



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Eléctrica

**Movilidad eléctrica en el transporte de
viajeros.
ANEXOS**

Autor:

Vallejo Luis, Raúl

Tutor:

**Alonso Ripoll, Francisco Javier
Departamento de Ingeniería
Eléctrica**

Valladolid, Julio de 2019

1. Programa de simulación de la aceleración en el arranque:

```
% Irizar ie bus 12m *****
% Simulación de la aceleración de un autobús eléctrico.

% Elegimos el tiempo de la simulación:
t = linspace (0, 50, 501); % 0 a 50 segundos, con un paso de 0.1 seg.
v = zeros(1, 501); % 501 lecturas de velocidad
d = zeros(1, 501); % 501 lecturas de distancia
dT = 0.1; % 0.1 segundos de paso de tiempo

% Parámetros del autobús:
m = 18000; % masa del vehículo en kg
G = 7; % Relación de transmisión
r = 0.45; % Radio de las ruedas (m)
ng = 0.95; % Rendimiento de la transmisión
urr = 0.008; % Coeficiente de resistencia a la rodadura para autobuses
A = (3.209 - 0.34) * 2.55; % Área frontal del autobús
Cd = 0.29; % Coeficiente aerodinámico
g = 9.8; % Aceleración de la gravedad (m/s^2)

% Datos del motor:
Pmax = 180000; % Potencia del motor (W)
Tmax = 1500; % Par del motor (Nm)
vbase = (r/G) * (Pmax/Tmax); % Velocidad base del motor (m/s)
vmax = (r/G) * 3500 * (2*pi/60); % Velocidad máxima del motor (m/s)

% En este caso hay tres fases de aceleración.
for n = 1:500
    if v(n) < vbase
        v(n+1) = v(n) + dT * ((G/(r * m * 1.05)) * ng * Tmax - urr * g * (1/1.05) - 0.625 * A * Cd * (1/(1.05 * m)) * (v(n)^2));
    elseif v(n) > vmax
        % El controlador detiene cualquier aumento de velocidad una vez que
        % se alcance la velocidad máxima del vehículo.
        v(n+1) = v(n);
    else
        v(n+1) = v(n) + dT * (- urr * g * (1/1.05) + (1/(m * 1.05)) * ng * Pmax * (1/(v(n))) - 0.625 * A * Cd * (1/(1.05 * m)) * (v(n)^2));
    end
    d(n+1) = d(n) + dT * v(n);
end

v = v.*3.6; % Multiplicamos los valores de v por 3600(s/h)/1000(m/km) para
convertir las unidades de m/s a km/h
```

```
figure;  
subplot(1, 2, 1);  
plot (t, v);  
xlabel ('Tiempo (s)');  
ylabel ('Velocidad (km/h)');  
  
subplot(1, 2, 2);  
plot (t, d);  
xlabel ('Tiempo (s)');  
ylabel ('Distancia recorrida (m)');
```

2. Simulación de la autonomía del Irizar ie bus 12m:

```
% Simulación del autobús Irizar ie bus 12m mientras se le somete
% a un ciclo de conducción.
% El ciclo se continúa repitiendo hasta que la profundidad de descarga de
% la batería alcanza un 90%.

% Llamada al ciclo de conducción.
sfuds;
% Ciclo_ECE15;
% Ciclo_EUDC;
% Ciclo_NEDC;      % El ciclo NEDC es 4 veces el ECE15 y una vez el EUDC
% Ciclo_WLTP;      % Ciclo sustituto del NEDC
% Ciclo_ArtemisUrbano;

N = length(V); % Encontrar el número de lecturas

% Dividimos la velocidad entre 3.6 para pasarlas de km/h a m/s
V = V./3.6;

% Introducimos los datos del vehículo.
masa = 18000;          % Masa del vehículo.
area = (3.209 - 0.34) * 2.55; % Área frontal del vehículo
Cd = 0.29;             % Coeficiente aerodinámico
G = 7;                % Relación de transmisión
r = 0.45;             % Radio de las ruedas (m)
Gratio = G/r;
G_eff = 0.95;         % Rendimiento
Regen_ratio = 0.5;

bat_type = 'IL';      % Elegir el tipo de batería: IL, NC, LA
NoCeldas = 194;       % Número de celdas de la batería
Capacidad = 643;      % Capacidad de la batería.
k = 1.12;             % Coeficiente de Peukert
Pac = 25000 ;         % Potencia de accesorios (W)

% Constantes de eficiencia del motor
kc = 0.1;             % Pérdidas en el cobre
ki = 0.005;          % Pérdidas en el hierro
kw = 0.0000025;      % Pérdidas aerodinámicas
ConL = 800;           % Pérdidas por constantes electrónicas

% Algunas constantes que aun han de ser calculadas:
urr = 0.008;          % Coeficiente de resistencia a la rodadura para
autobuses
Frr = urr * masa * 9.8;
Rin = (0.022/Capacidad) * NoCeldas;
Rin = Rin + 0.05;     % Aumentamos la resistencia por los cables de conexión
PeuCap = ((Capacidad/10)^k) * 10;
```

```
DoD_end = zeros(1,100);
CR_end = zeros(1,100);
D_end = zeros(1,100);

DoD = zeros(1,N);      % Tasa de descarga
CR = zeros(1,N);      % Carga que sale de la batería, corregida con el coef.
                        % Peukert
D = zeros(1,N); % Distancia recorrida en km.

CY = 1;
DD = 0; % Valor inicial

while DD < 0.9 %Fijamos como profundidad de decarga para que pare el
programa el 90%

    % Comenzamos un ciclo. *****
    % Hacemos la llamada al script
    % un_ciclo que es el que se encarga de
    % simular un ciclo del vehículo.

    un_ciclo;

    % Un ciclo completo se ha simulado.
    % Ahora actualizamos los valores.

    DoD_end(CY) = DoD(N);
    CR_end(CY) = CR(N);
    D_end(CY) = D(N);

    % Reseteamos los valores de las matrices para que estén listos para la
    % siguiente iteración.

    DoD(1) = DoD(N);
    CR(1) = CR(N);
    D(1) = D(N);
    DD = DoD_end(CY);

    % Fin de un ciclo *****

    CY=CY+1;
end

figure;
plot(D_end, DoD_end, 'k+');
ylabel('Tasa de descarga');
xlabel('Distancia recorrida (km)');
title('Autonomía del Irizar ie bus 12m');
```

3. Simulación de un ciclo:

```
% *****
%
%                               UN CICLO
% *****
% Este script realiza un ciclo de un vehículo con N puntos,
% empleando las baterías de Plomo Ácido, Níquel Cadmio e
% Ion-Litio. Todas las variables deben estar definidas en
% el programa de llamada.
% *****

for C = 2:N
    aceleracion = V(C) - V(C-1);
    % Fuerza de tracción del vehículo.
    Fad = 0.5 * 1.25 * area * Cd * V(C)^2; % Fuerza aerodinámica.
    Fhc = 0; % Fuerza de subir una pendiente.
    Fla = 1.05 * masa * aceleracion; % Fuerza para la aceleración lineal.
    % La Fwa (Fuerza para la aceleración angular) puede ser omitida si
    % aumentamos la masa del vehículo un 5% en la Fla.

    % Potencia de tracción.
    Pte = (Frr + Fad + Fhc + Fla) * V(C);

    omega = Gratio * V(C);

    if omega == 0 %Vehículo parado
        Pte = 0;
        Pmot_in = 0; % No hay potencia en el motor.
        Par = 0;
        eff_motor = 0.5; % Valor ficticio, para asegurarnos de que no es cero
    elseif omega > 0 %Vehículo en movimiento
        if Pte < 0
            Pte = Regen_ratio * Pte; % Se reduce la potencia, porque no es ideal el motor.
        end

        % Ahora se calcula la potencia del motor, que es diferente de la
        % potencia de las ruedas, ya que hay que tener en cuenta el
        % rendimiento de la transmisión.
        if Pte >= 0
            Pmot_out = Pte/G_eff; % Potencia del motor < Potencia del eje
        elseif Pte < 0
            Pmot_out = Pte * G_eff; % Potencia del motor disminuida si está frenando el
            motor
        end
    end

    Par = Pmot_out / omega; % Ecuación de tracción: Potencia = Par * velocidad
    if Par > 0
```

MOVILIDAD ELÉCTRICA EN EL TRANSPORTE DE VIAJEROS

```
eff_mot = (Par * omega)/((Par * omega) + ((Par^2) * kc) + (omega * ki) +  
((omega^3) * kw) + ConL);  
elseif Par < 0  
eff_mot = (- Par * omega)/((- Par * omega) + ((Par^2) * kc) + (omega * ki) +  
((omega^3) * kw) + ConL);  
end  
  
if Pmot_out >= 0  
Pmot_in = Pmot_out/eff_mot;  
elseif Pmot_out < 0  
Pmot_in = Pmot_out * eff_mot;  
end  
  
end  
  
Pbat = Pmot_in + Pac;  
  
% Elección del tipo de batería:  
if bat_type == 'NC'  
E = open_circuit_voltage_NiCd (DoD(C-1), NoCeldas);  
elseif bat_type=='LA'  
E = open_circuit_voltage_PbA (DoD(C-1), NoCeldas);  
elseif bat_type=='IL'  
E = open_circuit_voltage_IonLi (DoD(C-1), NoCeldas);  
else  
error('Invalid battery type');  
end  
  
if Pbat > 0  
I = (E - ((E * E) - (4 * Rin * Pbat))^0.5)/(2 * Rin);  
CR(C) = CR(C-1) + ((I^k)/3600);  
elseif Pbat == 0  
I = 0;  
elseif Pbat < 0  
% Al utilizar el frenado regenerativo, se considera el doble de resistencia  
% interna.  
Pbat = - 1 * Pbat;  
I = ( - E + (E * E + (4 * 2 * Rin * Pbat))^0.5)/(2 * 2 * Rin);  
CR(C) = CR(C-1) - (I/3600);  
29/06/19 19:20 F:\Raul HP\UNI\4º Curso\2º ... \un_ciclo.m 3 of 3  
end  
DoD(C) = CR(C)/PeuCap;  
if DoD(C) > 1  
DoD(C) = 1;  
end  
  
% Como estamos hablando de intervalos de tiempo de un segundo, la  
% distancia recorrida en metros, es la misma que la velocidad.  
% Dividimos por 1000 para pasar a km.
```


$D(C) = D(C-1) + (V(C)/1000);$

% Datos para gráficas:

% Par / Velocidad

XDATA(C) = omega;

YDATA(C) = Par;

% Potencia del motor / Tiempo

% XDATA(C) = C;

% YDATA(C) = Pmot_out;

% Aceleración / Tiempo

% XDATA(C) = C;

% YDATA(C) = aceleracion;

% Eficiencia del motor / Tiempo

% XDATA(C) = C;

% YDATA(C) = eff_mot;

% Intensidad / Tiempo

% XDATA(C) = C;

% YDATA(C) = I;

% Tensión de la batería / Tiempo

% XDATA(C) = C;

% YDATA(C) = E;

end

4. Curva de tracción del Irizar ie bus 12 m:

%% Curva de tracción del motor síncrono de imanes permanentes

% Datos del motor:

Pot_max = 180000; % Potencia en W

M_max = 1500; % Par en Nm

Wb = Pot_max/M_max;

Wmax = round(3500 * 2*pi/60);

W_1 = (0:01:Wb);

W_2 = (Wb+1:01:Wmax);

t = length(W_1);

tt = length(W_2);

% Curva potencia

for n = 1 : t

 P_1(n) = W_1(n) * M_max;

end

for n = 1 : tt

 P_2(n) = Pot_max;

end

W = [W_1, W_2];

P = [P_1, P_2];

P = P/1000; % (KW)

figure; yyaxis right; plot(W, P)

ylim([0 300]);

title ('Curva de tracción del Irizar ie bus 12m');

xlabel('Velocidad (rad/s)');

ylabel('Potencia (KW)');

% Curva par

for n=1:t

 M_1(n) = M_max;

end

for n=1:tt

 M_2(n) = Pot_max /W_2(n);

end

W = [W_1,W_2];

M = [M_1,M_2]; %(Nm)

hold on; yyaxis left; plot(W, M);

ylabel('Par (Nm)');