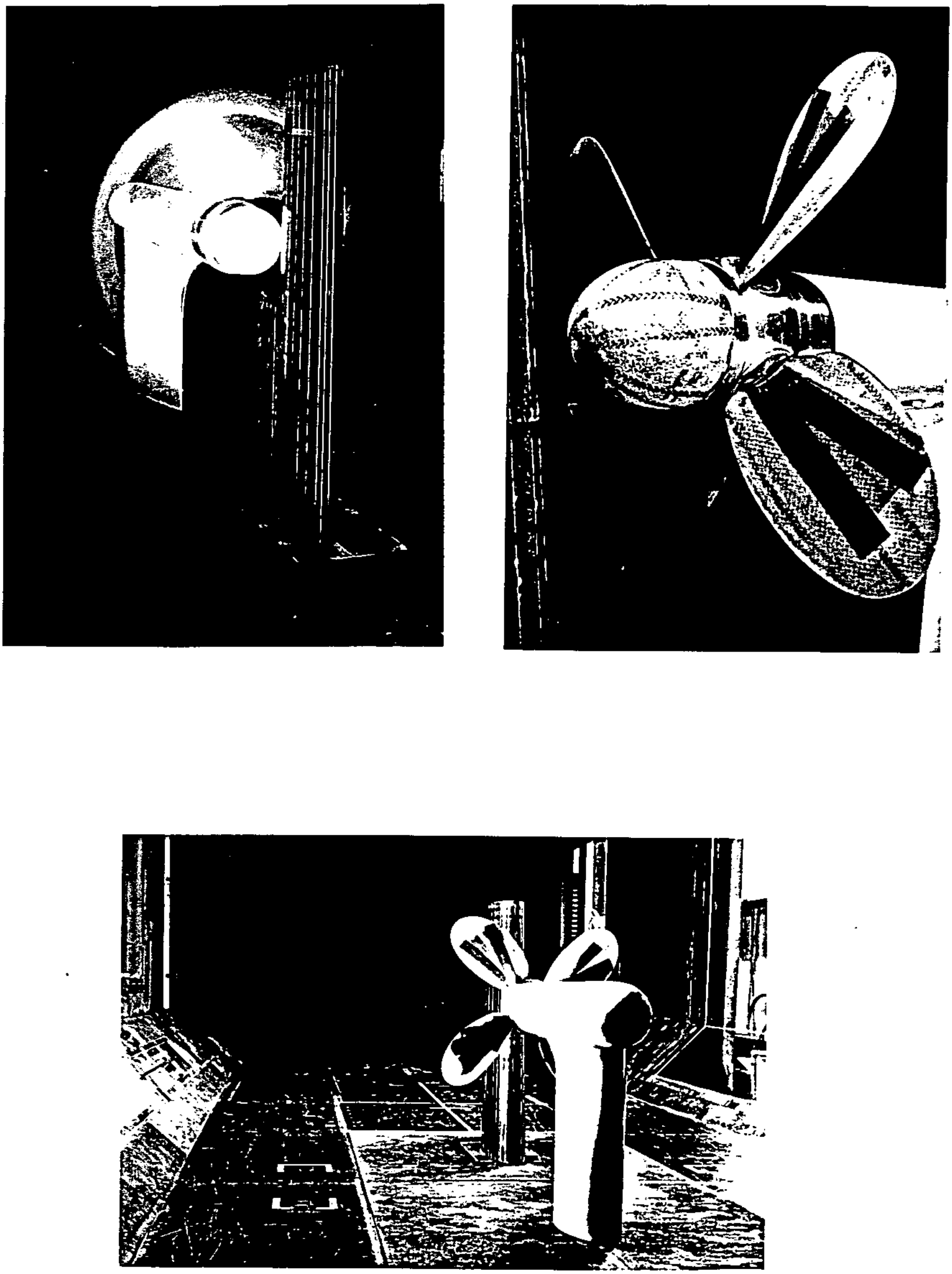


Figure 21 Photographs of Faired Propeller Rig

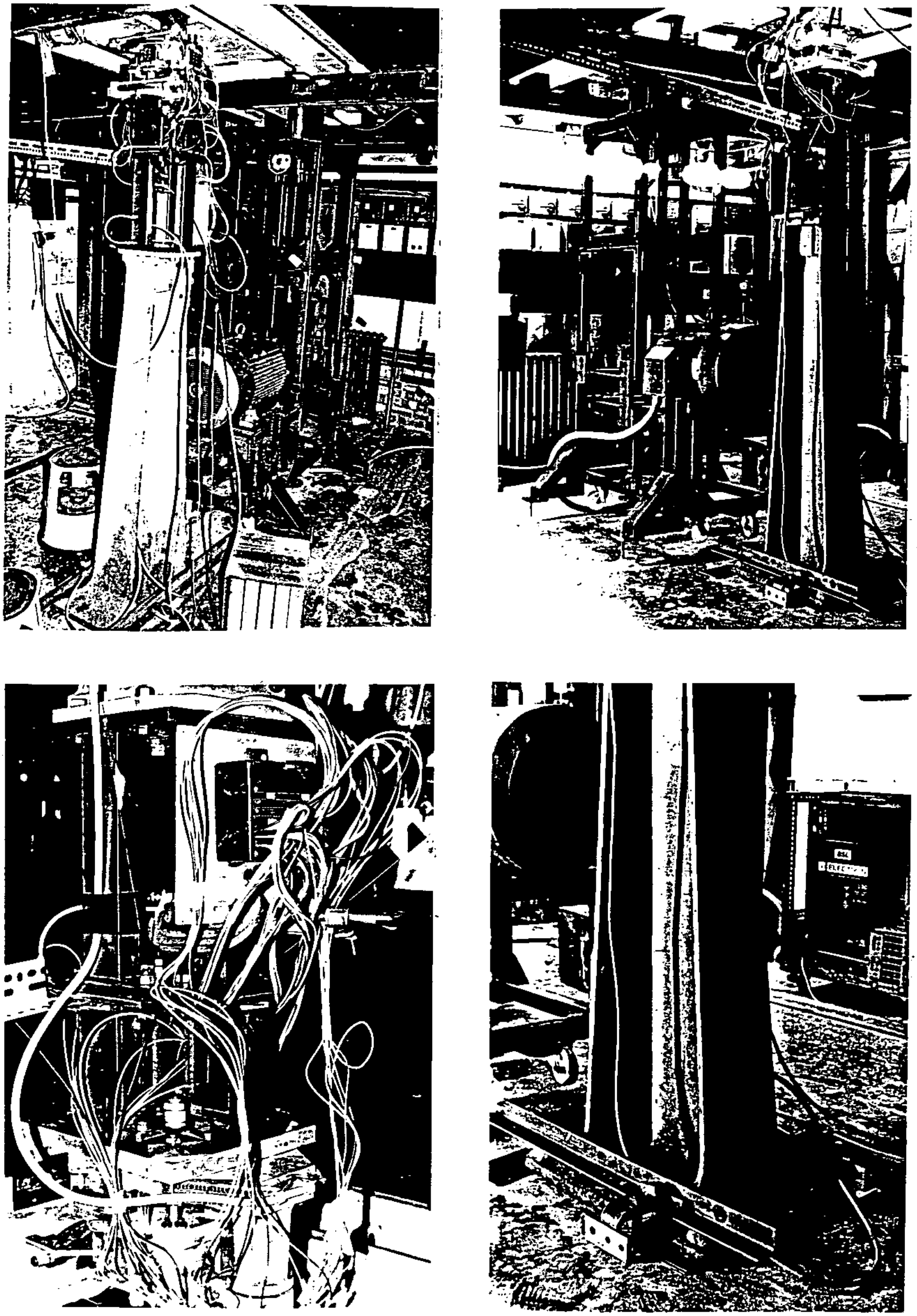


Figure 22 Photographs of Rudder Rig

Table 1 Required Operating Conditions for Propeller and Rudder Interactions

PROPELLER:

Revolutions: 0-3,000 RPM
Pitch Ratio: 0.4 - 1.4 (Design 1.0)
Max. Power : 30 kW
Diameter : 800mm

No. Of Blades : 4
Blade Area Ratio: 40%
Profile : Modified Wageningen B4-40

Hub/Diameter Ratio: 0.25
CentreLine Height : 0.6 - 0.9 m

RUDDER:

	Semi-Balanced Skeg/All-Movable	Rectangular All-Movable	Rectangular All-Movable
	RAP1	RAP2	RAP3
Mean Chord (m)	0.667	0.667	0.667
Span (m)	1.0	1.0	1.2
Taper Ratio	0.8	1.0	1.0

Wind Speed for Rudder Chord Reynolds No of 1000000 = 30 m/s

Table 2 Schedule of Work During Design and Construction of Rudder and Propeller Rig

Item \ Time	1989 March	April	May	June	July	August	September	October	November	December	1990 January	February	March
PHASE I	PD	PD											C W
Torque-Thrust Dynamometer			D		M								
Instrumentation Framework			D		M								
PHASE II													
Above-Wind-Tunnel Floor Assembly					D		M						
Lay-Shaft assembly					D		M						
PHASE III													
Top Frame							D		M				
Middle Frame							D		M				
Motor Frame							D		M				
Leg assembly							D		M				
Propeller Hub							D		M				
Propeller Blades									NC		M		

PD Preliminary Design
D Detailed Design
M Manufacture of Work
NC Numerical Control Data Generation
C Commissioning
W Wind Tunnel Tests