



**Corso di Laurea Magistrale in
Ingegneria Elettronica**

Gestione Energetica dei Veicoli a Trazione Elettrica e Ibrida

A.A. 2020/2021

**Relazione progettuale sulla realizzazione di un sistema
di gestione energetica di un veicolo ibrido**

Studenti:

Mario Andrea Sangiovanni 214599

Andrea Alecce 214611

Andrea Migali 214593

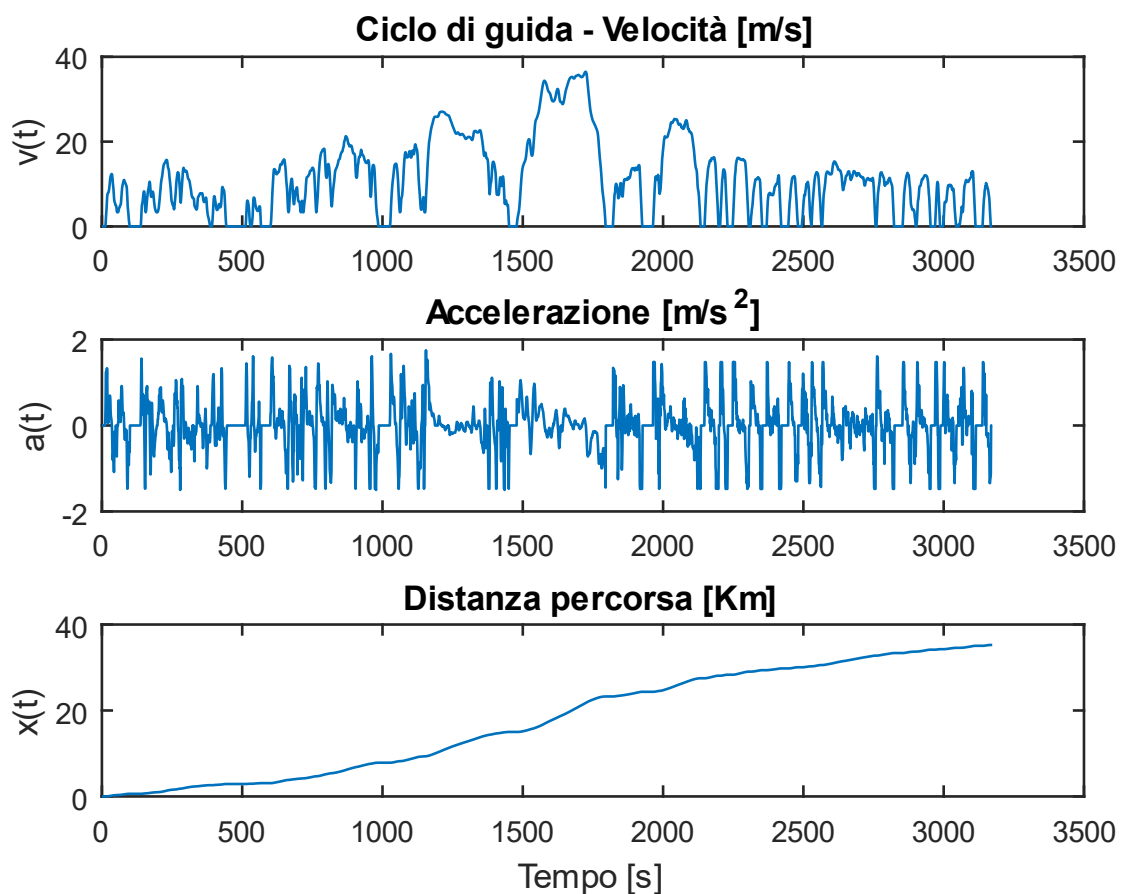
Prof.:

Marco Ricci

INTRODUZIONE

L'obiettivo del progetto consiste nella realizzazione di un sistema di gestione energetica di un veicolo ibrido. In particolare, il software, a partire dai parametri relativi alla dinamica del veicolo, ai motori a combustione ed elettrico, e al ciclo di guida, fornisce in uscita la traiettoria ottimale di controllo del motore elettrico. Più in dettaglio, il principale parametro di controllo è rappresentato dalla coppia del motore elettrico e la traiettoria ottimale è quella che porta al valore desiderato di Stato di Carica finale della batteria, minimizzando la potenza richiesta dal motore a combustione interna.

Le simulazioni sono state effettuate prendendo in considerazione i vari cicli di guida standard utilizzati normalmente in ambito industriale. In particolare, i risultati riportati in questa relazione sono riferiti a due diversi cicli, il WLTC e il FUDS, che sono stati concatenati per formare un unico ciclo di durata maggiore. I cicli forniscono la velocità istantanea del veicolo. A partire da essi è possibile ricavare accelerazione istantanea e distanza percorsa. I dati ottenuti sono illustrati nei grafici seguenti:



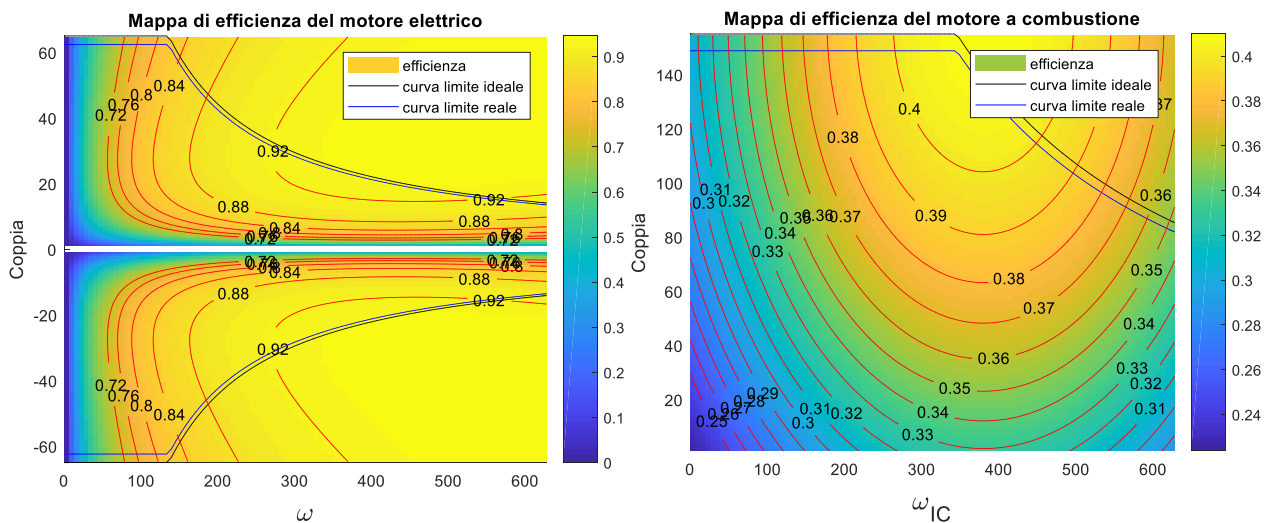
I parametri utilizzati e relativi alla dinamica del veicolo sono i seguenti:

Frontal area, A_f	2.33 m ²
Drag coefficient, C_d	0.26
Air density, ρ_{air}	1.22 kg/m ³
Roll. resist. coeff, c_{roll}	0.024
Total vehicle mass, M	1370 kg
Wheel radius, R_{wh}	0.32 m
Final drive ratio, g_f	4.113

I parametri relativi alla batteria presente sul veicolo simulato sono i seguenti:

Nominal charge capacity, Q_{nom}	6.5 Ah = 23400 C
Max voltage V_{max}	248 V
Energy capacity $E_{batt} = Q_{nom} V_{oc,nom}$	5.8 MJ = 1.6 kWh
Coulomb efficiency, η_{coul}	0.95
Max current (charge or discharge) I_{max}	130 A
Max power (charge or discharge) $P_{batt,max}$	31 kW

Per quanto riguarda i due motori, elettrico e a combustione, questi sono caratterizzati dalle rispettive mappe di efficienza, illustrate nei grafici seguenti:



La mappa di efficienza del motore elettrico presenta una curva limite caratterizzata da un tratto a coppia costante fino ad un valore di w critico e da un tratto a potenza costante, da w critico in poi. Il valore di efficienza per ogni punto $[w, T]$, nel caso di trazione, è ottenuto dalla seguente formula:

$$\eta = \frac{w * T}{w * T + w^3 * K_w + w * K_i + T^2 * K_c + C}$$

Mentre nel caso di frenata:

$$\eta = w * T - (w^3 * K_w + w * K_i + T^2 * K_c)$$

I parametri utilizzati hanno i seguenti valori:

$$\begin{aligned} Kc &= 0.40; \\ Ki &= 8e - 3; \\ Kw &= 0.2e - 5; \\ C &= 300; \end{aligned}$$

Nel caso del motore a combustione, invece, la mappa di efficienza presenta una curva limite approssimata nella quale è presente un tratto a coppia costante e uno a potenza costante. Per il calcolo dell'efficienza è stata utilizzata la relazione

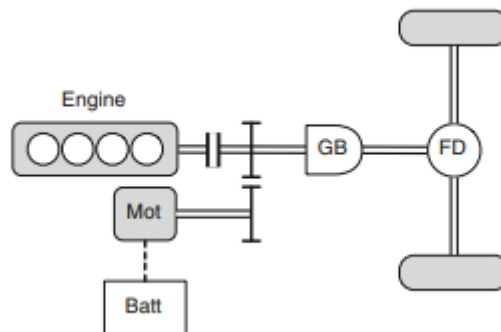
$$\eta_{ICE} = -(0.8e - 8) * (w - 3630/60 * 2 * \pi)^2 - (3.2e - 6) * (T - 160)^2 + 0.41;$$

Nella quale i valori di w e T sono calcolati tenendo conto del ciclo di guida e del rapporto di trasmissione utilizzato.

DESCRIZIONE TEORICA

Il modello di riferimento preso in considerazione per la simulazione è caratterizzato da un veicolo ibrido con architettura parallela (visibile in figura) e accoppiamento in coppia. In queste condizioni, le velocità angolari dei due motori coincidono e vale la relazione:

$$T_{pwt} = g_{tr}(i_{tr}) \cdot g_{fd} \cdot (T_{eng} + T_{mot})$$



Un altro modello importante da tenere in considerazione è il modello di scarica della batteria. È necessario tenere conto del fatto che, maggiore è la corrente erogata, maggiore è il rate di scarica della batteria. Per fare questo abbiamo utilizzato il modello di Peukert per il quale si considera, nel caso di batteria che eroga corrente, una variazione dello stato di carica pari a:

$$D_{SOC_p}(i) = ((I_p(i))^k) * dt / Q_0$$

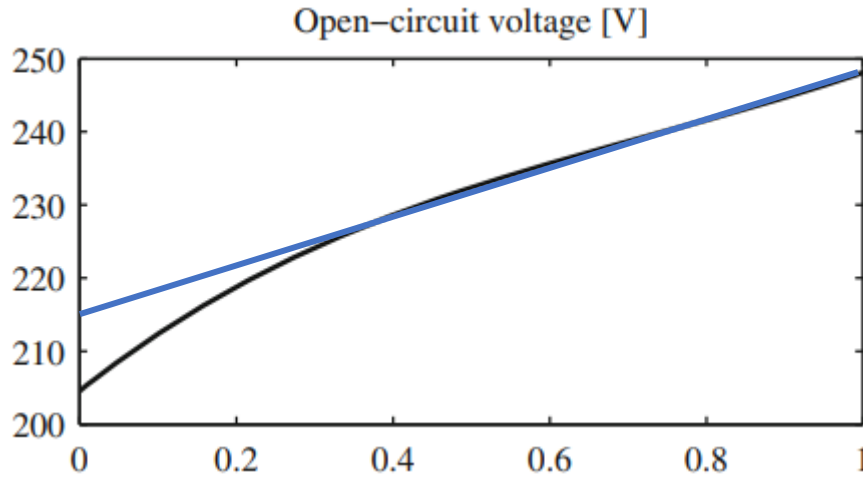
Se invece la batteria viene ricaricata, la variazione dello stato di carica è data da:

$$D_{SOC_n}(i) = (I_n(i)^{1/k}) * dt / Q_0$$

dove $K = 1.01$.

Inoltre, è necessario modellare la variazione della tensione ai capi della batteria in funzione dello stato di carica. Per fare questo, la curva reale di una batteria è stata approssimata con una retta data da

$$V(i) = 215 + 33 * (SOC(i))$$



L'approssimazione lineare utilizzata si basa sull'ipotesi che durante il normale utilizzo nel ciclo di guida, lo stato di carica non scenderà a valori troppo bassi.

Definiti i modelli di riferimento, il primo passo consiste nel calcolare la forza di trazione richiesta al veicolo sulla base dei parametri del veicolo, del ciclo di guida e del bilancio delle forze in gioco:

$$F_{tr}(i) = \mu_{r0} * m * g * \cos(\alpha) + R_g + (0.5 * A * C_d * \rho * (v(i)^2) + (m_{eff} * a(i)));$$

Calcolata la forza di trazione, è possibile risalire alla potenza richiesta nota la velocità fissata dal ciclo di guida e, di conseguenza, alla coppia richiesta. Questa coppia deve essere ripartita tra motore a combustione e motore elettrico. A partire dalla coppia richiesta dal motore elettrico e considerando la sua mappa di efficienza, oltre al modello di Peukert, si può risalire alla variazione dello stato di carica della batteria.

L'ottimizzazione della gestione energetica del veicolo, col fine di minimizzare il consumo di carburante e raggiungere uno stato di carica finale della batteria fissato, consiste proprio nel determinare, istante per istante, il valore ottimale della coppia del motore elettrico e quindi la suddivisione della coppia tra i due motori. Per fare questo, è stata utilizzata una tecnica di ottimizzazione numerica basata sul principio di minimalità di Pontryagin e quindi sulla minimizzazione di una funzione Hamiltoniana. Questa funzione contiene al suo interno la dinamica del sistema e la funzione costo istantaneo.

Il costo istantaneo è rappresentato dall'integrale nel tempo della potenza del motore a combustione, che è una funzione della coppia del motore e della sua velocità angolare:

$$J = Q_{lhv} \int_{t_0}^{t_f} \dot{m}_f(T_{eng}, \omega_{eng}) dt = \int_{t_0}^{t_f} P_{fuel}(T_{eng}, \omega_{eng}) dt$$

Per quanto riguarda la dinamica del sistema, questa è rappresentata dalla variazione nel tempo dello stato di carica della batteria. Tuttavia, risulta più comodo descrivere il sistema in termini di variazione temporale dell'energia della batteria, cioè in termini di potenza elettrochimica fornita/assorbita dalla batteria.

Si ottiene quindi un'espressione della funzione Hamiltoniana che contiene al suo interno la potenza del motore a combustione e quella della batteria:

$$\begin{aligned} H(T_{mot}, E_{ech}, T_{gb}, v_{veh}) \\ = P_{fuel}(T_{mot}, T_{gb}, v_{veh}) + (\lambda + w(SOC)) P_{ech}(T_{mot}, E_{ech}, v_{veh}) \end{aligned}$$

La funzione $w(SoC)$ rappresenta una penalità che permette di far sì che non ci sia una scarica eccessiva della batteria ad ogni istante temporale associando un peso maggiore alla P_{ech} quando il SoC all'istante

precedente è basso e un peso minore quando il SoC all'istante precedente è alto. Ad esempio, questa funzione può essere valutata come

$$w_{SOC} = 5 * (0.9 - SOC(i - 1))$$

Dato un valore di λ iniziale, ad ogni istante di tempo, il valore ottimale di coppia del motore elettrico è quello che minimizza la funzione Hamiltoniana. Questo valore può essere ricavato numericamente prendendo un vettore di possibili valori di coppia che rispettano i vincoli del motore, calcolando le corrispondenti Hamiltoniane e identificando quella minima. Il valore di coppia corrispondente a questa Hamiltoniana rappresenta il valore ottimo del parametro di controllo per quell'istante di tempo.

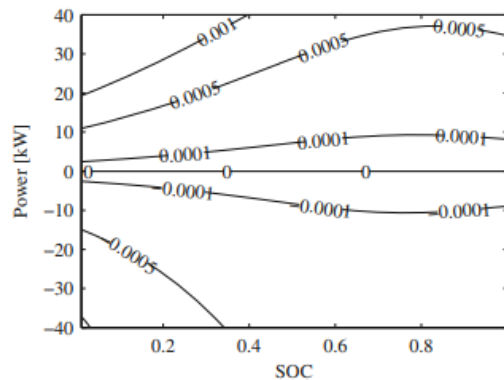
A partire da questo valore è possibile determinare tutti i rimanenti parametri del sistema, in particolare:

- efficienze dei due motori
- potenze dei due motori
- potenza che la batteria deve fornire/assorbire dal motore/generatore elettrico
- corrente che la batteria deve fornire/assorbire dal motore/generatore elettrico
- variazione dello stato di carica tenendo conto del modello di Peukert
- aggiornamento dello stato di carica attuale
- aggiornamento del valore di tensione ai capi della batteria

Per procedere con la simulazione all'istante di tempo successivo, bisogna aggiornare anche il valore di λ seguendo la relazione:

$$\dot{\lambda}(T_{mot}, E_{ech}, v_{veh}) = -\frac{\partial H}{\partial E_{ech}} = -\lambda \frac{\partial \dot{SOC}(P_{batt}, SOC)}{\partial SOC}$$

Il termine $\frac{\partial \dot{SOC}}{\partial SOC}$ varia in funzione di SoC e Pbatt nel modo illustrato nel seguente grafico:



Per effettuare le simulazioni, il suo andamento è stato approssimato supponendo che il SoC non scenda a valori troppo bassi e quindi l'evoluzione di λ è calcolata come:

$$P_{pk_n}(i) = I_n(i)^{1/k} * V(i);$$

$$P_{pk_p}(i) = I_p(i)^k * V(i);$$

$$dSOCp(i) = ((5e - 4)/(35e3)) * ((P_{pk_p}(i) - P_{pk_n}(i))/1e3);$$

$$\lambda(i + 1) = L1(i) - dt * \lambda(i) * dSOCp(i);$$

dove i termini P_{pk} rappresentano le potenze erogata ed assorbita dalla batteria secondo il modello di Peukert.

Procedendo in questo modo, si ottiene la traiettoria di controllo ottimo corrispondente al valore di λ iniziale preso in considerazione. Per questa traiettoria è possibile calcolare la funzione costo complessiva e il valore di SoC finale ottenuto.

A questo punto, il procedimento deve essere ripetuto per diversi valori di λ iniziali, applicando cioè il cosiddetto Shooting Method. Ad ognuno di questi valori corrisponderà una traiettoria di controllo ottimo che porterà ad un certo valore di SoC finale. Tra tutte queste traiettorie, quella ottima è quella che porta al valore di SoC finale più vicino a quello desiderato, con una certa tolleranza.

Per fare questo si può ricorrere al metodo della bisezione che, a partire dai due valori di λ iniziali le cui traiettorie portano rispettivamente al primo SoC finale maggiore del valore desiderato e al primo SoC finale minore di quello desiderato, ricava in maniera iterativa il valore di λ iniziale la cui traiettoria ottima porterà al valore di SoC desiderato.

IMPLEMENTAZIONE SOFTWARE

Il codice realizzato è suddiviso in sei scripts:

- “Progetto.m” : costituisce il corpo principale del codice da eseguire;
- “bisezione_soc.m”: implementa l’algoritmo di bisezione;
- “ciclo_tempo.m”: è richiamato nel file “bisezione_soc.m” ed implementa il ciclo temporale con il calcolo dei vari parametri del sistema;
- “mappa_eta_EL_gen.m”: contiene la descrizione della mappa di efficienza del motore elettrico nel caso di frenata rigenerativa;
- “mappa_eta_EL_mot.m”: contiene la descrizione della mappa di efficienza del motore elettrico nel caso di trazione;
- “mappa_eta_ICE.m”: contiene la descrizione della mappa di efficienza del motore a combustione;

Le mappe di efficienza tengono conto anche dei rapporti di trasmissione.

Progetto.m

La prima parte del codice richiama i due cicli di guida utilizzati, li concatena e definisce i vari parametri utilizzati nel resto del codice.

A questo punto, l’evoluzione dell’algoritmo è descritta attraverso due cicli for innestati. Il primo implementa lo Shooting Method sui valori iniziali di λ , il secondo calcola le traiettorie temporali e quindi l’evoluzione nel tempo del sistema.

```
x(i) = x(i-1) + dt*(v(i)+v(i-1))/2;  
mu_r=mu_r0*(1+v(i)/100);  
Rr0=mu_r*m*g;  
Rr=Rr0*cosd(alpha); %resistenza al rotolamento  
Ftr(i)=Rr+Rg+(0.5*A*Cd*ro*(v(i)^2)+(m_eff*a(i)));
```

Questa prima parte del ciclo temporale calcola la forza di trazione, la velocità angolare e la potenza totale da dividere tra i due motori. Da questa si risale alla coppia totale.

```
Ftr_p(i)=max(0,Ftr(i));  
Ftr_n(i)=abs(min(0,Ftr(i)));
```

```
P_p(i)=Ftr_p(i)*v(i);  
P_n(i)=Ftr_n(i)*v(i);
```

```
P_eff_p(i)=P_p(i)/eff_mech;%P_eff=potenza effettiva  
P_eff_n(i)=P_n(i)*eff_mech;
```

```
w(i)=v(i)/r*G0; %w motore  
v_rpm=(60/(2*pi))*w(i);
```

La sezione successiva del codice si occupa del calcolo del valore ottimo di coppia del motore elettrico per quell’istante di tempo attraverso la minimizzazione dell’Hamiltoniana. Queste operazioni vengono effettuate in maniera separata per il caso di trazione, cioè con forza di trazione positiva, e per quello di

frenata rigenerativa, cioè con forza di trazione negativa. In quest'ultimo caso, la funzione Hamiltoniana non contiene il termine legato al motore a combustione in quanto in questa fase la sua potenza è nulla.

```

if (Ftr(i)>0)
    if ((T_p(i)/(G*G0))>Tmax_el_w)
        Tstop=Tmax_el_w;
    else
        Tstop=T_p(i)/(G*G0);
    end
    T_opt=linspace(0,Tstop,1000);
    T_ice=T_p(i)/(G*G0)-T_opt;
    eff_motore_ic_vec=mappa_eta_ICE(w(i),T_ice,E0);
    Pfuel_vec=(T_ice*w(i))./eff_motore_ic_vec;
    eff_motore_el_vec=mappa_eta_EL_mot(w(i),T_opt,E0);
    P_batt_p_vec=T_opt*w(i)./eff_motore_el_vec;
    I_p_vec=P_batt_p_vec./V(i-1);
    D_SOC_p_vec=((I_p_vec).^k)*dt/Q0;
    SOC_vec=SOC(i-1)-D_SOC_p_vec;
    V_vec=215+m1.*(SOC_vec);
    P_pk_p_vec=I_p_vec.^k.*V_vec;
    SOCp_vec=-(1/Q0)*(V_vec/(2*R0)-sqrt((V_vec/(2*R0)).^2 ...
        -(P_pk_p_vec/R0)));
    Pech_vec=-V_vec*Q0.*SOCp_vec;
    w_SOC=5*(0.9-SOC(i-1));
    H_vec=Pfuel_vec+(L1(i)+w_SOC).*Pech_vec;
    [Hmin,min_index]=min(H_vec);
    Tel_p(i)=T_opt(min_index);
    Tel_n(i)=0;
else
    else
        if (T_n(i)>Tmax_el_w)
            Tstop=-Tmax_el_w;
        else
            Tstop=-T_n(i);
        end
        T_opt=linspace(Tstop,0,1000);
        eff_motore_el_vec=mappa_eta_EL_gen(w(i),T_opt,E0);
        P_batt_n_vec=T_opt*w(i)./eff_motore_el_vec;
        I_n_vec=-P_batt_n_vec./V(i-1);
        D_SOC_n_vec=((I_n_vec).^k)*dt/Q0;
        SOC_vec=SOC(i-1)+D_SOC_n_vec;
        SOC_vec(find(SOC_vec>1))=1;
        V_vec=215+m1.*(SOC_vec);
        P_pk_n_vec=I_n_vec.^k.*V_vec;
        SOCp_vec=-(1/Q0)*(V_vec/(2*R0)-sqrt((V_vec/(2*R0)).^2 ...
            -(P_pk_n_vec/R0)));
        Pech_vec=-V_vec*Q0.*SOCp_vec;
        w_SOC=5*(0.9-SOC(i-1));
        H_vec=(L1(i)+w_SOC).*Pech_vec;
        [Hmin,min_index]=min(H_vec);
        Tel_p(i)=0;
        Tel_n(i)=-T_opt(min_index);
    end
end

```

A questo punto, calcolato il valore ottimo di coppia del motore elettrico, si procede con l'aggiornamento di tutti i parametri del sistema fino ad arrivare al nuovo valore di stato di carica e al successivo aggiornamento del valore di λ . L'aggiornamento di questi parametri è effettuato rispettando alcuni vincoli. Infatti, oltre ai vincoli sulla coppia massima dei motori, considerati in precedenza, è necessario considerare i limiti relativi alla potenza massima della batteria, alla corrente massima della batteria e al valore massimo di SoC.

```

P_batt_p(i)=Tel_p(i)*w(i)/eff_motore_el(i);
if (P_batt_p(i)>PbattMax)
    P_batt_p(i)=PbattMax;
end
if (isnan(P_batt_p(i)))
    P_batt_p(i)=0;
end
P_batt_n(i)=Tel_n(i)*w(i)*eff_gen(i); %calcolo la potenza fornita alla batteria
if (isnan(P_batt_n(i)))
    P_batt_n(i)=0;
end

I_p(i)=P_batt_p(i)/V(i-1); % corrente erogata dalla batteria
if (I_p(i)>IbattMax)
    I_p(i)=IbattMax;
end
I_n(i)=P_batt_n(i)/V(i-1); % corrente fornita alla batteria

D_SOC_p(i)=((I_p(i)).^k)*dt/Q0;
D_SOC_n(i)=(I_n(i).^(1/k))*dt/Q0;
SOC(i)=SOC(i-1)-D_SOC_p(i)+D_SOC_n(i); % Con frenata rigenerativa
if (SOC(i)>1)
    SOC(i)=1;
end
Q(i)=Q0*SOC(i); %carica all'istante i
V(i)=215+m1*(SOC(i)); %tensione nominale all'istante i
P_pk_n(i)=I_n(i).^(1/k).*V(i);
P_pk_p(i)=I_p(i).^k.*V(i);
dSOCp(i)=(5e-4)/(35e3))*((P_pk_p(i)-P_pk_n(i))/1e3);
L1(i+1)=L1(i)-dt*L1(i)*dSOCp(i);

```


Questi calcoli concludono il codice relativo al ciclo temporale. Per quanto riguarda invece il ciclo relativo allo

```
SOC_end(j)=SOC(end);
f_SOC_target(j)=SOC_end(j)-SOC_target;
J(j)=dt*sum(P_ic);
figure(1);
plot(t,SOC*100,'color',[0, 0.4470, 0.7410]);
hold on;
end
```

shooting method, rimane da calcolare il valore della funzione costo e lo scostamento tra il SoC finale e quello desiderato.

bisezione_soc.m

Lo script “bisezione_soc” parte dalla funzione f_SOC_target che, per come è definita, ha uno zero nel punto in cui il SoC finale coincide con il SoC target. La bisezione ha il compito di trovare il λ iniziale ottimale corrispondente a questo punto.

```
f_p=find(f_SOC_target>0);
L_sup=Ll_0_vec(f_p(1));
L_inf=Ll_0_vec(f_p(1)-1);
Ll=ones(1,length(t))*L_sup;
ciclo_tempo;
SOC_fu=SOC(end)-SOC_target;

Ll=ones(1,length(t))*L_inf;
ciclo_tempo;
SOC_fl=SOC(end)-SOC_target;

Lm=0.5*(L_sup+L_inf);
Ll=ones(1,length(t))*Lm;
ciclo_tempo;
SOC_fm=SOC(end)-SOC_target;
```

Il primo passo consiste nell'estrarre i due valori di λ iniziali usati precedentemente le cui traiettorie hanno portato a valori di SoC finali immediatamente superiore e inferiore a quello desiderato. Questi valori sono stati definiti come Lsup ed Linf. Per iniziare le iterazioni della bisezione (per un totale fissato di dieci iterazioni), bisogna calcolare il punto Lm, medio tra Lsup ed Linf.

Per ognuno di questi tre punti, bisogna ricalcolare il ciclo temporale e valutare il valore di SoC finale a cui portano. Con questo valore si calcola poi la funzione f_SOC_target, cioè la differenza tra SOC finale e SOC target.

```
fm=SOC_fu*SOC_fl;
fu=SOC_fu*SOC_fm;
fl=SOC_fm*SOC_fl;
if fl>0
    L_inf=L_m;
elseif fu>0
    L_sup=L_m;
else
    L0=L_m;
    break
end
```

Per aggiornare i valori di Lsup/Linf da usare durante l'iterazione successiva, è necessario calcolare le funzioni di attraversamento. Sulla base dei loro valori si possono avere tre condizioni:

- Linf=Lm se il punto è compreso tra Lm ed Lsup
- Lsup=Lm se il punto è compreso tra Linf ed Lm
- L0=Lm se il punto è in Lm

Se non si verifica nessuna delle due condizioni $fl > 0$ ed $fu > 0$, allora il valore di λ iniziale ottimo è proprio Lm e quindi il ciclo della bisezione si arresta.

Se questo non accade, terminate le dieci iterazioni, rimangono i due punti L_{inf} , L_{sup} , con i quali è possibile calcolare un nuovo L_m . Fatto ciò, è possibile ricalcolare, attraverso il ciclo temporale, per ognuno dei tre punti, il SoC finale e prendere il punto il cui SoC finale si avvicina di più al SoC target.

```

if(fm~=0)
    L0=0.5*(L_sup+L_inf);
    L1=ones(1,length(t))*L0;
    ciclo_tempo;
    SOC_fm=SOC(end);
end

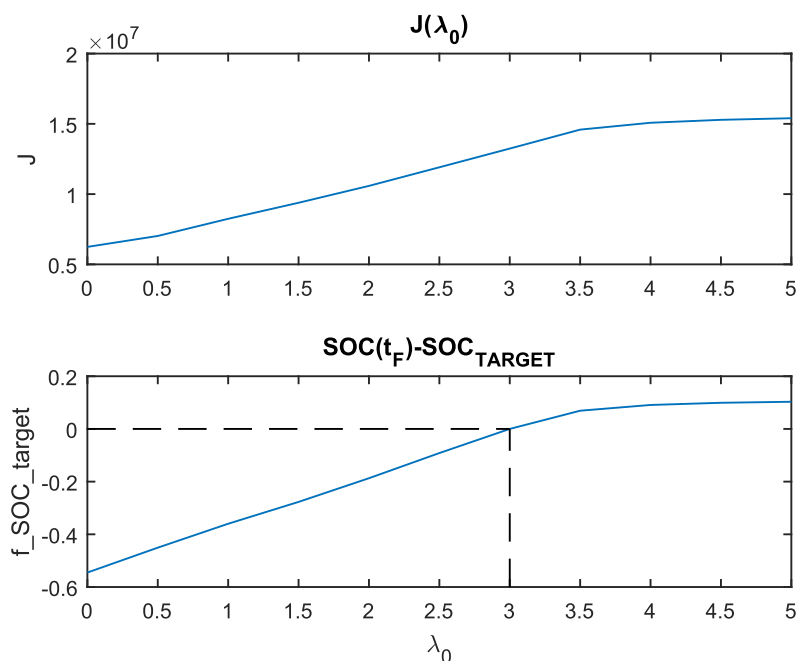
[min_val, indice]=min([abs(SOC_fu-SOC_target),abs(SOC_fl-SOC_target),abs(SOC_fm-SOC_target)]);
if(indice==1)
    L0=L_sup;
elseif(indice==2)
    L0=L_inf;
else
    L0=L_m;
end

L1=ones(1,length(t))*L0;
ciclo_tempo;

```

ANALISI DEI RISULTATI

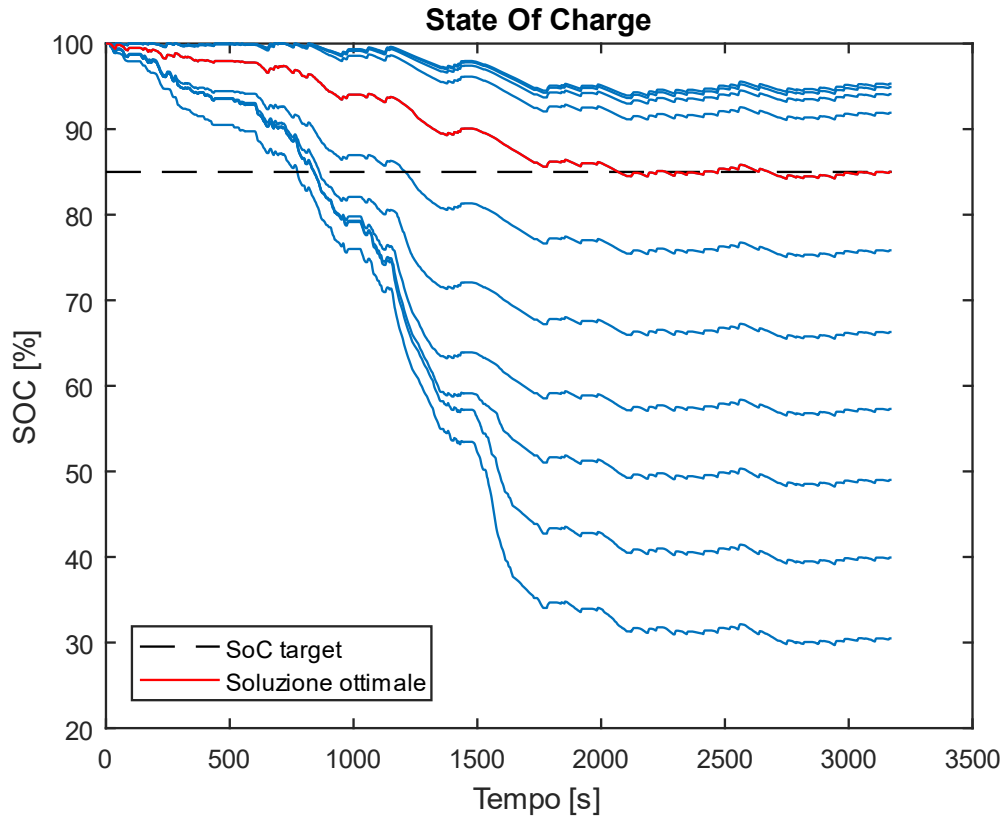
Dai risultati ottenuti dalla simulazione è possibile osservare come procede il processo di ricerca della soluzione ottimale. Un primo grafico da analizzare è relativo all'andamento della funzione costo e dello scostamento tra SoC finale e SoC target rispetto al valore iniziale di λ .



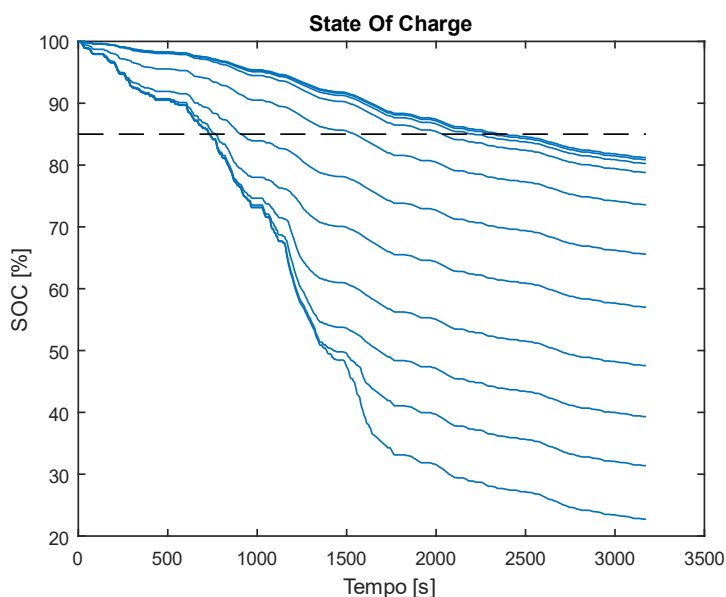
La funzione costo ha un andamento inizialmente crescente con λ per poi assestarsi su un valore costante. Il tratto costante corrisponde ai valori più alti di λ e quindi alle traiettorie caratterizzate da un maggiore costo sull'uso del motore elettrico. In questi casi, quindi, viene utilizzato quasi esclusivamente il motore a combustione per l'intero ciclo e quindi la funzione costo, data dall'integrale della potenza del motore a combustione, assume il suo valore limite per quel ciclo di guida. Nel tratto lineare invece c'è una certa suddivisione tra motore a combustione e motore elettrico, quindi sicuramente il λ iniziale ottimo si trova in questa zona. Questo comportamento è dimostrato dal grafico relativo allo scostamento tra SoC finale e SoC

target: questa funzione attraversa lo zero per $\lambda = 3$ (valore ottenuto dopo la bisezione) che ricade proprio nel tratto lineare della funzione costo.

A questo punto, è possibile analizzare le traiettorie di stato di carica ottenute per i vari valori iniziali di λ e per il valore ottimale individuato:

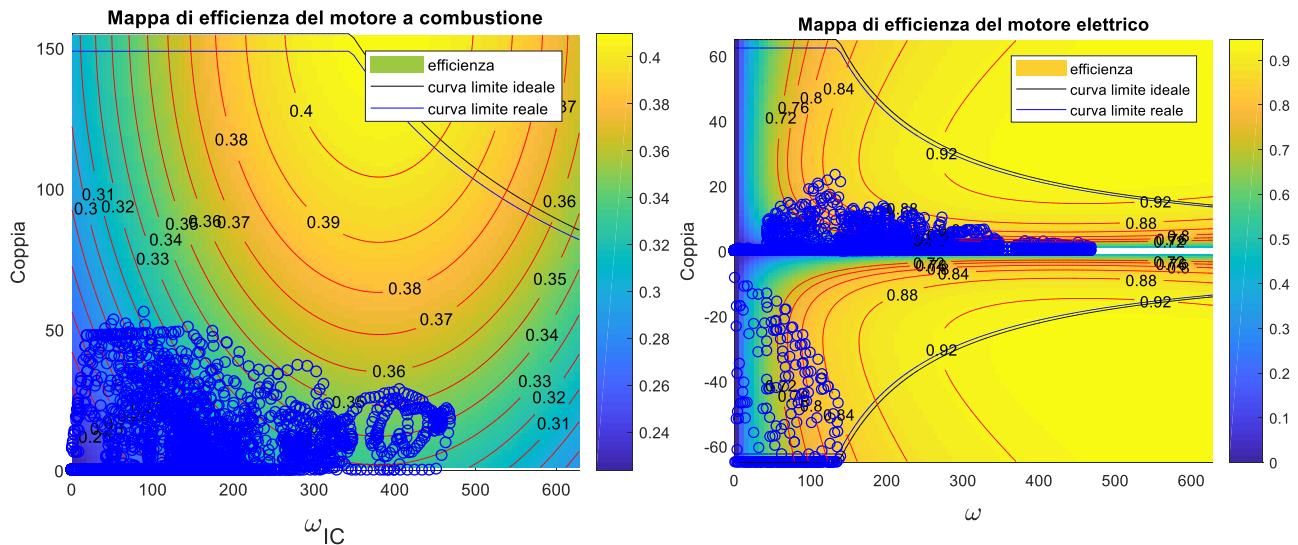


Dall'analisi delle curve si può dedurre che quelle con SoC finali più alti corrispondono ai valori più grandi di λ iniziale e quindi alle strategie che prediligono l'uso del motore a combustione e che sono quelle con la funzione costo più alta. Invece, le curve con SoC finali più bassi sono quelle corrispondenti ai valori di λ iniziale più bassi per i quali si è fatto largo uso del motore elettrico, il che ha portato a scaricare di più la batteria. Queste curve sono quelle con funzione costo più bassa perché il motore a combustione viene usato poco.



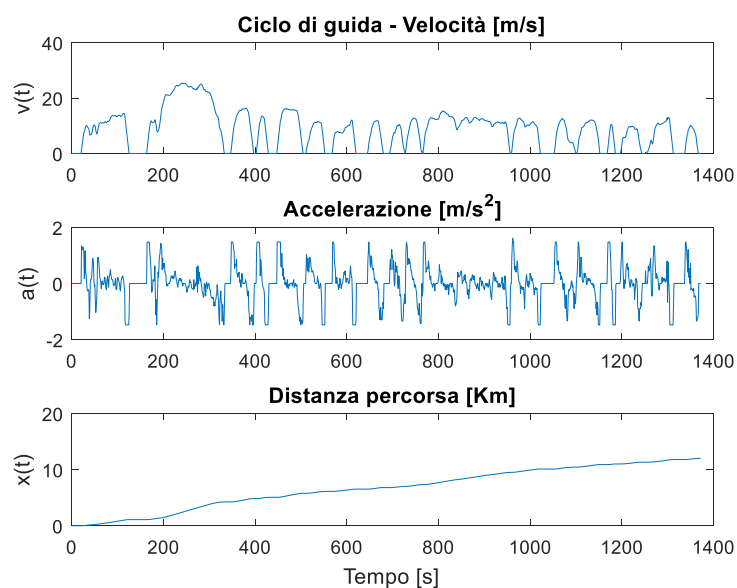
Effettuando le stesse simulazioni, ma in assenza di frenata rigenerativa, si può notare come il sistema non sia in grado di trovare la soluzione ottimale che permette di rispettare il vincolo sul SOC finale desiderato. In generale, tutte le traiettorie hanno una pendenza maggiore e portano ovviamente a valori di SOC finali più bassi.

Per avere una visione più completa della suddivisione ottimale ottenuta tra motore elettrico e motore a combustione è possibile osservare le mappe di efficienza dei due motori sulle quali sono riportati i punti di lavoro ($[w, T]$) relativi al loro funzionamento nel ciclo di guida simulato e per la strategia di controllo ottima individuata.

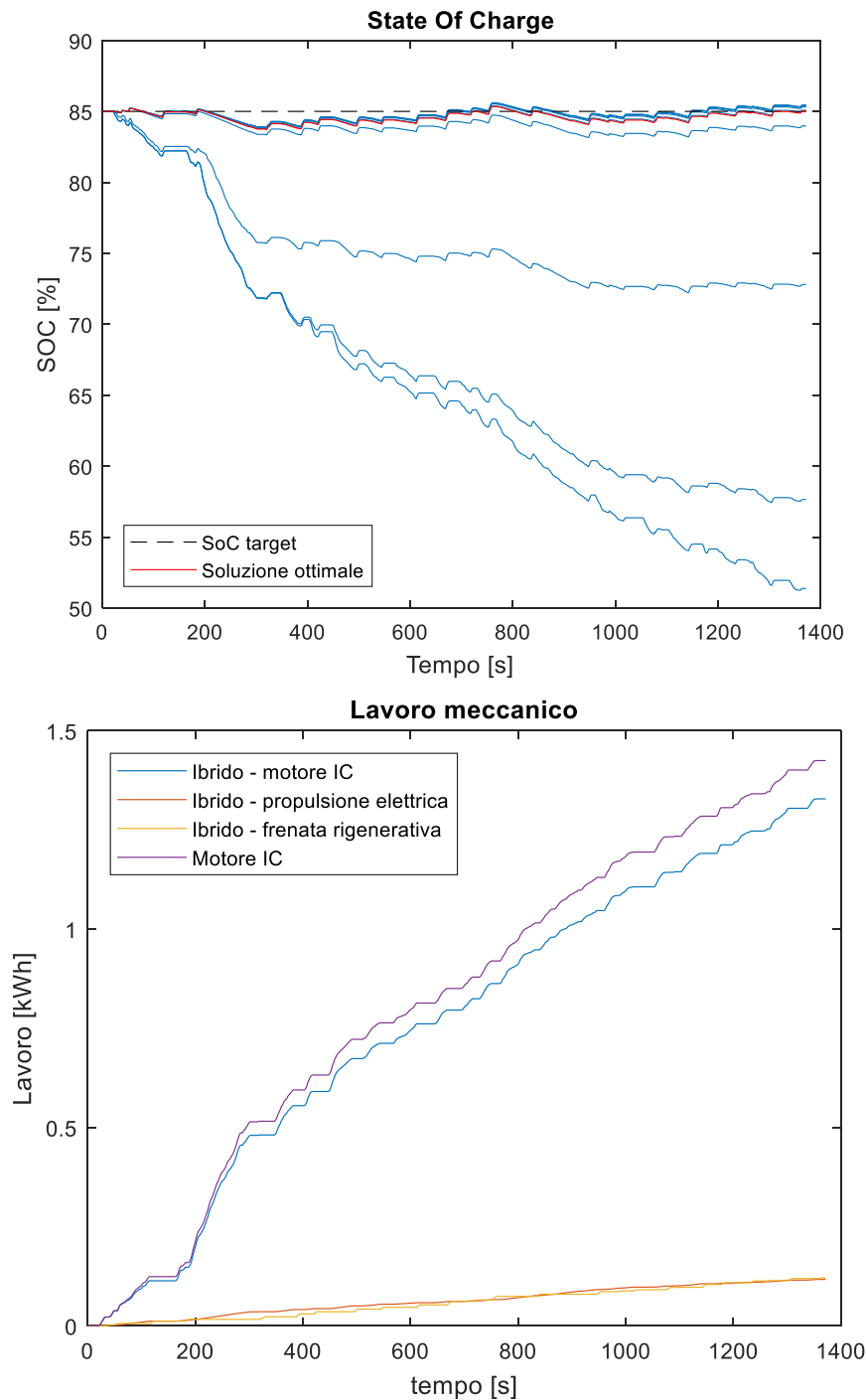


La mappa del motore elettrico, a differenza di quella del motore a combustione, contiene una zona a coppia negativa: rappresenta la condizione in cui il motore elettrico è utilizzato come generatore di energia per ricaricare la batteria (frenata rigenerativa). Alcuni dei punti presenti in questa zona si trovano sulla curva limite, cioè corrispondono alla condizione in cui la frenata imporrebbe una coppia molto elevata che però il generatore non è in grado di sfruttare totalmente.

Una ulteriore simulazione è stata effettuata sulla base del ciclo di guida FUDS, riportato in figura, con lo scopo di confrontare l'energia spesa per percorrere l'intero ciclo usando esclusivamente il motore a combustione con l'energia spesa dallo stesso motore nel caso in cui però si fa uso anche del motore elettrico, della frenata rigenerativa e dell'ottimizzazione numerica. In particolare, per poter fare questo confronto, il motore elettrico deve essere utilizzato in maniera tale da garantire l'uguaglianza tra il SOC di partenza e quello finale. In questo modo, l'energia spesa dal motore elettrico è uguale a quella recuperata in frenata e quindi ci sarà un vantaggio sull'energia spesa dal motore a combustione.



I seguenti grafici illustrano i risultati ottenuti da questa simulazione:



Una prima cosa che si può notare è che le due curve che rappresentano l'energia spesa dal motore elettrico in fase di trazione e quella recuperata in fase di frenata sono quasi uguali. Per quanto riguarda invece il motore a combustione, nel caso in cui non si utilizza proprio il motore elettrico, il lavoro totale risulta essere pari a 1.425 kWh . Quando invece il motore a combustione è associato a quello elettrico, il lavoro totale è pari a 1.328 kWh . Complessivamente, quindi, l'utilizzo del motore elettrico e dell'ottimizzazione numerica permette in questo caso di avere un risparmio sull'energia spesa dal motore a combustione del 6.8%.