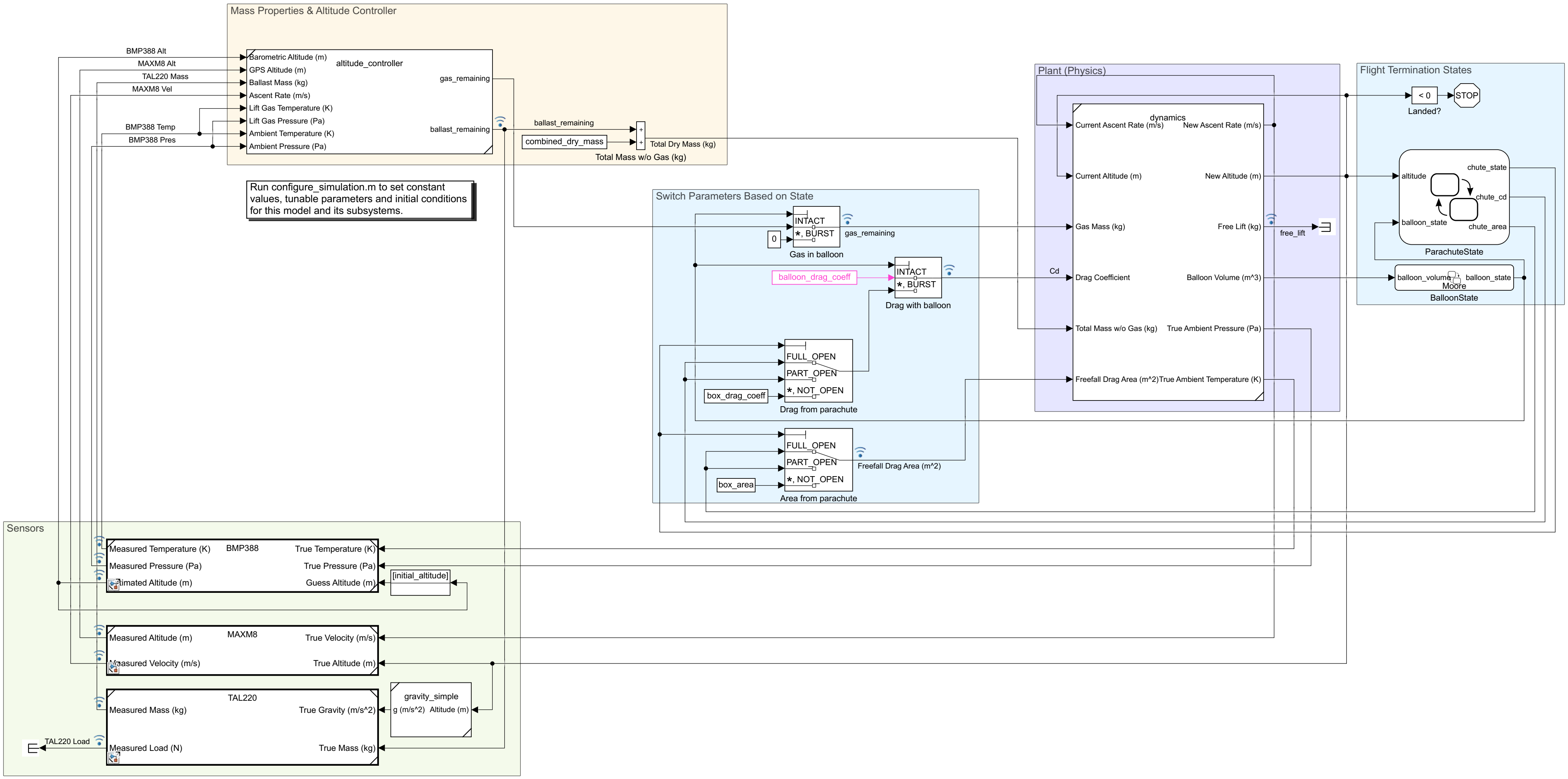
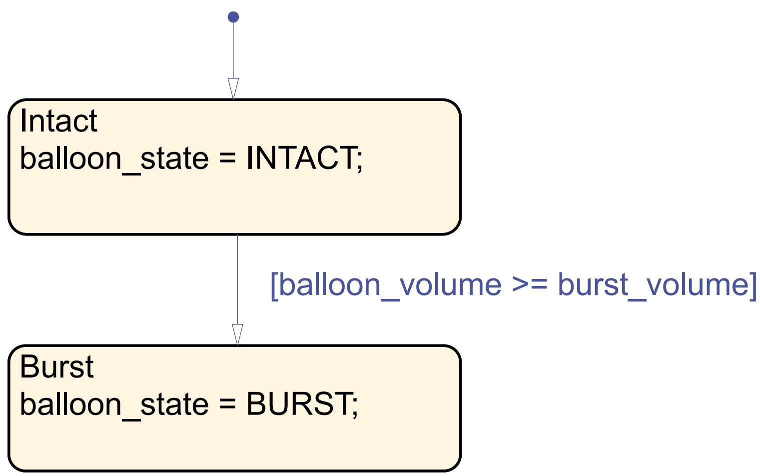
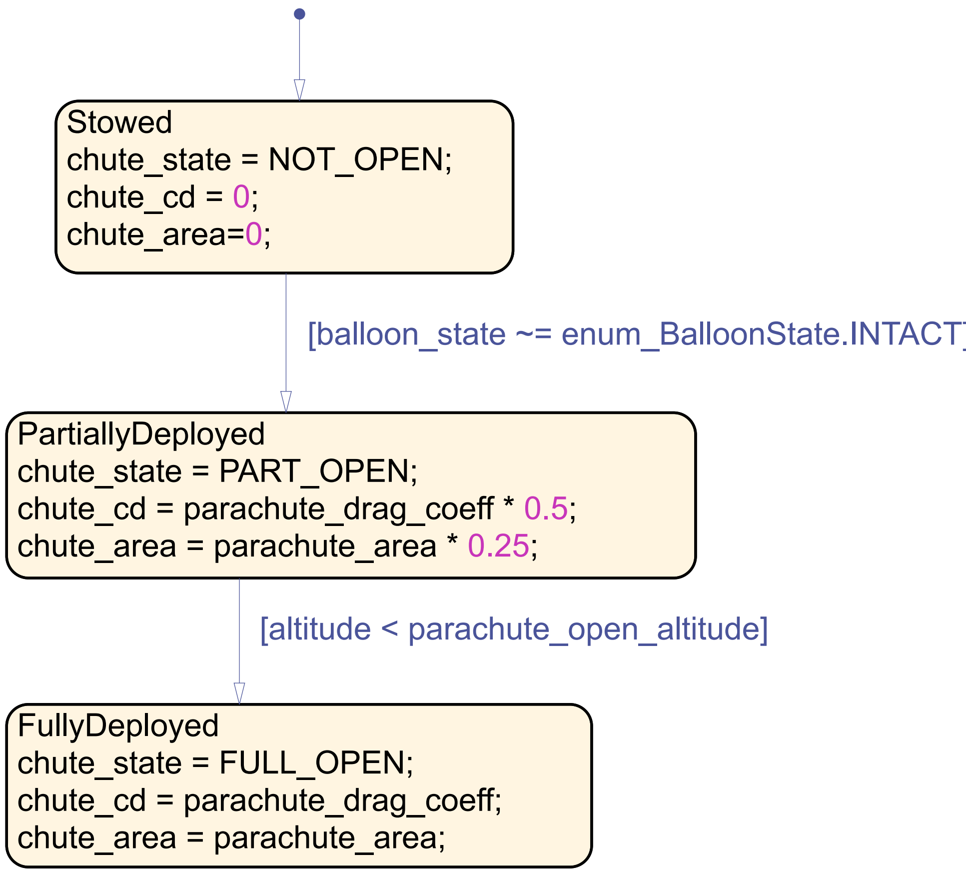
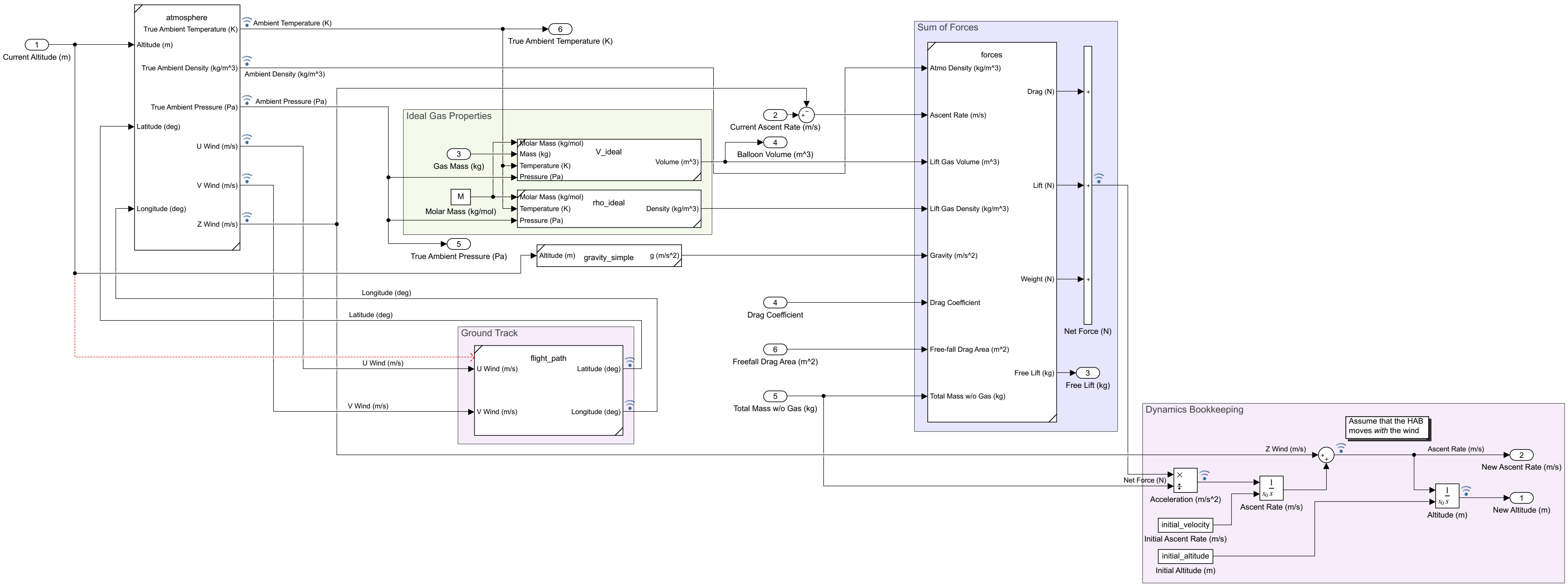
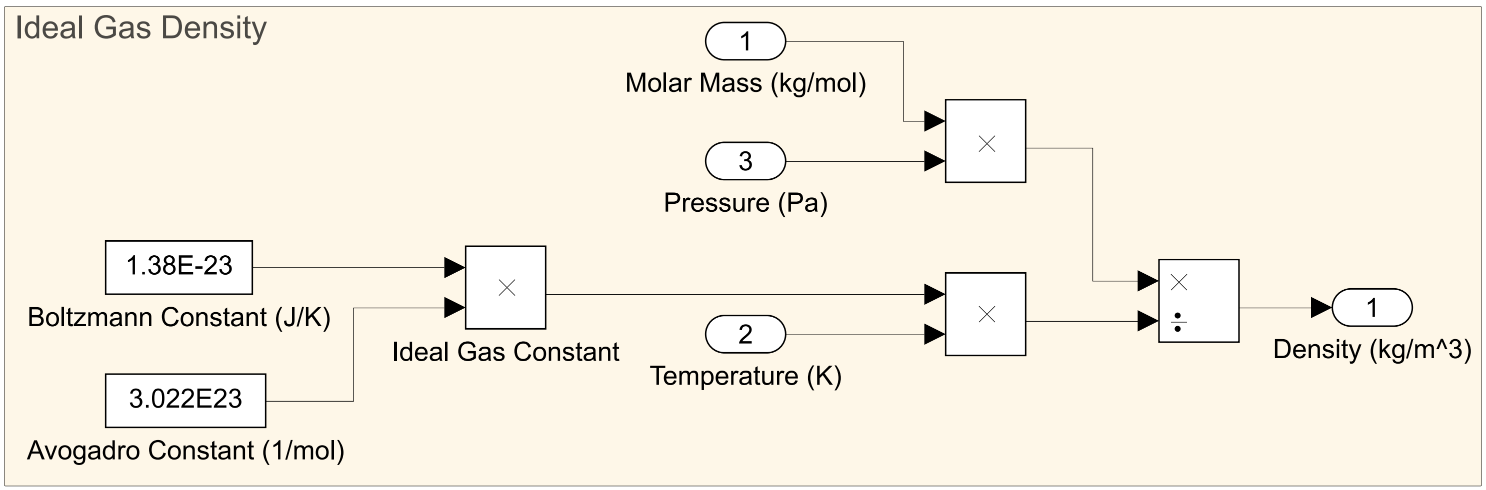


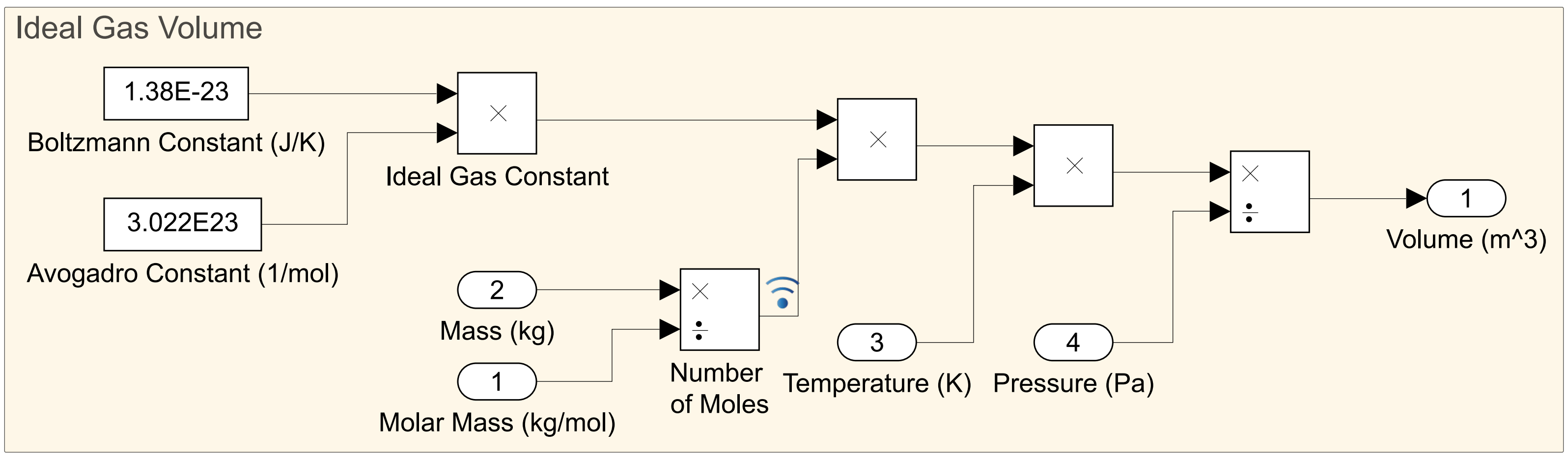
Mass Properties & Altitude Controller
altitude_controller
Barometric Altitude (m)
GPS Altitude (m)
Ballast Mass (kg)
Ascent Rate (m/s)
Lift Gas Temperature (K)
Lift Gas Pressure (Pa)
Ambient Temperature (K)
Ambient Pressure (Pa)
gas_remaining
ballast_remaining
BMP388 Alt
MAXM8 Alt
TAL220 Mass
MAXM8 Vel
BMP388 Temp
BMP388 Pres
combined_dry_mass
Total Dry Mass (kg)
Total Mass w/o Gas (kg)
Run configure_simulation.m to set constant values, tunable parameters and initial conditions for this model and its subsystems.
Switch Parameters Based on State
INTACT
*, BURST
Gas in balloon
balloon_drag_coeff
Drag with balloon
FULL_OPEN
PART_OPEN
*, NOT_OPEN
box_drag_coeff
Drag from parachute
box_area
Area from parachute
Freefall Drag Area (m^2)
Plant (Physics)
dynamics
Current Ascent Rate (m/s)
Current Altitude (m)
Gas Mass (kg)
Cd
Drag Coefficient
Total Mass w/o Gas (kg)
Freefall Drag Area (m^2)
New Ascent Rate (m/s)
New Altitude (m)
Free Lift (kg)
free_lift
Balloon Volume (m^3)
True Ambient Pressure (Pa)
True Ambient Temperature (K)
Flight Termination States
< 0
STOP
Landed?
altitude
balloon_state
chute_state
chute_cd
chute_area
ParachuteState
balloon_volume
Moore
BalloonState
Sensors
BMP388
Measured Temperature (K)
Measured Pressure (Pa)
Estimated Altitude (m)
True Temperature (K)
True Pressure (Pa)
Guess Altitude (m)
[initial_altitude]
MAXM8
Measured Altitude (m)
Measured Velocity (m/s)
True Velocity (m/s)
True Altitude (m)
TAL220
Measured Mass (kg)
Measured Load (N)
TAL220 Load
True Gravity (m/s^2)
True Mass (kg)
gravity_simple
g (m/s^2)
Altitude (m)


Intact
balloon_state = INTACT;
[balloon_volume >= burst_volume]
Burst
balloon_state = BURST;

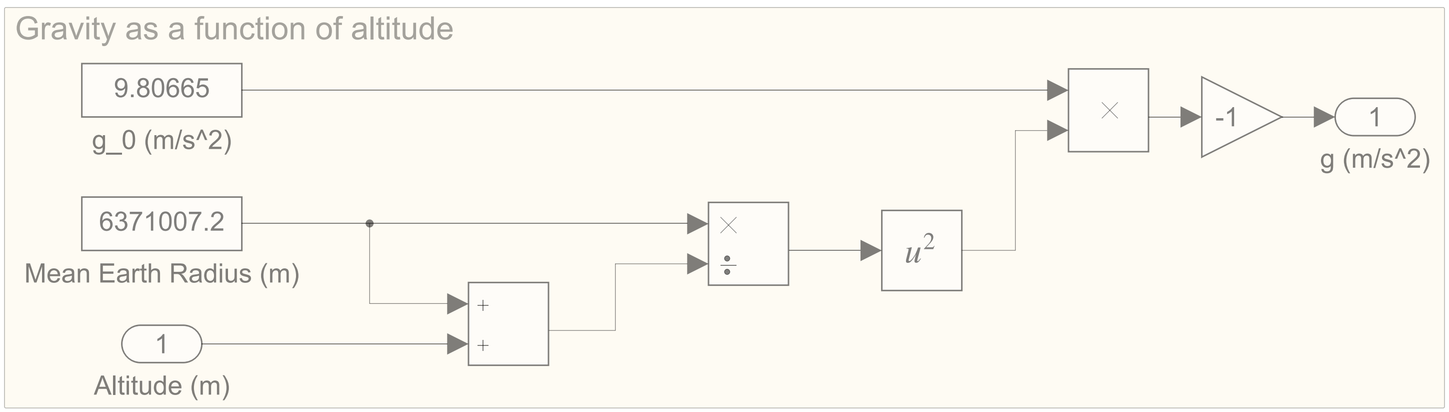

Stowed
chute_state = NOT_OPEN;
chute_cd = 0;
chute_area=0;
[balloon_state ~= enum_BalloonState.INTACT]
PartiallyDeployed
chute_state = PART_OPEN;
chute_cd = parachute_drag_coeff * 0.5;
chute_area = parachute_area * 0.25;
[altitude < parachute_open_altitude]
FullyDeployed
chute_state = FULL_OPEN;
chute_cd = parachute_drag_coeff;
chute_area = parachute_area;

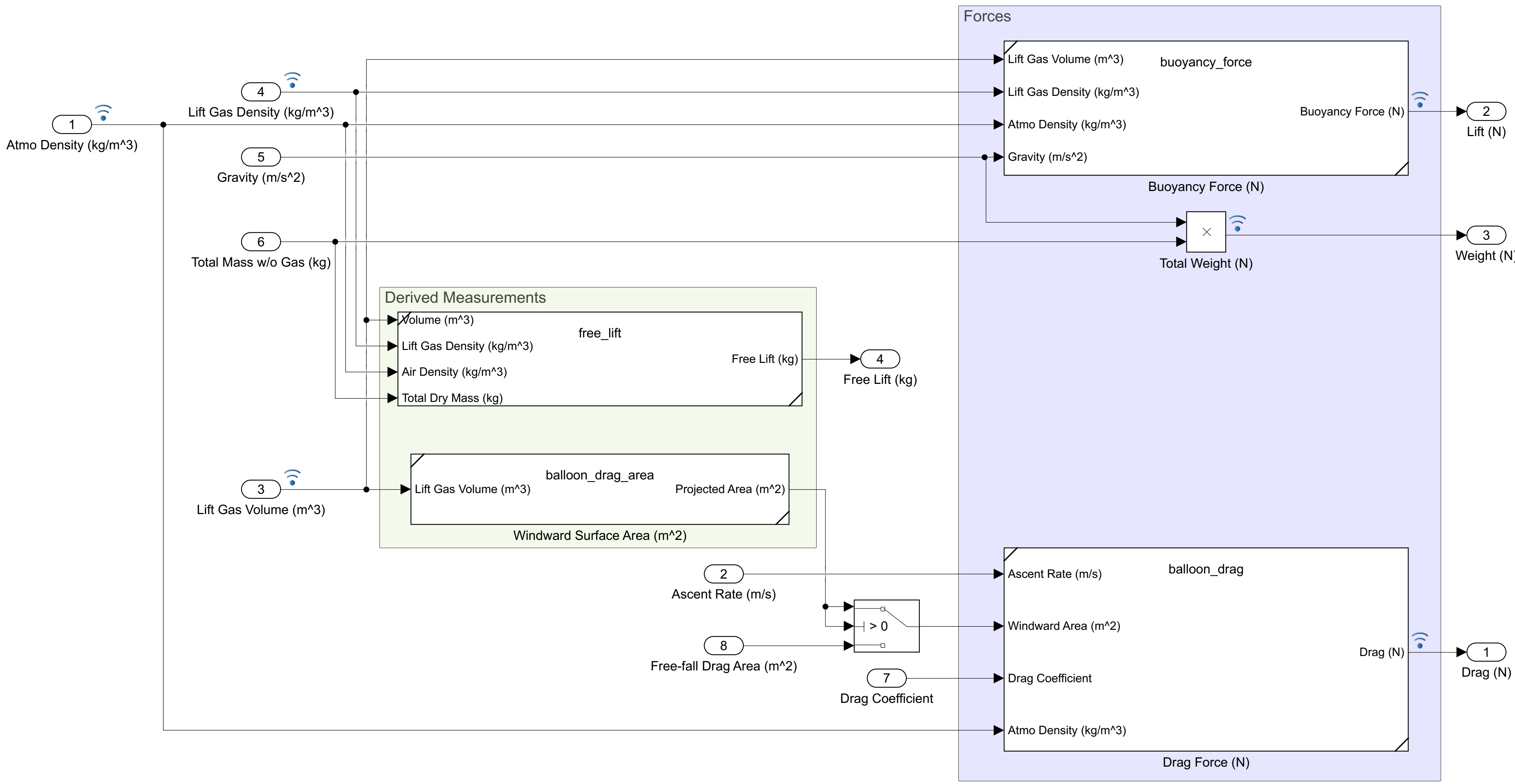

Current Altitude (m)
atmosphere
True Ambient Temperature (K)
Altitude (m)
True Ambient Density (kg/m^3)
True Ambient Pressure (Pa)
Latitude (deg)
U Wind (m/s)
V Wind (m/s)
Longitude (deg)
Z Wind (m/s)
Ambient Temperature (K)
Ambient Density (kg/m^3)
Ambient Pressure (Pa)
True Ambient Temperature (K)
Ideal Gas Properties
Gas Mass (kg)
Molar Mass (kg/mol)
V_ideal
Molar Mass (kg/mol)
Mass (kg)
Temperature (K)
Pressure (Pa)
Volume (m^3)
rho_ideal
Density (kg/m^3)
Current Ascent Rate (m/s)
Balloon Volume (m^3)
True Ambient Pressure (Pa)
gravity_simple
g (m/s^2)
Ground Track
flight_path
Sum of Forces
forces
Atmo Density (kg/m^3)
Ascent Rate (m/s)
Lift Gas Volume (m^3)
Lift Gas Density (kg/m^3)
Gravity (m/s^2)
Drag Coefficient
Free-fall Drag Area (m^2)
Total Mass w/o Gas (kg)
Drag (N)
Lift (N)
Weight (N)
Free Lift (kg)
Net Force (N)
Free Lift (kg)
Drag Coefficient
Freefall Drag Area (m^2)
Total Mass w/o Gas (kg)
Dynamics Bookkeeping
Assume that the HAB moves *with* the wind
Z Wind (m/s)
Ascent Rate (m/s)
New Ascent Rate (m/s)
Net Force (N)
Acceleration (m/s^2)
Ascent Rate (m/s)
Altitude (m)
New Altitude (m)
initial_velocity
Initial Ascent Rate (m/s)
initial_altitude
Initial Altitude (m)


Ideal Gas Density
1
Molar Mass (kg/mol)
3
Pressure (Pa)
1.38E-23
Boltzmann Constant (J/K)
3.022E23
Avogadro Constant (1/mol)
Ideal Gas Constant
2
Temperature (K)
1
Density (kg/m^3)

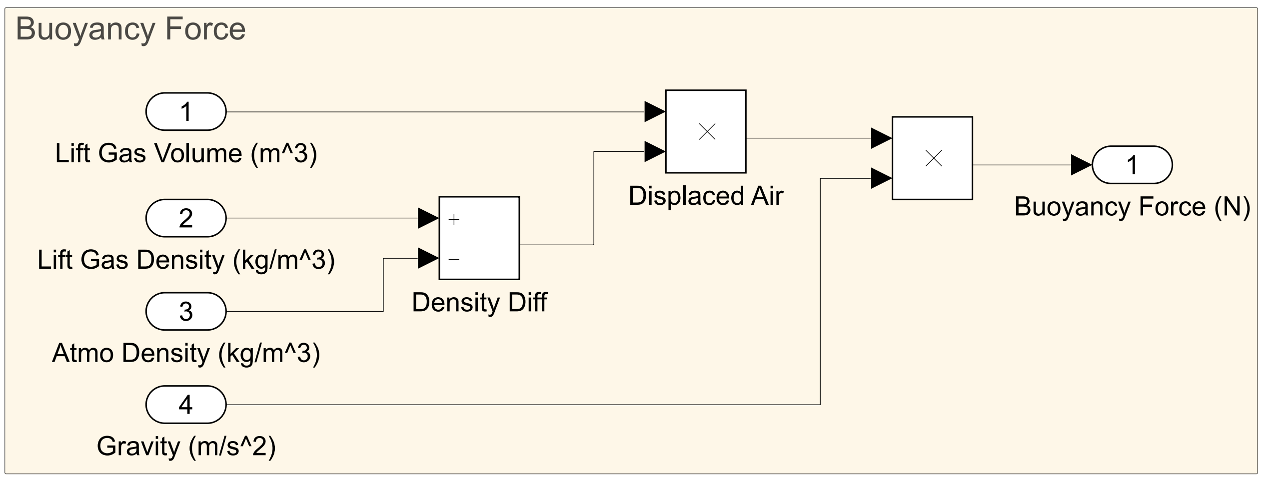


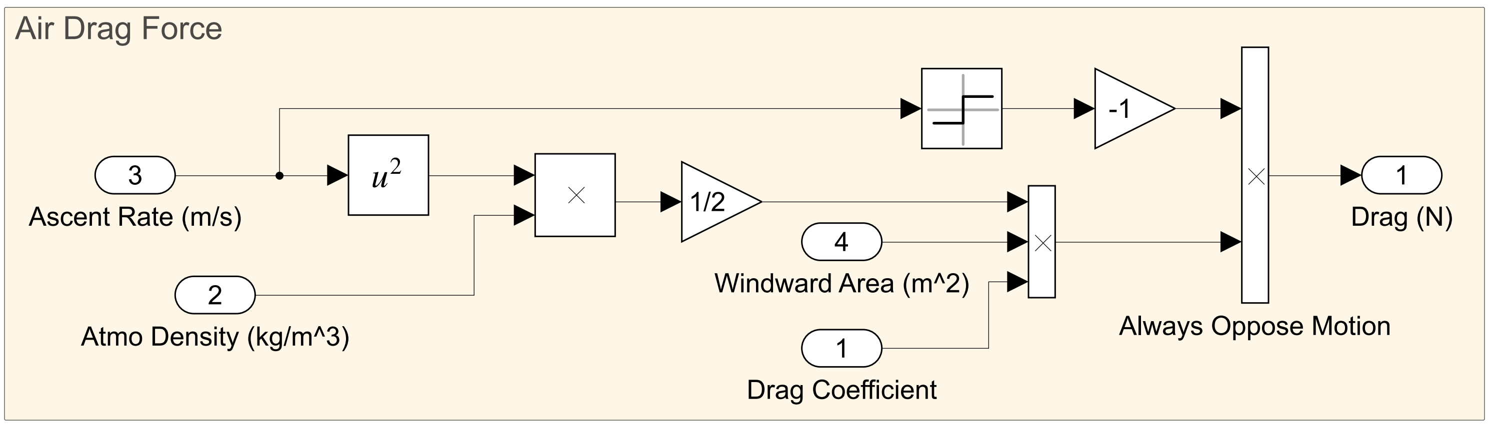
Ideal Gas Volume
1.38E-23
Boltzmann Constant (J/K)
3.022E23
Avogadro Constant (1/mol)
Ideal Gas Constant
2
Mass (kg)
1
Molar Mass (kg/mol)
Number of Moles
3
Temperature (K)
4
Pressure (Pa)
1
Volume (m^3)

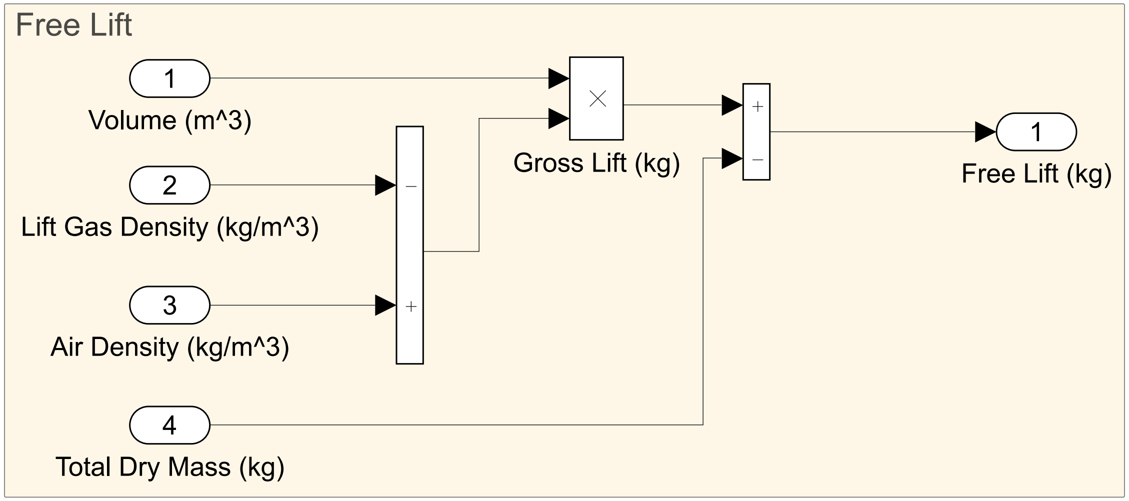

Gravity as a function of altitude
9.80665
g_0 (m/s^2)
6371007.2
Mean Earth Radius (m)
1
Altitude (m)
+
+
×
÷
u^2
×
-1
1
g (m/s^2)

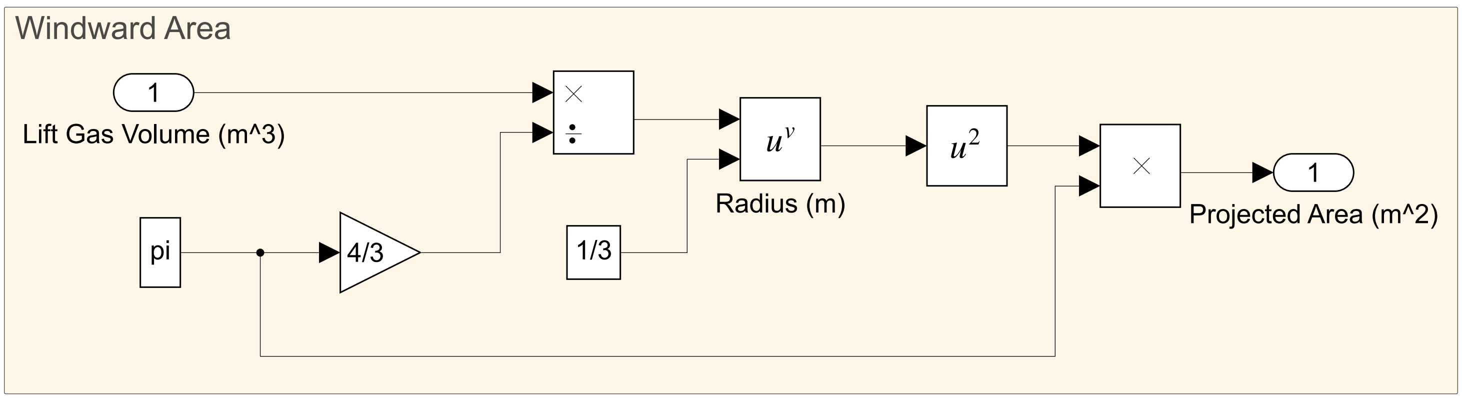


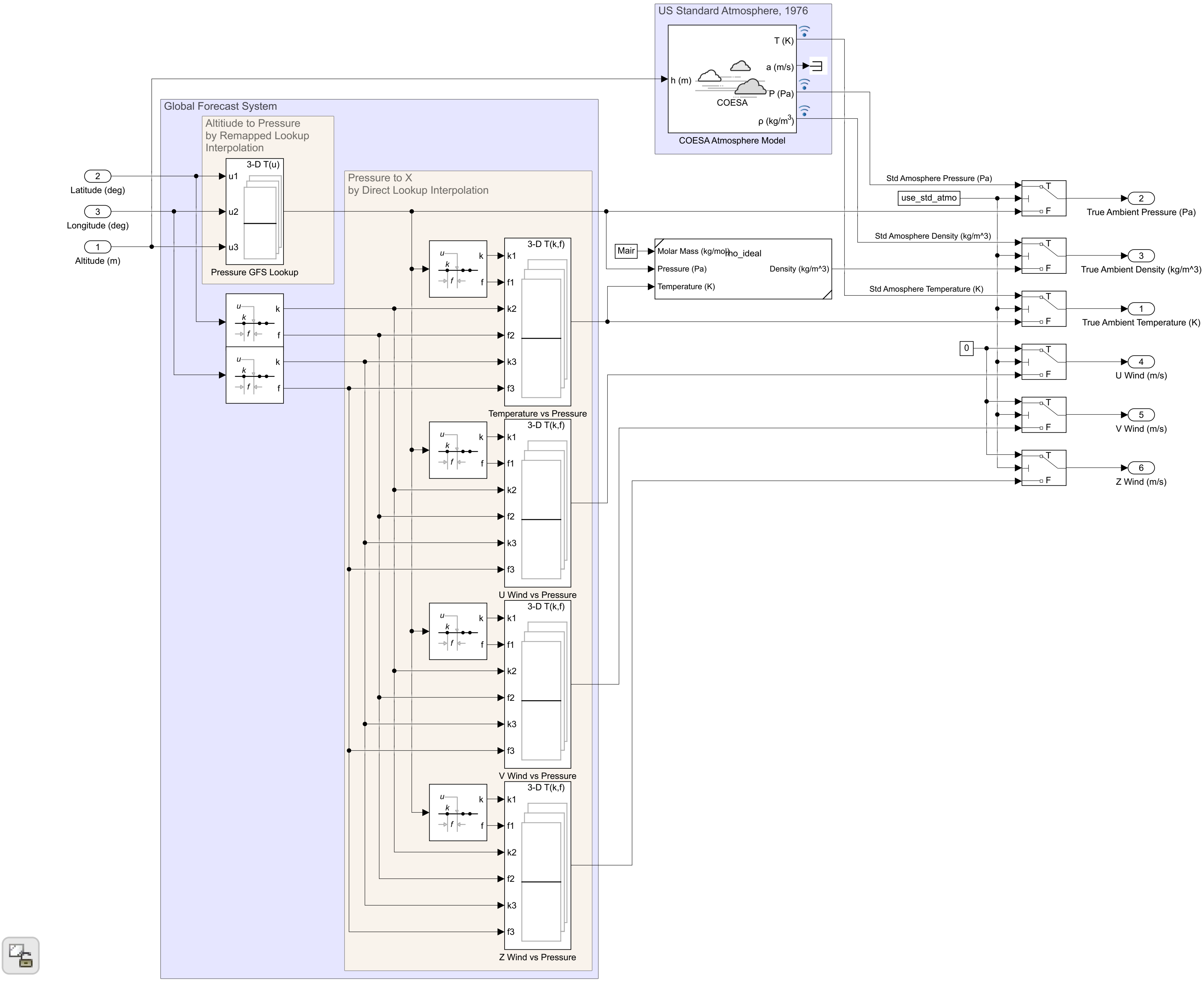
Forces
1
Atmo Density (kg/m^3)
4
Lift Gas Density (kg/m^3)
5
Gravity (m/s^2)
6
Total Mass w/o Gas (kg)
3
Lift Gas Volume (m^3)
Lift Gas Volume (m^3)
Lift Gas Density (kg/m^3)
Atmo Density (kg/m^3)
Gravity (m/s^2)
buoyancy_force
Buoyancy Force (N)
Buoyancy Force (N)
2
Lift (N)
Total Weight (N)
3
Weight (N)
Derived Measurements
Volume (m^3)
Lift Gas Density (kg/m^3)
Air Density (kg/m^3)
Total Dry Mass (kg)
free_lift
Free Lift (kg)
4
Free Lift (kg)
balloon_drag_area
Lift Gas Volume (m^3)
Projected Area (m^2)
Windward Surface Area (m^2)
2
Ascent Rate (m/s)
> 0
8
Free-fall Drag Area (m^2)
7
Drag Coefficient
balloon_drag
Ascent Rate (m/s)
Windward Area (m^2)
Drag Coefficient
Atmo Density (kg/m^3)
Drag (N)
Drag Force (N)
1
Drag (N)


Buoyancy Force
1
Lift Gas Volume (m^3)
2
Lift Gas Density (kg/m^3)
3
Atmo Density (kg/m^3)
4
Gravity (m/s^2)
Density Diff
Displaced Air
1
Buoyancy Force (N)

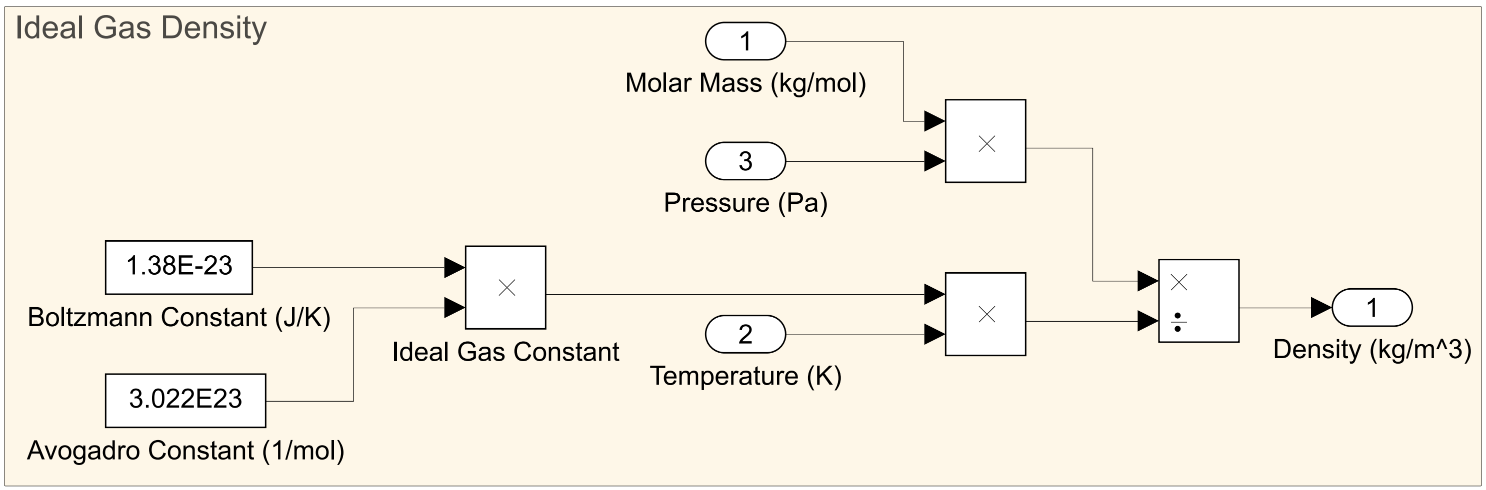

Air Drag Force
3
Ascent Rate (m/s)
2
Atmo Density (kg/m^3)
u^2
1/2
4
Windward Area (m^2)
1
Drag Coefficient
-1
Always Oppose Motion
1
Drag (N)

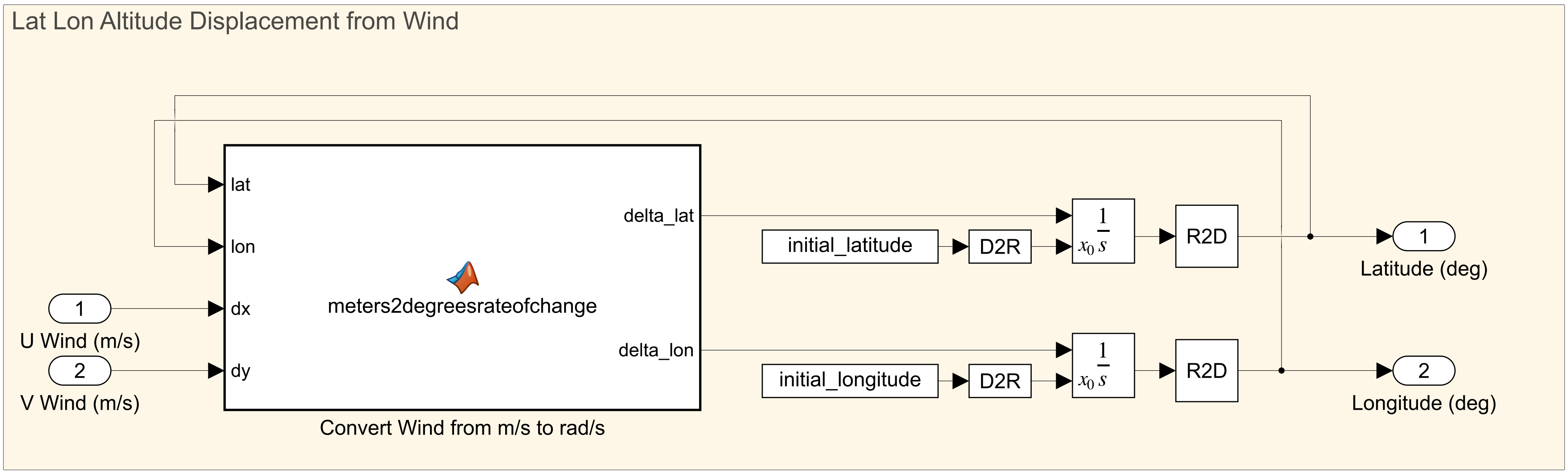

Free Lift
1
Volume (m^3)
2
Lift Gas Density (kg/m^3)
3
Air Density (kg/m^3)
4
Total Dry Mass (kg)
Gross Lift (kg)
1
Free Lift (kg)


Windward Area
1
Lift Gas Volume (m^3)
pi
4/3
1/3
Radius (m)
1
Projected Area (m^2)



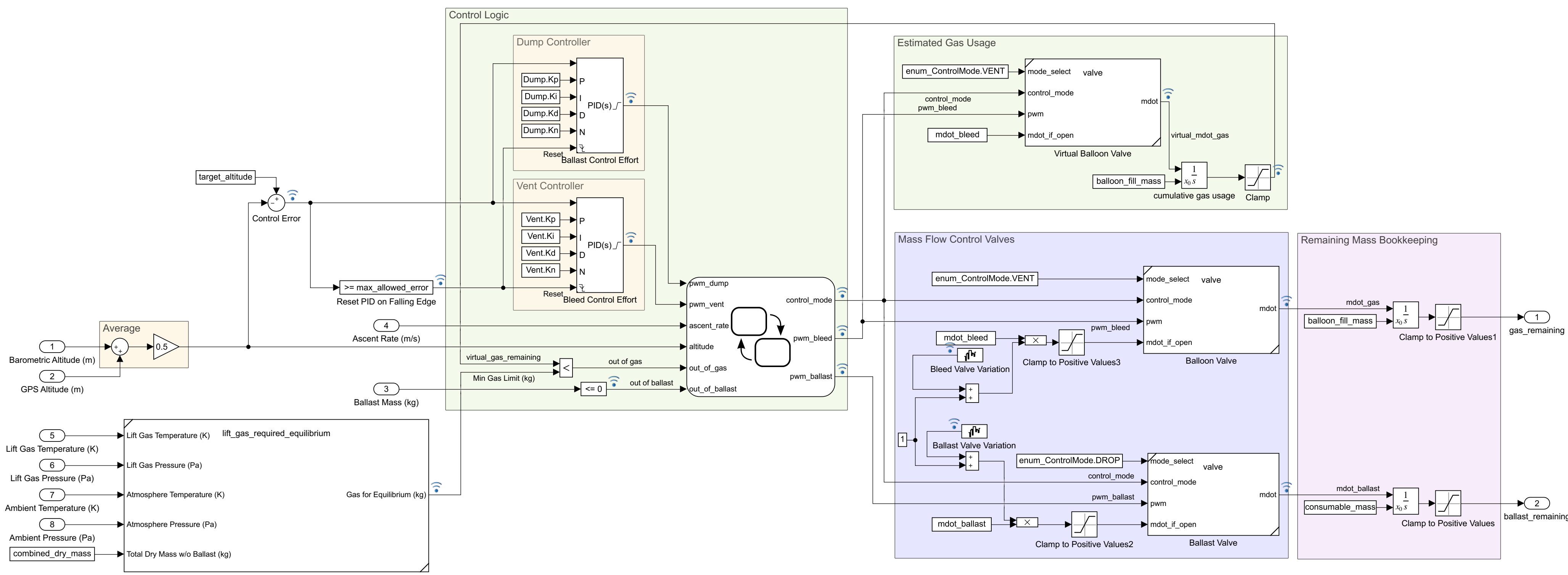
US Standard Atmosphere, 1976
T (K)
a (m/s)
h (m)
COESA
P (Pa)
ρ (kg/m^3)
COESA Atmosphere Model
Global Forecast System
Altitiude to Pressure
by Remapped Lookup
Interpolation
3-D T(u)
u1
u2
u3
Pressure GFS Lookup
Pressure to X
by Direct Lookup Interpolation
2
Latitude (deg)
3
Longitude (deg)
1
Altitude (m)
3-D T(k,f)
k1
f1
k2
f2
k3
f3
Temperature vs Pressure
U Wind vs Pressure
V Wind vs Pressure
Z Wind vs Pressure
Mair
Molar Mass (kg/mol)
rho_ideal
Pressure (Pa)
Temperature (K)
Density (kg/m^3)
Std Amosphere Pressure (Pa)
use_std_atmo
Std Amosphere Density (kg/m^3)
Std Amosphere Temperature (K)
0
2
True Ambient Pressure (Pa)
3
True Ambient Density (kg/m^3)
1
True Ambient Temperature (K)
4
U Wind (m/s)
5
V Wind (m/s)
6
Z Wind (m/s)

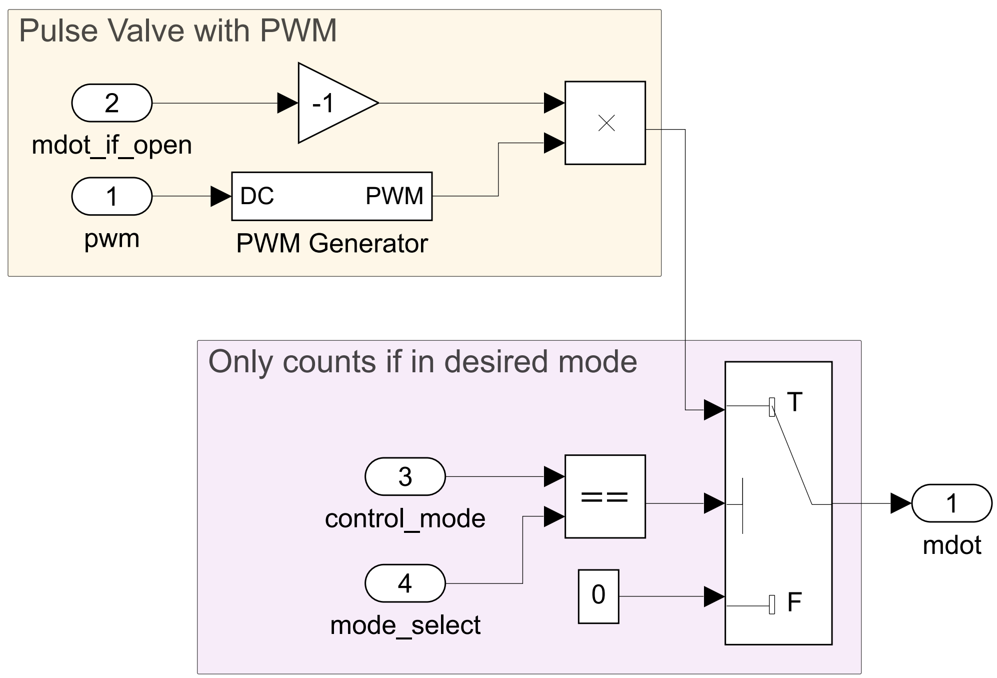

Ideal Gas Density
1
Molar Mass (kg/mol)
3
Pressure (Pa)
1.38E-23
Boltzmann Constant (J/K)
3.022E23
Avogadro Constant (1/mol)
Ideal Gas Constant
2
Temperature (K)
1
Density (kg/m^3)

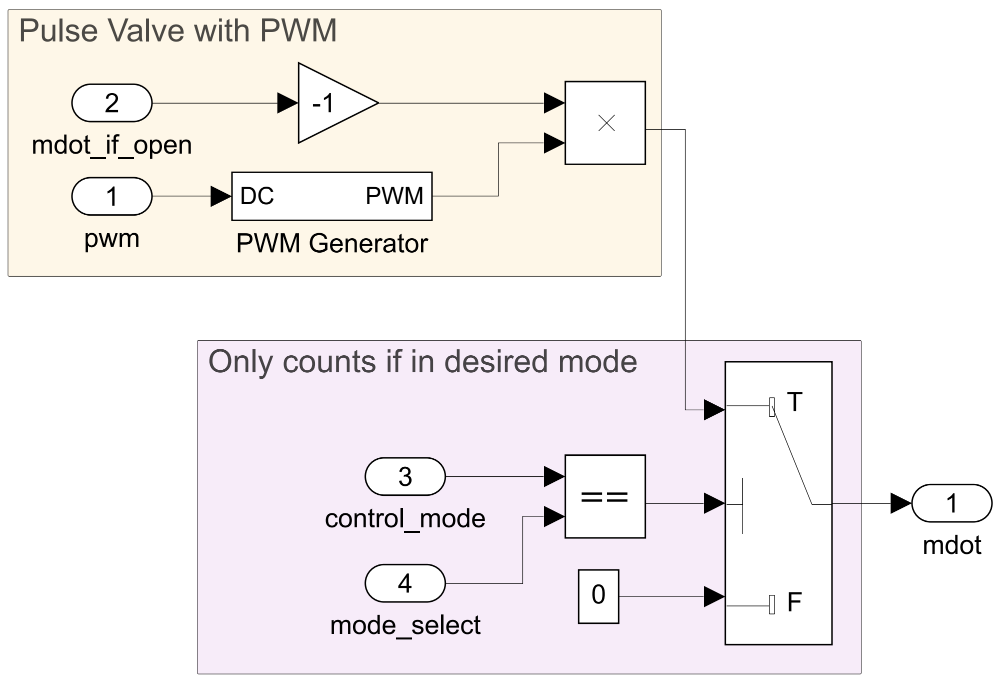


Lat Lon Altitude Displacement from Wind
lat
lon
dx
dy
delta_lat
delta_lon
meters2degreesrateofchange
Convert Wind from m/s to rad/s
1
U Wind (m/s)
2
V Wind (m/s)
initial_latitude
initial_longitude
D2R
D2R
R2D
R2D
1
Latitude (deg)
2
Longitude (deg)

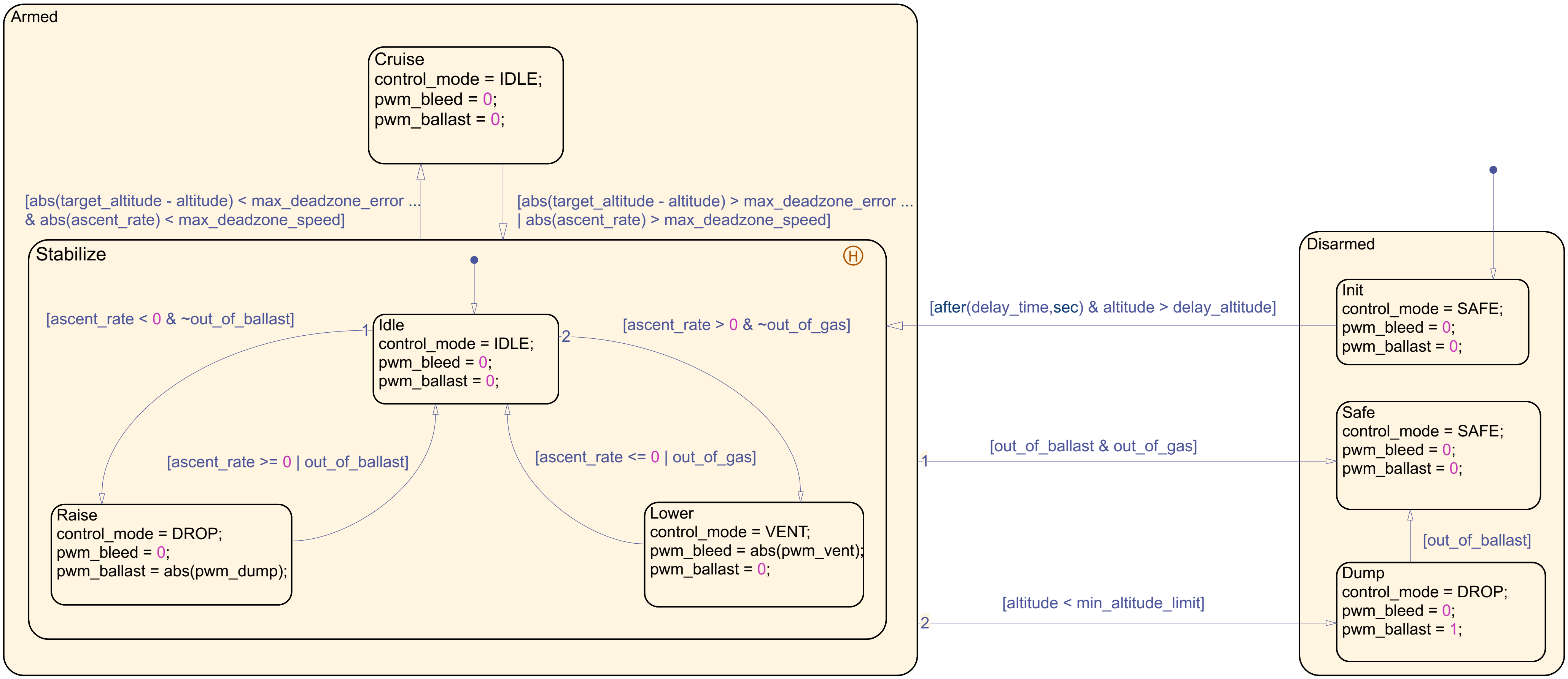
```
function [delta_lat, delta_lon] = meters2degreesrateofchange(lat,lon,dx,dy)
% convert a cartesian rate of change (m/s) to lat-lon rate of change (degs)
rEarth = 6371007.2;
if dy == 0
    % discontinuity in arctan!
    bearing = pi / 2;
else
    % unit vector of velocity
    bearing = atan(dx/dy);
end

% which quadrant of unit circle are we moving in?
% correct it so we always have the angle with respect to the x axis
if (dx > 0) && (dy > 0)
    bearing = (pi/2) - bearing;
elseif (dx > 0) && (dy < 0)
    bearing = (pi/2) - bearing;
elseif (dx < 0) && (dy <= 0)
    bearing = (pi/2) + bearing;
end
velocity_magnitude = norm(dx,dy);

delta_lat = cos(lat) * sin(velocity_magnitude / rEarth) * cos(bearing);
delta_lon = atan2(sin(bearing) * sin(velocity_magnitude / rEarth) * cos(delta_lat), ...
                cos(velocity_magnitude / rEarth) - sin(lat) * sin(delta_lat));
end
```

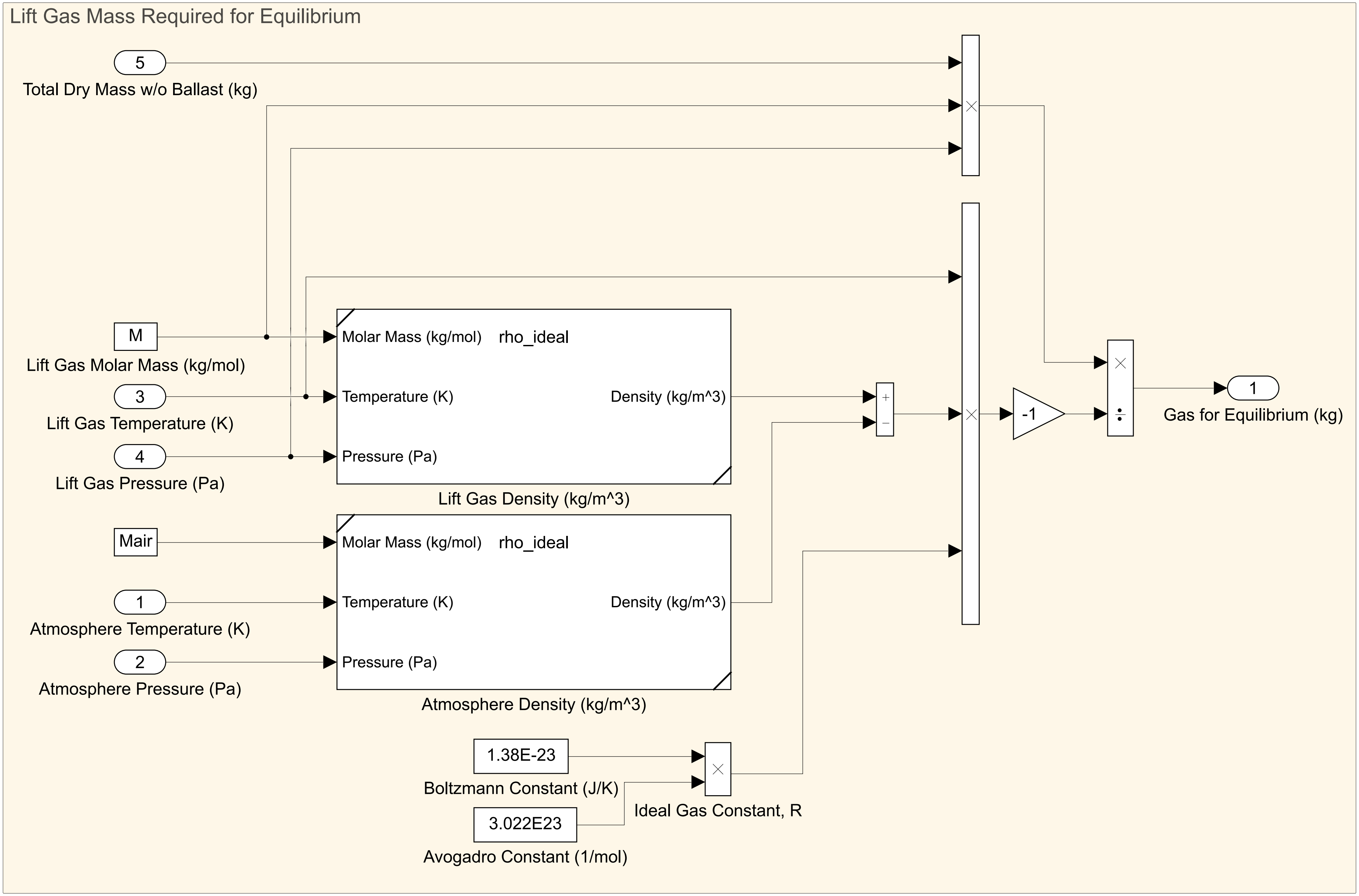

Control Logic
Dump Controller
Dump.Kp
Dump.Ki
Dump.Kd
Dump.Kn
P
I
D
N
PID(s)
Reset
Ballast Control Effort
Vent Controller
Vent.Kp
Vent.Ki
Vent.Kd
Vent.Kn
Bleed Control Effort
target_altitude
Control Error
>= max_allowed_error
Reset PID on Falling Edge
pwm_dump
pwm_vent
ascent_rate
altitude
out_of_gas
out_of_ballast
control_mode
pwm_bleed
pwm_ballast
virtual_gas_remaining
Min Gas Limit (kg)
out of gas
<= 0
out of ballast
1
Barometric Altitude (m)
2
GPS Altitude (m)
Average
0.5
4
Ascent Rate (m/s)
3
Ballast Mass (kg)
5
Lift Gas Temperature (K)
6
Lift Gas Pressure (Pa)
7
Ambient Temperature (K)
8
Ambient Pressure (Pa)
combined_dry_mass
lift_gas_required_equilibrium
Lift Gas Temperature (K)
Lift Gas Pressure (Pa)
Atmosphere Temperature (K)
Atmosphere Pressure (Pa)
Total Dry Mass w/o Ballast (kg)
Gas for Equilibrium (kg)
Estimated Gas Usage
enum_ControlMode.VENT
mode_select
valve
control_mode
pwm
mdot
mdot_bleed
mdot_if_open
Virtual Balloon Valve
virtual_mdot_gas
balloon_fill_mass
cumulative gas usage
Clamp
Mass Flow Control Valves
enum_ControlMode.VENT
Balloon Valve
Bleed Valve Variation
Clamp to Positive Values3
Ballast Valve Variation
enum_ControlMode.DROP
mdot_ballast
Clamp to Positive Values2
Ballast Valve
Remaining Mass Bookkeeping
mdot_gas
balloon_fill_mass
Clamp to Positive Values1
gas_remaining
mdot_ballast
consumable_mass
Clamp to Positive Values
ballast_remaining

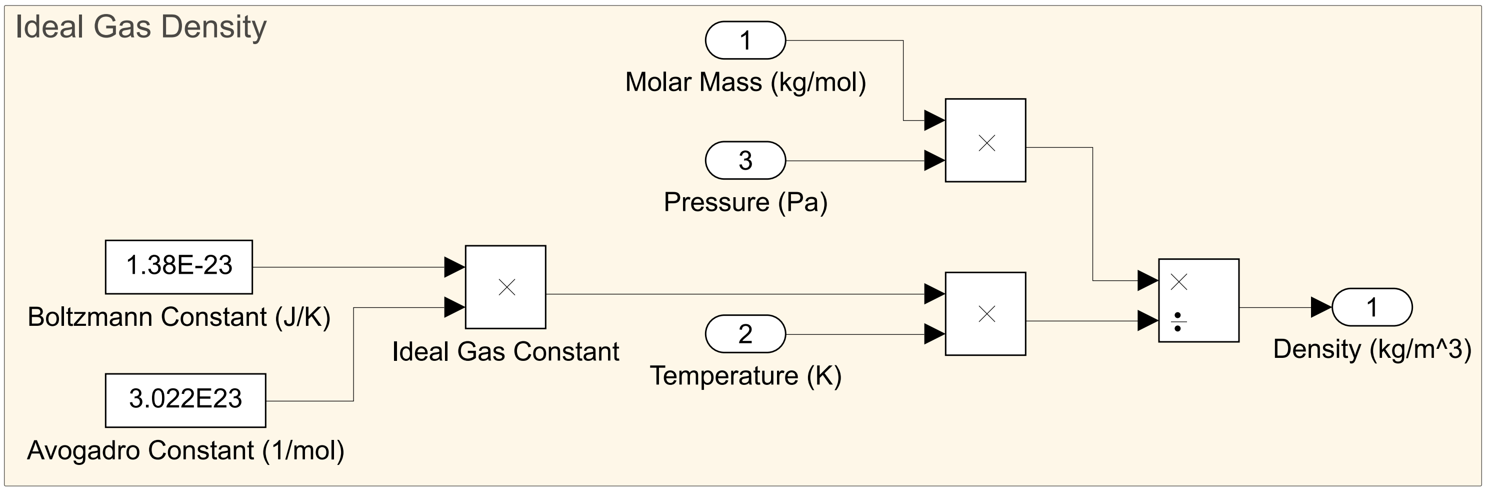

Pulse Valve with PWM
2
mdot_if_open
-1
1
pwm
DC
PWM
PWM Generator
Only counts if in desired mode
T
3
control_mode
==
4
mode_select
0
F
1
mdot

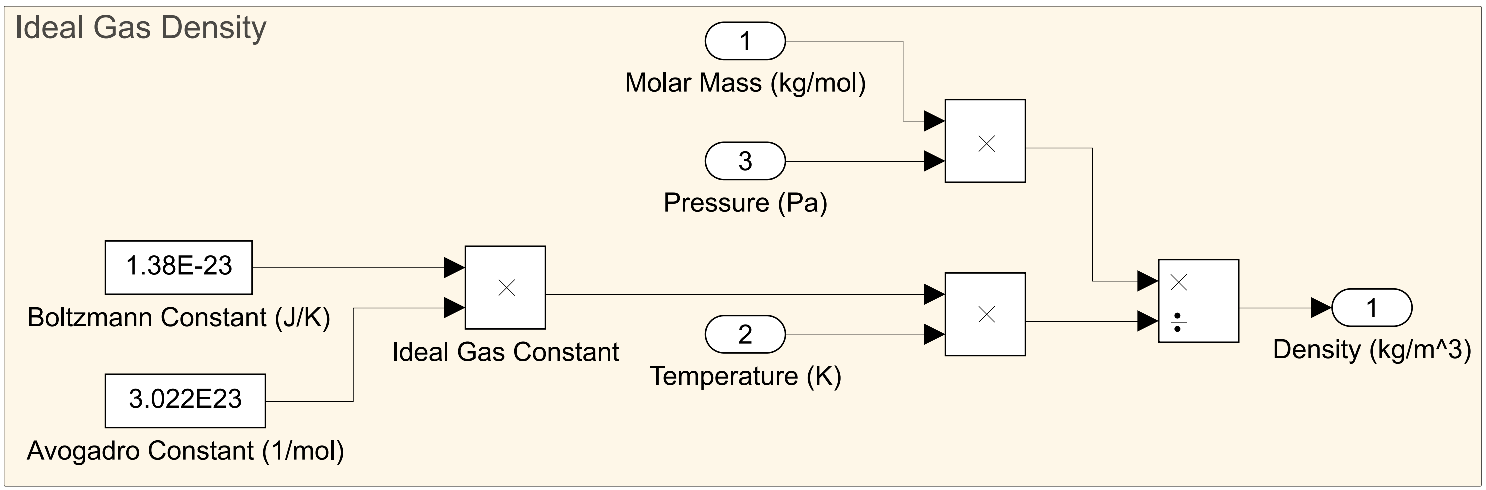


Pulse Valve with PWM
2
mdot_if_open
-1
1
pwm
DC
PWM
PWM Generator
Only counts if in desired mode
T
3
control_mode
==
4
mode_select
0
F
1
mdot

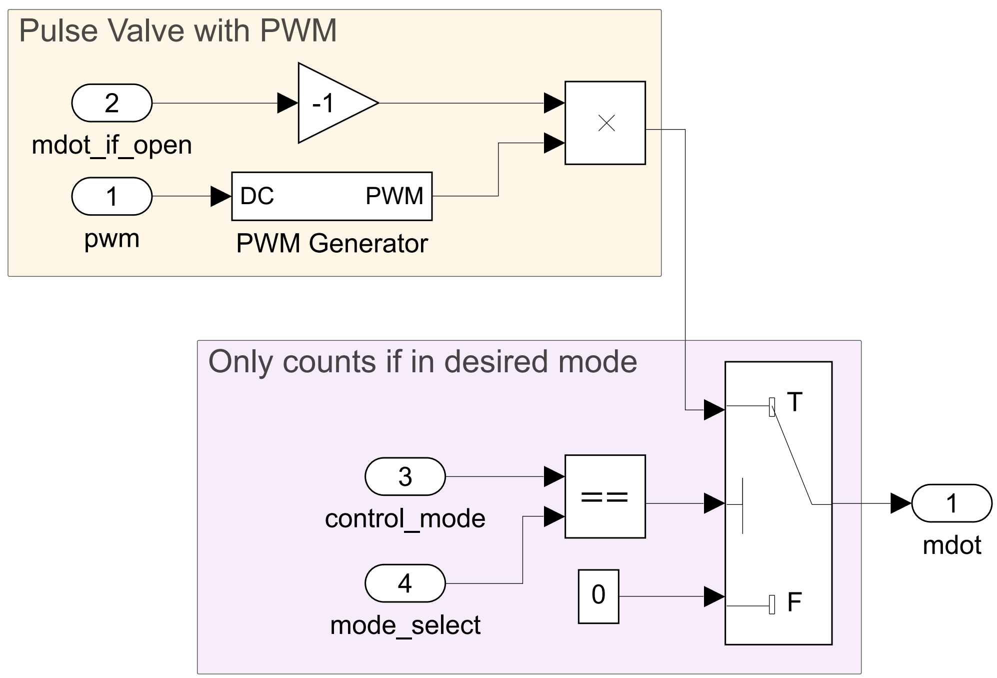

Armed
Cruise
control_mode = IDLE;
pwm_bleed = 0;
pwm_ballast = 0;
[abs(target_altitude - altitude) < max_deadzone_error ...
& abs(ascent_rate) < max_deadzone_speed]
[abs(target_altitude - altitude) > max_deadzone_error ...
| abs(ascent_rate) > max_deadzone_speed]
Stabilize
H
[ascent_rate < 0 & ~out_of_ballast]
1
Idle
control_mode = IDLE;
pwm_bleed = 0;
pwm_ballast = 0;
2
[ascent_rate > 0 & ~out_of_gas]
[ascent_rate >= 0 | out_of_ballast]
[ascent_rate <= 0 | out_of_gas]
Raise
control_mode = DROP;
pwm_bleed = 0;
pwm_ballast = abs(pwm_dump);
Lower
control_mode = VENT;
pwm_bleed = abs(pwm_vent);
pwm_ballast = 0;
[after(delay_time,sec) & altitude > delay_altitude]
1
[out_of_ballast & out_of_gas]
2
[altitude < min_altitude_limit]
Disarmed
Init
control_mode = SAFE;
pwm_bleed = 0;
pwm_ballast = 0;
Safe
control_mode = SAFE;
pwm_bleed = 0;
pwm_ballast = 0;
[out_of_ballast]
Dump
control_mode = DROP;
pwm_bleed = 0;
pwm_ballast = 1;

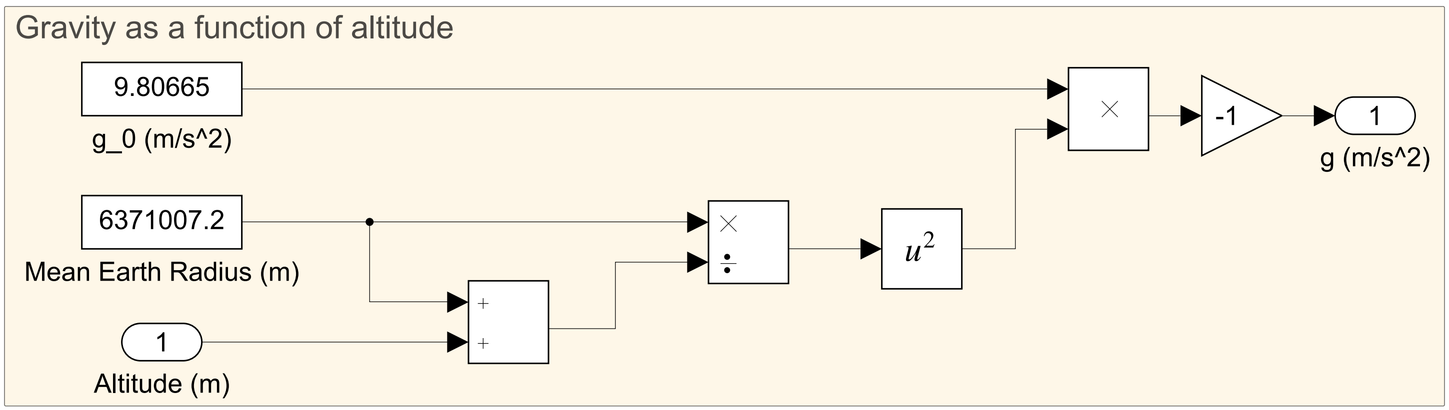

Lift Gas Mass Required for Equilibrium
5
Total Dry Mass w/o Ballast (kg)
M
Lift Gas Molar Mass (kg/mol)
3
Lift Gas Temperature (K)
4
Lift Gas Pressure (Pa)
Molar Mass (kg/mol)
rho_ideal
Temperature (K)
Density (kg/m^3)
Pressure (Pa)
Lift Gas Density (kg/m^3)
Mair
1
Atmosphere Temperature (K)
2
Atmosphere Pressure (Pa)
Molar Mass (kg/mol)
rho_ideal
Temperature (K)
Density (kg/m^3)
Pressure (Pa)
Atmosphere Density (kg/m^3)
1.38E-23
Boltzmann Constant (J/K)
3.022E23
Avogadro Constant (1/mol)
Ideal Gas Constant, R
-1
1
Gas for Equilibrium (kg)


Ideal Gas Density
1
Molar Mass (kg/mol)
3
Pressure (Pa)
1.38E-23
Boltzmann Constant (J/K)
3.022E23
Avogadro Constant (1/mol)
Ideal Gas Constant
2
Temperature (K)
1
Density (kg/m^3)



Ideal Gas Density
1
Molar Mass (kg/mol)
3
Pressure (Pa)
1.38E-23
Boltzmann Constant (J/K)
Ideal Gas Constant
3.022E23
Avogadro Constant (1/mol)
2
Temperature (K)
1
Density (kg/m^3)


Pulse Valve with PWM
2
mdot_if_open
-1
1
pwm
DC
PWM
PWM Generator
Only counts if in desired mode
T
3
control_mode
==
4
mode_select
0
F
1
mdot


Gravity as a function of altitude
9.80665
g_0 (m/s^2)
6371007.2
Mean Earth Radius (m)
1
Altitude (m)
+
+
×
÷
u^2
×
-1
1
g (m/s^2)