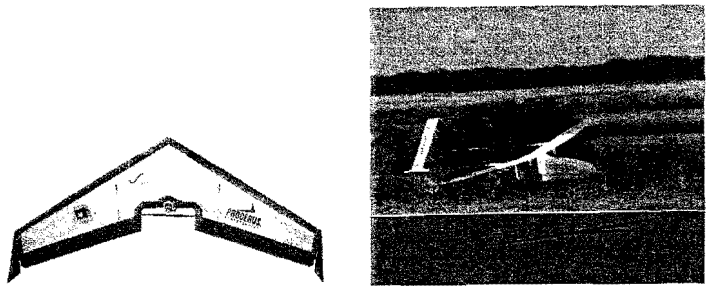
```
55        real_T *x0 = ssGetContStates(S);
56        x0[0] = 0.0;
57        x0[1] = 0.0;
58    }
59
60    /* Function: mdlOutputs
61     *      output function
62     */
63    static void mdlOutputs(SimStruct *S, int_T tid)
64    {
65        real_T*y    = ssGetOutputPortRealSignal(S,0);
66        real_T*x    = ssGetContStates(S);
67        InputRealPtrsType uPtrs
68         = ssGetInputPortRealSignalPtrs(S,0);
69
70        UNUSED_ARG(tid); /* not used */
71        const real_T *wn   = wn_PARAM(S);
72        y[0] = wn[0]*wn[0]*x[1];
73    }
74
75    #define MDL_DERIVATIVES
76    /* Function: mdlDerivatives
77     *      Calculate state-space derivatives
78     */
79    static void mdlDerivatives(SimStruct *S)
80    {
81        real_T *dx   = ssGetdX(S);
82        real_T *x    = ssGetContStates(S);
83        InputRealPtrsType uPtrs
84         = ssGetInputPortRealSignalPtrs(S,0);
85
86        const real_T *zeta = zeta_PARAM(S);
87        const real_T  *wn   = wn_PARAM(S);
88        dx[0] = -2*zeta[0]*wn[0]*x[0] - wn[0]*wn[0]*x[1] + U(0);
89        dx[1] = x[0];
90    }
91
92    /* Function: mdlTerminate
93     *    No termination needed.
94     */
95    static void mdlTerminate(SimStruct *S)
96    {
97        UNUSED_ARG(S); /* unused input argument */
98    }
99
100   #ifdef  MATLAB_MEX_FILE
101   #include "simulink.c"
102   #else
103   #include "cg_sfun.h"
104   #endif
```

APPENDIX E

Airframe Parameters

This appendix gives the physical parameters for two small unmanned aircraft: a Zagi flying wing, shown in figure E.1(a), and the Aerosonde UAV, shown in figure E.1(b). Mass, geometry, propulsion, and aerodynamic parameters for the Zagi flying wing are given in table E.1. Mass, geometry, propulsion, and aerodynamic parameters for the Aerosonde are given in table E.2 [129].



(a) (b)

Figure E.1 (a) The Zagi airframe. (b) The Aerosonde UAV.

E.1 Zagi Flying Wing

TABLE E.1
Parameters for a Zagi flying wing

		Longitudinal			
Parameter	*Value*	*Coef.*	*Value*	*Lateral Coef.*	*Value*
m	1.56 kg	C_{L_0}	0.09167	C_{Y_0}	0
J_x	0.1147 kg m^2	C_{D_0}	0.01631	C_{l_0}	0
J_y	0.0576 kg m^2	C_{m_0}	−0.02338	C_{n_0}	0
J_z	0.1712 kg m^2	C_{L_α}	3.5016	C_{Y_β}	−0.07359
J_{xz}	0.0015 kg m^2	C_{D_α}	0.2108	C_{l_β}	−0.02854
S	0.2589 m^2	C_{m_α}	−0.5675	C_{n_β}	−0.00040
b	1.4224 m	C_{L_q}	2.8932	C_{Y_p}	0

TABLE E.1
Continued.

		Longitudinal			
Parameter	*Value*	*Coef.*	*Value*	*Lateral Coef.*	*Value*
c	0.3302 m	C_{D_q}	0	C_{l_p}	−0.3209
S_{prop}	0.0314 m^2	C_{m_q}	−1.3990	C_{n_p}	−0.01297
ρ	1.2682 kg/m^3	$C_{L_{\delta_e}}$	0.2724	C_{Y_r}	0
k_{motor}	20	$C_{D_{\delta_e}}$	0.3045	C_{l_r}	0.03066
k_{T_p}	0	$C_{m_{\delta_e}}$	−0.3254	C_{n_r}	−0.00434
k_Ω	0	C_{prop}	1.0	$C_{Y_{\delta_a}}$	0
e	0.9	M	50	$C_{l_{\delta_a}}$	0.1682
		α_0	0.4712	$C_{n_{\delta_a}}$	−0.00328
		ϵ	0.1592		
		C_{D_p}	0.0254		

E.2 Aerosonde UAV

TABLE E.2
Aerodynamic coefficients for the Aerosonde UAV

		Longitudinal		Lateral	
Parameter	*Value*	*Coef.*	*Value*	*Coef.*	*Value*
m	13.5 kg	C_{L_0}	0.28	C_{Y_0}	0
J_x	0.8244 kg-m^2	C_{D_0}	0.03	C_{l_0}	0
J_y	1.135 kg-m^2	C_{m_0}	−0.02338	C_{n_0}	0
J_z	1.759 kg-m^2	C_{L_α}	3.45	C_{Y_β}	−0.98
J_{xz}	0.1204 kg-m^2	C_{D_α}	0.30	C_{l_β}	−0.12
S	0.55 m^2	C_{m_α}	−0.38	C_{n_β}	0.25
b	2.8956 m	C_{L_q}	0	C_{Y_p}	0
c	0.18994 m	C_{D_q}	0	C_{l_p}	−0.26
S_{prop}	0.2027 m^2	C_{m_q}	−3.6	C_{n_p}	0.022
ρ	1.2682 kg/m^3	$C_{L_{\delta_e}}$	−0.36	C_{Y_r}	0
k_{motor}	80	$C_{D_{\delta_e}}$	0	C_{l_r}	0.14
k_{T_p}	0	$C_{m_{\delta_e}}$	−0.5	C_{n_r}	−0.35
k_Ω	0	C_{prop}	1.0	$C_{Y_{\delta_a}}$	0
e	0.9	M	50	$C_{l_{\delta_a}}$	0.08
		α_0	0.4712	$C_{n_{\delta_a}}$	0.06
		ϵ	0.1592	$C_{Y_{\delta_r}}$	−0.17
		C_{D_p}	0.0437	$C_{l_{\delta_r}}$	0.105
		$C_{n_{\delta_r}}$	−0.032		

APPENDIX F

Trim and Linearization in Simulink

F.1 Using the Simulink `trim` Command

Simulink provides a built-in routine for computing trim conditions for general Simulink diagrams. Useful instructions for using this command can be obtained by typing `help trim` at the Matlab prompt. As described in section 5.3, given the parameters V_a^*, γ^*, and R^*, the objective is to find x^* and u^* such that $\dot{x}^* = f(x^*, u^*)$ where x and u are defined in equations (5.17) and (5.18), $\dot{x}^*$ is given by equation (5.21), and where $f(x, u)$ is defined by the right-hand side of equations (5.1)–(5.12).

The format for the Simulink `trim` command is

```
[X,U,Y,DX]=TRIM('SYS',X0,U0,Y0,IX,IU,IY,DX0,IDX),
```

where `X` is the computed trim state x^*, `U` is the computed trim input u^*, `Y` is the computed trim output y^*, and `DX` is the computed derivative of the state $\dot{x}^*$. The system is specified by the Simulink model `SYS.mdl`, where the state of the model is defined by the union of all of the states in the subsystems of `SYS.mdl` and the inputs and outputs are defined by Simulink `Inports` and `Outports` respectively. Figure F.1 shows a Simulink model that could be used to compute aircraft trim. The inputs to the system as specified by the four `Inports` are the servo commands `delta_e`, `delta_a`, `delta_r`, and `delta_t`. The states of this block are the states of the Simulink model, which in our case are $\xi = (p_n, p_e, p_d, u, v, w, \phi, \theta, \psi, p, q, r)^\top$, and the outputs are specified by the three `Outports` as the airspeed V_a, the angle of attack α, and the sideslip angle β. Our purpose in specifying V_a, α, and β as outputs is that we wish to force the Simulink `trim` command to maintain $V_a = V_a^*$ and α^* is often a quantity of interest. If we have access to a rudder, then we can enforce a coordinated turn by forcing the trim command to maintain $\beta^* = 0$. If a rudder is not available, then β will not necessarily be zero in a turn.

Since the trim calculation problem reduces to solving a system of nonlinear algebraic equations, which may have many solutions, the Simulink `trim` command requires that initial guesses for the state `X0`, input `U0`, output `Y0`, and derivative of the state `DX0` be specified. If we know from the outset, that some of the states, inputs, outputs, or derivatives of states are fixed and specified by their initial conditions, then those constraints are indicated by the index vectors `IX`, `IU`, `IY`, and `IDX`.