```
function [control_gains,
linear_terms]=CalculateControlGainsSimpleSLC_Nondim_Ttwistor(aircraft_paramet
ers, trim_definition, trim_variables)
%
% function [control_gains,
linear_terms]=CalculateControlGainsSimpleSLC_Nondim(aircraft_parameters)
%
% This function determines the control gains that are required for
% SimpleSLCAutopilot.m. It assumes that the aircraft_parameters structure
% has the NONDIMENSIONAL aircraft coefficients.
%

g=aircraft_parameters.g;
control_gains.g=g;

density = stdatmo(trim_definition(3));
Va = trim_definition(1);

%%%%%%%%
%%%% Control parameters
%%%%%%%%
control_gains.max_roll      = 45*pi/180;
control_gains.max_roll_rate = 45*pi/180;
control_gains.max_pitch     = 30*pi/180;

control_gains.max_da        = 30*pi/180;
control_gains.max_dr        = 30*pi/180;
control_gains.max_de        = 20*pi/180;

%%%%%%%%
% roll hold gains
zeta_roll = 1.2; %%%<---------- STUDENT SELECT
e_phi_max = control_gains.max_roll; % used by saturation method to select
proportional gain, assume never give step commanded of greater than full
roll limit


QS = 0.5*density*Va*Va*aircraft_parameters.S;

a_phi1 =
-QS*aircraft_parameters.b*aircraft_parameters.Clp*aircraft_parameters.b/
(2*Va);
a_phi2 = QS*aircraft_parameters.b*aircraft_parameters.Clda;

control_gains.Kp_roll = 3*(control_gains.max_da/e_phi_max)*sign(a_phi2);
wn_roll = sqrt(abs(a_phi2*control_gains.Kp_roll));

control_gains.Kd_roll = (2*zeta_roll*wn_roll - a_phi1)/a_phi2;
control_gains.Ki_roll = -0.6; %%%<---------- STUDENT SELECT

den_phi2 = [1 a_phi1+a_phi2*control_gains.Kd_roll
a_phi2*control_gains.Kp_roll a_phi2*control_gains.Ki_roll];
```

```
% damp(roots(den_phi2))
% stepinfo(roots(den_phi2))


%%%%%%%
% course hold gains
wn_chi = (1/50)*wn_roll; %%%<---------- STUDENT SELECT
zeta_chi = 1.2; %%%<---------- STUDENT SELECT

control_gains.Kp_course = 2*zeta_chi*wn_chi*Va/g;
control_gains.Ki_course = wn_chi*wn_chi*Va/g;

%%%%%%%%%
% sideslip hold gains
e_beta_max = 2*control_gains.max_roll; % used by saturation method to select
proportional gain, assume never give step commanded of greater than full
roll limit
zeta_beta = 5; %%%<---------- STUDENT SELECT

a_beta1 = -density*Va*aircraft_parameters.S*aircraft_parameters.CYbeta/
(2*aircraft_parameters.m);
a_beta2 = density*Va*aircraft_parameters.S*aircraft_parameters.CYdr/
(2*aircraft_parameters.m);

control_gains.Kp_beta = (control_gains.max_dr / e_beta_max)*sign(a_beta2);
control_gains.Ki_beta = (1/a_beta2)*((a_beta1 +
a_beta2*control_gains.Kp_beta)/(2*zeta_beta))^2;
control_gains.Kd_beta = 1; %%%<---------- STUDENT SELECT NOT USED IN
SIDESLIP HOLD AUTOPILOT

wn_beta = sqrt(a_beta2*control_gains.Ki_beta);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Longitudinal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%
% pitch hold
e_theta_max = 2*control_gains.max_pitch; % used by saturation method to
select proportional gain, assume never give step commanded of greater than
full pitch limit
zeta_pitch = 1.5; %%%<---------- STUDENT SELECT

a_theta1 =
-density*Va*aircraft_parameters.c*aircraft_parameters.S*aircraft_parameters.C
mq*aircraft_parameters.c/(4*aircraft_parameters.Iy);
a_theta2 =
-density*Va*Va*aircraft_parameters.c*aircraft_parameters.S*aircraft_parameter
s.Cmalpha/(2*aircraft_parameters.Iy);
a_theta3 =
density*Va*Va*aircraft_parameters.c*aircraft_parameters.S*aircraft_parameters
.Cmde/(2*aircraft_parameters.Iy);
```

```
control_gains.Kp_pitch = (control_gains.max_de / e_theta_max)*sign(a_theta3);
wn_pitch = sqrt(a_theta2 + abs(control_gains.Kp_pitch*a_theta3));

control_gains.Kd_pitch = (2*zeta_pitch*wn_pitch - a_theta1)/a_theta3;

%%%%%%%%
% height hold
Kpitch_DC = a_theta3*control_gains.Kp_pitch/
(a_theta3*control_gains.Kp_pitch+a_theta2);
wn_height = (1/100)*wn_pitch; %%%<---------- STUDENT SELECT
zeta_height = 0.8; %%%<---------- STUDENT SELECT

control_gains.Kp_height = 2*zeta_height*wn_height/(Kpitch_DC*Va);
control_gains.Ki_height = wn_height*wn_height/(Kpitch_DC*Va);

%%%%%%%%
% height control state machine parameters
control_gains.Kpitch_DC = Kpitch_DC;
control_gains.takeoff_height = 1675; %%%<---------- FREW SELECTED
control_gains.takeoff_pitch = 6*pi/180; %%%<---------- FREW SELECTED
control_gains.height_hold_limit = 25; %%%<---------- FREW SELECTED
control_gains.climb_throttle = 0.75; %%%<---------- FREW SELECTED

%%%%%%%%%
% airspeed (from pitch)
alpha = trim_variables(1);
de = trim_variables(2);
dt = trim_variables(3);

CLtrim = aircraft_parameters.CL0 + aircraft_parameters.CLalpha*alpha +
aircraft_parameters.CLde*de;
CDtrim = aircraft_parameters.CDmin+aircraft_parameters.K*(CLtrim-
aircraft_parameters.CLmin)^2;

dCDdCL = 2*aircraft_parameters.K*(CLtrim-aircraft_parameters.CLmin);
CDalpha = dCDdCL*aircraft_parameters.CLalpha;
```

For new engine model

```
a_v1 = (density*Va*aircraft_parameters.S/aircraft_parameters.m)*(CDtrim) -
...

density*aircraft_parameters.Sprop*aircraft_parameters.Cprop*(2*(dt-1)*Va
+ (aircraft_parameters.kmotor-2*aircraft_parameters.kmotor*dt))/
aircraft_parameters.m;

a_v2 =
density*aircraft_parameters.Sprop*aircraft_parameters.Cprop*((2*dt-1)*Va*Va
+ (aircraft_parameters.kmotor-4*aircraft_parameters.kmotor*dt)*Va + ...
       2*aircraft_parameters.kmotor*aircraft_parameters.kmotor*dt)/
aircraft_parameters.m;

wn_airspeed = (1/50)*wn_pitch; %%%<---------- STUDENT SELECT
```

```
zeta_airspeed = 2; %%%<---------- STUDENT SELECT

control_gains.Kp_speed_pitch = (a_v1 - 2*zeta_airspeed*wn_airspeed)/
(Kpitch_DC*aircraft_parameters.g);
control_gains.Ki_speed_pitch = -wn_airspeed*wn_airspeed/
(Kpitch_DC*aircraft_parameters.g);

%%%%%%%%%
% airspeed (from throttle)
wn_airspeed = (1/50)*wn_pitch; %%%<---------- STUDENT SELECT
zeta_airspeed = 2; %%%<---------- STUDENT SELECT
control_gains.Kp_speed_throttle = (2*zeta_airspeed*wn_airspeed-a_v1)/a_v2;
control_gains.Ki_speed_throttle = wn_airspeed*wn_airspeed/a_v2;

%%%%%%%%%
% Linear terms for simulations
linear_terms.a_phi1  = a_phi1;
linear_terms.a_phi2  = a_phi2;

linear_terms.a_beta1  = a_beta1;
linear_terms.a_bet2  = a_beta2;

linear_terms.a_theta1  = a_theta1;
linear_terms.a_theta2  = a_theta2;
linear_terms.a_theta3  = a_theta3;

linear_terms.a_v1  = a_v1;
linear_terms.a_v2  = a_v2;
```

```
Not enough input arguments.

Error in CalculateControlGainsSimpleSLC_Nondim_Ttwistor (line 10)
g=aircraft_parameters.g;
```

Published with MATLAB® R2023b