T-70-01874

Bell Aerospace Company DIVISION OF textron

POST OFFICE BOX 1 BUFFALO, NEW YORK 14240

FINAL FLIGHT CONTROL SOFTWARE PACKAGE FOR DIGITAL FLIGHT CONTROL AND LANDING SYSTEM

CONTRACT NAS 12-2074

REPORT NO. 6200-933011

MAY 1970

CONTENTS

ILLUSTRATIONS

TABLES

LIST OF SYMBOLS

Symbol	Description	Type
g	Gravitational acceleration at sea level	Input constant
k_p	Roll rate damping gain	Input constant
k_q	Pitch rate damping gain	Input constant
k_r	Yaw rate damping gain	Input constant
$k_{V_{cx}\delta_{ep}}$	Forward velocity command sensitivity on electric stick pitch input	Input constant
$k_{V_{cx}\delta_{eps}}$	Forward velocity command sensitivity on sidearm controller pitch input	Input constant
$k_{V'_{cx}}$	Forward velocity integral bypass gain	Input constant
$k_{\dot{V}_{cx}\delta_{ep}}$	Forward acceleration command sensitivity on electric stick pitch input	Input constant
$k_{V_{cy}\delta_{ap}}$	Lateral velocity command sensitivity on electric stick roll input	Input constant
$k_{V_{cy}\delta_{aps}}$	Lateral velocity command sensitivity on sidearm controller roll input	Input constant
$k_{V'_{cy}}$	Lateral velocity integral bypass gain	Input constant
$k_{\dot{V}_{cy}\delta_{ap}}$	Lateral acceleration command sensitivity on electric stick roll input	Input constant
$k_{V_{cz}\delta_{cp}}$	Vertical velocity command sensitivity on collective stick input	Input constant
k_{V_x}	Forward velocity feedback gain	Input constant
$k_{V'_x}$	Forward velocity integral gain	Input constant
$k_{\dot{V}_x}$	Forward acceleration feed forward gain	Input constant
k_{V_y}	Lateral velocity feedback gain	Input constant
$k_{V'_y}$	Lateral velocity integral gain	Input constant
$k_{\dot{V}_y}$	Lateral acceleration feed forward gain	Input constant
k_{V_z}	Vertical velocity feedback gain	Input constant
$k_{V'_z}$	Vertical velocity integral gain	Input constant
k_β	Sideslip feedback gain	Input constant
$k_{\delta_a\delta_{ap}}$	Cyclic roll rate command sensitivity on electric stick roll input	Input constant
$k_{\delta_c\phi}$	Gain on roll command feed forward into collective	Input constant
$k_{\delta_e\delta_{ep}}$	Differential collective pitch rate command sensitivity on electric stick pitch input	Input constant
$k_{\delta_r\delta_{rp}}$	Differential cyclic yaw rate sensitivity on rudder pedal input	Input constant
$k_{\delta_r\phi}$	Gain on roll command feed forward into rudder for coordinated turns	Input constant
$k_{\Delta V_{xd}}$	Forward velocity command error sensitivity on flight director horizontal needle	Input constant
$k_{\Delta V_{yd}}$	Lateral velocity command error sensitivity on flight director vertical needle	Input constant
$k_{\Delta V_{zd}}$	Vertical velocity command error sensitivity on flight director collective bug	Input constant
$k_{\Delta\delta_{ad}}$	Cyclic roll rate command error sensitivity on flight director vertical needle	Input constant
$k_{\Delta\delta_{cd}}$	Collective command error sensitivity on flight director collective bug	Input constant
$k_{\Delta\delta_{ed}}$	Differential collective/pitch rate command error sensitivity on flight director horizontal needle	Input constant
$k_{\Delta\theta_d}$	Pitch attitude command error sensitivity on flight director horizontal needle	Input constant
$k_{\Delta\phi_d}$	Roll attitude command error sensitivity on flight director vertical needle	Input constant
k_θ	Pitch attitude feedback gain	Input constant
$k_{\theta_c\delta_{ep}}$	Pitch attitude command sensitivity on electric stick pitch input	Input constant

LIST OF SYMBOLS (CONT.)

Symbol	Description	Type
$k_{\theta H}$	Hysteresis compensation gain in pitch	Program variable
$k_{\theta H\,max}$	Maximum hysteresis compensation gain in pitch	Input constant
$k_{\theta'}$	Pitch trimming integrator gain	Input constant
$k_{\xi_c \delta_{ap}}$	Course command sensitivity on electric stick roll input	Input constant
$k_{\xi_c \delta_{ap}}$	Course rate command sensitivity on electric stick roll input	Input constant
k_{ϕ}	Roll attitude feedback gain	Input constant
$k_{\phi_c \delta_{ap}}$	Roll attitude command sensitivity on electric stick roll input	Input constant
$k_{\phi H}$	Hysteresis compensation gain in pitch	Program variable
$k_{\phi H\,max}$	Maximum hysteresis compensation gain in roll	Input constant
$k_{\phi'}$	Roll trimming integrator gain	Input constant
k_{ψ}	Yaw attitude feedback gain	Input constant
$k_{\psi H}$	Hysteresis compensation gain in yaw	Program variable
$k_{\psi H\,max}$	Maximum hysteresis compensation gain in yaw	Input constant
$k_{\psi'}$	Yaw trimming integrator gain	Input constant
$k_{\dot{\psi}}$	Yaw rate command feed forward gain	Input constant
$k_{\dot{\psi}_c \delta_{rp}}$	Yaw rate command sensitivity on rudder pedal input	Input constant
$k_{\dot{\psi}_c \phi_c}$	Gain on yaw rate command due to roll command	Program variable
$k_{\dot{\psi}_c \phi_c\,max}$	Maximum gain on yaw rate command due to roll command	Input constant
p	Angular rate about the x body axis	Input variable
q	Angular rate about the y body axis	Input variable
r	Angular rate about the z body axis	Input variable
s	Laplacian operator	Program operation
V_a	Sensed airspeed	Input variable
V_{af}	Filtered airspeed	Program variable
V_a'	A forward speed used for switching	Program variable
V_T	Switching value of Va′	Input constant
V_x^{an}	Forward velocity in approach navigation frame (ANF)	Input variable
V_y^{an}	Lateral velocity in approach navigation frame (ANF)	Input variable
V_x^h	Forward velocity in vertical heading frame (VHF)	Input variable
V_{xo}^h	Initial value of V_x^h	Program variable
V_y^h	Lateral velocity in vertical heading frame (VHF)	Input variable
V_{yo}^h	Initial value of V_y^h	Program variable
V_y^{ξ}	Lateral velocity in vertical course frame (VCF)	Program variable
V_{yo}^{ξ}	Initial value of V_y^{ξ}	Program variable
V_y'	A lateral velocity (either V_y^h or V_y^{ξ})	Program variable
V_{yo}'	Initial value of Vy′	Program variable
V_z^h	Vertical velocity in vertical heading frame (VHF)	Input variable
V_{zo}^h	Initial value of V_z^h	Program variable
$Z_{\delta_c\,avg}$	Average vertical acceleration per inch of collective command	Input constant
β	Sideslip angle	Input variable
δ_{ac}	Cyclic command to EISS	Output variable

LIST OF SYMBOLS (CONT.)

Symbol	Description	Type
δ_{cc}	Collective command to EISS	Output variable
δ_{ec}	Differential collective command to EISS	Output variable
δ_{rc}	Differential cyclid command to EISS	Output variable
δ_{ad}	Vertical needle display command to flight director	Output variable
δ_{cd}	Collective bug display command to flight director	Output variable
δ_{ed}	Horizontal needle display command to flight director	Output variable
δ_{ap}	Electric stick roll input	Input variable
δ_{apo}	Initial value of δ_{ap}	Program variable
δ_{aps}	Sidearm controller roll input	Input variable
δ_{apso}	Initial value of δ_{aps}	Program variable
δ_{cp}	Collective stick input	Input variable
δ_{cpo}	Initial value of δ_{cp}	Program variable
δ_{ep}	Electric stick pitch input	Input variable
δ_{epo}	Initial value of δ_{ep}	Program variable
δ_{eps}	Sidearm controller pitch input	Input variable
δ_{epso}	Initial value of δ_{eps}	Program variable
δ_{rp}	Rudder pedal input	Input variable
δ_{rpo}	Initial value of δ_{rp}	Program variable
$\Delta V_{at}'$	Switching threshold on Va'	Input constant
ΔV_{cx}^{h}	Incremental forward velocity command in VHF	Program variable
$\Delta \dot{V}_{cx}^{h}$	Incremental forward acceleration comand in VHF	Program variable
ΔV_{cy}^{h}	Incremental lateral velocity command in VHF	Program variable
ΔV_{cy}^{ξ}	Incremental lateral velocity command in VCF	Program variable
$\Delta V_{cy}'$	An incremental lateral velocity command (either ΔV_{cy}^{h} or ΔV_{cy}^{ξ})	Program variable
$\Delta \dot{V}_{cy}^{h}$	Incremental lateral acceleration command in VHF	Program variable
$\Delta \dot{V}_{cy}^{\xi}$	Incremental lateral acceleration command in VCF	Program variable
ΔV_{cz}^{h}	Incremental vertical velocity command in VHF	Program variable
ΔV_{x}^{h}	Forward velocity error in VHF	Input or program variable
$\Delta V_{y}'$	A lateral velocity error (either ΔV_{y}^{h} or ΔV_{y}^{ξ})	Program variable
ΔV_{z}^{h}	Vertical velocity error in VHF	Input or program variable
$\Delta\delta_{ac}$	Incremental cyclic command	Program variable
$\Delta\delta_{aH}$	Expected hysteresis in cyclic channel	Input constant
$\Delta\delta_{ap}$	Incremental electric stick roll input	Program variable
$\Delta\delta_{aps}$	Incremental sidearm controller roll input	Program variable
$\Delta\delta_{ast}$	Threshold on sidearm controller pitch input	Input constant
$\Delta\delta_{at}$	Threshold on electric stick roll input	Input constant
$\Delta\delta_{a\dot{\phi}c}$	Incremental cyclic roll rate command	Program variable
$\Delta\delta_{cc}$	Incremental collective command	Program variable
$\Delta\delta_{cp}$	Incremental collective stick input	Program variable
$\Delta\delta_{c\phi}$	Incremental collective command due to roll	Program variable

LIST OF SYMBOLS

Symbol	Description	Type
$\Delta\delta_{ec}$	Incremental differential collective command	Program variable
$\Delta\delta_{eH}$	Expected hysteresis in differential collective channel	Input constant
$\Delta\delta_{ep}$	Incremental electric stick pitch input	Program variable
$\Delta\delta_{eps}$	Incremental sidearm controller pitch input	Program variable
$\Delta\delta_{est}$	Threshold on sidearm controller pitch input	Input constant
$\Delta\delta_{et}$	Threshold on electric stick pitch input	Input constant
$\Delta\delta_{e\dot{\theta}_c}$	Incremental differential collective pitch rate command	Program variable
$\Delta\delta_{jc}$	An incremental command in the jth EISS channel	Program variable
$\Delta\delta_{jc\ prev}$	A previous value of $\Delta\delta_{jc}$	Program variable
$\Delta\delta_{jr}$	A reference on the jth EISS channel	Program variable
$\Delta\delta_{jr\ prev}$	A previous value of $\Delta\delta_{jr}$	Program variable
$\Delta\delta_{rc}$	Incremental differential cyclic command	Program variable
$\Delta\delta_{rH}$	Expected hysteresis in differential cyclic channel	Input constant
$\Delta\delta_{rp}$	Incremental rudder pedal input	Program variable
$\Delta\delta_{rt}$	Threshold on rudder pedal input	Input constant
$\Delta\delta_{r\beta}$	Incremental rudder command due to sideslip	Program variable
$\Delta\delta_{r\phi}$	Incremental rudder command due to roll	Program variable
$\Delta\delta_{r\dot{\psi}c}$	Incremental differential cyclic yaw rate command	Program variable
$\Delta\theta_c$	Incremental pitch attitude command	Program variable
$\Delta\theta_{max}$	Maximum pitch attitude command increment	Input constant
$\Delta\xi_c$	Incremental course command	Program variable
$\Delta\dot{\xi}_c$	Incremental course rate command	Program variable
$\Delta\phi_c$	Incremental roll attitude command	Program variable
θ	Euler pitch attitude	Input variable
θ_{cl}	Limited pitch attitude command	Program variable
θ_{max}	Maximum pitch attitude command limit	Program variable
θ_{min}	Minimum pitch attitude command limit	Program variable
θ_o	Initial value of θ	Program variable
θ_{trim}	Pitch trim attitude	Program variable
ξ	Course	Program variable
τ_{V_a}	Airspeed filter time constant	Input constant
τ_β	Sideslip filter time constant	Input constant
$\tau_{\dot{\psi}_c}$	Yaw rate command filter time constant	Input constant
ϕ	Euler roll attitude	Input variable
ϕ_c	Roll attitude command	Program variable
ϕ_{cl}	Limited roll attitude command	Program variable
ϕ_{max}	Maximum roll attitude command limit	Input constant
ϕ_o	Initial value of ϕ	Program variable
ψ	Euler yaw attitude	Input variable
ψ_o	Initial value of ψ	Program variable
$\dot{\psi}_c$	Yaw rate command	Program variable
$\dot{\psi}_{c\phi}$	Yaw rate command due to roll	Program variable
$\dot{\psi}_{max}$	Maximum yaw rate command limit	Input constant

I. INTRODUCTION

This document describes the software required to mechanize the flight control system on an airborne computer and is submitted in compliance with Item 2(a) of NASA/ERC Contract NAS12-2074. The majority of this description is presented in the form of a detailed engineering equations and functional block diagrams and only those functions that are peculiar to the digital nature of the system are presented in the form of digital flow diagrams and logic equations. It is believed that this type of description will give the airborne computer programmer a better understanding of the system involved and will enable him to develop the most efficient digital program for the airborne computer that is selected.

A detailed description of the development and performance of this system is contained in the final report on this contract, Bell Report No. 6200-933013.

II. FUNCTIONAL DESCRIPTION OF FLIGHT CONTROL SYSTEM

A. MODES

There are eight pilot selectable modes of operation of the flight control laws: (1) Disengage, (2) SAS, (3) Attitude I, (4) Attitude II, (5) Velocity I, (6) Velocity II, (7) Velocity III, and (8) Automatic. In the first of these modes, the system is inactive except for mode sampling and initialization activities. The next six modes are manual modes of operation where command errors are normally displayed to the pilot on flight director needles and the pilot nulls these errors by inputting commands to the flight control laws through the electric stick or sidearm controller, rudder pedals, and collective stick. The last mode of operation is a completely automatic mode where incremental velocity error inputs to the flight control laws are obtained from the guidance laws. In this mode, actual velocity errors are displayed on the flight director needles for monitoring purposes. The types of control, the form and source of the commands, and the form of the displays for each of these modes, except the Disengage mode, are listed in Table 1. In this table, the following abbreviations are used:

ES	Electric Stick.
CS	Collective Stick.
RP	Rudder Pedals.
SAC	Side-arm Controller.
GUID.	Guidance.

All of the manual modes and the automatic modes of operation of the flight control laws are designed to be used in conjunction with either the Guidance I or Guidance II modes. The manual flight control modes can also be used when the guidance is in the Disengage mode although no command errors will be available on the flight director needles in this case. The automatic flight control mode cannot be used when the guidance is in the Disengage mode. In addition, all of these possible modes of operation apply in both the flight phase to hover and in the landing flight phase.

B. FLIGHT CONTROL LAWS

1. General

The flight control laws for the various modes of operation have been divided into command laws and control laws. The command laws convert the pilot control inputs into command inputs to the flight control laws for the selected mode of operation. There are separate command laws for the SAS Mode, the Attitude Modes, and each of the Velocity Modes. By doing this, it is only necessary to include a single set of velocity control laws for the three Velocity Modes since the control inputs in each mode can be converted into velocity commands that are compatible with these.

The control laws generate the incremental output commands to the EISS. In addition, in conjunction with the command laws, they also generate the display information in each mode. As a result of this, no separate display laws are required. The command laws have been developed in a cascading manner where there are separate laws for the SAS, attitude, and velocity loops. Each of

TABLE I
CONTROL MODE DEFINITIONS

Mode	Command and Source				Type of Control				Display		
	DIFF. COLL.	CYCLIC	COLL.	DIFF. CYCLIC	DIFF. COLL.	CYCLIC	COLL.	DIFF. CYCLIC	HOR. NEEDLE	VERT. NEEDLE	COLL. BUG
SAS	$\Delta\delta_{e\dot{\theta}_c}$ (ES)	$\Delta\delta_{a\dot{\phi}_c}$ (ES)	$\Delta\delta_{cc}$ (CS)	$\Delta\delta_{r\dot{\psi}_c}$ (RP)	Pitch Rate Damping	Roll Rate Damping	Direct	Sideslip hold, $V_a' > V_T$ Yaw Rate Damping, $V_a' < V_T$	$\Delta\delta_{e\dot{\theta}cd}$	$\Delta\delta_{a\dot{\phi}cd}$	$\Delta\delta_{cd}$
Att. I	$\Delta\theta_c$ (ES)	$\Delta\phi_c$ (ES)	$\Delta\delta_{cc}$ (CS)	$\dot{\psi}_c$ (RP)	Pitch Attitude Hold	Roll Attitude Hold	Direct	Sideslip Hold, $V_a' > V_T$ Heading Hold, $V_a' < V_T$	$\Delta\theta_{cd}$	$\Delta\phi_{cd}$	$\Delta\delta_{cd}$
Att. II	$\Delta\theta_c$ (ES)	$\Delta\phi_c$ (ES)	ΔV^h_{cz} (CS)	$\dot{\psi}_c$ (RP)	Pitch Attitude Hold	Roll Attitude Hold	Vertical Velocity Hold	Sideslip Hold, $V_a' > V_T$ Heading Hold, $V_a' < V_T$	$\Delta\theta_{cd}$	$\Delta\phi_{cd}$	ΔV_{czd}
Vel. I	$\Delta\dot{V}^h_{cx}$ (ES)	$\Delta\dot{V}^h_{cy}$ or $\Delta\dot{\xi}_c$ (ES)	ΔV^h_{cz} (CS)	$\dot{\psi}_c$ (RP)	Forward Velocity Rate Hold	Lateral Velocity Rate or Course Rate Hold	Vertical Velocity Hold	Sideslip Hold, $V_a' > V_T$ Heading Hold, $V_a' < V_T$	ΔV_{cxd}	ΔV_{cyd}	ΔV_{czd}
Vel. II	ΔV^h_{cx} (ES)	ΔV^h_{cy} or $\Delta\xi_c$ (ES)	ΔV^h_{cz} (CS)	$\dot{\psi}_c$ (RP)	Forward Velocity Hold	Lateral Velocity or Course Hold	Vertical Velocity Hold	Sideslip Hold, $V_a' > V_T$ Heading Hold, $V_a' < V_T$	ΔV_{cxd}	ΔV_{cyd}	ΔV_{czd}
Vel. III	ΔV^h_{cx} (SAC)	ΔV^h_{cy} (SAC)	ΔV^h_{cz} (CS)	$\dot{\psi}_c$ (RP)	Forward Velocity Hold	Lateral Velocity Hold	Vertical Velocity Hold	Sideslip Hold, $V_a' > V_T$ Heading Hold, $V_a' < V_T$	ΔV_{cxd}	ΔV_{cyd}	ΔV_{czd}
Auto	ΔV^h_x (GUID.)	ΔV^h_y GUID.)	ΔV^h_z (GUID.)	$\dot{\psi}_c$ (FCL)	Forward Velocity Hold	Lateral Velocity Hold	Vertical Velocity Hold	Sideslip Hold, $V_a' > V_T$ Heading Hold, $V_a' < V_T$ to Hover Yaw into Wind, Hover	ΔV_{xd}	ΔV_{yd}	ΔV_{zd}

these control laws receives its input either from the corresponding command law or next outer loop depending on the mode of operation. This was done to eliminate the necessity of duplicating the inner loops in the digital flight control program sections for the higher order modes.

This separation of the command and control laws also permits the laws for the outer loops to be updated at a slower rate than is used for those in the inner loops. As a result of this, the command and control laws have been grouped into fast and slow loop computations for efficiency in the airborne program. In general, the fast loop computations contain the: (1) SAS loop control laws, (2) SAS command laws, and (3) attitude control laws. The slow loop computations contain the: (1) attitude command laws, (2) velocity control laws, and (3) velocity command laws. In some cases, certain equations in the inner loops can be updated in the slow loop computations. These will be specifically pointed out in the following sections.

2. SAS Laws

a. Command Laws

The SAS command laws convert the incremental electric stick pitch, electric stick roll and rudder pedal inputs into inches of attitude rate command in the pitch, roll, and yaw channels respectively. The collective stick input is converted directly to a collective command in SAS.

Pitch SAS Command

$$\Delta\delta_{e\dot{\theta}_c} = k_{\delta_e \delta_{ep}} \Delta\delta_{ep}$$

Roll SAS Command

$$\Delta\delta_{a\dot{\phi}_c} = k_{\delta_a \delta_{ap}} \Delta\delta_{ap}$$

Yaw SAS Command

$$\Delta\delta_{r\dot{\psi}_c} = k_{\delta_r \delta_{rp}} \Delta\delta_{rp}$$

Collective Command

$$\Delta\delta_{cc} = \Delta\delta_{cp}$$

b. Control Laws

The SAS control laws use inches of attitude rate command inputs and attitude rate, sideslip, and roll feedback to generate incremental commands in inches of pitch roll, and yaw control.

Pitch SAS Control

$$\Delta\delta_{ec} = k_{\theta H} (\Delta\delta_{e\dot{\theta}_c} - k_q q)(1 + \frac{k_\theta'}{s}),$$

where $k_{\theta H}$ is a gain for hysteresis compensation and is defined as,

$$k_{\theta H} = \frac{|\Delta\delta_{e\dot{\theta}_c} - k_q q| + \Delta\delta_{eH}}{|\Delta\delta_{e\dot{\theta}_c} - k_q q|} \leq k_{\theta H\ max}$$

Roll SAS Control

$$\Delta\delta_{ac} = k_{\phi H} (\Delta\delta_{a\dot{\phi}_c} - k_p p) (1 + \frac{k_\phi'}{s}),$$

where $k_{\phi H}$ is gain for hysteresis compensation and is defined as,

$$k_{\phi H} = \frac{|\Delta\delta_{a\dot{\phi}_c} - k_p p| + \Delta\delta_{aH}}{|\Delta\delta_{a\dot{\phi}_c} - k_p p|} \leq k_{\phi H\ max}$$

Yaw SAS Control

$$\Delta\delta_{rc} = [k_{\psi H} (\Delta\delta_{r\dot{\psi}_c} - k_r r + \Delta\delta_{r\beta}) + \Delta\delta_{r\phi}] (1 + \frac{k_\psi'}{s})$$

where $\Delta\delta_{r\beta} = -\frac{k_\beta \beta}{T_\beta s + 1}$, if V_a' is high

$= 0,$ if V_a' is low

$\Delta\delta_{r\phi} = 0,$ if in SAS mode or if V_a' is low

$= k_{\delta_r \phi} \phi_{cl},$ if not in SAS mode and V_a' is high

$$k_{\psi H} = \frac{|\Delta\delta_{r\dot{\psi}_c} - k_r r + \Delta\delta_{r\beta}| + \Delta\delta_{rH}}{|\Delta\delta_{r\dot{\psi}_c} - k_r r + \Delta\delta_{r\beta}|} \leq k_{\psi H\ max}$$

The speed, V_a', at which the sideslip feedback is switched and which is used in all subsequent speed dependent switching is defined as,

$V_a' = V_{af},$ if $V_{af} \leq V_x^h$

$= V_x^h,$ if $V_{af} > V_x^h$

where $V_{af} = \frac{V_a}{\tau_{V_a} s + 1}$

It is recommended that the computation of this speed be made in the slow loop computations since it does not change rapidly relative to the SAS commands.

3. Attitude Loops

a. Command Laws

The pitch and roll attitude command laws convert the incremental electric stick pitch and electric stick roll inputs into pitch and roll attitude commands respectively. The yaw rate command laws convert the rudder pedal input into a yaw rate command in the manual modes. In the Auto mode, they generate a yaw rate command from the roll attitude command in the roll attitude control laws. In the Attitude I mode, the collective stick input is converted directly to a collective command as in the SAS mode. In the Attitude II mode, the collective command is converted into a vertical velocity command as described in the next section on velocity command laws.

Pitch Attitude Command

$$\Delta\theta_c = k_{\theta_c \delta_{ep}} \Delta\delta_{ep}$$

Roll Attitude Command

$$\Delta\phi_c = k_{\phi_c \delta_{ap}} \Delta\delta_{ap}$$

Manual Yaw Rate Command

$$\dot{\psi}_c = k_{\dot{\psi}_c \delta_{rp}} \Delta\delta_{rp}$$

Auto Yaw Rate Command

$$\dot{\psi}_{c\phi} = \frac{k_{\dot{\psi}_c \phi_c} \phi_{cl}}{\tau_{\dot{\psi}_c} s + 1}, \quad \text{if in hover or land phase of guidance}$$

$$= 0., \quad \text{if in any other phase of guidance}$$

Where:

$$k_{\dot{\psi}_c \phi_c} = \frac{g}{|V_n^h|} \leq k_{\dot{\psi}_c \phi_c \max}$$

$$\dot{\psi}_c = \dot{\psi}_{c\phi}, \quad \text{if } |\dot{\psi}_{c\phi}| \leq \dot{\psi}_{max}$$

$$= \dot{\psi}_{max}, \quad \text{if } \dot{\psi}_{c\phi} > \dot{\psi}_{max}$$

$$= \dot{\psi}_{max}, \quad \text{if } \dot{\psi}_{c\phi} < -\dot{\psi}_{max}$$

b. Control Laws

The attitude control laws use the incremental attitude commands or attitude rate command and attitude feedback to generate inches of attitude rate commands.

Pitch Attitude Control

$$\Delta\delta_{e\dot{\theta}_c} = k_\theta (\theta_{cl} - \theta),$$

where:

$$\theta_{cl} = \theta_0 + \Delta\theta_c, \text{ if } \theta_{min} \leq (\theta_0 + \Delta\theta_c) \leq \theta_{max}$$

$$= \theta_{max}, \quad \text{if } (\theta_0 + \Delta\theta_c) > \theta_{max}$$

$$= \theta_{min}, \quad \text{if } (\theta_0 + \Delta\theta_c) < \theta_{min}$$

The upper and lower limits on the pitch attitude command are defined as,

$$\theta_{max} = \theta_{trim} + \Delta\theta_{max}$$

$$\theta_{min} = \theta_{trim} - \Delta\theta_{max}$$

The trim pitch attitude required to define these limits is approximated by,

$$\theta_{trim} = 0.1438, \qquad \text{if } V_{af} \leq V_T$$

$$= 0.1625 - 0.297 \left(\frac{V_{af}}{236}\right)^2, \text{ if } V_{af} > V_T$$

It is recommended that the computation of the trim pitch attitude be made in the slow loop computations since it does not change rapidly relative to the attitude commands.

Roll Attitude Control

$$\Delta\delta_{a\dot{\phi}_c} = k_\phi (\phi_{cl} - \phi),$$

where

$$\phi_{cl} = \phi_0 + \Delta\phi_c, \quad \text{if } |\phi_0 + \Delta\phi_c| \leq \phi_{max}$$

$$= \phi_{max} \text{ sgn } \Delta\phi_c, \text{ if } |\phi_0 + \Delta\phi_c| > \phi_{max}$$

Yaw Rate Control

$$\Delta\delta_{r\dot{\psi}_c} = k_\psi \left(\frac{\dot{\psi}_c}{s} + \psi_0 - \psi\right) + k_{\dot{\psi}} \dot{\psi}_c, \quad \text{if } V_a' \text{ is low}$$

$$= k_r \dot{\psi}_c, \quad \text{if } V_a' \text{ is high and Attitude or Velocity Modes are selected}$$

$$= 0, \quad \text{if } V_a' \text{ is high and Auto Mode is selected.}$$

4. Velocity Loops

a. Command Laws

(1) General

The velocity command laws convert control inputs into incremental forward, lateral, and vertical velocity commands. The forward and lateral velocity command laws are dependent on the velocity mode selected since both the pilot controls used and the form of the control inputs change with the mode of operation. The vertical velocity command law is independent of the mode of operation.

(2) Forward and Lateral Velocity Commands

(a) Velocity I Mode

In this mode, incremental forward velocity rate is commanded through the pitch axis of the electric stick and either incremental lateral velocity rate or course rate, depending on speed, is commanded through the roll axis of the electric stick. These inputs are converted to velocity commands as follows:

Forward Velocity I Command

$$\Delta V_{cx}^{h} = (k_{\dot{V}_x} + \frac{1}{s})\, \Delta \dot{V}_{cx}^{h}$$

where:

$$\Delta \dot{V}_{cx}^{h} = k_{\dot{V}_{cx}\, \delta_{ep}}\, \Delta\delta_{ep}$$

Lateral Velocity I Command

$$\Delta V_{cy}' = (k_{\dot{V}_y} + \frac{1}{s})\, \Delta \dot{V}_{cy}'$$

where:

$$\Delta V'_{cy} = (k_{\dot{V}_y} + \frac{1}{s})\, \Delta \dot{V}_{cy'}^{h}, \quad \text{if } V_a' \text{ is low}$$

$$= (k_{\dot{V}_y} + \frac{1}{s})\, \Delta \dot{V}_{cy'}^{\xi}, \quad \text{if } V_a' \text{ is high}$$

$$\Delta \dot{V}_{cy}^{h} = k_{\dot{V}_{cy}\, \delta_{ap}}\, \Delta\delta_{ap}$$

$$\Delta \dot{V}_{cy}^{\xi} = V_a'\, \Delta\dot{\xi}_c$$

$$\Delta\dot{\xi}_c = k_{\dot{\xi}_c\, \delta_{ap}}\, \Delta\delta_{ap}$$

(b) Velocity II Mode

In this mode, incremental forward velocity with integral bypass is commanded through the pitch axis of the electric stick and either incremental lateral velocity or course, depending on speed, with integral bypass is commanded through the roll axis of the electric stick. These inputs are converted to incremental velocity commands as follows.

Forward Velocity II Command

$$\Delta V_{cx}^{h} = \left(1 + \frac{k_{V_{cx}}'}{s}\right) k_{V_{cx}\,\delta_{ep}} \Delta\delta_{ep}$$

Lateral Velocity II Command

$$\Delta V_{cy}' = \left(1 + \frac{k_{V_{cy}}'}{s}\right) \Delta V_{cy'}^{h}, \quad \text{if } V_a' \text{ is low}$$

$$= \left(1 + \frac{k_{V_{cy}}'}{s}\right) \Delta V_{cy'}^{\xi}, \quad \text{if } V_a' \text{ is high}$$

where:

$$\Delta V_{cy}^{h} = k_{V_{cy}\,\delta_{ap}} \Delta\delta_{ap}$$

$$\Delta V_{cy}^{\xi} = V_a' \Delta\xi_c$$

$$\Delta\xi_c = k_{\xi_c\,\delta_{ap}} \Delta\delta_{ap}$$

(c) Velocity III Mode

In this mode, incremental forward velocity with integral bypass is commanded through the pitch axis of the sidearm controller and incremental lateral velocity with integral bypass is commanded through the roll axis of the sidearm controller. These outputs are converted into incremental velocity commands as follows:

Forward Velocity III Command

$$\Delta V_{cx}^{h} = \left(1 + \frac{k_{V_{cx}}'}{s}\right) k_{V_{cx}\,\delta_{eps}} \Delta\delta_{eps}$$

Lateral Velocity III Command

$$\Delta V_{cy}' = \Delta V_{cy}^{h} = \left(1 + \frac{k_{V_{cy}}'}{s}\right) k_{V_{cy}\,\delta_{aps}} \Delta\delta_{aps}.$$

(3). Vertical Velocity Command Law

In all modes where it is used, the vertical velocity command law converts the incremental collective stick input into an incremental vertical velocity command.

Vertical Velocity Command

$$\Delta V^h_{cz} = k_{V_{cz}\,\delta_{cp}} \Delta\delta_{cp}$$

b. Control Laws

The velocity control laws use the incremental velocity commands and velocity feedback to generate velocity errors in all modes except the Automatic Mode where the velocity errors are obtained directly from the guidance laws. The forward and lateral velocity errors are then used to generate incremental pitch and roll commands respectively. The vertical velocity error is used to generate an incremental collective command in inches of control.

Forward Velocity Control Equation

$$\Delta\theta_c = k_{V_x} \left(1 + \frac{k_{V_x}'}{s}\right) \Delta V^h_x$$

Forward Velocity Control Input

$$\Delta V^h_x = V^h_{xo} + \Delta V^h_{cx} - V^h_{x'}, \quad \text{if in manual modes}$$

$$= \Delta V^h_x \text{ from guidance,} \quad \text{if in Auto Mode.}$$

Lateral Velocity Control Equation

$$\Delta\phi_c = k_{V_y} \left(1 + \frac{k_{V_y}'}{s}\right) \Delta V_y'$$

Lateral Velocity Control Input

$$\Delta V_y' = \Delta V^h_y \text{ from guidance,} \quad \text{if in Auto Mode}$$

$$= V_{yo}' + \Delta V_{cy}' - V_y', \quad \text{if in Manual Modes}$$

$$V_y' = V^h_y, \quad \text{if } V_a' \text{ is low or if in Vel. III Mode}$$

$$= V_a'\,\xi, \quad \text{if } V_a' \text{ is high and if not in Vel. III Mode}$$

$$\xi = \tan^{-1}\left(V^{an}_y / V^{an}_x\right)$$

Vertical Velocity Control Equation

$$\Delta\delta_{cc} = k_{V_z}\left(1 + \frac{k_{V_z}'}{s}\right)\Delta V_z^h + \Delta\delta_{c\phi}$$

where: $$\Delta\delta_{c\phi} = k_{\delta_c\phi}(1 - \cos\phi)$$

Vertical Velocity Control Input

$$\Delta V_z^h = V_{zo}^h + \Delta V_{cz}^h - V_z^h, \quad \text{if in manual modes}$$

$$= \Delta V_z^h \text{ from guidance}, \quad \text{if in Auto Mode}$$

C. MODE CONTROL

The logic and controls required to select the appropriate command and control laws for each mode of operation, except the Disengage Mode, are shown in Figures 1 and 2 for the longitudinal and lateral axes respectively. Since the control laws have been developed in a cascading manner, it can be seen from these figures that the required EISS commands in each mode can be generated simply by switching in the control laws for the number of loops involved and the command laws corresponding to the highest order loop involved.

It can also be seen from these figures that the required display information in each mode, except the Auto Mode, is generated by differencing the commands that the automatic system would be using, if it were engaged, with the commands that are generated from the pilot's inputs and multiplying the resulting command errors by appropriate gains. The commands that the automatic system would be using, if it were engaged, are generated by using the guidance command inputs and the outer flight control loops which are of a higher order than those for the flight control mode selected.

The outer flight control loops which are required to generate the appropriate display information in each mode are automatically switched in by the same switches that control the output commands to the EISS. With this arrangement, no additional laws are required to generate the required display information. In the Auto Mode, the velocity errors obtained from the guidance laws are multiplied by appropriate gains and displayed directly.

In the Disengage Mode, no EISS commands are generated and it is only necessary to set the display commands (δ_{ed}, δ_{ad}, and δ_{cd}) to zero and to perform certain initialization functions as described in the next section.

It should be noted that the flight control mode that controls this switching is to be decoded from an input data word as described in Section II.E. Although the decoded flight control mode will normally be equal to the pilot selected mode, it can be different when the guidance is in the Disengage Mode.

D. INITIALIZATION

Whenever any switching takes place in the flight control system, various parameters in the flight control laws must be initialized. A complete initialization of these parameters is required

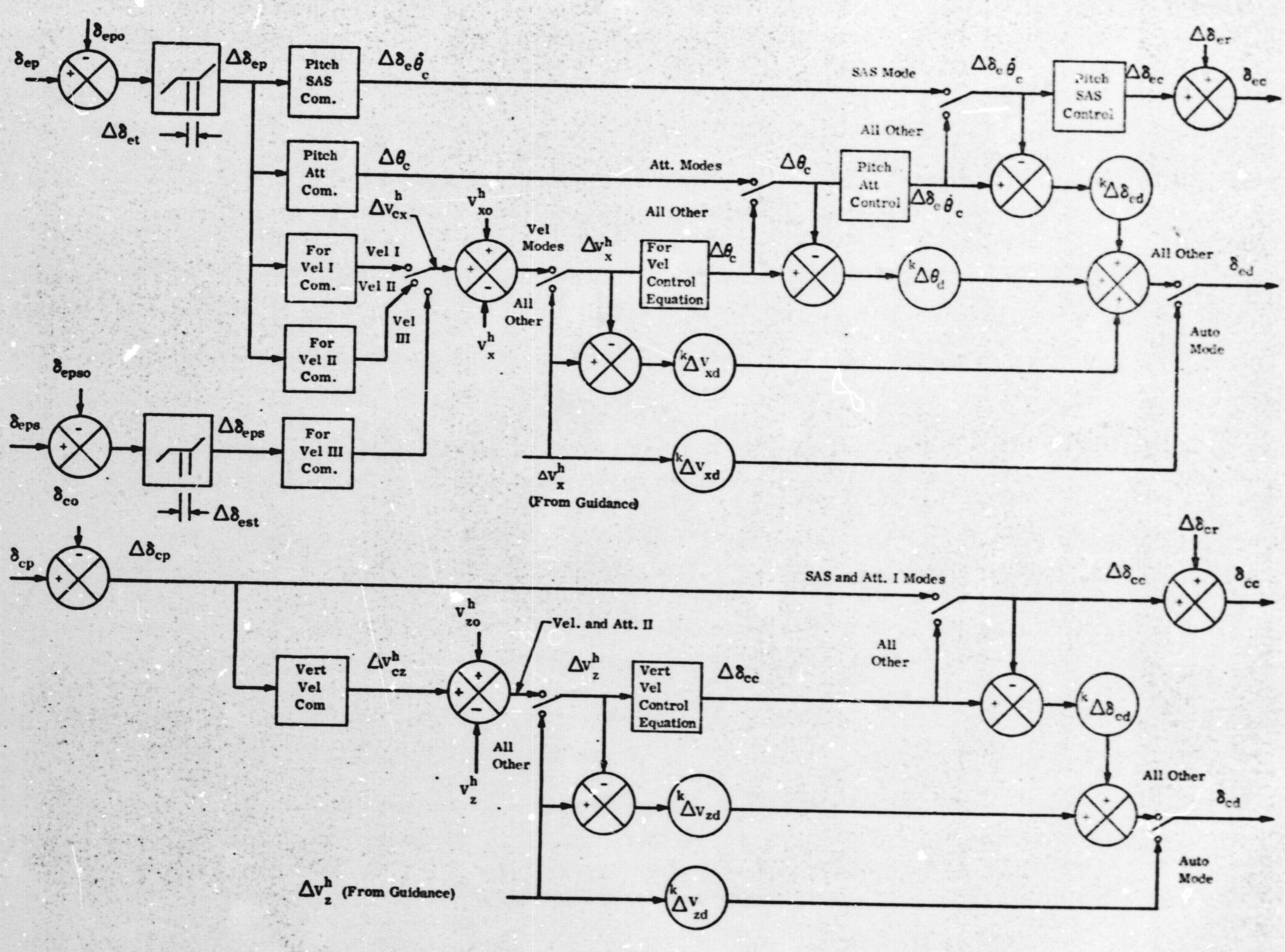


Figure 1. Longitudinal Flight Control Laws

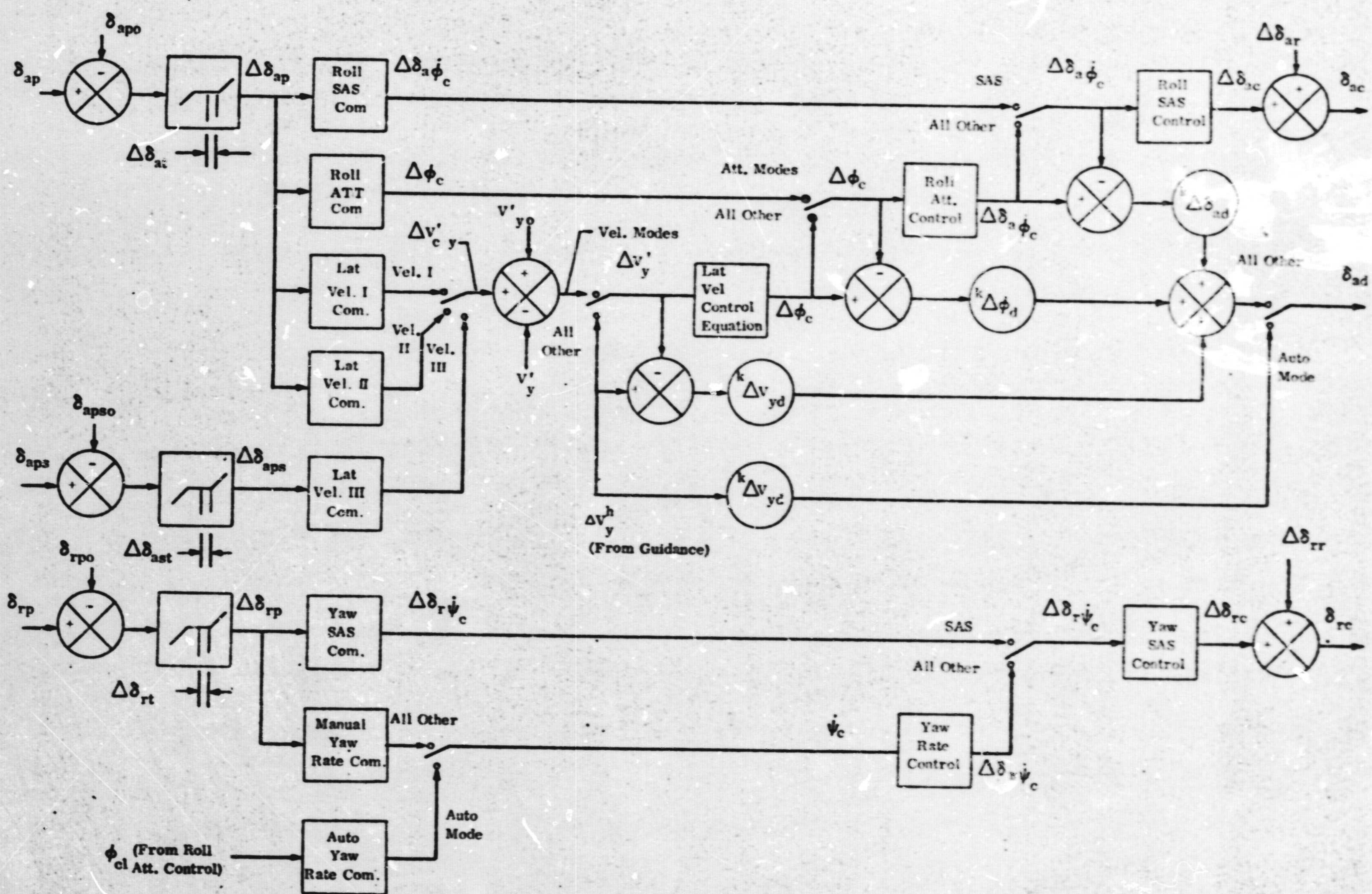


Figure 2. Lateral Flight Control Laws

whenever the decoded flight control mode changes, or the decoded guidance mode (Guid. I or Guid. II) changes. A partial initialization of these parameters is required whenever the system changes from the high speed mode to the low speed mode of operation or vice versa.

For a complete initialization of the system, the initialization is as follows:

(1) The initial conditions on the electric stick inputs (δ_{epo} and δ_{apo}), sidearm controller inputs (δ_{epso} and δ_{apso}), rudder pedal input (δ_{rpo}), and collective stick input (δ_{cpo}) must be set equal to the current values of these respective inputs.

(2) The initial conditions on the helicopter attitudes (ψ_0, θ_0, and ϕ_0), and velocities (V^h_{xo}, V^h_{yo}, V^ξ_{yo}, and V^h_{zo}) must be set equal to the current values of these flight variables.

(3) The previous values of the inputs and outputs of all integrator difference equations must be set equal to zero.

In addition, once a complete initialization has taken place, the guidance laws, the proper commands for the flight control mode selected, and the flight control laws must be gone through once in sequential order until the incremental EISS commands ($\Delta\delta_{ec}$, $\Delta\delta_{cc}$, $\Delta\delta_{ac}$, and $\Delta\delta_{rc}$) have been computed. On this cycle, the command references must then be reset before the final EISS commands are computed. This resetting is necessary to prevent any transients from occuring in the EISS commands as a result of a mode switch. The resetting is defined in general as,

$$\Delta\delta_{jr} = \Delta\delta_{jr\ prev} - (\Delta\delta_{jc} - \Delta\delta_{jc\ prev}),$$

where the subscript j refers to the channel (differential collective, collective, etc.) and the subscript prev refers to a previous value of a variable. The final EISS commands must then be computed using the updated values of these references. Once they are reset, they are to be left unchanged until another initialization takes place.

A partial initialization of the system is required whenever a switch is made from the high speed to the low speed mode of operation of the system or vice versa. Since this switch does not affect the pitch or collective channels, the initial conditions in these channels must not be initialized in this case and the command references in these channels must not be reset. In the roll and rudder channels, all initial conditions must be set as for a complete initialization except for the initial conditions on the roll inputs from the stick and sidearm controller (δ_{apo} and δ_{apso}) and on the rudder pedal input (δ_{rpo}). These initial conditions must be left unchanged. In addition, the command references on these channels ($\Delta\delta_{ar}$ and $\Delta\delta_{rr}$) must be reset as for a complete initialization.

In addition to the initialization just described, certain other parameters must be initialized when the system is in the Disengage Mode. When this occurs, the previous values of the difference equation inputs and outputs for the sideslip filter and airspeed filter must be set equal to the current sideslip and airspeed values respectively. In addition, the previous values of the command references in all channels must be set to zero.

E. INPUTS AND OUTPUTS

1. Inputs

a. Variable

The description, source, and units of all variable inputs required by the flight control system are listed in Table 2.

b. Discretes

Two discrete inputs are required from the cockpit mode controls: (1) the flight control mode and (2) the guidance mode. These will be externally coded into a single input data word for the airborne computer. This word must be decoded as required in the airborne computer. To enable the decoded form to be specified, it is assumed that, subject to final coder specification: (1) from right to left, the first three bits of this word will be used to specify the flight control mode and the next two bits will be used to specify the guidance modes and (2) the modes will be normally decoded from this word as specified in Table 3. As can be seen from this table, the decoded flight control mode is normally independent of the guidance mode and equal to the pilot selected flight control mode. However, in the event that the Auto Mode of flight control is selected while the guidance is in the Disengage Mode, the decoded flight mode is to remain at the last decoded mode instead of being decoded as Auto as shown in Table 3. In addition, in the event that the Disengage Mode of guidance is selected while the flight control is in the Auto Mode, the decoded flight control is to be set to Disengage instead of being decoded as Auto as shown in Table 3. In this case, an output warning signal is to be transmitted to the cockpit displays as described in Section III.E.2.

In addition to the two discretes from the cockpit mode controls, the Auto Yaw Rate Command section of the flight control system also requires a discrete from the guidance laws to determine if the vehicle is in the hover and land phases of flight or in the phases prior to hover. This discrete will be directly available from the guidance laws in the airborne computer and; therefore, is not a part of the coded input data word.

c. Constant

The description, values, and units of all constant inputs required by the flight control system are listed in Table 4. For cross referencing purposes, these are grouped according to the command and control laws where they are most directly used.

2. Outputs

a. Variable

The description and destination of all of the variable outputs of the flight control system are listed in Table 5. All of these outputs are generated in units of inches in the flight control system.

TABLE 2
VARIABLE INPUTS

Input	Description	Source	Units
p	Angular rate about the x body axis	Navigation System	Rad/sec
q	Angular rate about the y body axis	Navigation System	Rad/sec
r	Angular rate about the z body axis	Navigation System	Rad/sec
ψ	Euler yaw angle between body axis and heading vertical frame	Navigation System	Rad
θ	Euler pitch angle between body axis and heading vertical frame	Navigation System	Rad
ϕ	Euler roll angle between body axis and heading vertical frame	Navigation System	Rad
V_x^{an}	Longitudinal velocity in the approach navigation frame	Navigation System	Ft/sec
V_y^{an}	Lateral velocity in the approach navigation frame	Navigation System	Ft/sec
V_z^{an}	Vertical velocity in the approach navigation frame	Navigation System	Ft/sec
V_a	Total airspeed of the aircraft	Airspeed sensor	Ft/sec
β	Sideslip angle in the x, y body axis plane between airspeed vector and the x body axis	Beta vane sensor	Rad
δ_{ep}	Electric stick pitch input	Electric stick	In.
δ_{ap}	Electric stick roll input	Electric stick	In.
δ_{cp}	Collective stick input	Collective stick	In.
δ_{rp}	Rudder pedal input	Rudder Pedals	In.
δ_{eps}	Sidearm controller pitch input	Sidearm controller	In.
δ_{aps}	Sidearm controller roll input	Sidearm controller	In.
ΔV_x^h	Forward velocity error in the heading vertical frame	Guidance Laws	Ft/sec
ΔV_y^h	Lateral velocity error in the heading vertical frame	Guidance Laws	Ft/sec
ΔV_z^h	Vertical velocity error in the heading vertical frame	Guidance Laws	Ft/sec

TABLE 3

NORMAL INPUT DATA WORD DECODING

Coded Input Word			Decoded Modes	
Spares	Guidance Bits	Flight Control Bits	Guidance	Flight Control
	0 0		Disengage	
	0 1		Guidance I	
	1 0		Guidance II	
		0 0 0		Disengage
		0 0 1		SAS
		0 1 0		Attitude I
		0 1 1		Attitude II
		1 0 0		Velocity I
		1 0 1		Velocity II
		1 1 0		Velocity III
		1 1 1		Automatic

TABLE 4
CONSTANT INPUTS

Input	Section	Description	Value and Units
$\Delta\delta_{et}$	Control Thresholds	Threshold on electric stick pitch input	0.1 in.
$\Delta\delta_{at}$		Threshold on electric stick roll input	0.1 in.
$\Delta\delta_{rt}$		Threshold on rudder pedal input	0.1 in.
$\Delta\delta_{est}$		Threshold on sidearm controller pitch input	0.02 in.
$\Delta\delta_{ast}$		Threshold on sidearm controller roll input	0.02 in.
$k_{\delta_e\delta_{ep}}$	SAS Command Laws	Electric stick pitch sensitivity for SAS	1.0 in/in.
$k_{\delta_a\delta_{ap}}$		Electric stick roll sensitivity for SAS	1.0 in/in.
$k_{\delta_r\delta_{rp}}$		Rudder pedal sensitivity for SAS	1.0 in/in.
k_q	Pitch SAS Control	Pitch rate damping gain	6.5 in/rad/sec
$k_{\theta'}$		Pitch trimming integrator gain	0.2 in/rad/sec
$\Delta\delta_{eH}$		Expected hysteresis in differential collective channel	0.1 in.
$k_{\theta H\,max}$		Maximum hysteresis compensation gain in pitch	2.0
k_p	Roll SAS Control	Roll rate damping gain	7.5 in/rad/sec
$k_{\phi'}$		Roll trimming integrator gain	0.2 in/rad/sec
$\Delta\delta_{aH}$		Expected hysteresis in cyclic channel	0.1 in
$k_{\phi H\,max}$		Maximum hysteresis compensation gain in roll	2.0
k_r	Yaw SAS Control	Yaw rate damping gain	15.0 in/rad/sec
k_β		Sideslip feedback gain	19.0 in/rad
τ_β		Sideslip filter time constant	0.5 sec
$k_{\delta_r\phi}$		Gain on roll command feed forward into rudder for coordinated turns	2.3 in/rad
$k_{\psi'}$		Yaw trimming integrator gain	0.2 in/rad/sec
$\Delta\delta_{rH}$		Expected hysteresis in differential cyclic channel	0.1 in.
$k_{\psi H\,max}$		Maximum hysteresis compensation gain in yaw	2.0
$k_{\theta_c\delta_{ep}}$	Attitude Command Laws	Pitch command sensitivity on electric stick pitch input	0.145 rad/in.
$k_{\phi_c\delta_{ap}}$		Roll command sensitivity on electric stick roll input	0.298 rad/in.
$k_{\dot\psi_c\delta_{rp}}$		Yaw rate command sensitivity on rudder pedal input	0.128 rad/sec/in.
$\tau_{\dot\psi}$		Yaw rate command time constant in Auto Mode	6.0 sec
$k_{\dot\psi\phi\,max}$		Maximum gain on yaw rate command due to roll command in Auto Mode	10.0 rad/sec/rad
$\dot\psi_{c\,max}$		Maximum yaw rate command in Auto Mode	0.35 rad/sec
k_θ	Attitude Control Laws	Pitch attitude feedback gain	13.5 in/rad.
$\Delta\theta_{max}$		Maximum pitch command increment about trim	0.174 rad
k_ϕ		Roll attitude feedback gain	15.0 in/rad
ϕ_{max}		Roll command limit	0.785 rad
k_ψ		Yaw attitude feedback gain	14.0 in/rad
$k_{\dot\psi}$		Yaw rate command feed forward gain	15.0 in/rad/sec
$k_{\dot V_{cx}\delta_{ep}}$	Velocity I Commands	Forward acceleration command sensitivity on electric stick pitch input	-1.67 ft/sec²/in.
$k_{\dot V_x}$		Forward acceleration feed forward gain	0.6 ft/sec/ft/sec²
$k_{\dot V_{cy}\delta_{ap}}$		Lateral acceleration command sensitivity on electric stick roll input	1.67 ft/sec²/in.
$k_{\dot\xi_c\delta_{ap}}$		Course rate command sensitivity on electric stick roll input	0.0282 rad/sec/in.
$k_{\dot V_y}$		Lateral acceleration feed forward gain	0.6 ft/sec/ft/sec²

TABLE 4. (CONT)

Input	Section	Description	Value and Units
$k_{V_{cx}\delta_{ep}}$	Velocity II Commands	Forward velocity command sensitivity on electric stick pitch input	-6.67 ft/sec/in.
$k_{V'_{cx}}$		Forward velocity command integral bypass gain	0.2 ft/sec/ft
$k_{V_{cy}\delta_{ap}}$		Lateral velocity command sensitivity on electric stick roll input	6.67 ft/sec/in.
$k_{V'_{cy}}$		Lateral velocity command integral bypass gain	0.2 ft/sec/ft
$k_{\xi_c\delta_{ap}}$		Course command sensitivity on electric stick roll input	0.113 rad/in.
$k_{V_{cx}\delta_{eps}}$	Velocity III Commands	Forward velocity command sensitivity on sidearm controller pitch input	-80.0 ft/sec/in.
$k_{V_{cy}\delta_{aps}}$		Lateral velocity command sensitivity on sidearm controller roll input	24.0 ft/sec/in.
$k_{V_{cz}\delta_{cc}}$	Vertical Vel Command	Vertical velocity command sensitivity on collective stick input	-6.25 ft/sec/in.
k_{V_x}	Velocity Control Laws	Forward velocity feedback gain	-0.015 rad/ft/sec
$k_{V'_x}$		Forward velocity integral gain	0.1 rad/ft
k_{V_y}		Lateral velocity feedback gain	0.015 rad/ft/sec
$k_{V'_y}$		Lateral velocity integral gain	0.1 rad/ft
k_{V_z}		Vertical velocity feedback gain	-0.2 in/ft/sec
$k_{V'_z}$		Vertical velocity integral gain	1.0 in/ft
$k_{\delta_c\phi}$		Gain on roll command feed forward into collective	3 in.
τ_{V_a}	Miscellaneous	Airspeed filter time constant	2.0 sec
$\Delta V'_{at}$		Deadzone on speed, V'_a	5.0 ft/sec
V_T		Switch value of V_a'	51.0 ft/sec
$k_{\Delta\delta_{ed}}$	Display Sensitivities	Differential collective pitch rate command error sensitivity on flight director horizontal needle	0.4 in/in.
$k_{\Delta\delta_{ad}}$		Cyclic roll rate command error sensitivity on flight director vertical needle	0.4 in/in.
$k_{\Delta\delta_{cd}}$		Collective command error sensitivity on flight director collective bug	0.5 in/in.
$k_{\Delta\theta_d}$		Pitch attitude command error sensitivity on flight director horizontal needle	2.87 in/rad
$k_{\Delta\phi_d}$		Roll attitude command error sensitivity on flight director vertical needle	1.43 in/rad
$k_{\Delta V_{xd}}$		Forward velocity command error sensitivity on flight director horizontal needle	-0.05 in/ft/sec
$k_{\Delta V_{yd}}$		Lateral velocity command error sensitivity on flight director vertical needle	+0.025 in/ft/sec
$k_{\Delta V_{zd}}$		Vertical velocity command error sensitivity on flight director collective bug	-0.1 in/ft/sec

TABLE 5

VARIABLE OUTPUTS

Output	Description	Destination
δ_{ec}	Differential collective command	EISS differential collective channel
δ_{cc}	Collective command	EISS collective channel
δ_{ac}	Cyclic command	EISS cyclic channel
δ_{rc}	Differential cyclic command	EISS differential cyclic channel
δ_{ed}	Horizontal needle display command	Flight director horizontal needle
δ_{cd}	Collective bug display command	Flight director collective bug
δ_{ad}	Vertical needle display command	Flight director vertical needle

b. Discrete

There are four discrete outputs to the cockpit: (1) the decoded flight control mode, (2) the heading hold indicator, (3) the disengage warning signal, and (4) the flight director status indicator. The decoded flight control mode discrete is to be used to light the flight control mode selector in the cockpit that corresponds to the decoded mode that is actually being used in the flight control system. As discussed previously, this will be equal to the pilot selected mode except in certain cases when the guidance is in the Disengage mode. The heading hold discrete is to be used to light the cockpit heading hold indicator when the flight control system is in the heading hold mode. The disengage warning discrete is to be used to light the cockpit disengage warning light whenever the flight control system disengages itself. As discussed previously, this will occur in the abnormal event that the guidance system is disengaged while the flight control system is in the Auto Mode. The flight director status discrete is to be used to activate the flight director off flags whenever either the guidance or flight control system is in the Disengage Mode.

These discretes must be coded into a single output data word in the airborne computer. This output data word will then be externally decoded as required to accomplish the required cockpit functions. To enable the coding form to be specified, it is assumed that, subject to final decoder specifications: (1) from right to left, the first three bits of this word will be used for the decoded flight control mode discrete, the fourth bit will be used for the heading holding discrete, the fifth bit will be used for the disengage warning discrete, and the sixth bit will be used for the flight director-status discrete and (2) the bit pattern of the output data word will be coded as specified in Table 6. As shown in this table, all of the discretes are independent and; therefore, the bits corresponding to each can be coded independently.

TABLE 6
DISCRETE OUTPUT CODING

	Coded Output Word					
Specifications For Coding	Spares	Flight Director Bit	Disengage Warning Bit	Heading Hold Bit	Flight Control Mode Bits	Cockpit Use
If decoded flight control mode is Disengage					0 0 0	Disengage light on
If decoded flight control mode is SAS					0 0 1	SAS light on
If decoded flight control mode is Attitude I					0 1 0	Attitude I light on
If decoded flight control mode is Attitude II					0 1 1	Attitude II light on
If decoded flight control mode is Velocity I					1 0 0	Velocity I light on
If decoded flight control mode is Velocity II					1 0 1	Velocity II light on
If decoded flight control mode is Velocity III					1 1 0	Velocity III light on
If decoded flight control mode is Auto					1 1 1	Auto light on
If decoded flight control mode is Auto or SAS or if V_a' is high				0		Heading hold light off
If decoded flight control mode is not Auto or SAS and if V_a' is low				1		Heading hold light on
For all conditions other than following condition			0			Disengage warning light off
If decoded flight control mode is Disengage because Disengage mode of guidance is selected while Auto mode of flight control is selected.			1			Disengage warning light on
If decoded flight control or guidance mode is Disengage		0				Flight director off
If decoded flight control and guidance modes are not Disengage		1				Flight director on

III. DIGITAL DESCRIPTION OF FLIGHT CONTROL LAWS

A. GENERAL REQUIREMENTS

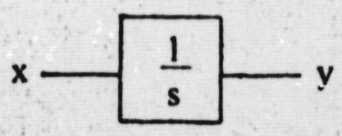
A main program must be provided to control entry to the flight control subroutine and to generate certain inputs needed by the flight control subroutine. Prior to the first entry after execution, the main program must set the previous value of the decoded guidance mode to Disengaged. After the flight control subroutine is finished, it will return to the main program. The main program can then either enter an idle loop or do other computations until it is time to enter the flight control subroutine again. The main program must establish the relative frequencies at which the slow and fast computational loops will be updated. The SAS command laws and the SAS and attitude control laws will be updated at the higher frequency. The attitude and velocity command laws and the velocity control laws will be updated at the lower frequency. In addition, discrete inputs will be sensed and discrete outputs will be generated at the lower frequency. It is recommended that the higher frequency be an integral multiple of the lower frequency and that the update frequencies be 32 times/sec for the fast loop and 8 times/sec for the slow loop. The main program can then set a code NV which can be tested each time the flight control subroutine is entered. If NV = 1, the slow loop will be updated before entering the fast loop. If NV = 0, only the fast loop will be updated. NV must be 1 the first time the flight control subroutine is entered.

The main program must also provide current problem time to the flight control package, either by monitoring a real time clock, counting interrupts, or any other method. Time is needed so that the incremental time between successive updates of each loop can be calculated and provided to the difference equation computations.

B. DIFFERENCE EQUATIONS

Whenever a transfer function in the Laplacian operator, s, domain is encountered in the flight control law specifications, a difference equation derived by Tustin's method should be used as described below.

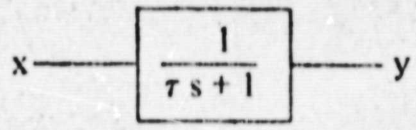
1. Integrator



$$y_i = y_{i-1} + \frac{\Delta t}{2}(x_i + x_{i-1})$$

where
- y_i = current output of integrator
- y_{i-1} = previous output of integrator
- x_i = current input to integrator
- x_{i-1} = previous input to integrator
- Δt = elapsed time between this update and previous update (seconds)

2. Filter



$$y_i = \frac{x_i + x_{i-1} + (a-1)\,y_{i-1}}{a+1}$$

where

a	=	$2\tau/\Delta t$
τ	=	filter time constant (seconds)
Δt	=	elapsed time (seconds)
y_i	=	current output of filter
y_{i-1}	=	previous output of filter
x_i	=	current input to filter
x_{i-1}	=	previous input to filter

C. FLOW DIAGRAMS

A flow diagram of the Flight Control Subroutine is shown in Figure 3. Figures 4 and 5 show the flow diagrams of the Fast Loop Subroutine and Slow Loop Subroutine respectively. Table 7 describes the logic codes used in these flow diagrams.

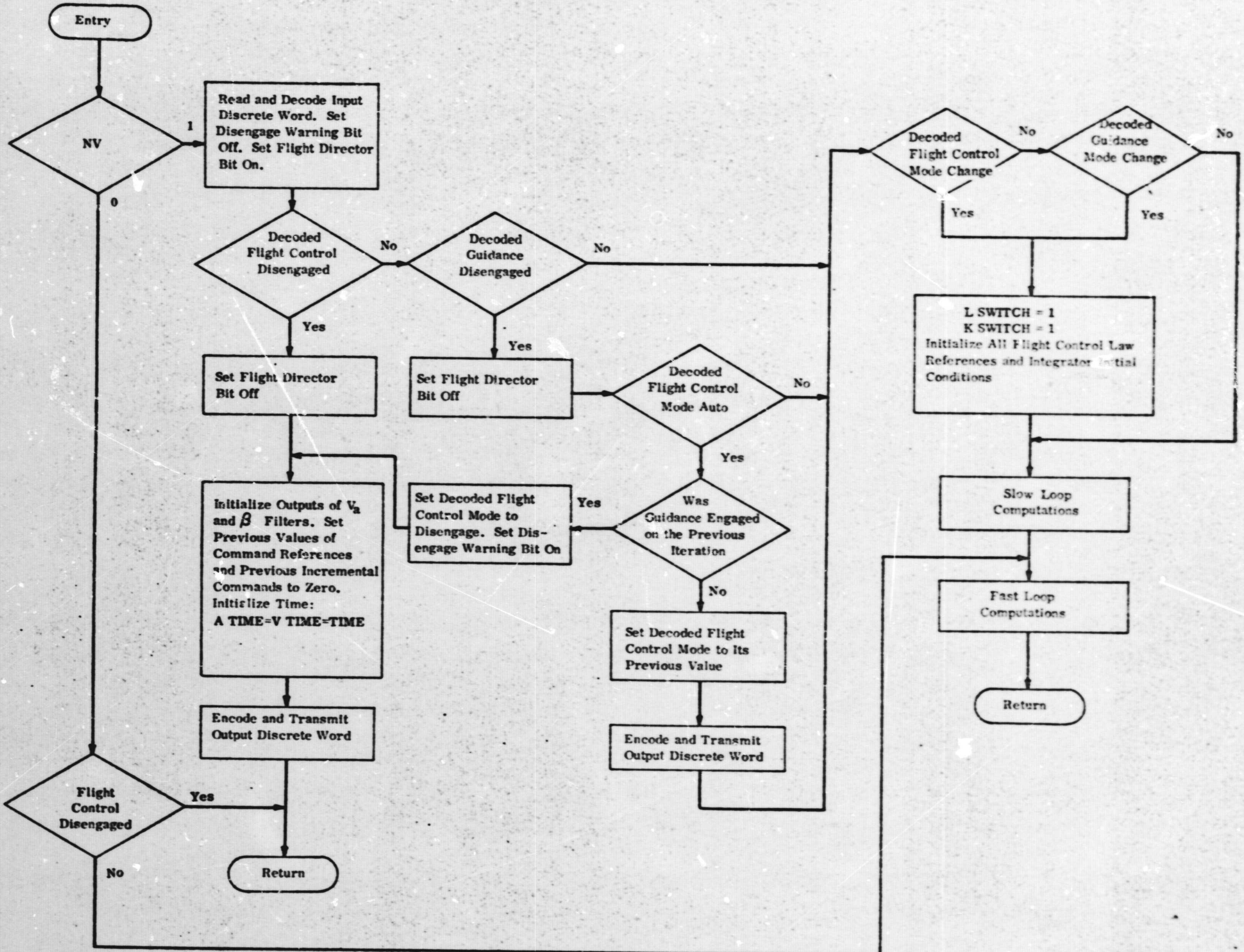


Figure 5. Flight Control Subroutine

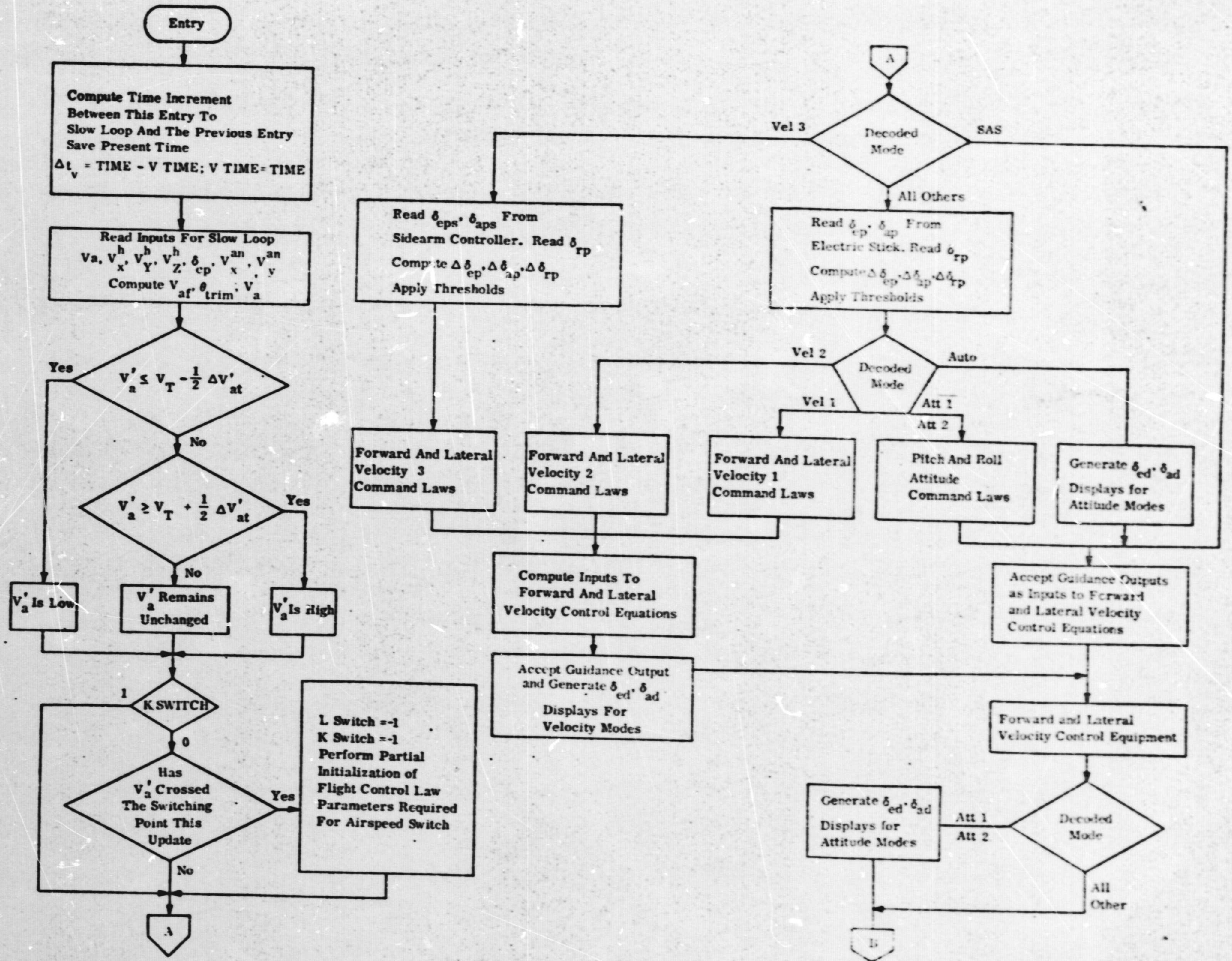


Figure 4. Slow Loop (Sheet 1 of 2)

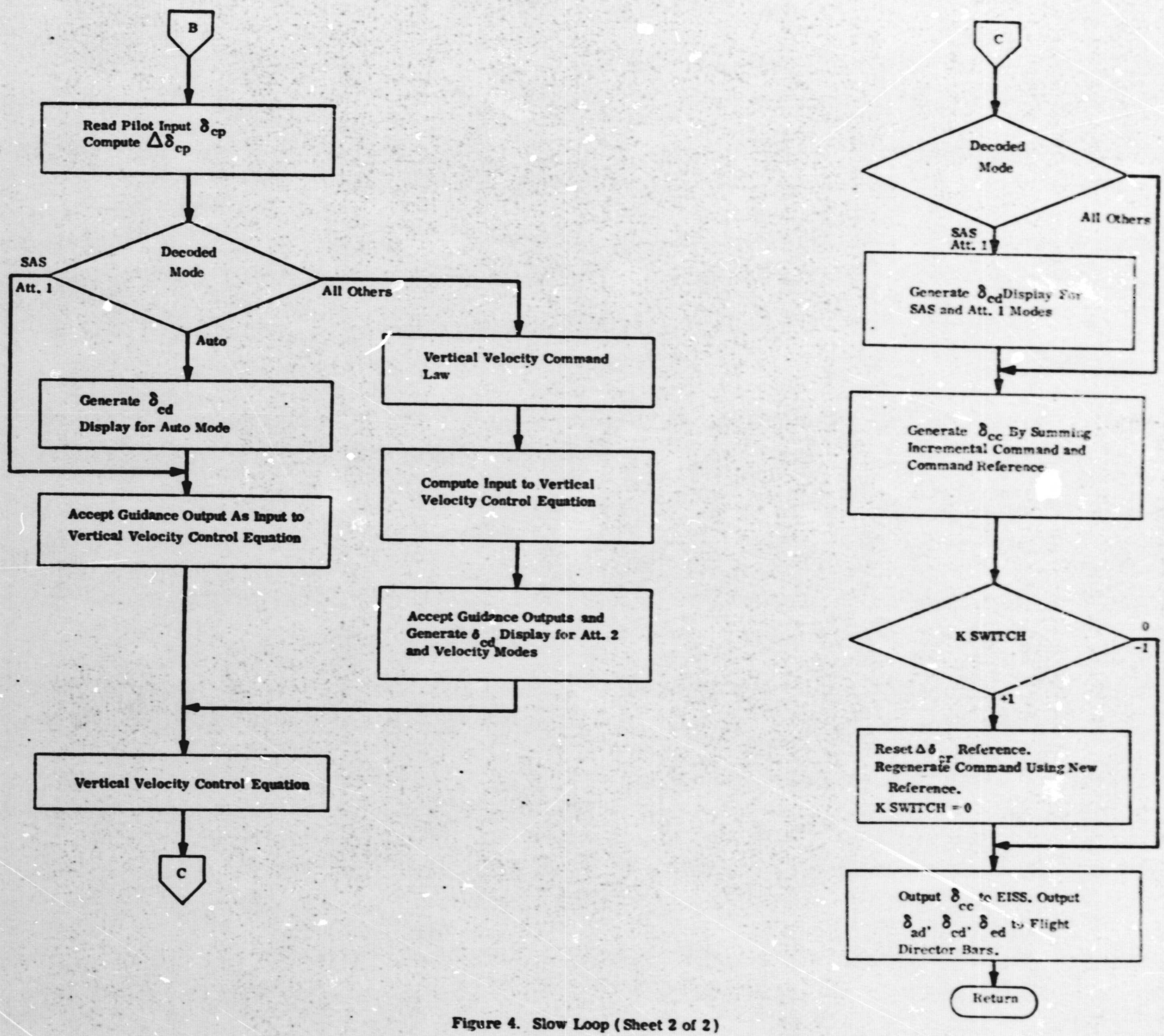


Figure 4. Slow Loop (Sheet 2 of 2)

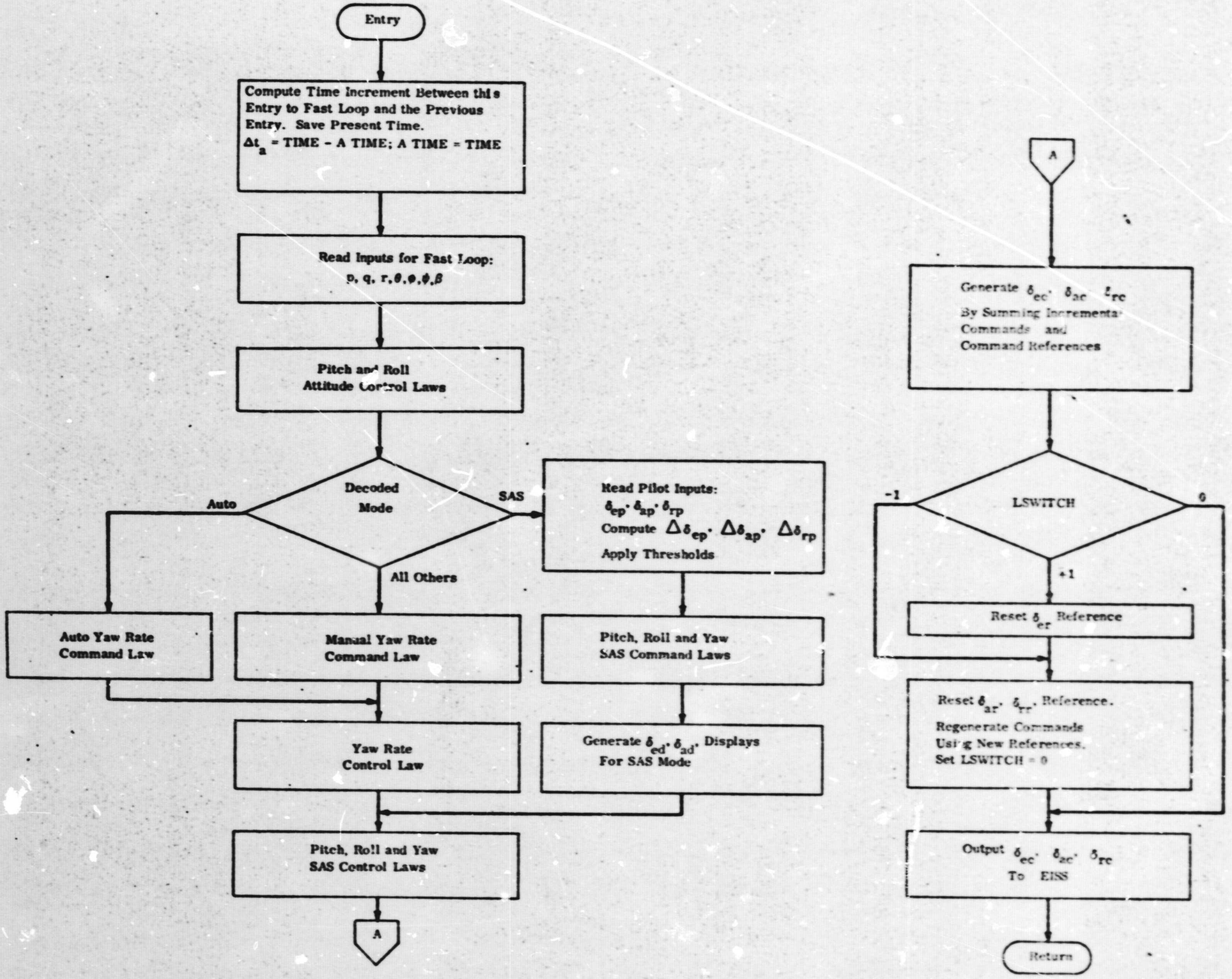


Figure 5. Fast Loop

TABLE 7

DEFINITION OF CODES USED IN FLOW DIAGRAMS

Code	Definition
NV	Set to 1 by main program, if both slow and fast loops are to be entered. Set to 0 if only fast loop is to be entered.
LSWITCH	Fast loop switch indicator. Set to 1 when a complete initialization is required, set to -1 when a partial initialization is required, set to 0 when no initialization is required.
KSWITCH	Slow loop switch indicator. Defined same as LSWITCH.
MODE	Flight control mode as decoded from input discrete word. Disengage, SAS,ATT. 1, ATT. 2, VEL. 1, VEL. 2, VEL. 3, AUTO.
TIME	Current problem time in seconds from main program.
ATIME	Time at the beginning of a fast loop update.
VTIME	Time at the beginning of a slow loop update.
Δt_a	Computation interval between successive updates of the fast loop.
Δt_v	Computation interval between successive updates of the slow loop.