**POLITECNICO DI MILANO**

**Scuola di Ingegneria Industriale e dell'Informazione**

**Corso di Laurea in Ingegneria Elettrica**



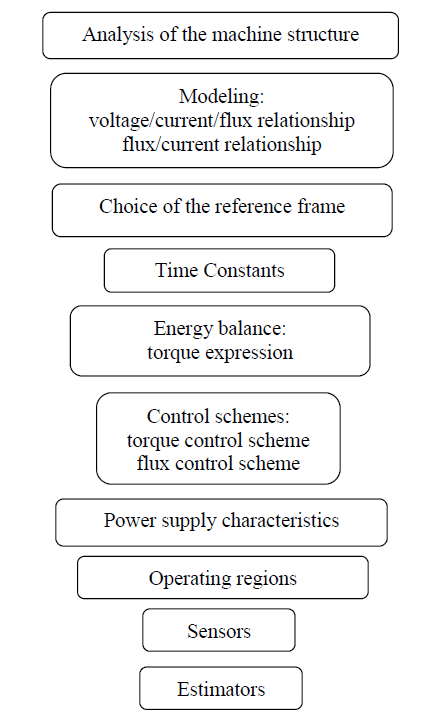
Simulation of a DC motor drive for a Tram with a reference speed profile

Alessandro Secchi

Matr. 944668

The DC machine control drive is very important since is the “base” for many other types of machine control.

Generally, before to study a drive, a general procedure requires to analyse the machine first:



The DC machine principle is based on the following two points:

1. An excitation winding realized on the stator, generates a constant excitation magnetic field.

2. An armature winding mounted on the rotor, connected to a commutator, which, through the brushes, realizes the rectification of ac electromotive forces into a rather constant voltage proportional to the mechanical speed and to the flux 𝛹𝑎ⅇ. The main function of the commutator in the DC machine is to collect the current from the armature conductor as well as supplies the current to the load using brushes and also provides uni-directional torque for DC-motor.

There is a fundamental point in the DC machine:

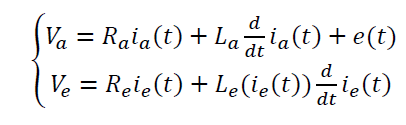
When the current-carrying conductor placed in a magnetic field rotates and cuts the flux of the magnetic field, according to the electromagnetic induction phenomenon “when the conductor cuts the magnetic field, EMF is induced in the conductor”.

In a generator, the EMF of rotation can be called the generated emf, in the motor, the emf of rotation can be called as counter or back emf.

The back emf opposes the supply voltage. The supply voltage induces the current in the coil which rotates the armature. The electrical work required by the motor for causing the current against the back emf is converted into mechanical energy. And that energy is induced in the armature of the motor. Thus, we can say that energy conversion in the DC motor is possible only because of the back emf.

**General electrical equation for separately excited DC motor**

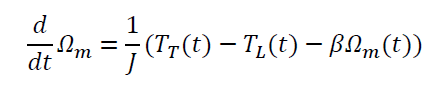
The equations necessary to model the machine are the following:



Where:



And:



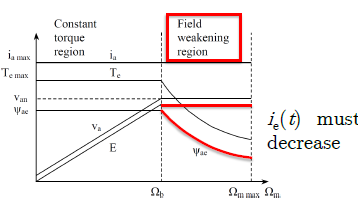
Where:

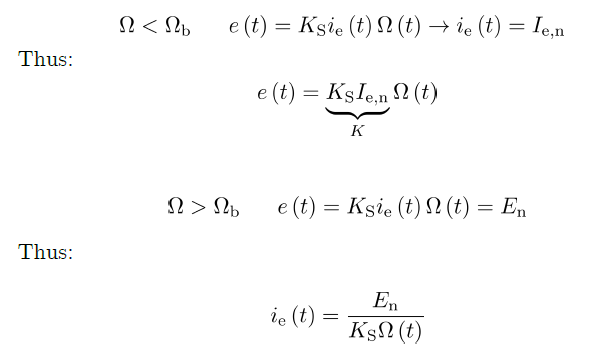


The excitation flux takes place mainly in ferromagnetic material, consequently the relationship between excitation flux and the excitation current is not linear. Therefore, the excitation inductance 𝐿𝑒(𝑖𝑒(𝑡)) is a function of the excitation current.

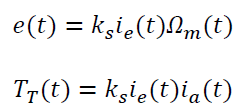
While the armature field lines pass through high air gaps and the reluctance of armature flux path inside the ferromagnetic material is negligible respect to the reluctance of the air-gap path. This implies that the armature inductance 𝐿𝑎 is considered constant.

For this simulation, the saturation of the excitation flux is neglected and the linearity between the flux and the excitation current is admitted.



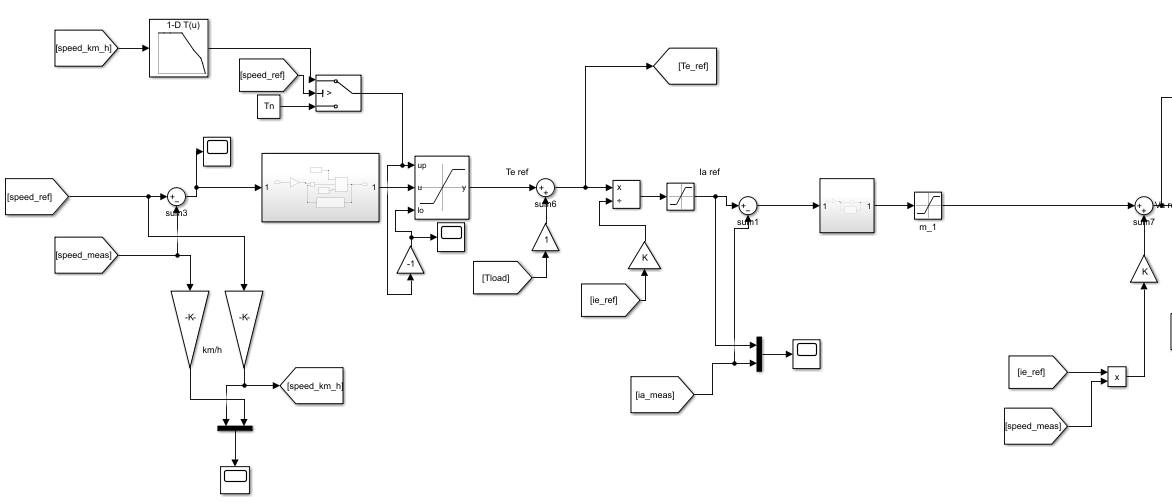


So the main two equation to remember are:

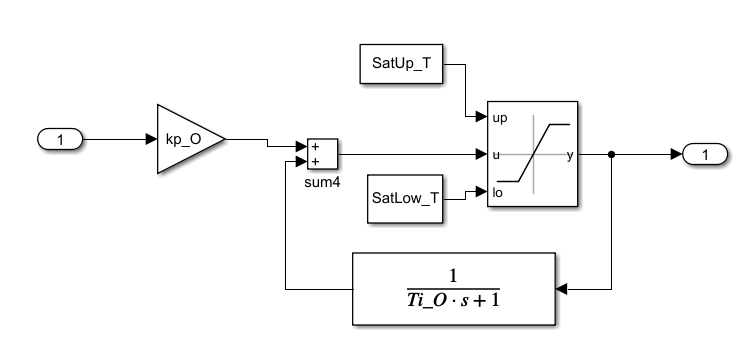


Where 𝑘𝑠 includes the linearized link between excitation current and torque.

Therefore the control unit of the DC machine (referred to the torque control) is:

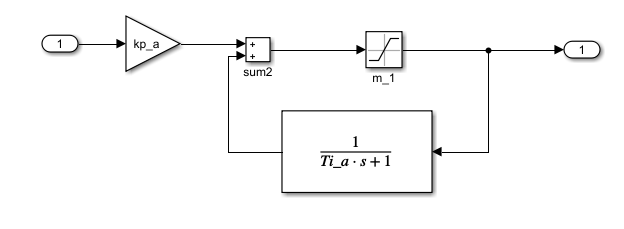


Where the saturation constrains are placed inside the regulators by considering the constant torque region and the field weakening region:

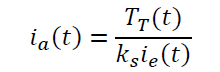


The controller above includes torque saturation (Tn under the base speed and Tn\_weak if over the base speed) and anti-windup properties.

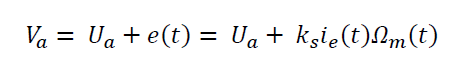
While the current regulator includes a saturation on the armature voltage:



Then starting from the reference torque, the reference armature current is computed according to:



By balancing the output of the control voltage with the back emf, the reference armature voltage is computed:



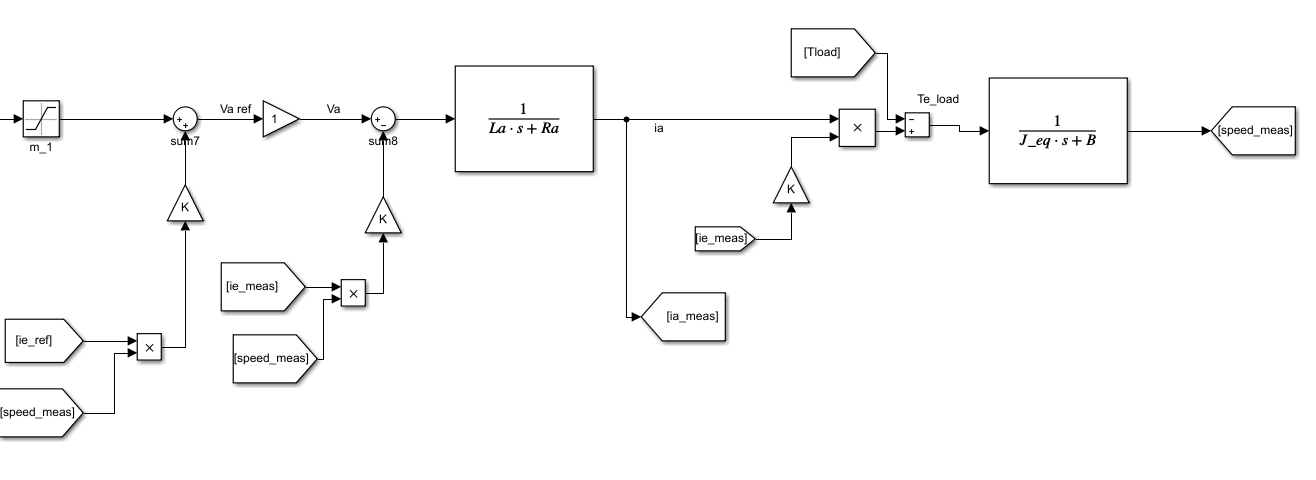
In this way electrical and mechanical dynamics are decoupled through e(t).

Note: decoupling through a compensation term is possible if a speed sensor or a good speed estimation is available.

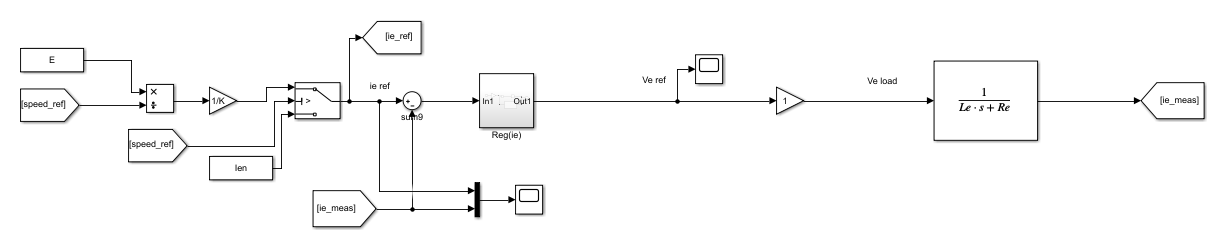
Since the dynamics are decoupled, the two PI controllers can be designed independently and considering the cascade constrains (10\*ωΩ < ωi). (Please check the code at the end of the report).



The power unit includes a power supply that is considered so fast to be approximated as a unity gain.

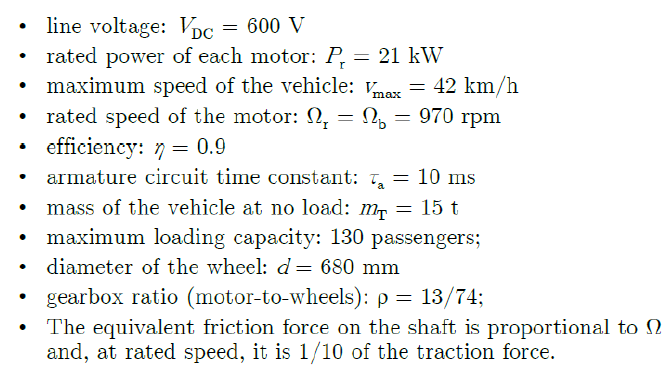


The control scheme of the flux control is modified according to the linearization



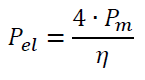
The studied vehicle is the tramway vehicle “Carelli1928” by ATM company (that is still in operation in Milan and San Francisco). It is moved by four DC motors with series excitation.

It is characterized by these parameters:

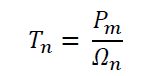


From these values, we can compute all the value we need to model the system:

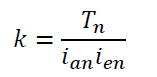
The electric power is calculated from the mechanical power and the efficiency (considering four motors that behave like one equivalent DC motor with separate excitation).



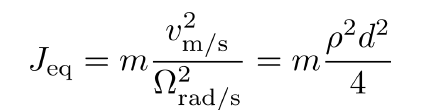
The nominal torque is computed from the mechanical power and the motor speed:

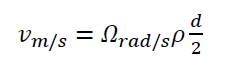


The machine torque constant is:



The mechanical parameters of the machine are calculated as following (for details, please check code at the end of the report).



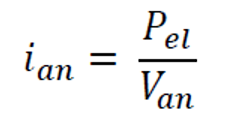




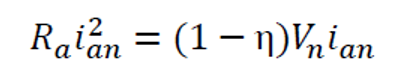
It is important to outline that the mass includes the mass of the vehicle at not load and the mass of 130 passengers that is the maximum loading capacity:



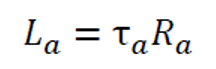
The armature parameters formulas are:



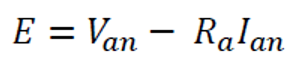
Moreover, knowing the efficiency the armature resistance is computed:



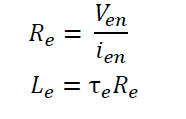
From the armature resistance and the armature time constant, the armature inductance is:



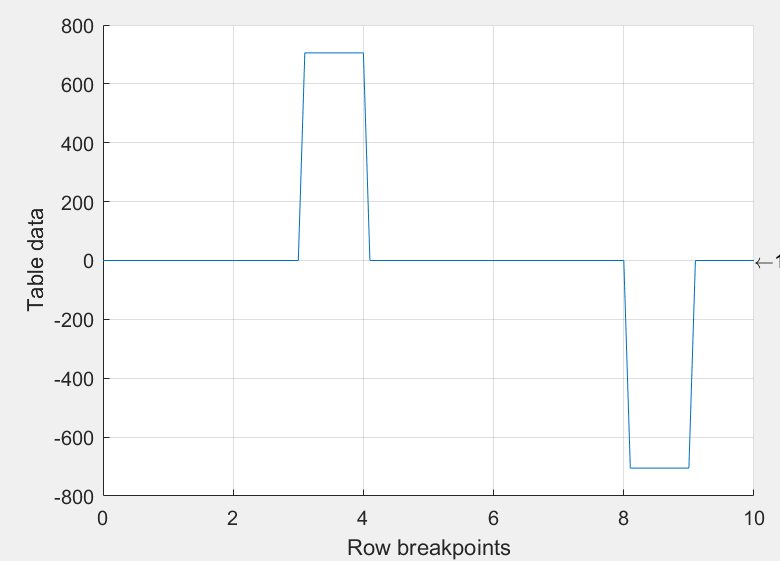
The electromotive force is:



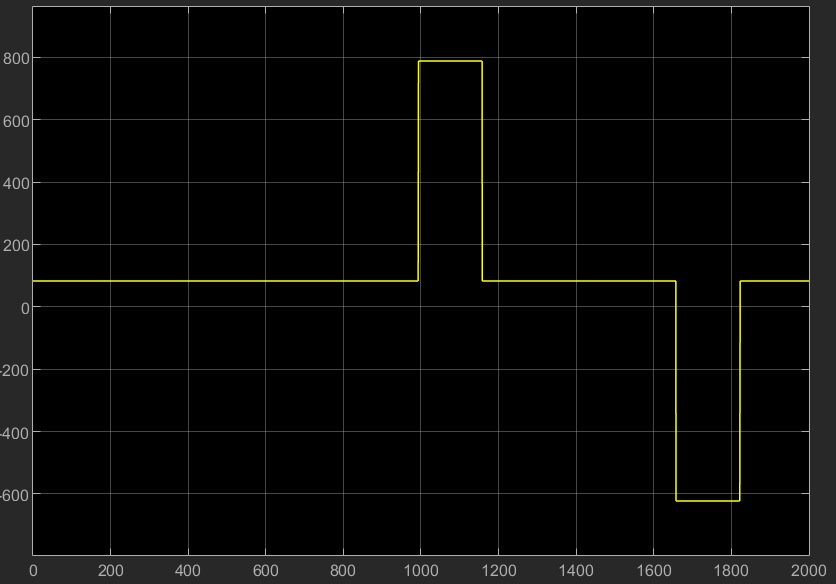
The excitation parameters are:



For the simulation of a DC tram electrical drive, the voltage Va is regulated between 0 and 600 V and the tram will have to run considering also a slop torque and a friction torque (Tn/10):

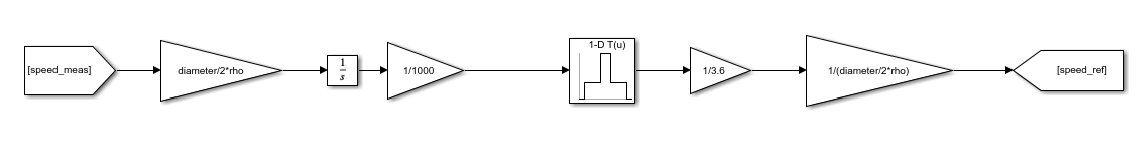


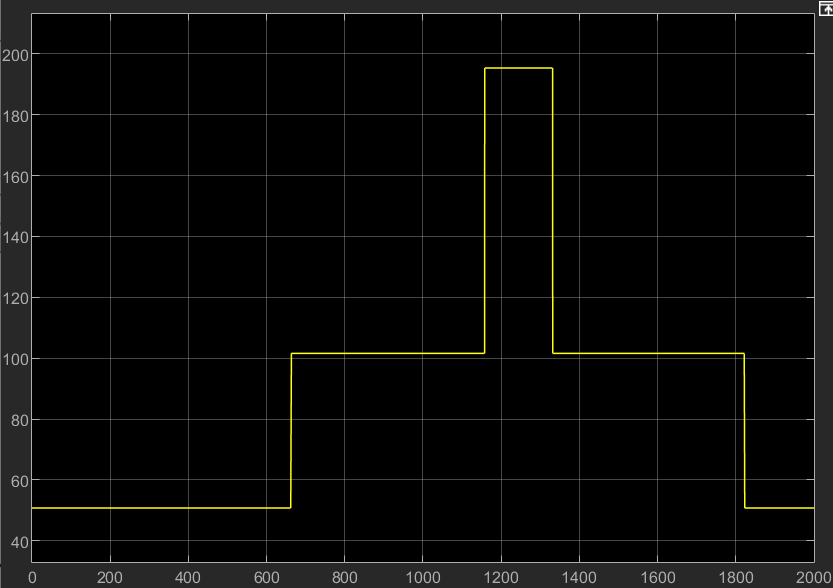
In particular in the time frame considered the load torque + the friction torque will have the following profile:

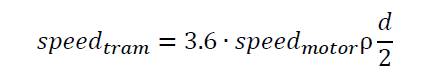


From the figure above it can be seen that there is first a positive slope (uphill) and then, when the torque becomes negative it means that the slope is negative (the tram is going downhill).

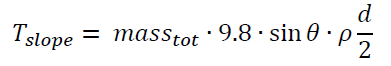
The speed profile on Simulink can be implemented in this way with a lookup table:

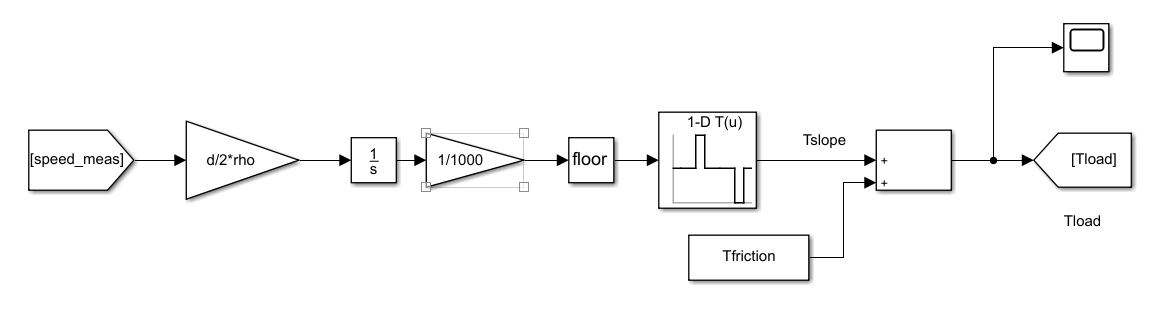






The torque generated by the slope is computed with the following equation:





**Final results:**

It’s interesting to notice that if the effect of the slope on the tram speed profile is compensated with a feedforward term after the speed controller, the speed profile is better fulfilled than without considering the estimated load torque.

The figure shows the comparison between the two cases.

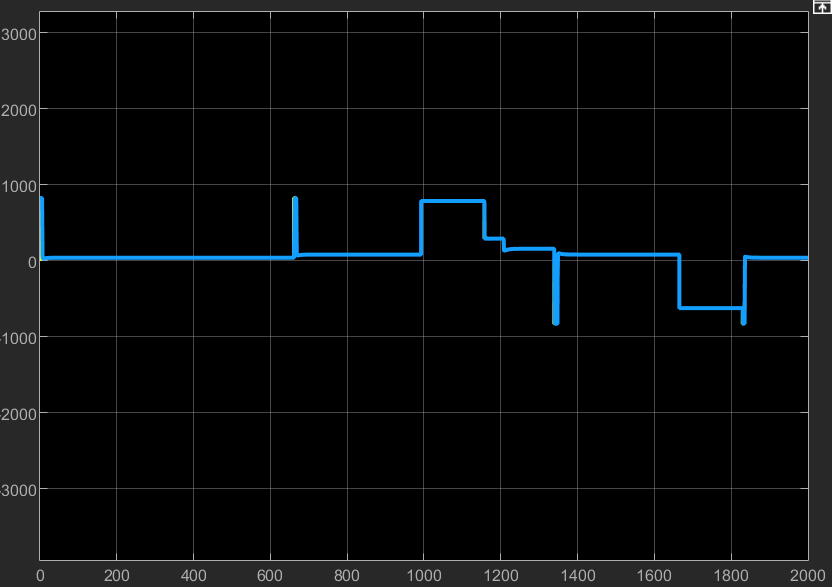
Without the feedforward term:



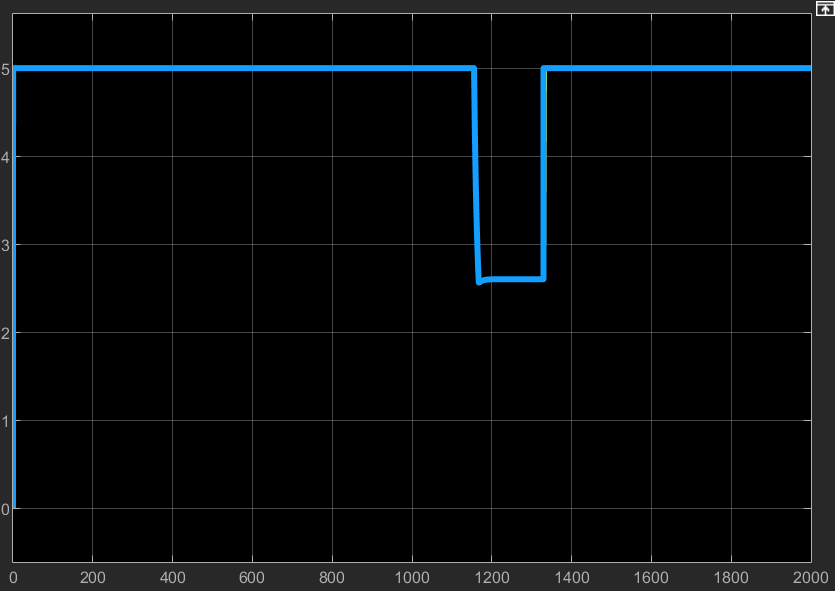
With feedforward load torque term:



The electrical torque of the motor follows the reference torque and does not exceed the limits of the operating regions as shown in figure:



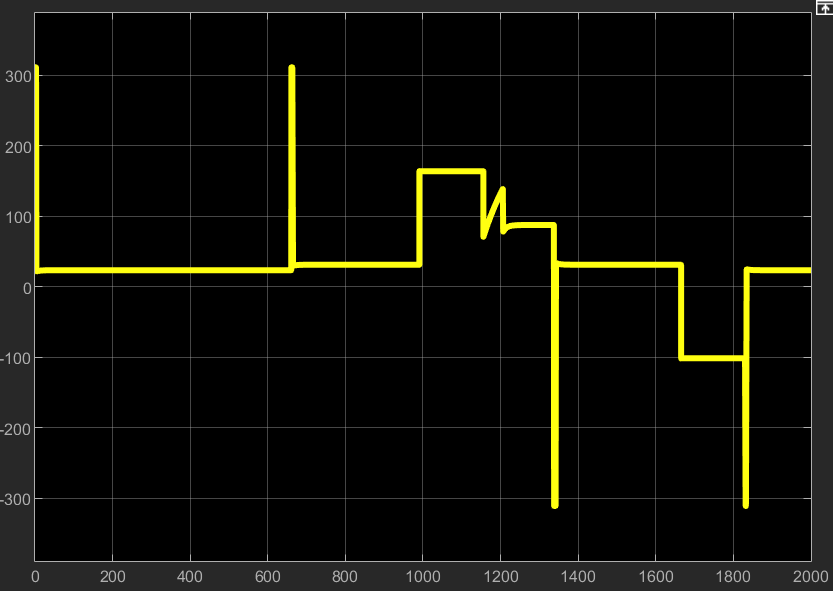
Moreover it is possible to observe also the field weakening region for the excitation current when the speed exceeds the base speed.



If we add the friction torque (Tn/10) the tram seems to not be able to win the high resistive torque over the base speed. Infact, the picture show that the tram is struggling in correspondence of the Tslope (1000s).



This happen since the saturation blocks limit the max torque and current up to the limit value of the motor. If we increase this value and we allow short instants with higher currents (up to 2 time Ian) and up to 3 time for the max torque:





The tram can accomplish its journey:

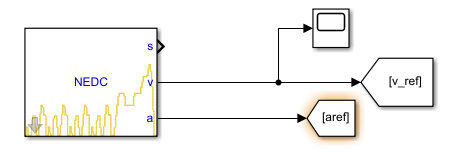


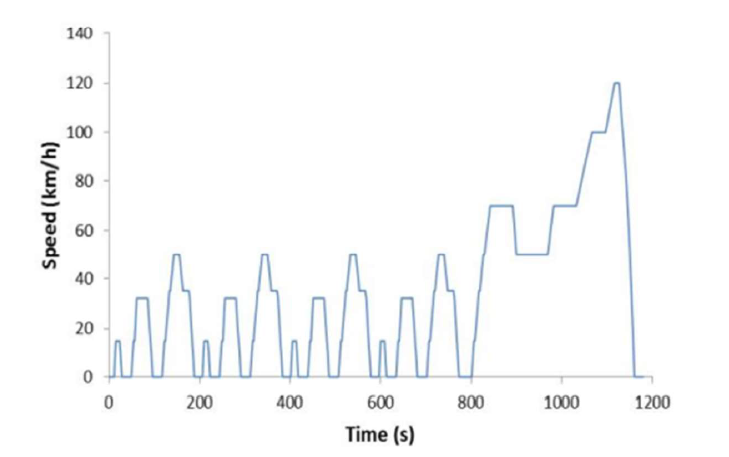
**ADDITIONAL PART:**

If we assume that the tram has its own battery and is not connected with the pantograph, we can add the model of the battery in order to plot the energy consumption, battery temperature and SoC (State of charge).

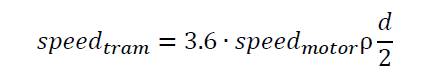
For these simulations, the EV battery cell considered is a NMC Li-ion that has the specifications reported in the code at the end of the report.

Moreover the speed profile considered is given by a typical driving cycle:



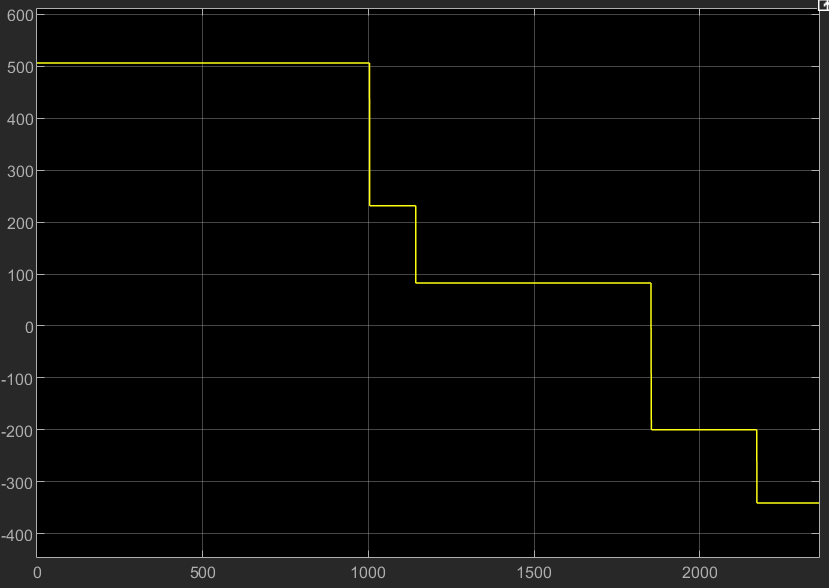


In particular this speed has been used as our speed reference in the model, and the simulation lasts 2 times this cycle so 1180s\*2=2360s.

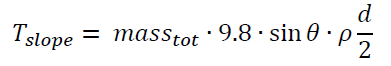


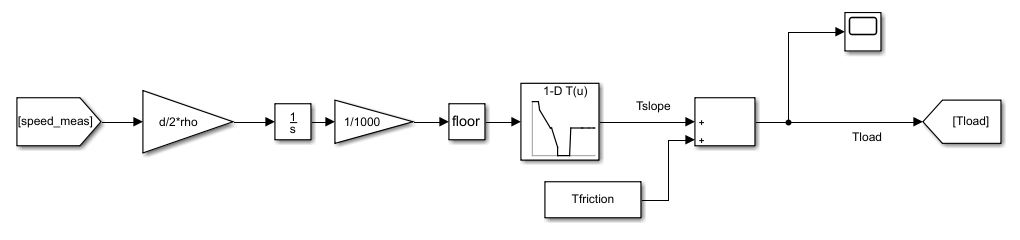
The slope torque follow an arbitrary profile (pretty demanding to test an extreme case).

From the next figure it can be seen that there is first a positive slope (uphill) and then, when the torque becomes negative it means that the slope is negative (the tram is going downhill).



The torque generated by the slope is computed with the following equation:





**Final results:**

It’s interesting to notice that if the effect of the slope on the tram speed profile is compensated with a feedforward term after the speed controller, the speed profile is better fulfilled than without considering the estimated load torque.

The figure shows the comparison between the two cases.

Without the feedforward term:

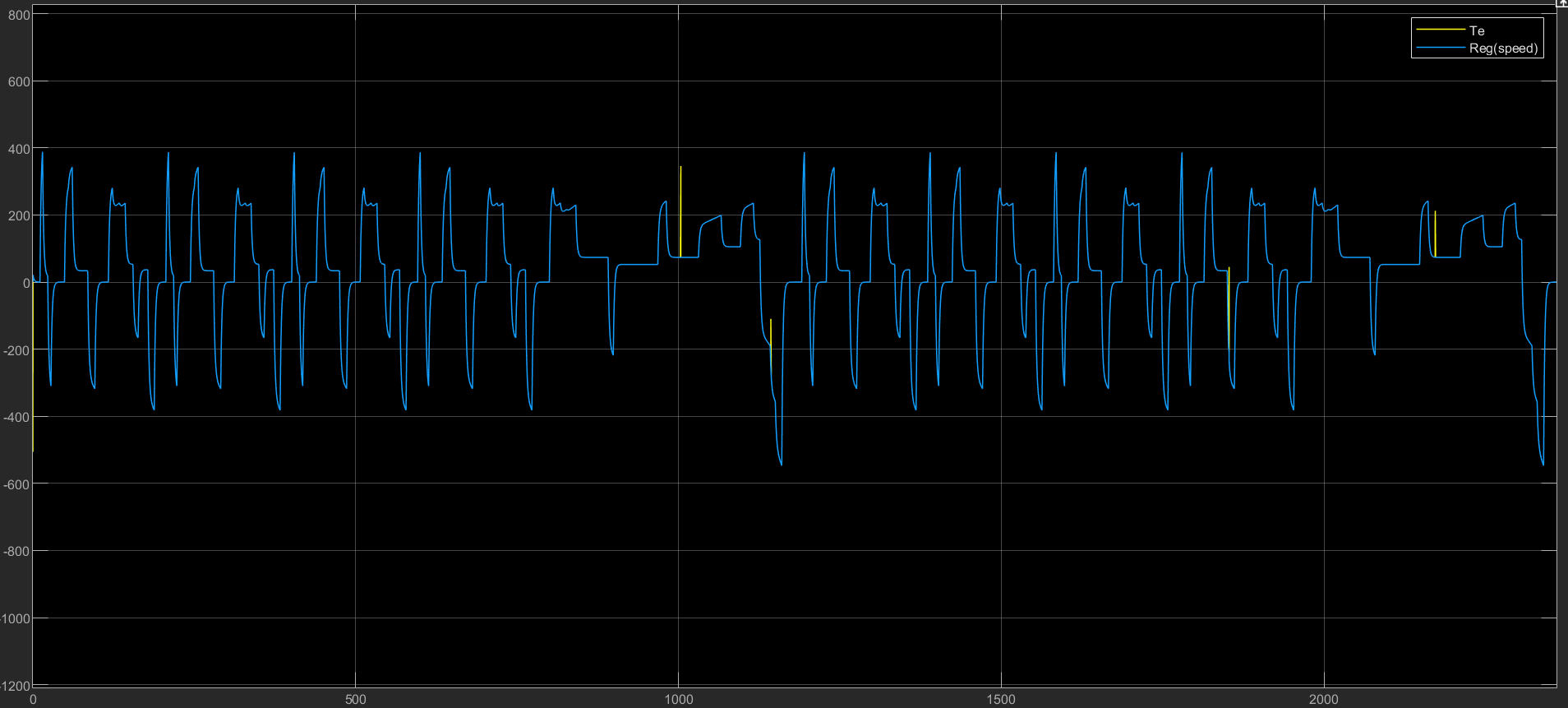


With feedforward load torque term:



At the beginning of the driving cycle it can be seen that without the load torque compensation, the speed profile tends to go under zero when the slope torque is high (the tram in that instant “uphill” tends to have a negative velocity, while with the feedforward it can “predict” the slope and provide the right torque to follow the reference speed).

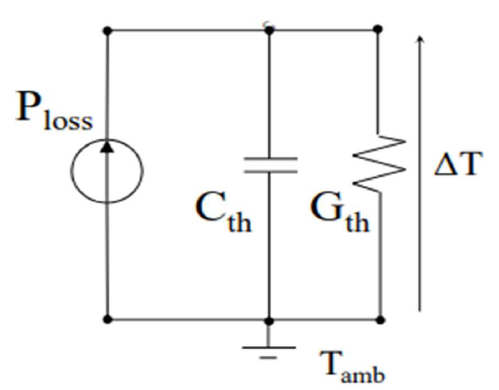
The vehicle measured speed suddenly follows every reference speed variation, increasing and decreasing the electrical torque. The behaviour of the torque is showed in this figure.



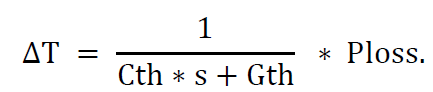
Moreover it is possible to observe also the field weakening region for the excitation current when the speed exceeds the base speed.



For the thermal model of the battery pack, a simplified first-order model is adopted:



The transfer function associated to the model is:



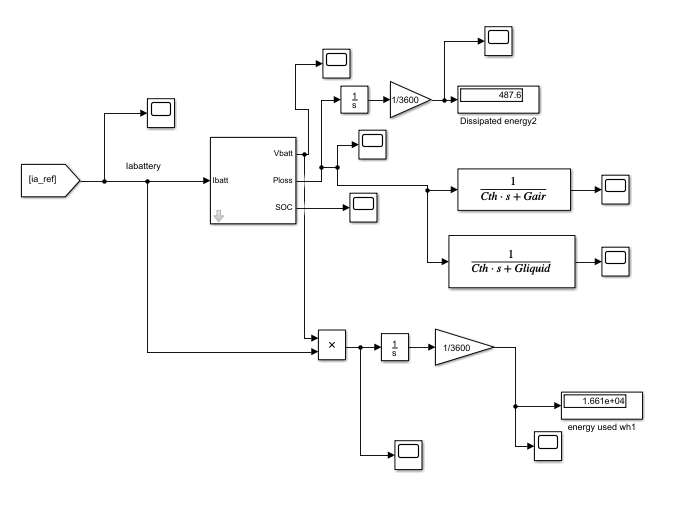
Where Δ𝑇 represents the overtemperature produced during the charging process, Cth

and Gth the thermal parameters of the battery and Ploss the battery power losses.

In order to dissipate the heat, a cooling system is needed.

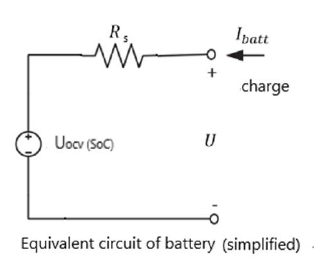
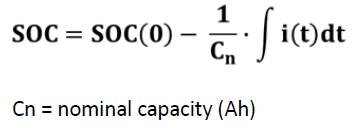
There are 2 possible cooling methods which are: air or liquid cooling.

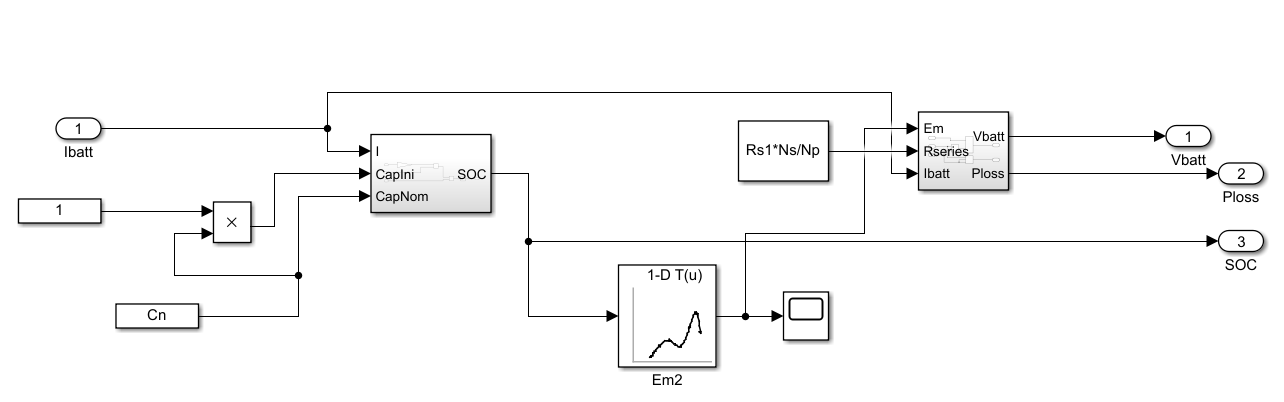
The model drawn in Simulink is:

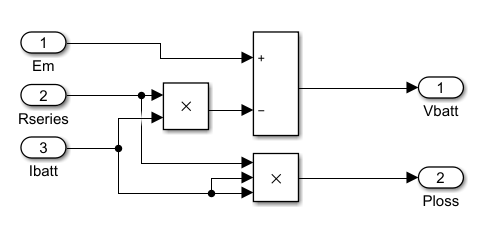
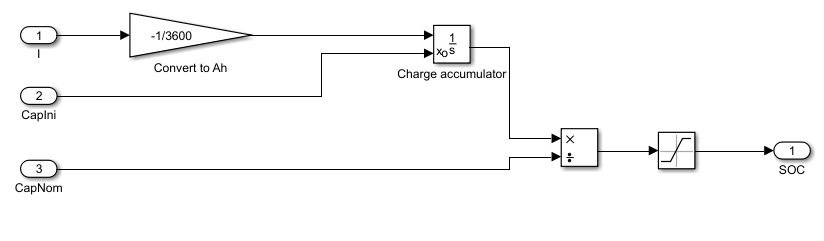


This is a simplified way since the input of the battery comes directly from the ia\_ref block. In alternative I could compute the I battery from Te\*omega that gives the electrical power at the motor shaft and then multiplied (Pem>0) or divided (Pem<0) by the efficiency of the motor in order to find the Pe (and the Pbattery).

Inside the battery model the following equation is implemented and the Vbattery is found from the equivalent circuit of the battery:

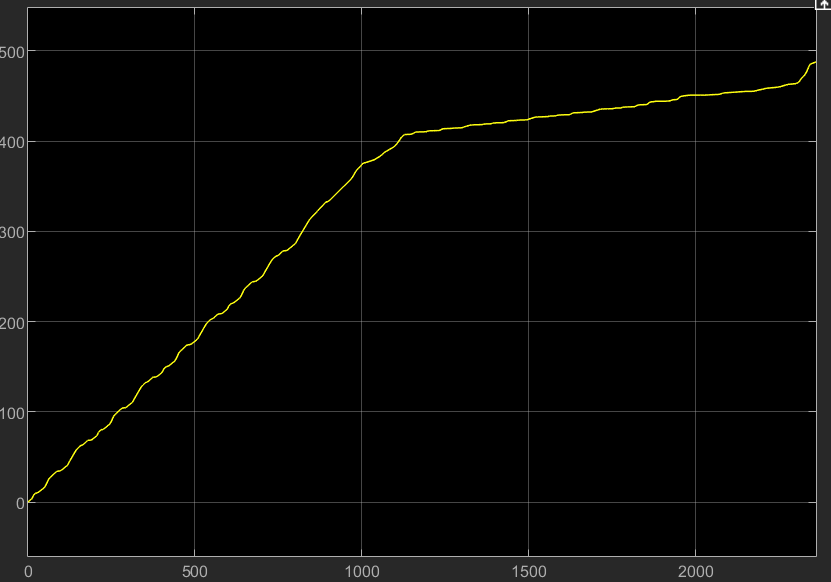






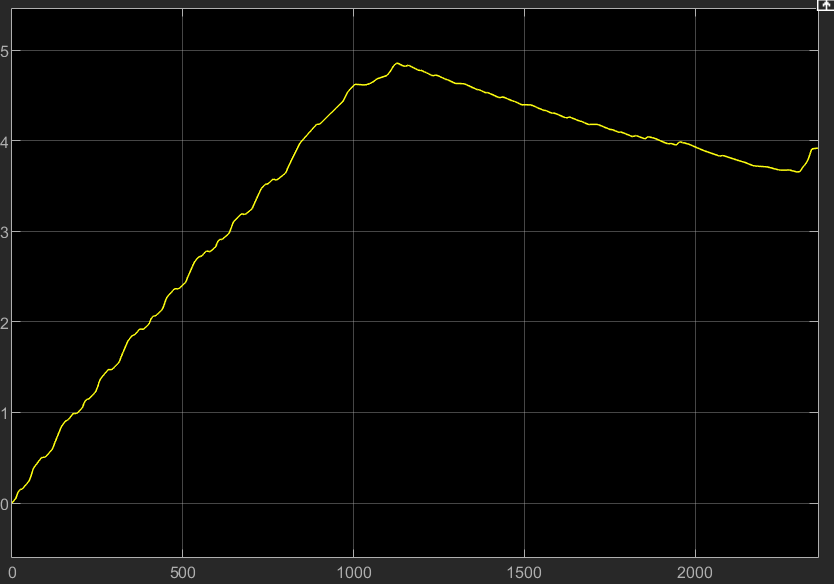
The results are now reported:

The dissipated energy is equal to 487Wh.

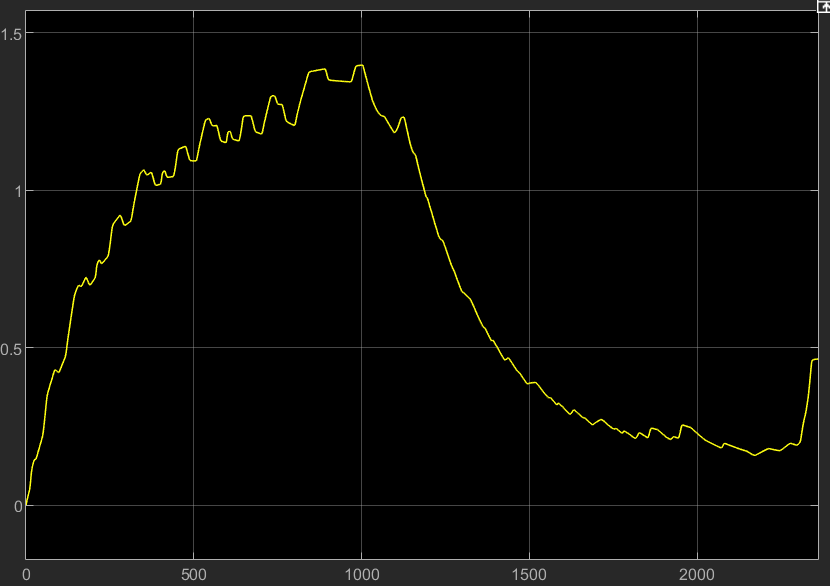


The overtemperature graphs of the battery for both the cooling methods: air and liquid heat exchangers respectively:

Air cooling

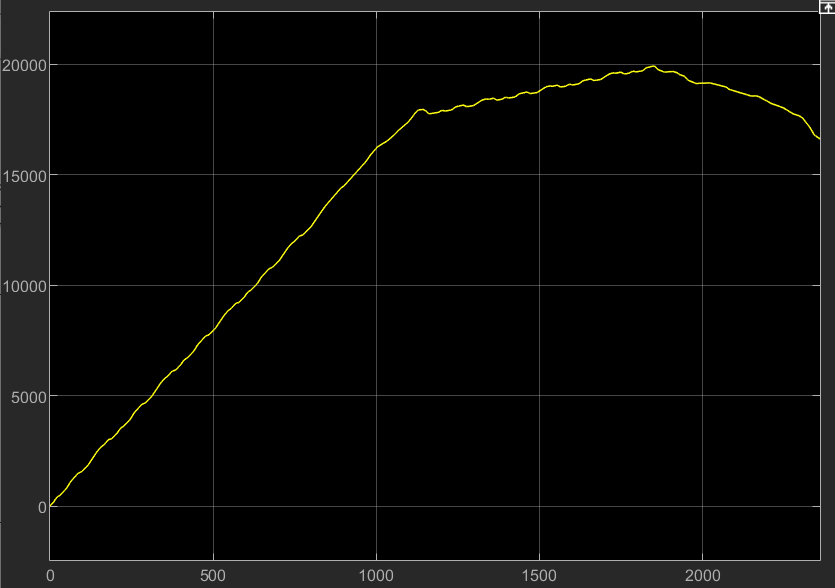


Liquid cooling



At the end of the cycle the overtemperature for the air-cooling method is about 4°C, while for the liquid-cooling system is 0.5°C.

The energy consumption is now reported and at the end is equal to 16.6kWh.



The State of Charge shows that the SoC from 100% is reduced at the end of the cycle to 80%.



From the above results it can be seen that:

* The tram is starting uphill and therefore the first cycle requires a positive torque (and current) while in the last part the tram is moving downhill and therefore there is a “regenerative breaking” that recharge the battery
* Since the speed profile is “periodic” and during the deceleration (negative torque), the battery is recharging, the power used by the battery is about 850W (mainly the one for the auxiliary’s services).

MATLAB CODE:

clc

clear all

%% DC machine parameters

%electrical parameters

%armature

Vdc = 600;

Van =Vdc;

Pmec = 21e3; %W

Pmec\_eq = Pmec\*4;

efficiency = 0.9;

Pel\_eq = Pmec\_eq/efficiency;

Ian = Pel\_eq/Van; % Rotor nominal current

tau\_a = 10e-3; %equivalent tao a

%Ra\*Ian^2 = (1-efficiency)\*Van\*Ian

Ra = (1-efficiency)\*Van/Ian; % Armature resistance [Ohm]

La = tau\_a\*Ra; % Armature inductance [H]

%excitation

Ven = 60;

Ien = 5;

tau\_e = 0.1;

Re = Ven/Ien;

Le = Re\*tau\_e;

E = Van-Ra\*Ian;

%mechanical parameters

speed\_max = 42; %km/h

Or = 970; %rpm of motor

Ob = Or;

rated\_speed\_motor = Or\*(2\*pi)/60; %rad/s

d = 680e-3; %diameter in m

rho = 13/74; %gearbox ratio

speed\_tram = rated\_speed\_motor\*d/2\*rho\*3.6; %km/h

Tn = Pmec\_eq/rated\_speed\_motor; % Nominal torque provided by the machine N\*m

K = Tn/(Ian\*Ien); % Machine torque constant

m\_noload = 15e3; %kg

n\_passengers\_max = 130;

average\_weight\_person = 70;

m\_full\_load = m\_noload + n\_passengers\_max\*average\_weight\_person;

J\_eq = m\_full\_load\*((speed\_tram/3.6)^2)/(rated\_speed\_motor^2);

Tfriction = Tn/10; %at rated speed

distance = [0,1,1.1,3,3.1,4,4.1,6,6.1,8,8.1,9,9.1,9.9,10];

speed\_ref = [speed\_tram/2,speed\_tram/2, speed\_tram,speed\_tram,speed\_tram,speed\_tram,speed\_max,speed\_max,speed\_tram,speed\_tram,speed\_tram,speed\_tram,speed\_tram/2,speed\_tram/2,0 ];

slope = [3,3,2,0,0,-2,-3,-3,0,0,0,0,0,0,0];

theta = atan(slope./100);

Tslope = m\_full\_load\*9.81\*sin(theta)\*d/2\*rho;

Tload = Tslope + Tfriction;

B = Tfriction/rated\_speed\_motor; % friction coefficient < Or

tau\_mec=J\_eq/B;

%% PI controller design parameters

s=tf('s');

%Gi

tau\_a\_desired=tau\_a;

wc\_a=2\*pi/tau\_a\_desired;

%Ge

tau\_e\_desired=tau\_e;

wc\_e=2\*pi/tau\_e\_desired;

%GO

tau\_mec\_desired=tau\_mec/10;

wc\_O=2\*pi/tau\_mec\_desired;

%tf

Gi = 1/(Ra+La\*s);

Ge = 1/(Re+Le\*s);

GO = 1/(B+J\_eq\*s);

%% Pidtool

phase\_m=90;

%pidtool(Gi)

opt=pidtuneOptions('PhaseMargin', phase\_m);

par\_regi=pidtune(Gi,'PI',wc\_a,opt);

ki\_a=par\_regi.Ki;

kp\_a=par\_regi.Kp;

Regi=kp\_a+ki\_a/s

Ti\_a=kp\_a/ki\_a;

%tf open loop

Li=Regi\*Gi;

%tf close loop

Fi=Li/(1+Li);

% figure

% bode(Li);

% figure

% bode(Fi);

% figure

% margin(Li);

%pidtool(Ge)

opt=pidtuneOptions('PhaseMargin', phase\_m);

par\_rege=pidtune(Ge,'PI',wc\_a,opt);

ki\_e=par\_rege.Ki;

kp\_e=par\_rege.Kp;

Rege=kp\_e+ki\_e/s

Ti\_e=kp\_e/ki\_e;

%tf open loop

L\_e=Rege\*Ge;

%tf close loop

Fe=L\_e/(1+L\_e);

% figure

% bode(L\_e);

% figure

% bode(Fe);

% figure

% margin(L\_e);

%pidtool(GO)

par\_reg\_speed=pidtune(GO,'PI',wc\_O,opt);

ki\_O=par\_reg\_speed.Ki;

kp\_O=par\_reg\_speed.Kp;

RegO=kp\_O+ki\_O/s

Ti\_O=kp\_O/ki\_O;

%tf open loop

LO=RegO\*GO;

%tf close loop

FO=LO/(1+LO);

% figure

% bode(LO);

% figure

% bode(FO);

% figure

% margin(LO);

%% saturation

SatUp\_Va = Van; % [A]

SatLow\_Va = -Van;

SatUp\_Ve = Ven\*1.1;

SatLow\_Ve = 0;

%%BATTERY Li-ion cell specifications

Cn1=51; %Ah

Rs1=3.6e-3; %ohm

%quantity

Ns=100; %cells in serie

Ntot=300; %cells

Np=3; %3 branches in parallel

%complete battery

Cn=Np\*Cn1; %Nominal capacity in Ah

%cooling system

A\_exchange=0.5; %m^2;

Kair=200; %W/(m^2\*K)

Kliquid=2000; %W/(m^2\*K)

Gair=Kair\*A\_exchange; %W/K

Gliquid=Kliquid\*A\_exchange; %W/K

Cth=240e3; %j/k battery thermal capacity