A Real-time Distributed Optimal Autopilot

G S Virk and J M Tahir

Department of Control Engineering
University of Sheffield
Mappin Street
Sheffield S1 3JD
U K

Research Report No 398

June 1990

Abstract

A new distributed optimal autopilot for aircraft flight control is presented. A parallel processing approach is taken where the longitudinal and lateral motions with cross-coupling effects are handled in different processors. The proposed control algorithm is implemented on a T800 transputer network programmed using Parallel C, and it is shown that real-time performance is possible.

1 Introduction

An aircraft in flight is a nonlinear time varying system, see Babister [1] and Blakelock [2], which can be demanding to control adequately over its entire flight regime. In principle, nonlinear optimal control techniques can provide the ability to design suitable strategies but these are rarely implemented due to the large computational requirements for solving such problems in real-time. For example the general nonlinear optimal control problem requires several steps to be performed iteratively for calculating the solution. The main steps in this procedure are:

Step 1: Linearise the formulated optimal control problem about the current operating conditions.

Step 2: Solve the linearised problem to give a "descent direction".

Step 3: If the optimal solution to the nonlinear problem has been deduced, stop.
Else calculate a suitable step size in the "descent direction", and determine a better operating condition. Then go to Step 1.

It is clear that steps $1-3$ are repeated until the desired solution has been obtained. Several theoretical (or conceptual) algorithms, see [3]– [5], have been proposed for calculating the solution of such problems using various methods including steepest descent, conjugate gradient, strong variational, penalty function and quasi-linearisation approaches. Some have been made implementable by reducing the need for exact solution at intermediate points but they still require vast levels of numerical processing that are difficult to satisfy in many real-time computer control applications.

When considering optimal control of aircraft systems the situation is further complicated because large dimensional mathematical models are necessary for adequate representation and fast sample rates are normally required. It is expected that a sampling rate of 200 Hz is necessary to achieve good real-time control performance, and so the computations must be accomplished within a sampling interval of 5 $msec$. Unfortunately, due to the magnitude of the computing task, this is difficult to achieve using conventional sequential computing methods and alternative approaches need to be investigated. One of the main alternatives is to use parallel processing techniques to distribute the computing task over a network of processors which can communicate effectively with each other, see Bertsekas [6], Hockney and Jesshope [7], Patton [8] and Kung [9].

A recent development in parallel computer systems has been the Inmos transputer [10], which has a number of unique serial links that enable true parallelism to be introduced in computing systems relatively easily. The transputer's introduction has greatly excited computer specialists, engineers and application scientists, and significant research is underway to develop expertise and to implement parallel processing based solutions for many application areas. The majority of current computer users rely on sequential "flow diagram" type of programming which is rather limiting when multiple processing devices are present. Problems such

as deadlock, communication overheads (granularity), program termination and synchronisation all become important issues that need consideration and handling in suitable ways.

In this paper we use a transputer network and parallel processing methods to determine a real-time optimal autopilot for an aircraft whose engineering data was supplied by British Aerospace, Brough [11]. Similiar reasoning as in Tahir and Virk [13], [14], where a distributed longitudinal autopilot is presented, is used to arrive at a suitable control strategy for the whole aircraft taking cross-coupling effects into account. We start our discussions by a statement of the mathematical model of the aircraft.

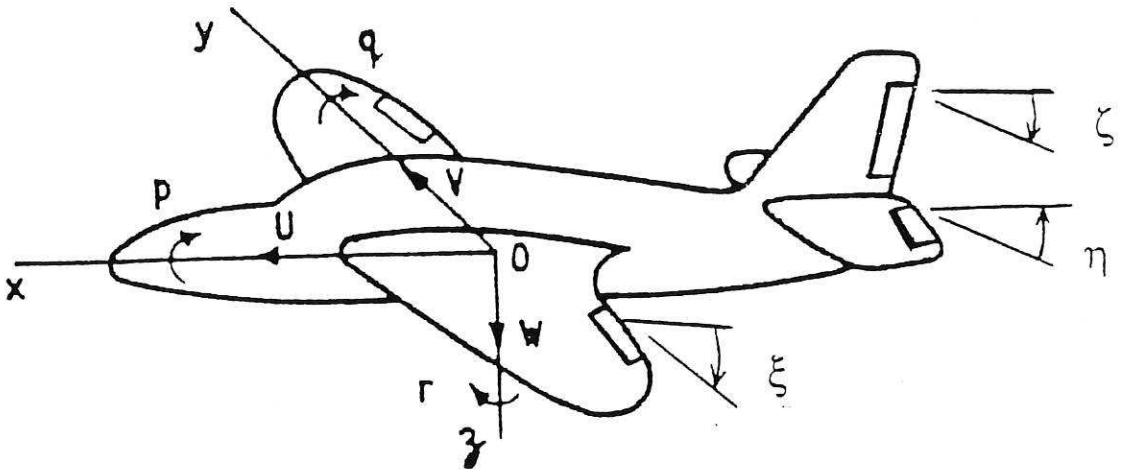

Figure 1: Aircraft in flight with notation and axes

2 Aircraft Equations

When considered as a rigid body (see Figure 1) the motion of the aircraft is defined by a set of nonlinear equations, see [1], [2]:

$$\dot{U}(t) = \frac{X_f(t)}{M} - Q(t)W(t) + R(t)V(t) \tag{1}$$

$$\dot{W}(t) = \frac{Z_f(t)}{M} + Q(t)U(t) - P(t)V(t) \tag{2}$$

$$\dot{Q}(t) = \left\{P_m(t) + (I_z - I_x)P(t)R(t) + I_{xz}\left(R^2(t) - P^2(t)\right)\right\}/I_y \tag{3}$$

$$\dot{\Theta}(t) = Q(t)\cos\Phi(t) - R(t)\sin\Phi(t) \tag{4}$$

$$\dot{H}(t) = U(t)\sin\Theta(t) - W(t)\cos\Theta(t)\cos\Phi(t) - V(t)\sin\Phi(t)\cos\Theta(t) \tag{5}$$

$$\dot{E}_s(t) = \bar{f}(E_s(t), V_r(t), H(t), \gamma(t)) \tag{6}$$

$$\dot{V}(t) = \frac{Y_f(t)}{M} + P(t)W(t) - R(t)U(t) \tag{7}$$

$$\dot{P}(t) = \left\{R_m(t) + (I_y - I_z)Q(t)R(t) + I_{xz}\left(\dot{R}(t) + P(t)Q(t)\right)\right\}/I_x \tag{8}$$

$$\dot{R}(t) = \left\{Y_m(t) + (I_x - I_y)P(t)Q(t) + I_{xz}\left(\dot{P}(t) - Q(t)r(t)\right)\right\}/I_z \tag{9}$$

$$\dot{\Phi}(t) = P(t) + (Q(t)\sin\Phi(t) + R(t)\cos\Phi(t))\tan\Theta(t) \tag{10}$$

$$\dot{\Psi}(t) = \{Q(t)\sin\Phi(t) + R(t)\cos\Phi(t)\}/\cos\Theta(t) \tag{11}$$

or in compact form as

$$\dot{x}\left(t\right) = f\left(x\left(t\right), u_{in}\left(t\right)\right) \tag{12}$$

where the state vector $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is made up of variables related with the longitudinal motion, $\left(x_1 = [U, W, Q, \theta, H, E_s]^T\right)$ and those connected with the lateral motion $\left(x_2 = [V, P, R, \Phi, \Psi]^T\right)$, and the control input vector $\left(u_{in} = [\eta, \gamma, \xi, \zeta]^T\right)$. The notation used is defined in the appendix. These equations can be written in a linearised form to highlight the cross-coupling terms between the longitudinal and lateral motions as

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \\ \dot{e}_s \\ \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \left[\begin{array}{cccccc|ccccc} f_{1u} & f_{1w} & f_{1q} & f_{1\theta} & 0 & f_{1es} & R & 0 & V & 0 & 0 \\ f_{2u} & f_{2w} & f_{2q} & f_{2\theta} & 0 & f_{2es} & -P & -V & 0 & f_{2\phi} & 0 \\ f_{3u} & f_{3w} & f_{3q} & f_{3\theta} & 0 & f_{3es} & 0 & f_{3p} & f_{3r} & f_{3\psi} & 0 \\ 0 & 0 & f_{4q} & 0 & 0 & 0 & 0 & 0 & f_{4r} & f_{4\phi} & 0 \\ f_{5u} & f_{5w} & 0 & f_{5\theta} & 0 & 0 & f_{5v} & 0 & 0 & f_{5\phi} & 0 \\ f_{6u} & 0 & 0 & 0 & f_{6h} & f_{6es} & 0 & 0 & 0 & 0 & 0 \\ \hline f_{7u} & f_{7w} & 0 & f_{7\theta} & 0 & 0 & f_{7v} & f_{7p} & f_{7r} & f_{7\phi} & 0 \\ f_{8u} & f_{8w} & f_{8q} & 0 & 0 & 0 & f_{8v} & f_{8p} & f_{8r} & 0 & 0 \\ f_{9u} & f_{9w} & f_{9q} & 0 & 0 & f_{9es} & f_{9v} & f_{9p} & f_{9r} & 0 & 0 \\ 0 & 0 & f_{10q} & f_{10\theta} & 0 & 0 & 0 & f_{10p} & f_{10r} & f_{10\phi} & 0 \\ 0 & 0 & f_{11q} & f_{11\theta} & 0 & 0 & 0 & 0 & f_{11r} & f_{11\phi} & 0 \end{array}\right] \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \\ e_s \\ v \\ p \\ r \\ \phi \\ \psi \end{bmatrix}$$

$$+ \begin{bmatrix} f_{1\eta} & 0 & 0 & 0 \\ f_{2\eta} & 0 & 0 & 0 \\ f_{3\eta} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & f_{6\gamma} & 0 & 0 \\ 0 & 0 & 0 & f_{7\zeta} \\ 0 & 0 & f_{8\xi} & f_{8\zeta} \\ 0 & 0 & f_{9\xi} & f_{9\zeta} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \gamma \\ \xi \\ \zeta \end{bmatrix} \tag{13}$$

where $f_{1a} = \partial f_1 / \partial a$ is the partial derivative of the first element of function f, i.e. f_1, with respect to state a, etc, and $u, w, q, \ldots, \psi$ are the state perturbations about the linearised point. In compact form this reduces to:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} u_{in}(t) \tag{14}$$

where A_{11}, A_{12}, A_{21} and A_{22}, are the partial derivations of the nonlinear functions f in equation (12) with respect to the states and B_1 and B_2 are the derivatives with respect to the controls. Hence A_{12} and A_{21} represents the cross-coupling terms, and if these are absent the longitudinal and lateral motions can be easily decoupled and handled independently of each other. It is well known that the cross-coupling terms can be removed by assuming:

(i) the aircraft is in straight and unaccelerated flight and then disturbed by deflections of the control surfaces.

(ii) the elevator deflection causes only a pitching moment about the OY axis and causes no rolling or yawing moments.

(iii) the aileron and rudder deflections causes rotations only about the OX and OZ axes respectively.

These assumptions are not strictly valid in many modern aircraft and the situation is worsening with designs evolving towards having more weight concentrated in the fuselage and the wings becoming thinner and shorter (and hence lighter). This weight shift is causing the cross-coupling effects to be increased considerably since the various moments of inertia are changing, and hence necessitating the need for consideration of these effects in the controller design. As more weight is concentrated along the longitudinal axis, the moment of inertia about the OX axis, I_x decreases while the moments of inertia about the OY and OZ axes increase. This phenomena increases the interaction between the longitudinal and lateral motions, and can best be seen by examining the basic moment equations (3), (8) and (9). As I_x becomes much smaller than I_y and I_z the moment of inertia difference terms $(I_z - I_x)$ and $(I_x - I_y)$ become large. Hence if a rolling moment is introduced it results in some yawing moment, and the term $P(t)R(t)(I_z - I_x)$ may become large enough to cause considerable pitching.

Another factor that must be considered is the aerodynamic cross-coupling effects that are present. For example, the lateral aerodynamic derivatives are proportional to the angle of attack (α) which is dependent upon longitudinal states (i.e, $\alpha = \tan^{-1}\frac{W}{U}$). Furthermore, in our controller design, we use pointwise linearisations for the aircraft over short intervals of time, and so the aircraft may not be in straight and level unaccelerated flight at the linearisation instants as assumed above.

Hence the above assumptions are not valid and an alternative method for breaking the aircraft equations is needed for improved system performance. Equation (13) shows that the longitudinal motion can be separated if we assume that the lateral state perturbations $\left([v, p, r, \phi, \psi]^T\right)$ have small values which can be neglected. In the same way we can separate the lateral motion equations by neglecting the longitudinal perturbations. Therefore if X is a vector of the state variable then

$$X(t) = X_0 + x(t) \tag{15}$$

where X_0 is the value of the state vector at the linearisation instant, $x(t)$ is the state vector perturbations. We now present out procedure for solving the aircraft optimal autopilot design problem.

2.1 Longitudinal Motion

The lateral state perturbations are assumed to be small and that their effect can be neglected during an interval T_l between two succesive linearisations. Hence $\dot{V} = \dot{P} = \dot{R} = \dot{\Phi} = \dot{\Psi} = 0$, and this leads to the following longitudinal equations (not showing time dependence for convenience):

$$\dot{U} = \frac{X_f}{M} - QW + (R_0 V_0)^* \tag{16}$$

$$\dot{W} = \frac{Z_f}{M} + QU - (P_0 V_0)^* \tag{17}$$

$$\dot{Q} = \left\{P_m + (I_z - I_x) P_0 R_0 + I_{xz}\left(R_0^2 - P_0^2\right)\right\}/I_y \tag{18}$$

$$\dot{\Theta} = Q\cos\Phi_0 - (R_0 \sin\Phi_0)^\dagger \tag{19}$$

$$\dot{H} = U\sin\Theta - W\cos\Theta\cos\Phi_0 - V_0\cos\Theta\sin\Phi_0 \quad (20)$$

$$\dot{E}_s = \bar{f}(E_s, V_r, H, \gamma) \quad (21)$$

2.2 Lateral Motion

In the same way if we neglect the effects of longitudinal perturbations, then $\dot{U} = \dot{W} = \dot{Q} = \dot{\Theta} = \dot{H} = \dot{E}_s = 0$, and we have the following lateral equations:

$$\dot{V} = \frac{Y_f}{M} + PW_0 - RU_0 \quad (22)$$

$$\dot{P} = \left\{R_m + (I_y - I_z)RQ_0 + I_{xz}\left(\dot{R} + PQ_0\right)\right\}/I_x \quad (23)$$

$$\dot{R} = \left\{Y_m + (I_x - I_y)PQ_0 + I_{xz}\left(\dot{P} - RQ_0\right)\right\}/I_z \quad (24)$$

$$\dot{\Phi} = P + (Q_0\sin\Phi + R\cos\Phi)\tan\Theta_0 \quad (25)$$

$$\dot{\Psi} = \{Q_0\sin\Phi + R\cos\Phi\}/\cos\Theta_0 \quad (26)$$

These sets of equations will be used for short time intervals T_l over which the aircraft is linearised and suitable optimal control laws designed. It is clear that in this way cross-coupling effects between the two motions are allowed for but are assumed to be constant over the linearised intervals. For example when considering the longitudinal motion, the lateral variables (V, P, R, Φ, Ψ) are assumed to be constant at their values $(V_0, P_0, R_0, \Phi_0, \Psi_0)$ when the linearisation is performed. In this case, the lateral variables are found to appear as separate constant terms which disappear in the linearisation process and so have no effect on the longitudinal feedback gains.

In the lateral motion equations (22)-(26) the (assumed constant) longitudinal terms are combined with the (assumed varying) lateral terms. Therefore $(U_0, W_0, Q_0, H_0, E_{s0})$ will not vanish in general from the linearised lateral equations and hence effect the feedback gains calculated allowing for the cross-coupling effects.

A solution for handling the various cross-coupling terms in equations (16)-(26) is possible along the following lines:

(i) The terms marked * in equations (16) and (17) normally have a small effect on the longitudinal motion and can therefore be removed from the equations without effecting the performance significantly.

(ii) The inertial cross-coupling term in equation (18) is normally the most important term. It can be neutralised by applying an equal and opposite amount of pitching moment using the elevator. Such a result can be achieved by setting

$$\Delta\eta = \left\{PR(I_x - I_z) + I_{xz}\left(P^2 - R^2\right)\right\}/\frac{\partial P_m}{\partial \eta} \quad (27)$$

very The term marked † in equation (19) can also be eliminated by changing the pitch rate Q by an amount ΔQ which cancels this cross-coupling effect and so

$$\Delta Q = R\tan\Phi \quad (28)$$

This ΔQ can be added to the demanded pitch rate Q_d causing an error which generates (when multiplied by the corresponding feedback gain) an elevator control action dependent upon yaw rate and roll angle to keep the pitch angle and aircraft height at desired values.

In this way it is not necessary to wait until significant errors accumulate in the longitudinal states before remedial action can be taken since such an action can be computed and applied as soon as the errors arise in the lateral attitude angles and rates.

3 Optimal Control Problem

The approach taken here is to formulate an optimal control problem for the complete linearised aircraft, separate the motions into the longitudinal and lateral dynamics, taking into account the cross-coupling effects, and solve the two subproblems using a multi-transputer network in real-time.

The nonlinear aircraft system, equation (12) can be linearised about an operating point $(X_0, U_{in,0})$ and the equations written as

$$\dot{e} = Ae(t) + B\Delta u(t) \tag{29}$$

where the A and B matrices are of the form shown in equation (14), e is the error in the system states and Δu is the control deviation from the operating point $U_{in,0}$. The elements of the A and B matrices were obtained using the aircraft engineering data supplied by British Aerospace, Brough, [11]. Equation (29) is assumed to be time invariant for a short interval T_l over which optimisation is performed, cross-coupling effects assumed to be constant and handled as discussed above, and the resulting control actions applied to a simulation of the nonlinear time-varying aircraft. The receding horizon technique is used so that real-time, on-line optimal control of the aircraft is possible. Starting at time t_0 a linear quadratic performance index having the form

$$\begin{aligned} J(u, t_0) &= \frac{1}{2} \int_{t_0}^{t_0+T} \left[e^T(t) Q e(t) + \Delta u^T(t) R \Delta u(t) \right] dt \\ &+ \frac{1}{2} \left[e^T(t_0+T) F e(t_0+T) \right] \end{aligned} \tag{30}$$

is considered and minimised, where Q and F are 6×6 positive semi-definite matrices for the longitudinal mode and 5×5 positive semi-definite matrices for the lateral mode, R is a 2×2 positive definite matrix for each mode and T is the length of the horizon interval (in seconds).

It is well known that the receding horizon optimal control law, see Kwon and Bruckstein [12], is given for our problem by

$$\Delta u^*(t) = -R^{-1} B^T P(t_0, t_0+T) e(t) \tag{31}$$

$$u^*(t) = U_0 + \Delta u^*(t) \tag{32}$$

where $P(\cdot)$ is computed by solving the following Riccati equation backwards

$$\dot{P}(t) = -P(t)A - A^T P(t) - Q + P(t) B R^{-1} B^T P(t) \tag{33}$$

$$P(t_0+T, t_0+T) = F \tag{34}$$

Once the time t has reached $t_0 + T_l$, a new linearising interval of length T_l can be started by setting $t_0 = t_0 + T_l$ and repeating the cycle indefinitely.

4 Real-time Parallel Implementation

The above optimisation and control procedure can be stated in the following algorithm form:

Step 0: Initialise control U_c, initial time t_0, receding horizon length T, linearisation interval T_l, sampling interval T_s, demanded state x_d.

Step 1: Linearise the aircraft equations ((16) – (21) or (22) – (26)) about the current state X_0 and the midpoint control U_c.

Step 2: Integrate the Riccati equation from t_0+T to t_0, and store the gains $P_T = P(t_0, t_0+T)$.

Step 3: Set $t = t_0$, and compute the control $u_{in}(t), t \in [t_0, t_0 + T_l]$ using step 4.

Step 4: Calculate

$$\Delta u(t) = -R^{-1}B^T P_T e(t) \quad (35)$$

$$\text{where state error } e(t) = x(t) - x_d \quad (36)$$

$$\text{set } u_{in}(t) = U_c + \Delta u(t) \quad (37)$$

$$\text{set } t = t + T_s \quad (38)$$

Step 5: Calculate $x(t)$ by integrating the nonlinear aircraft equations (1) - (11) from initial condition $x(t - T_s)$ and using the new control $u_{in}(t)$.

Step 6: If $t \geq t_0 + T_l$, set $X_0 = x(t), t_0 = t$ and go to Step 1.
Else go to Step 4.

Note that the aircraft linearisations are performed about the control U_c, where the control surfaces are at the centre (zero) position, and the throttle control is at the centre (0.5) position. The use of U_c instead of the current control $u_{in}(t)$ at the instant of linearisation is to prevent the integral action in the calculated control from equation (37), caused by the high repetition rate of the linearisation.

When the above algorithm is used for controlling the longitudinal motion, the extra controlling terms needed to account for the inertial cross-coupling $\Delta\eta$ from equation (27) should be added to the calculated elevator control input from equation (37). In addition the pitch rate ΔQ from equation (28) should be added to the desired pitch rate as discussed earlier.

Clearly for real-time performance all these computations have to be performed iteratively within the time scales of the aircraft. A sampling rate of approximately 200 Hz is usually required to achieve satisfactory control. Hence the above algorithm should be processed to provide control updates every 5 $msec$. It has been shown in Tahir and Virk [13], [14], that the aircraft can be adequately controlled using a time invariant linearised model for a specified time interval of T_l seconds, which can be made equal to the execution time for Steps 1 and 2 of the above algorithm. A transputer network system as shown in Figure 2 is used to provide real-time control as follows:

Transputer T0 is the aircraft simulator.

Transputers T1 and T2 handle the lateral mation, that is provide aileron (ξ) and rudder (ζ) control actions. Transputer T1 performs these control updates by using the gains P_T stored in, and supplied by Transputer T2. T2 performs the linearisations and solves the Riccati integration, that is steps 1 and 2 of the above algorithm for the lateral mode.

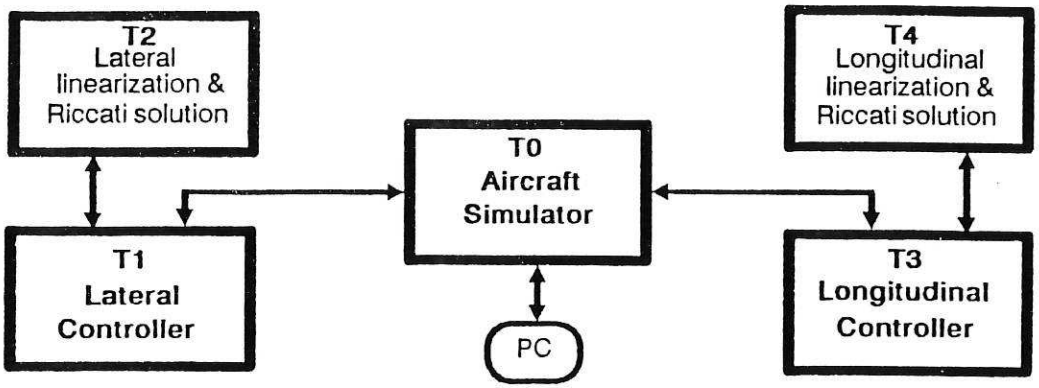

Figure 2: Transputer system for autopilot implementation

Transputers T3 and T4 control the longitudinal motion of the aircraft, that is provide the elevator (η) and engine throttle (γ) controls. T3 performs the same operations as T1, and transputer T4, the same task as transputer T2, but for the longitudinal mode.

Both the lateral and longitudinal autopilots read the complete state from the simulator through transputer T1 and T3 respectively, so that the cross-coupling effects can be allowed for as discussed earlier. In practice an observer is necessary to estimate the states which are not directly measured for use in such a control system design implementation.

For a receding horizon interval T of 0.5 *sec*, and if the integration step used in step 2 is 10 *msec* then it is found that the lateral Riccati equations can be up dated every 40 *msec* ($= T_{l,lat}$)and the longitudinal ones every 50 *msec* ($= T_{l,long}$). These figures can be reduced if the linearisation and Riccati solution transputers (T2 and T4) can share their computations with extra processors - see Tahir and Virk [13] for the computational complexity and detailed timings.

5 Results and Conclusions

The real-time algorithm was coded on a T800 transputer network of Figure 2 using Parallel C. Using the state vector $x = [U, W, Q, \theta, H, E_s, V, P, R, \Phi, \Psi]^T$, an initial state

$$x_{ic} = [\ 150,\ 5,\ 0,\ 0.033,\ 5000,\ 6615,\ 0,\ 0,\ 0,\ 0,\ 0.2]^T$$

and a desired state of

$$x_d = [\ 150,\ 5,\ 0,\ 0.033,\ 5000,\ 6615,\ 0,\ 0,\ 0,\ 0,\ 0]^T$$

is assumed, and so the aircraft is required to change its direction by 11.5°.

The following weighting matrices were used.

Longitudinal motion autopilot

$$\begin{aligned} Q_{long} &= diag\left[\ 5,\ 0,\ 500,\ 1000,\ 0.6,\ 5 \times 10^{-5}\right], \\ R_{long} &= diag[\ 5,\ 1], \\ F_{long} &= 0 \end{aligned}$$

Lateral motion autopilot

$$
\begin{aligned}
Q_{lat} &= diag[\,0.1,\ 10,\ 1000,\ 15,\ 1500], \\
R_{lat} &= diag[\,1,\ 50], \\
F_{lat} &= 0
\end{aligned}
$$

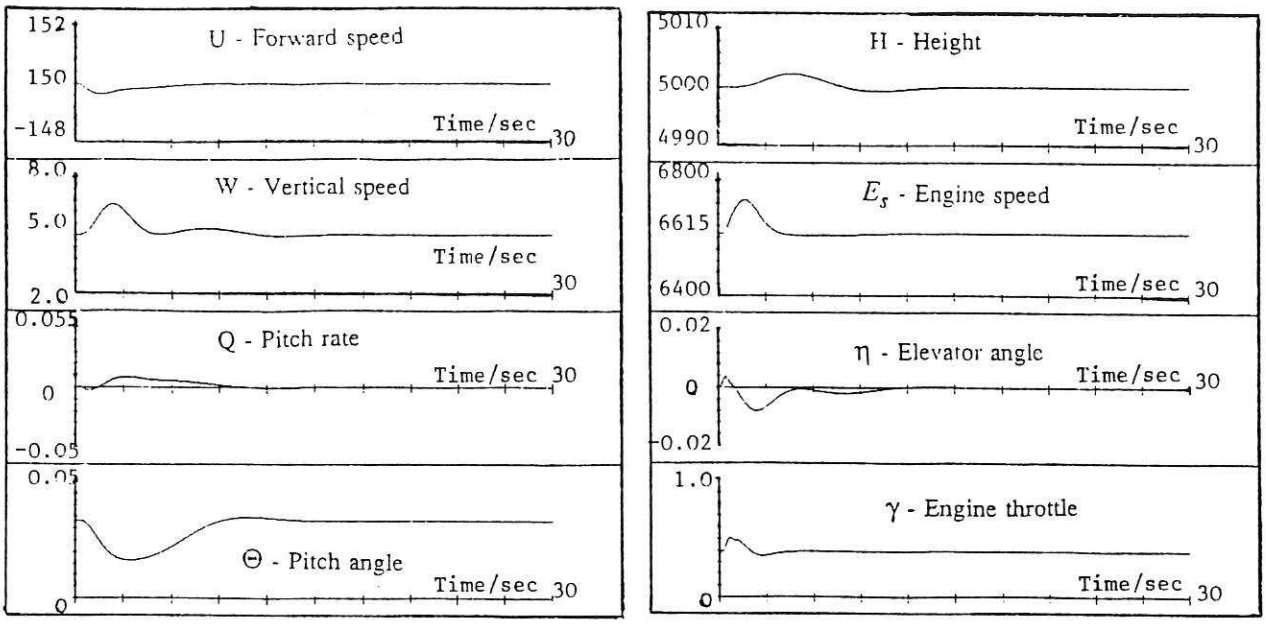

Figure 3: Longitudinal aircraft motion under optimal control

The optimal trajectories and controls for the longitudinal motion are shown in Figure 3 and the lateral ones in Figure 4. From these we can observe the following:

(i) The elevator positive deflection in the first few seconds is due to the additional cross-coupling control $\Delta\eta$, while later on the elevator responds optimally to reduce the errors in the pitch angle and aircraft height.

(ii) The reduction in forward speed U is caused mainly by the positive pitch rate and slightly by the cross-coupling term (RV). This reduction in U is remedied optimally by a slight increase in throttle control.

(iii) The response to an error in the yaw direction (Ψ) is acceptable and the negative error in the roll angle (Φ) reduces the angle of sideslip using the bank to turn idea. The amount of roll can be reduced by increasing the weighting elements q_{11}, q_{33} and r_{11} in the lateral mode, but at the expense of increasing the settling time for the yaw angle heading, see Virk and Tahir [15].

Overall, the results shown in Figures 3 and 4 are adequate. Hence the linear time invariant aircraft model, and the cross-coupling assumptions are valid and useful when considering problems of this kind. Moreover the cross-coupling problem has been solved without requiring any physical link between the two controllers and hence they can work separately to achieve real-time performance.

It has been shown that such performance levels are possible by splitting the computational task functionally into smaller sub-tasks which are processed on different devices. The

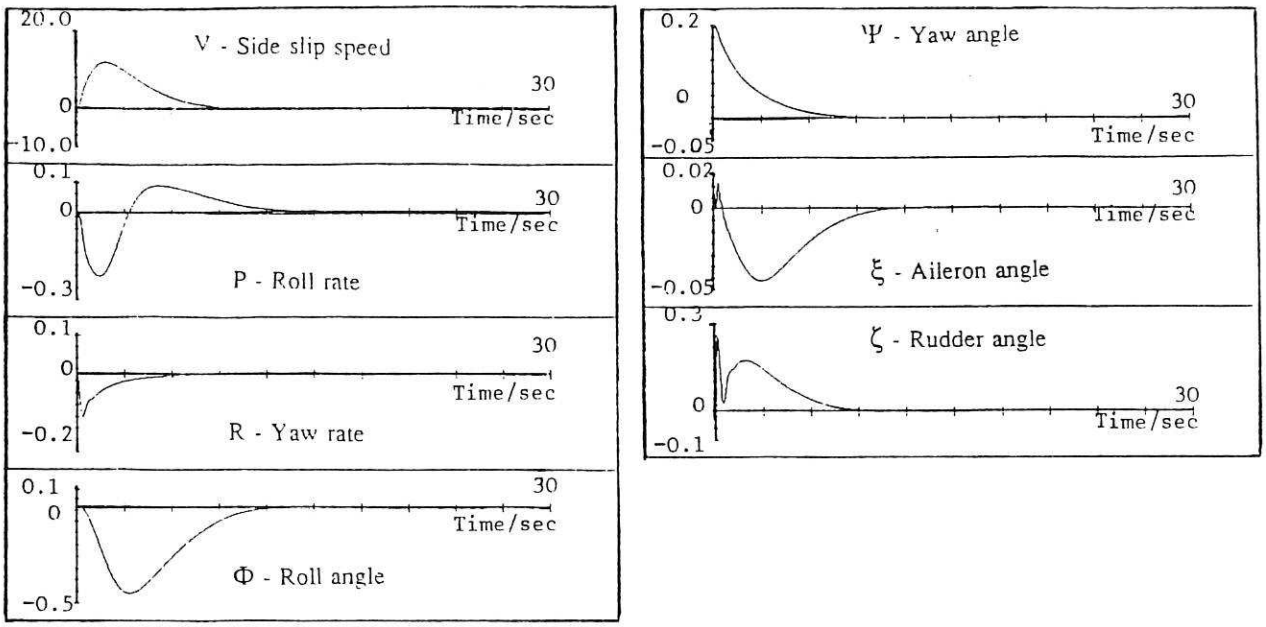

Figure 4: Lateral aircraft motion under optimal control

individual processing devices can be configured into suitable architectures for optimising the communication aspects for the application being considered.

It is imperative that such flexibility be available for the effective employment of parallel processing techniques to a wide variety of applications.

Acknowledgement

The authors would like to express their gratitude to British Aerospace (Military Aircraft) Ltd, Brough for providing the aircraft engineering data.

References

[1] Babister A W, Aircraft dynamic stability and response, Pergamon Press 1980.

[2] Blakelock J H, Automatic control of aircraft and missiles, John Wiley & Sons 1965.

[3] Polak E, Computational methods in optimisation, Academic Press, 1970.

[4] Gruver W A and Sachs E, Algorithmic methods in optimal control, Pitman, 1980.

[5] Dyer P and MacReynolds S R, The computation and theory of optimal control, Academic Press, 1970.

[6] Bertsekas D P and Tsitsiklis , Parallel and distributed computation: numerical methods, Prentice-Hall, 1989.

[7] Hockney R W and Jesshope C R, Parallel Computers 2: Architecture programming and algorithms, Adams Hilger, Bristol, 1988.

[8] Patton P C, Multiprocessors: Architectures and applications, Computer, Vol 18, No 6, pp 29-40, 1985.

[9] Kung H T, Why systolic architectures, IEEE Computer, pp 37-46, 1982.

[10] Inmos Ltd, IMS T800 Transputer engineering data, 1988.

[11] British Aerospace aircraft engineering data, (private correspondence).

[12] Kwon W H, Bruckstein A M and Kailath T, Stabilising state feedback design via the moving horizon method, Int J Control, Vol.37, No.3, pp 631-643, 1983.

[13] Tahir J,M and Virk G S, A real-time distributed algorithm for an aircraft longitidinal optimal autopilot, Concurrancy: Practice and Experience, Vol 2(2), pp 109-121, 1990.

[14] Virk G S and Tahir J M, Parallel optimal control algorithms for Aircraft, IEE Colloquium on navigation, guidance and control in aerospace. Digest No.1989/142, 3/1 - 3/5, November 1989.

[15] Virk G S and Tahir J M, Selection of weights in optimal control, Research Report 397, Department of Control Engineering, University of Sheffield, 1990.

Appendix

Table of notations

U	m/sec	forward component of aircraft velocity
W	m/sec	downward component of aircraft velocity
V	m/sec	velocity of sideslip
V_r	m/sec	aircraft relative velocity
Q	rad/sec	pitch rate
P	rad/sec	roll rate
R	rad/sec	yaw rate
Θ	rad	pitch attitude angle
Φ	rad	roll attitude angle
Ψ	rad	yaw attitude angle
H	$metres$	aircraft height
E_s	rev/min	engine speed
η	rad	elevator deflection
ξ	rad	aileron deflection
ζ	rad	rudder deflection
γ	dimensionless	engine power setting
X_f	N	total force acting along OX axis
Y_f	N	total force acting along OY axis
Z_f	N	total force acting along OZ axis
P_m	Nm	total pitching moment
Y_m	Nm	total yawing moment
R_m	Nm	total rolling moment
M	kg	total aircraft mass
I_x	kgm^2	moment of inertia about OX axis
I_y	kgm^2	moment of inertia about OY axis
I_z	kgm^2	moment of inertia about OZ axis
I_{xz}	kgm^2	the cross product of inertia about OZX axes
T_s	ms	sampling period
T_l	ms	time between successive linearizations
T	sec	horizon depth
x		vector of state variables
x_d		vector of desired states
u_{in}		vector of control variables
α	rad	angle of attack