PowerInForwardFlight_main_tail_rotor.m user guide

Rosa Castiello, Raffaella Scarano

Contents

1 Introduction

1.1 Required power of main rotor in forward flight

The required power of main rotor in forward flight is calculated as:

$$P_{request} = P_i + P_0 + P_{fus}$$

Where:

$P_{request}$ is the power required;

P_i is the induced power;

P_0 is the parasitic power;

P_{fus} is the power absorbed by the fuselage.

$$P_0 = \pi\rho\frac{\sigma\overline{C_d}}{8}\Omega^3R^5[1+k\mu^2] \qquad \text{con} \qquad \mu = \frac{V_\infty cos\alpha}{\Omega R}$$

Assuming α ≅ 0:

$$P_i = T(V_\infty sin\alpha + w) \cong Tw$$

Where, being in fast forward flight, T = W.

From Glauert's theory, that underly the rotor model in non-axial flow, it is derived that:

$$T = 2\dot{m}w = 2A\rho V'w$$

From the inversion of which we can derive w, then:

$$P_i = \frac{W^2}{2\rho AV'} \cong \frac{W^2}{2\rho AV_\infty}$$

In order to calculate the induced power more accurately, the equation describing the operating curve at constant-thrust of the rotor in forward flight is:

$$\widetilde{P_\imath} = \widetilde{V_\infty}sin\alpha + \tilde{w} \cong \tilde{w}$$

We then solve axial induction by the equation:

$$\left(\widetilde{V_\infty}\tilde{w}sin\alpha + \tilde{w}^2\right)^2 + \widetilde{V_\infty}^2\tilde{w}^2cos\alpha^2 = 1$$

We go on to solve this second-degree equation in $\tilde{w}$, whose positive root is $\tilde{w}$. Then we obtain P_i from:

$$P_i = \widetilde{P_\imath}P_h$$

Where P_h is the power under hovering conditions.

The parasitic power of the fuselage is given by:

$$P_{fus} = f\frac{1}{2}\rho_\infty{V_\infty}^3$$

Where the factor f, which has the size of a surface, is called the ‘equivalent wet area’. A typical value is f/A ≅ 0.007.

1.2 Required power of tail rotor in forward flight

The required power by the tail rotor is given by:

$$P_{r\mathrm{tail}} = P_{indotta} + P_{parassita}$$

$P_{indotta}$ and $P_{parassita}$ were calculated in the same way as was shown in the previous paragraph for the main rotor.

1.3 The complete helicopter system: the total required power

The total power required by the helicopter engines for forward flight is:

$$P_{tot} = [P_{rprincipal} + P_{r\mathrm{tail}} + P_{aux}] \cdot \eta_{\mathrm{t}}$$

Where P_{aux} is the power required by the on-board auxiliary systems. It assumes values between 10kW and 30kW.

η_{t} is the coefficient that introduces transmission losses; it is assumed to be 1.03.

2 Inputs and outputs

2.1 Inputs

The function has the following inputs:

Altitude (h), asintotic velocity vector (Vinf), helicopter weigth (W),

Main rotor's radius (v_input(1)), Main rotor's Area (v_input(2)), Main rotor's Solidity (v_input(3)), Main rotor's angular velocity (v_input(4)), Cd of main rotor's blade element (v_input(5)),

Tail rotor's radius (v_input(6)), Tail rotor's Area (v_input(7)), Tail rotor's solidity (v_input(8)), Tail rotor's angular velocity (v_input(9)), Cd of tail rotor's blade element (v_input(10)),

Distance between main rotor and tail rotor (v_input(11)), correction factor (v_input(12)), Auxiliary Power (v_input(13)), Coefficient due to transmission losses (v_input(14)), Available Power at sea level (v_input(15)).

2.2 Outputs

The function has the following outputs:

Induced Power, Parasitic Power and Required Power by the main rotor, Parasitic Power by the fuselage,

Induced Power, Parasitic Power and Required Power by the tail rotor,

Total Required Power, Available Power, Maximum Speed, Speed of maximum endurance and Speed of maximum Range.

3 Function description

```
function [ Pind_B, Ppar_rp,Pfus,P_tot_rp,...
         Pind_rc,Ppar_rc,P_tot_rc,Ptot_req, Pdisp, ...
         V_NE, V_BE, V_BR] = PowerInForwardFlight_main_tail_rotor(h, Vinf, W,v_input)
% Required Power by the main rotor in forward flight
[~, ~, ~, rho]     = atmosisa(h);
[~, ~, ~, rho_sl]  = atmosisa(0);
```

The Available power varies with altitude:

$$P_{available} = P_{availableSL}\frac{\rho}{\rho_{SL}}$$

```
Pdisp = v_input(15)*rho/rho_sl;


%Total power of the main rotor
%Parasitic Power
mu_rp   = Vinf./(v_input(4)*v_input(1));
Ppar_rp = (v_input(3)*v_input(5)/8.*(1+4.7*mu_rp.^2)).*...
          (pi*rho*v_input(4)^3*v_input(1)^5); % Rotor BEMT
                                              % obs: k=4.7 for radial speeds


T = W;  % Hp small alpha, in forward flight
%Induced Power

%Impulsive Theory
Pind_A      = W^2./(2*rho*v_input(2).*Vinf); %Glauert Theory

w_hov       = 1/sqrt(2*rho)*sqrt(T/v_input(2));
Vinf_tilde = Vinf./w_hov;
w_tilde     = sqrt((- (Vinf_tilde.^2)./2 + sqrt(((Vinf_tilde.^2)./2).^2+1)));
w_rotore    = w_tilde*w_hov;
Pi_tilde    = w_tilde; %Pi_tilde = Vinf_tilde*sin(alpha) + w = w because sin(alpha)=0
                       % Hp: T costant
P_hov   = T*w_hov;

Pind_B = Pi_tilde*P_hov;

%Fusulage Power
f    = 0.007*v_input(2);         % f: equivalent wet area
Pfus = 0.5*rho.*Vinf.^3*f;
%real main rotor's request Power

P_tot_rp = v_input(12)*Pind_B + Ppar_rp + Pfus;



%Total Power of the tail rotor


mu_rc = Vinf./(v_input(9)*v_input(6));




% Induced Power
Q_rot_main      = P_tot_rp./(v_input(4));
T_tail_required = Q_rot_main/v_input(11);
whov_rc         = 1/sqrt(2*rho).*sqrt(T_tail_required./v_input(7)); % Impulsive Theory
Vinf_tilde_rc   = Vinf./whov_rc;
```

```
w_tilde_rc  = sqrt((- (Vinf_tilde_rc.^2)./2 +...
              sqrt(((Vinf_tilde_rc.^2)./2).^2+1)));
w_rotor_rc  = w_tilde_rc.*whov_rc;

Pind_rc     = T_tail_required.*w_rotor_rc;

%Parasitic Power
Ppar_rc  = (v_input(8)*v_input(10)/8.*(1+4.7*mu_rc.^2)).*...
            pi*rho*v_input(9)^3*v_input(6)^5;
% Total power of the tail rotor
P_tot_rc = v_input(12)*Pind_rc + Ppar_rc;

% Total power required by total elicopter system
% in forward flight

v_P_aus  = v_input(13)*ones(1,length(Vinf));
Ptot_req = (P_tot_rp+P_tot_rc+v_P_aus)*v_input(14);
```

Maximum Speed in forward flight is determined by comparing the available Power with the total required Power. It corresponds to the value of the speed for which the available power is equal to the total required power.

The speed of maximum endurance is obtained at the minimum value of required power.

The speed of maximum range is obtained at the minimum value of the $\frac{required\ power}{speed}$ ratio.

```
% Maximum speed in forward flight
for f=(length(Vinf)/2):length(Vinf)
  if(Pdisp>=Ptot_req(f-1)  && Pdisp<Ptot_req(f))
       V_NE = (Vinf(f)+Vinf(f-1))/2;
  end
end

% Speed of maximum endurance
for f=1:length(Vinf)
 if (Ptot_req(f)==min(Ptot_req))
       V_BE = Vinf(f);
 end
end

% Speed of maximum Range
for f=1:length(Vinf)
 if(Ptot_req(f)/Vinf(f)  == min(Ptot_req./Vinf))
      V_BR = Vinf(f);
 end
end
end
```

4 Test Case

A test case was performed in order to validate the function.

```
% DATA

  % 1) Aircraft
    Wto         = 9979.03*9.81;      % [N]
    Pd_sl       = 2110000;           % [W]
    h           = 0;                 % [m]

    T0   = 288; % K
    mu0  = 1.79e-5; % Kg/(ms)
    [Temp, a0, P0, rho] = atmosisa(h);
    mu_viscoso          = mu0*(Temp/T0)^1.5*(T0+110)/(Temp+110);  % Sutherlan's law

  % 2) Main rotor                       % Tail rotor
    N         = 4;                        N_rc     = 4;
    R_rp      = 8.18;                     R_rc     = 1.7;          % [m]
    A_rp      = pi*R_rp^2;                A_rc     = pi*R_rc^2;    % [m^2]
    c         = 0.53;                     c_rc     = 0.2;          % [m]
    sigma_rp = 0.082;                     sigma_rc = 0.188;
    Vtip      = 220.98;                   Vtip_rc  = 208.79;       % [m/s]
    Omega_rp = Vtip/R_rp;                 Omega_rc = Vtip_rc/R_rc;
    n         = Omega_rp/(2*pi);          n_rc     = Omega_rc/(2*pi);

                              b = 10.73;       % [m]

    theta1_rp = -18;                        theta1_rc = -17;         % [deg]
    Num       = 10;
    r_rp      = linspace(0.05,R_rp,Num);  r_rc      = linspace(0.05,R_rc,Num);
    r_adim_rp = r_rp/R_rp;                  r_adim_rc = r_rc./R_rc;

    Cd_rc=0.01;
    Cd_rp=0.01;

    k = 1.15;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Forward Flight-Required Power at different altitudes and weights
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


numero  = 120;
Vinf    = linspace(0,120,numero);
P_aus   = 25000;  % [W]
P_ausil = P_aus*ones(1,length(Vinf));
eta_rc   = 1.03;


%Available Power

v_h = linspace(0, 1800, length(Vinf));

[~, ~, ~, rho_sl] = atmosisa(0);

for j = 1:length(v_h)
  [a,b,c,density] = atmosisa(v_h(j));
  v_Pdisponibile(j)= Pd_sl*density/rho_sl;
end
figure
plot(v_h,v_Pdisponibile, 'k');
xlabel('h [m]'); ylabel('P_a_v_a_i [W]'); title('Available power at different h');
```

```
grid on;

%Total Power at different altitudes and weights



v_WTO = [0.75*Wto,0.90*Wto, Wto];


h = [0,1000,1800];


v_input = [R_rp, A_rp, sigma_rp, Omega_rp, Cd_rp, ...
          R_rc, A_rc, sigma_rc, Omega_rc, Cd_rc, ...
          braccio, k, P_aus, eta_rc, Pd_sl];

for i = 1:length(h)
    for j = 1:length(v_WTO)

    [Pindotta_B,  Pparassita_rp,Pfus,Ptot_rp,...
     Pindotta_rc,  Pparassita_rc, Ptot_rc,Ptot_richiesta(j,:),Pdisp,...
     V_NE,V_BE, V_BR] = PowerInForwardFlight_main_tail_rotor(h(i), Vinf, v_WTO(j),...
v_input);

    Pdisponibile(j,:) = Pdisp*ones(1,length(Vinf));

    if (h(i) == 0   &&   v_WTO(j) == Wto)

% Plot
    figure
    plot(Vinf,Pindotta_B,'k-.',Vinf,Pparassita_rp,'k--',Vinf,Pfus,'k:',Vinf,Ptot_rp,'k-
',Vinf,Pdisponibile(j,:),'.-k');
    hold on;
    assey = linspace(0,Ptot_rp(end),numero);
    xlabel('V_∞ [m/s]'); ylabel('P [W]'); title('Required power of the main rotor in
forward flight’);
    subtitle(['altitude h = ',num2str(h(i)), 'm']);
    legend('P_i_n_d,'P_p_a_r,'P_f_u_s_o_l_a_g_e,'P_t_o_t','P_a_v_a_i);
    grid on


    figure
    plot(Vinf,Pparassita_rc,'k-.',Vinf,Pindotta_rc,'k--',Vinf,Ptot_rc,'k-');
    title('Powers of the tail rotor'); xlabel('V_∞ [m/s]'); ylabel('P [W]');
    subtitle(['altitude h = ',num2str(h(i)), 'm']);
    legend('P_p_a_r,'P_i_n_d,'P_t_o_t');
     grid on;


     figure
     plot(Vinf,Ptot_rp,'k-.',Vinf,Ptot_rc,'k--
',Vinf,P_ausil,'k:',Vinf,Ptot_richiesta(j,:),'k-',Vinf,Pdisponibile(j,:),'.-k');
     hold on;
     xlabel('V_∞ [m/s]'); ylabel('P [W]'); title('Total required power in forward
flight');
     subtitle(['altitude h = ',num2str(h(i)), 'm']);
     legend('P_m_a_i_n_ _r_o_t_o_r','P_t_a_i_l_ _r_o_t_o_r', 'P_a_u_x', 'P_t_o_t',
'P_a_v_a_i');
     grid on

    end
 end


    figure
    plot(Vinf,Ptot_richiesta(j,:),'.-k');
    xlabel('V_∞ [m/s]'); ylabel('P [W]');
    subtitle(['altitude h = ',num2str(h(i)), 'm']);
    hold on;
    plot(Vinf,Pdisponibile(j,:),'k');
    assey = linspace(0,Ptot_richiesta(end),numero);
    plot(V_NE*ones(1,numero),assey,'k-.');
    plot(V_BE*ones(1,numero),assey,'k--');
    plot(V_BR*ones(1,numero),assey,'k');
    legend('P_t_o_t','P_d_i_s_p','V_N_E','V_B_E,'V_B_R');


    figure
      plot(Vinf, Ptot_richiesta(1,:), 'k- ',...
      Vinf, Ptot_richiesta(2,:), 'k--',...
      Vinf, Ptot_richiesta(3,:), 'k-.',...
      Vinf, Pdisponibile(j,:)  , '.-k ');

    title('Total required power in forward flight at different W_T_O')
    subtitle(['altitude h = ',num2str(h(i)), 'm']);
    xlabel('V_∞ [m/s]'); ylabel('P [W]');
    legend('75% WTO',...
           '90% WTO',...
            'WTO',...
            'P_a_v_a_i_l_a_b_l_e);

    grid on;

end
```

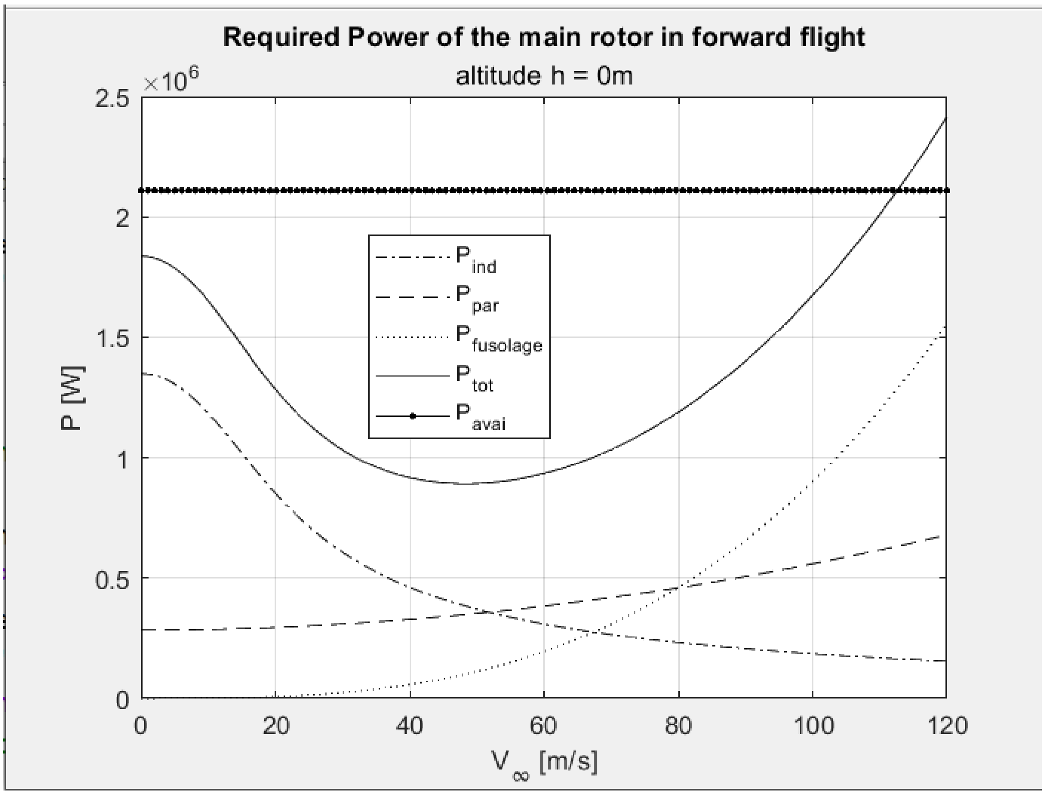

Figura 4.1: Powers of the main rotor in forward flight at Sea level.

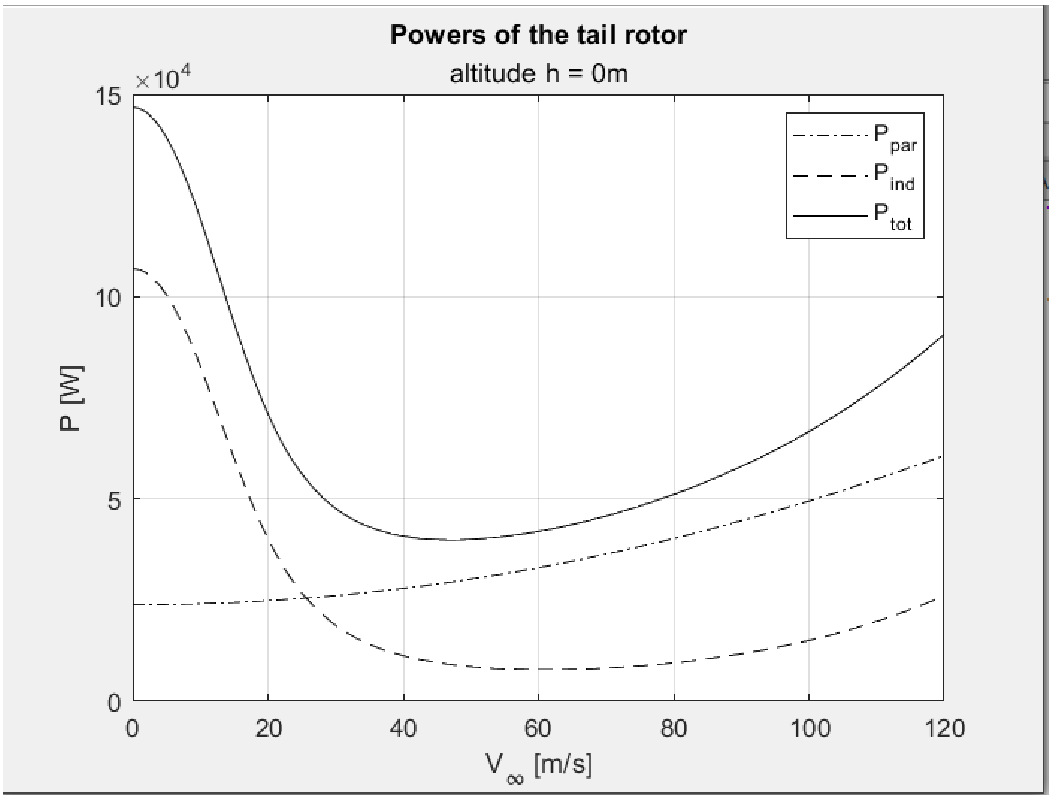

Figura 4.2: Powers of the tail rotor in forward flight at Sea level.

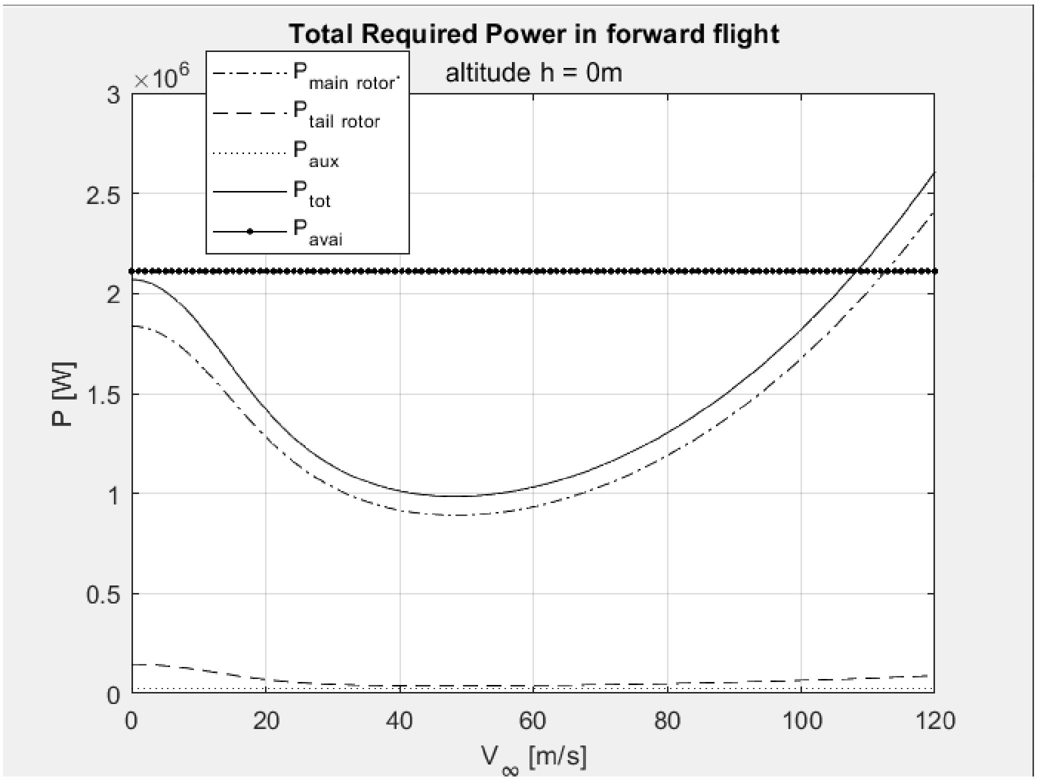

Figura 4.3: Total required powers in forward flight at Sea level.

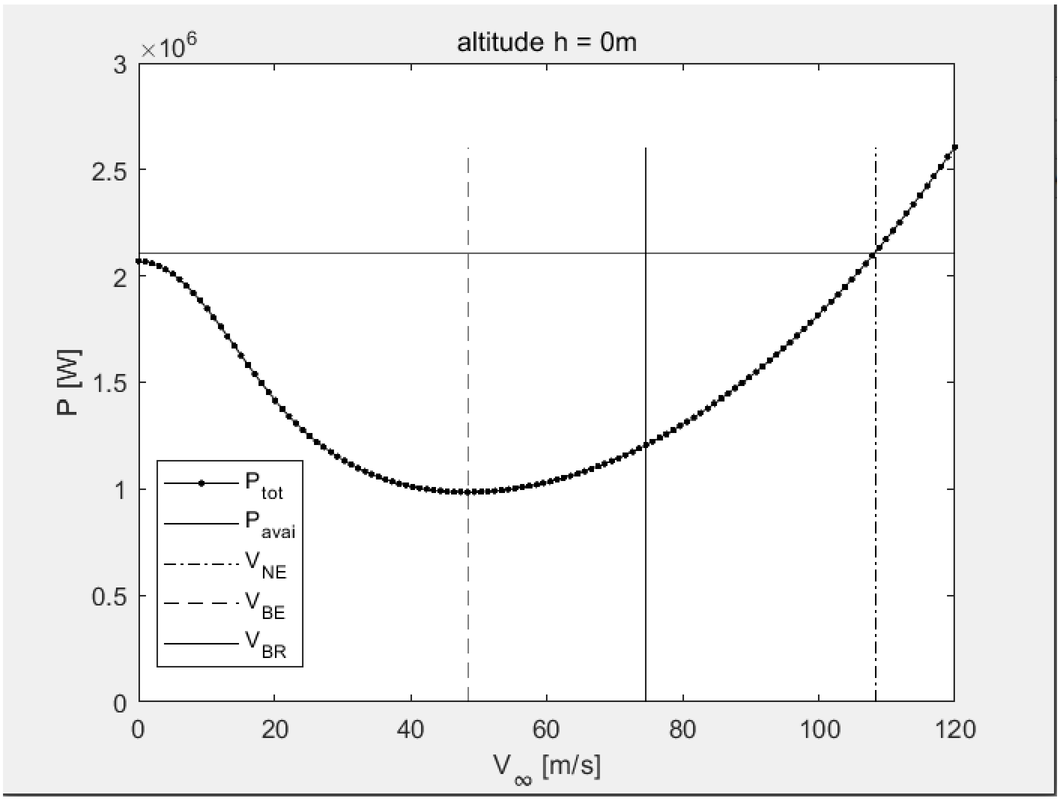

Figura 4.4: Maximum speed, speed of maximum endurance and speed of maximum range at Sea level.

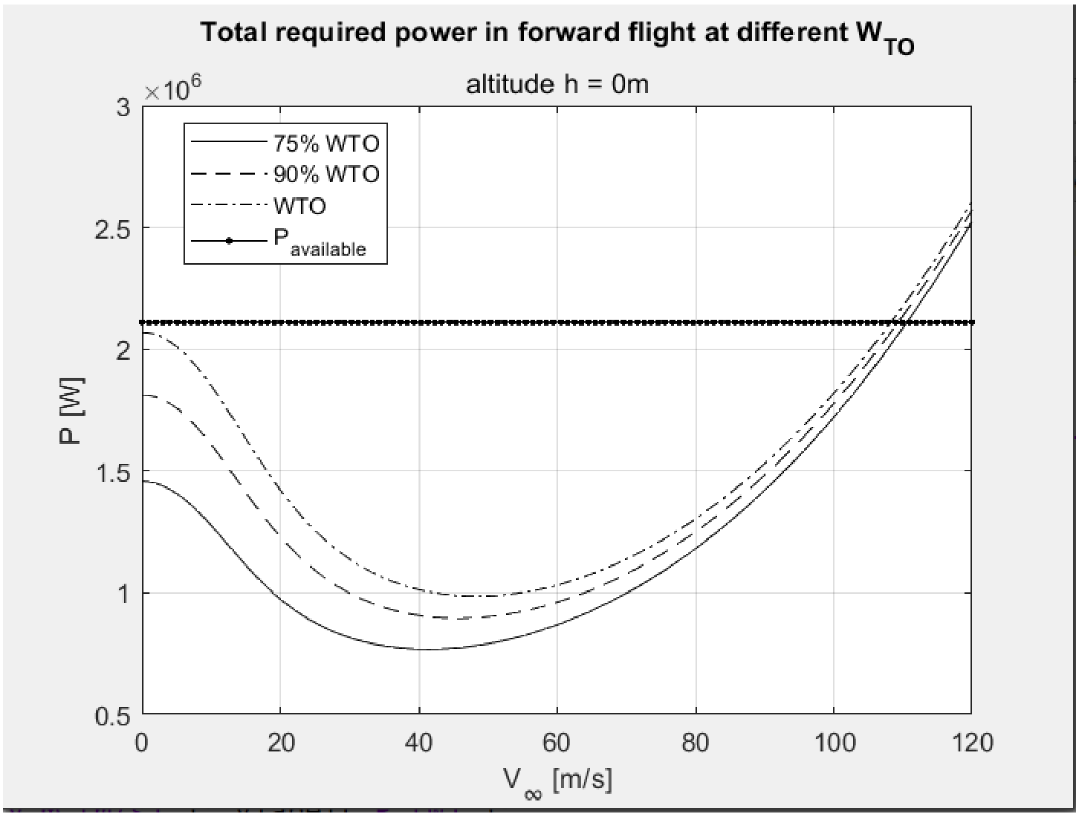

Figura 4.5: Total required power at different weights at Sea Level.

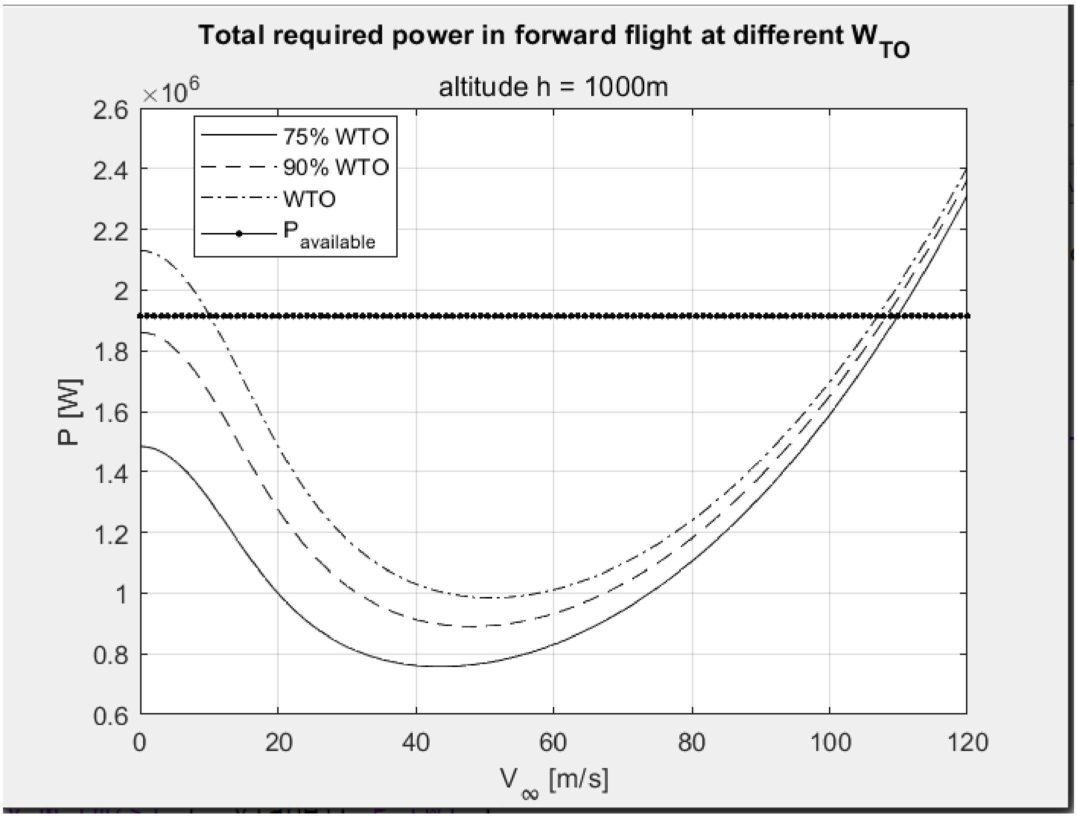

Figura 4.6: Total required power at different weights at 1000m.

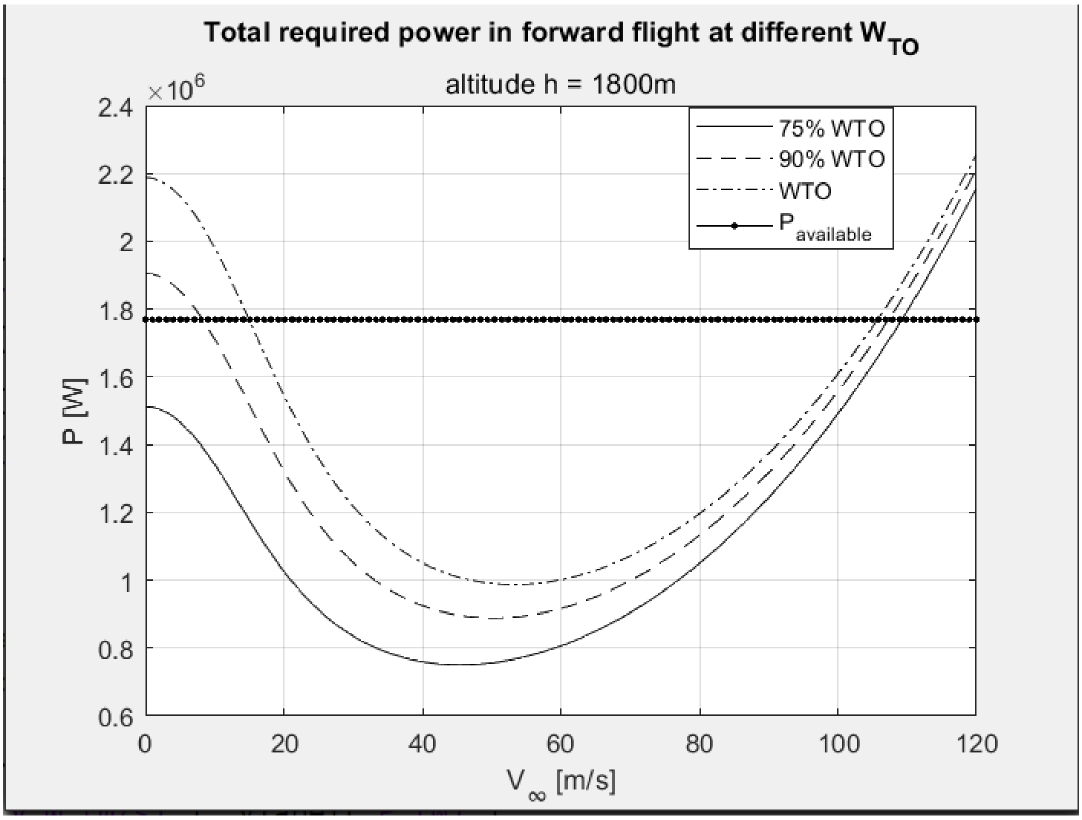

Figura 4.7: Total required power at different weights 1800m.

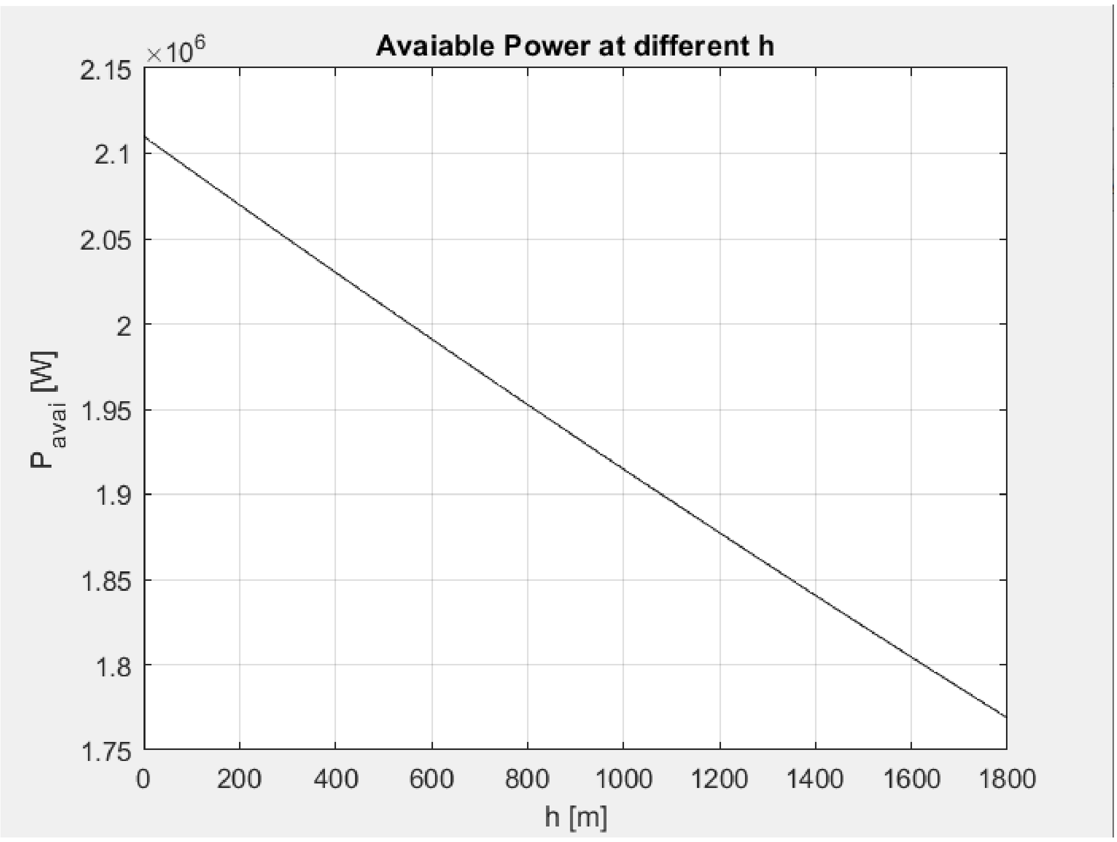

Figura 4.8: Available Power at different altitudes.