APPENDIX A

F-16 MODEL

This appendix contains the remainder of the data for the F-16 aircraft model given in Chapter 3. The usage of the lookup tables will be made evident by referring to the aircraft model. These data, Appendix B, and the other programs used in this book can be obtained on a floppy disc, at a nominal cost, from Dr. B. L. Stevens, 1051 Park Manor Terr., Marietta, GA 30064.

A.1 Mass Properties

$$\begin{aligned} \text{Weight (lbs)}: \quad & W = 20{,}500 \\ \text{Moment of Intertia (slug-ft}^2\text{)}: \quad & J_{xx} = 9{,}456 \\ & J_{yy} = 55{,}814 \\ & J_{zz} = 63{,}100 \\ & J_{xz} = \phantom{00{,}}982 \end{aligned}$$

A.2 Wing Dimensions

$$\text{Span} = 30 \text{ ft}$$

$$\text{Area} = 300 \text{ ft}^2$$

$$\text{m.a.c} = 11.32 \text{ ft}$$

A.3 Reference CG Location

$$X_{cg} = 0.35\bar{c}$$

Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, Third Edition.
Brian L. Stevens, Frank L. Lewis, Eric N. Johnson.

A.4 Control Surface Actuator Models

	deflection limit	rate limit	time const.
Elevator	$\pm 25.0°$,	$60°/s$,	0.0495 s lag
Ailerons	$\pm 21.5°$,	$80°/s$,	0.0495 s lag
Rudder	$\pm 30.0°$,	$120°/s$,	0.0495 s lag

A.5 Engine Angular Momentum

Assumed fixed at 160 slug-ft^2/s

A.6 Standard Atmosphere Model

```
SUBROUTINE ADC(VT,ALT,AMACH,QBAR)       ! air data computer
DATA R0/2.377E-3/                       ! sea-level density
TFAC = 1.0 - 0.703E-5 * ALT
T      = 519.0 * TFAC                               ! temperature
IF (ALT .GE. 35000.0) T= 390.0
RHO = R0 * (TFAC**4.14)                 ! density
AMACH= VT/SQRT(1.4*1716.3*T)            ! Mach number
QBAR = 0.5*RHO*VT*VT                    ! dynamic pressure
C      PS    = 1715.0 * RHO * T               ! static pressure
RETURN
END
```

A.7 Engine Model

The F-16 engine power response is modeled by a first-order lag (in function PDOT, given below). The rest of the model consists of the throttle gearing (in TGEAR) and the lookup tables for thrust as a function of operating power level, altitude, and Mach (in THRUST). In the thrust lookup tables the rows correspond to a Mach number variation from 0 to 1.0 in increments of 0.2, and the columns correspond to altitudes from 0 to 50,000 ft in increments of 10,000 ft. There is a table for each of the power levels “idle,” “military,” and “maximum.” The accompanying linear interpolation algorithm can extrapolate beyond the boundaries of a table, but the results may not be realistic.

```
FUNCTION TGEAR(THTL) ! Power command v. thtl. relationship
IF(THTL.LE.0.77) THEN
TGEAR = 64.94*THTL
ELSE
TGEAR = 217.38*THTL-117.38
END IF
RETURN
END
FUNCTION PDOT(P3,P1) ! PDOT= rate of change of power
IF (P1.GE.50.0) THEN ! P3= actual power, P1= power command
IF (P3.GE.50.0) THEN
T=5.0
P2=P1
```

```
ELSE
P2=60.0
T=RTAU(P2-P3)
END IF
ELSE
IF (P3.GE.50.0) THEN
T=5.0
P2=40.0
ELSE
P2=P1
T=RTAU(P2-P3)
END IF
END IF
PDOT=T*(P2-P3)
RETURN
END
FUNCTION RTAU(DP)    ! used by function PDOT
IF (DP.LE.25.0) THEN
RTAU=1.0     ! reciprocal time constant
ELSE IF (DP.GE.50.0)THEN
RTAU=0.1
ELSE
RTAU=1.9-.036*DP
END IF
RETURN
END
FUNCTION THRUST(POW,ALT,RMACH)    ! Engine thrust model
 REAL A(0:5,0:5), B(0:5,0:5), C(0:5,0:5)
 DATA A/
+ 1060.0, 670.0, 880.0, 1140.0, 1500.0, 1860.0,
+ 635.0, 425.0, 690.0, 1010.0, 1330.0, 1700.0,
+ 60.0, 25.0, 345.0, 755.0, 1130.0, 1525.0,
+ -1020.0, -710.0, -300.0, 350.0, 910.0, 1360.0,
+ -2700.0, -1900.0, -1300.0, -247.0,     600.0, 1100.0,
+ -3600.0, -1400.0, -595.0, -342.0,      -200.0,      700.0/
C     mil data now
DATA B/
+ 12680.0, 9150.0, 6200.0, 3950.0, 2450.0, 1400.0, + 12680.0,
9150.0, 6313.0, 4040.0, 2470.0, 1400.0, + 12610.0, 9312.0, 6610.0,
4290.0, 2600.0, 1560.0, + 12640.0, 9839.0, 7090.0, 4660.0, 2840.0,
1660.0, + 12390.0, 10176.0, 7750.0, 5320.0, 3250.0, 1930.0,
+ 11680.0, 9848.0, 8050.0, 6100.0, 3800.0, 2310.0/
C      max data now
DATA C/
+ 20000.0, 15000.0, 10800.0, 7000.0, 4000.0, 2500.0,
+ 21420.0, 15700.0, 11225.0, 7323.0, 4435.0, 2600.0,
+ 22700.0, 16860.0, 12250.0, 8154.0, 5000.0, 2835.0,
+ 24240.0, 18910.0, 13760.0, 9285.0, 5700.0, 3215.0,
+ 26070.0, 21075.0, 15975.0, 11115.0, 6860.0, 3950.0,
+ 28886.0, 23319.0, 18300.0, 13484.0, 8642.0,  5057.0/
C
H = .0001*ALT
I = INT(H)
IF(I.GE.5)I=4
```

```
DH= H-FLOAT(I)
RM= 5.0*RMACH
M = INT(RM)
IF(M.GE.5)M=4
DM= RM-FLOAT(M)
CDH=1.0-DH
S= B(I,M) *CDH + B(I+1,M) *DH
T= B(I,M+1)*CDH + B(I+1,M+1)*DH
TMIL= S + (T-S)*DM
IF( POW .LT. 50.0 ) THEN
S= A(I,M) *CDH + A(I+1,M) *DH
T= A(I,M+1)*CDH + A(I+1,M+1)*DH
TIDL= S + (T-S)*DM
THRUST=TIDL+(TMIL-TIDL)*POW*.02
ELSE
S= C(I,M) *CDH + C(I+1,M) *DH
T= C(I,M+1)*CDH + C(I+1,M+1)*DH
TMAX= S + (T-S)*DM
THRUST=TMIL+(TMAX-TMIL)*(POW-50.0)*.02
END IF
RETURN
END
```

A.8 Aerodynamic Data

The aerodynamic data tables and associated interpolation algorithms, given below, will provide values for the body-axis dimensionless aerodynamic coefficients of the F-16 model at arbitrary values of the independent variables. The angle-of-attack range of the tables is from $-10°$ to $45°$ in $5°$ increments, and the sideslip angle range is from $-30°$ to $30°$ in either $5°$ or $10°$ increments. The given interpolation algorithm interpolates linearly between the data points; it will extrapolate beyond the table boundaries, but the results may be unrealistic.

```
 SUBROUTINE DAMP(ALPHA, D) ! various damping coefficients
 REAL A(-2:9,9),D(9)
 DATA A/
&  -.267, -.110,  .308,   1.34,  2.08,  2.91,  2.76,
&   2.05,  1.50,  1.49,   1.83,  1.21,
&   .882,  .852,  .876,   .958,  .962,  .974,  .819,
&   .483,  .590,  1.21,  -.493, -1.04,
&  -.108, -.108, -.188,   .110,  .258,  .226,  .344,
&   .362,  .611,  .529,   .298, -2.27,
&  -8.80, -25.8, -28.9,  -31.4, -31.2, -30.7, -27.7,
&  -28.2, -29.0, -29.8,  -38.3, -35.3,
&  -.126, -.026,  .063,   .113,  .208,  .230,  .319,
&   .437,  .680,  .100,   .447, -.330,
&  -.360, -.359, -.443,  -.420, -.383, -.375, -.329,
&  -.294, -.230, -.210,  -.120, -.100,
&  -7.21, -.540, -5.23,  -5.26, -6.11, -6.64, -5.69,
&  -6.00, -6.20, -6.40,  -6.60, -6.00,
&  -.380, -.363, -.378,  -.386, -.370, -.453, -.550,
&  -.582, -.595, -.637,  -1.02, -.840,
```

```
&   .061,  .052,  .052,  -.012, -.013, -.024,  .050,
&   .150,  .130,  .158,   .240,  .150/
C
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
DO 1, I= 1,9
1      D(I)= A(K,I) + ABS(DA) * (A(L,I) - A(K,I))
END
C
C D1= CXq; D2= CYr; D3= CYp; D4= CZq; D5= Clr; D6= Clp
C D7= Cmq; D8= Cnr; D9= Cnp
FUNCTION CX(ALPHA,EL) ! x-axis aerodynamic force coeff.
REAL A(-2:9,-2:2)
DATA A/
& -.099, -.081, -.081, -.063, -.025, .044, .097,
&  .113,  .145,  .167,  .174,  .166,
& -.048, -.038, -.040, -.021,  .016, .083, .127,
&  .137,  .162,  .177,  .179,  .167,
& -.022, -.020, -.021, -.004,  .032, .094, .128,
&  .130,  .154,  .161,  .155,  .138,
& -.040, -.038, -.039, -.025,  .006, .062, .087,
&  .085,  .100,  .110,  .104,  .091,
& -.083, -.073, -.076, -.072, -.046, .012, .024,
&  .025,  .043,  .053,  .047,  .040/
C
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= EL/12.0
M= INT(S)
IF(M .LE. -2) M= -1
IF(M .GE. 2) M= 1
DE= S - FLOAT(M)
N= M + INT( SIGN(1.1,DE) )
T= A(K,M)
U= A(K,N)
V= T + ABS(DA) * (A(L,M) - T)
W= U + ABS(DA) * (A(L,N) - U)
CX= V + (W-V) * ABS(DE)
RETURN
END
FUNCTION CY(BETA,AIL,RDR) ! sideforce coefficient
CY= -.02*BETA + .021*(AIL/20.0) + .086*(RDR/30.0)
END
FUNCTION CZ(ALPHA,BETA,EL) ! z-axis force coeff.
REAL A(-2:9)
DATA A/ .770, .241, -.100, -.416, -.731, -1.053,
```

```
& -1.366, -1.646, -1.917, -2.120, -2.248, -2.229/
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= A(K) + ABS(DA) * (A(L) - A(K))
CZ= S*(1-(BETA/57.3)**2) - .19*(EL/25.0)
END
FUNCTION CM(ALPHA,EL) ! pitching moment coeff.
REAL A(-2:9,-2:2)
DATA A/
&  .205,  .168,  .186,  .196,  .213,  .251,  .245,
&  .238,  .252,  .231,  .198,  .192,
&  .081,  .077,  .107,  .110,  .110,  .141,  .127,
& .119,   .133,  .108,  .081,  .093,
& -.046, -.020, -.009, -.005, -.006,  .010,  .006,
& -.001, .014,  .000, -.013,  .032,
& -.174, -.145, -.121, -.127, -.129, -.102, -.097,
& -.113, -.087, -.084, -.069, -.006,
& -.259, -.202, -.184, -.193, -.199, -.150, -.160,
& -.167, -.104, -.076, -.041, -.005/
C      SAME INTERPOLATION AS CX ********************
C
FUNCTION CL(ALPHA,BETA) ! rolling moment coeff.
REAL A(-2:9,0:6)
DATA A/12*0,
& -.001, -.004, -.008, -.012, -.016, -.019, -.020,
& -.020, -.015, -.008, -.013, -.015,
& -.003, -.009, -.017, -.024, -.030, -.034, -.040,
& -.037, -.016, -.002, -.010, -.019,
& -.001, -.010, -.020, -.030, -.039, -.044, -.050,
& -.049, -.023, -.006, -.014, -.027,
&  .000, -.010, -.022, -.034, -.047, -.046, -.059,
& -.061, -.033, -.036, -.035, -.035,
&  .007, -.010, -.023, -.034, -.049, -.046, -.068,
& -.071, -.060, -.058, -.062, -.059,
&  .009, -.011, -.023, -.037, -.050, -.047, -.074,
& -.079, -.091, -.076, -.077, -.076/
C
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= .2* ABS(BETA)
M= INT(S)
IF(M .EQ. 0) M= 1
IF(M .GE. 6) M= 5
DB= S - FLOAT(M)
```

```
N= M + INT( SIGN(1.1,DB) )
T= A(K,M)
U= A(K,N)
V= T + ABS(DA) * (A(L,M) - T)
W= U + ABS(DA) * (A(L,N) - U)
DUM= V + (W-V) * ABS(DB)
CL= DUM + SIGN(1.0, BETA)
RETURN
END
FUNCTION CN(ALPHA,BETA)   ! yawing moment coeff.
REAL A(-2:9,0:6)
DATA A/12 *0,
& .018, .019,  .018,  .019,  .019, .018, .013,
& .007, .004, -.014, -.017, -.033,
& .038, .042,  .042,  .042,  .043, .039, .030,
& .017, .004, -.035, -.047, -.057,
& .056, .057,  .059,  .058,  .058, .053, .032,
& .012, .002, -.046, -.071, -.073,
& .064, .077,  .076,  .074,  .073, .057, .029,
& .007, .012, -.034, -.065, -.041,
& .074, .086,  .093,  .089,  .080, .062, .049,
& .022, .028, -.012, -.002, -.013,
& .079, .090,  .106,  .106,  .096, .080, .068,
& .030, .064,  .015,  .011, -.001/
C NOW USE SAME INTERPOLATION AS CL ********************
C
FUNCTION DLDA(ALPHA,BETA) ! rolling mom. due to ailerons
REAL A(-2:9,-3:3)
DATA A/-.041, -.052, -.053, -.056, -.050, -.056, -.082,
&  -.059,  -.042,  -.038,  -.027,  -.017,
&  -.041,  -.053,  -.053,  -.053,  -.050,  -.051,  -.066,
&  -.043,  -.038,  -.027,  -.023,  -.016,
&  -.042,  -.053,  -.052,  -.051,  -.049,  -.049,  -.043,
&  -.035,  -.026,  -.016,  -.018,  -.014,
&  -.040,  -.052,  -.051,  -.052,  -.048,  -.048,  -.042,
&  -.037,  -.031,  -.026,  -.017,  -.012,
&  -.043,  -.049,  -.048,  -.049,  -.043,  -.042,  -.042,
&  -.036,  -.025,  -.021,  -.016,  -.011,
&  -.044,  -.048,  -.048,  -.047,  -.042,  -.041,  -.020,
&  -.028,  -.013,  -.014,  -.011,  -.010,
&  -.043,  -.049,  -.047,  -.045,  -.042,  -.037,  -.003,
&  -.013,  -.010,  -.003,  -.007,  -.008/
S= 0.2 * ALPHA
K= INT(S)
IF(K .LE. -2) K= -1
IF(K .GE. 9) K= 8
DA= S - FLOAT(K)
L = K + INT( SIGN(1.1,DA) )
S= 0.1 * BETA
M= INT(S)
IF(M .eq. -3) M= -2
IF(M .GE. 3) M= 2
```

```
DB= S - FLOAT(M)
N= M + INT( SIGN(1.1,DB) )
T= A(K,M)
U= A(K,N)
V= T + ABS(DA) * (A(L,M) - T)
W= U + ABS(DA) * (A(L,N) - U)
DLDA= V + (W-V) * ABS(DB)
RETURN
END
FUNCTION DLDR(ALPHA,BETA) ! rolling moment due to rudder
REAL A(-2:9,-3:3) ! use same interpolation as DLDA
DATA A/ .005, .017, .014, .010, -.005, .009, .019,
& .005, -.000, -.005, -.011, .008,
& .007,  .016,  .014,  .014, .013,  .009,  .012,
& .005,  .000,  .004,  .009, .007,
& .013,  .013,  .011,  .012, .011,  .009,  .008,
& .005, -.002,  .005,  .003, .005,
& .018,  .015,  .015,  .014, .014,  .014,  .014,
& .015,  .013,  .011,  .006, .001,
& .015,  .014,  .013,  .013, .012,  .011,  .011,
& .010,  .008,  .008,  .007, .003,
& .021,  .011,  .010,  .011, .010,  .009,  .008,
& .010,  .006,  .005,  .000, .001,
& .023,  .010,  .011,  .011, .011,  .010,  .008,
& .010,  .006,  .014,  .020, .000/
C
FUNCTION DNDA(ALPHA,BETA) ! yawing moment due to ailerons
REAL A(-2:9,-3:3) ! use same interpolation as DLDA *
DATA A/ .001, -.027, -.017, -.013, -.012, -.016,  .001,
&  .017,   .011,   .017,   .008,   .016,
&  .002,  -.014,  -.016,  -.016,  -.014,   -.019,   -.021,
&  .002,   .012,   .015,   .015,   .011,
& -.006,  -.008,  -.006,  -.006,  -.005,   -.008,  -.005,
&  .007,   .004,   .007,   .006,   .006,
& -.011,  -.011,  -.010,  -.009,  -.008,   -.006,   .000,
&  .004,   .007,   .010,   .004,   .010,
& -.015,  -.015,  -.014,  -.012,  -.011,   -.008,   -.002,
&  .002,   .006,   .012,   .011,   .011,
& -.024,  -.010,  -.004,  -.002,  -.001,   .003,    .014,
&  .006,  -.001,   .004,   .004,   .006,
& -.022,   .002,  -.003,  -.005,  -.003,   -.001,  -.009,
& -.009,  -.001,   .003,  -.002,   .001/
C
FUNCTION DNDR(ALPHA,BETA) ! yawing moment due to rudder
REAL A(-2:9,-3:3)
DATA A/ -.018, -.052, -.052, -.052, -.054, -.049, -.059,
&  -.051,  -.030,  -.037,  -.026,  -.013,
&  -.028,  -.051,  -.043,  -.046,  -.045,  -.049,  -.057,
&  -.052,  -.030,  -.033,  -.030,  -.008,
&  -.037,  -.041,  -.038,  -.040,  -.040,  -.038,  -.037,
&  -.030,  -.027,  -.024,  -.019,  -.013,
&  -.048,  -.045,  -.045,  -.045,  -.044,  -.045,  -.047,
```

```
&  -.048,  -.049,  -.045,  -.033,  -.016,
&  -.043,  -.044,  -.041,  -.041,  -.040,  -.038,  -.034,
&  -.035,  -.035,  -.029,  -.022,  -.009,
&  -.052,  -.034,  -.036,  -.036,  -.035,  -.028,  -.024,
&  -.023,  -.020,  -.016,  -.010,  -.014,
&  -.062,  -.034,  -.027,  -.028,  -.027,  -.027,  -.023,
&  -.023,  -.019,  -.009,  -.025,  -.010/
C NOW USE SAME INTERPOLATION AS DLDA *******************
```